USING REVIEWS ON THE WEB TO PREDICT BOX OFFICE SUCCESS WITH MACHINE LEARNING METHODS

by

Burak Sivrikaya

B.S., Computer Engineering, Boğaziçi University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2019

ACKNOWLEDGEMENTS

Firstly, I express my deepest gratitude to my thesis supervisor Assoc. Prof. Arzucan Özgür. During my master education, she has always been supportive to me. It was really challenging for me to complete this thesis together with my busy work life; but she has been understanding about this and helped me to solve the problems that I encountered.

I would like to thank to my thesis committee members Prof. Tunga Güngör and Assoc. Prof. A. Cüneyd Tantuğ for accepting to be in my thesis committee.

I would also like to thank to Assist. Prof. Mustafa Gökçe Baydoğan who taught me the concepts I used in this study during one of his master courses that I've taken. He has been helpful to me whenever I had questions during his course and my thesis.

I am very grateful to my friend and colleague Özgür Sivrikaya for giving me the main idea of my research topic.

My special thanks goes to my brother Fikret Sivrikaya who has always been a great model to me. He has helped and given me ideas whenever I needed, from the beginning of my university life to the end of this study.

I dedicate this thesis to my family; firstly to my father, whom we lost years ago and who has always been proud of the successes of his sons. Today would be another proud day for him. To my lovely mom who always makes great sacrifices for us and to my brothers whose supports can not be described with words.

Finally, during all ups and downs of the last year, there were my close friends next to me. Some people helped me to learn that great things come after great sacrifice and patience. I am lucky to have them in my life.

ABSTRACT

USING REVIEWS ON THE WEB TO PREDICT BOX OFFICE SUCCESS WITH MACHINE LEARNING METHODS

Movie production is a serious undertaking that requires a significant amount of investment. The success of a movie, i.e., positive return on investment, can only be realized with a good performance at the box-office well after the movie's release at theaters. Ability to forecast box-office revenues, which is the amount of money a particular film generates, before a movie's release can decrease the financial risk of film producers. Movie distributors or movie theater organizers can make better decisions about the time and space they assign for a specific movie if they have accurate predictions. However, the box-office success of a movie relies on many factors, some of which are highly subjective, making accurate predictions a challenging task.

Before movies are released in theaters, their trailers, or previews, are made public by movie producers, for which the online social platforms such as YouTube are increasingly utilized as the distribution medium. Such platforms also provide the opportunity for the users to react to and comment on them.

In this thesis, we attempt at using user feedback on movie trailers on YouTube as additional features for box-office success prediction of movies with machine learning. Our results indicate that people's reactions to movie trailers provides a set of helpful features in making more accurate predictions on movie box-office success.

ÖZET

KULLANICI YORUMLARI ÜZERİNDEN MAKİNE ÖĞRENİMİ İLE FİLM GİŞE BAŞARISI TAHMİNİ

Dünya çapındaki film endüstrisinin büyümesiyle birlikte, film yapım bütçeleri de büyük artış göstermektedir. Yüksek bütçeli bir filmin, yapımcılar açısından karlılık sağlayabilmesi için, yüksek gişe rakamlarına ulaşması gerekmektedir. Bir filmin gişe başarısını önden tahmin edebilmek, film yapıcıları açısından finansal riskleri azaltacaktır. Bununla birlikte, film dağıtımcıları ve sinema organizatörleri, bir filmin vizyonda bulacağı yer ve zamana ilişkin kararlar alırken bu tahminleri kullanabilirler. Ancak, bir filmin gişe başarısını etkileyen birçok faktör vardır ve bu konuda başarılı tahminler yapabilmek zor bir iş olarak ortaya çıkmaktadır.

Filmler vizyona girmeden önce, filmden belli kesitlerin yer aldığı fragmanlar, yapımcılar tarafından tanıtım amacıyla yayınlanmaktır. İnternetteki en büyük video paylaşım platformlarından biri olan YouTube içerisinde de bu fragmanlar yer almaktadır. Bu site içerisinde fragmanlar yayınlandığında, kullanıcılar henüz film vizyona girmeden önce filmin içeriği hakkında fikir sahibi olabilir; fragmanla ilgili yorumlarını paylaşabilirler.

Bu tez çalışması, gişe başarısını tahmin etmeye yönelik önceki çalışmalarda kullanılan film özelliklerine ek olarak, kullanıcıların YouTube'daki film fragmanlarına yönelik yorum ve reaksiyonlarından elde edilecek bir takım yeni özelliklerle, daha iyi bir tahminleme yapılıp yapılamayacağını incelemektedir. Sonuçlar, buradan elde edilen yeni özelliklerin, gişe başarısı tahminleyen modellerin doğruluk oranını yükselttiğini ortaya koymaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

ANN	Artificial Neural Network
CART	Classification and Regression Tree
CV	Cross Validation
DT	Decision Tree
DAN2	Dynamic Artificial Neural Network
XGB	Extreme Gradient Boosting
IMDb	Internet Movie Database
kNN	k-Nearest Neighbor
NLP	Natural Language Processing
SVC	Support Vector Classifier
SVR	Support Vector Regression
RF	Random Forest
NLTK	The Natural Language Toolkit
VADER	Valence Aware Dictionary and Sentiment Reasoner

1. INTRODUCTION

The global film industry has been steadily growing, with increasing amounts of movies produced, money spent by film producers, and tickets sold to viewers. In the context of the film industry, the term "box office" is generally used as a synonym for the amount of money a particular film generates [8]. Ability to forecast box office revenues before and right after a movie's release can decrease the financial risk of film producers. Movie distributors or movie theater organizers can make better decisions about the time and space they assign for a specific movie if they have accurate predictions. Apart from these possible economic benefits, people might use these predictions to have an idea about a movie's potential box-office success and possibly take it into account as an additional reference in their decision to watch a particular movie.

Since the box-office success of a movie relies on many factors, some of which are highly subjective, making accurate predictions is a challenging task. At this point, there are various information sources related to movies that can be used in enriching the models and improving the accuracy of predictions. For example, thousands of online comments about movies or movie trailers might serve as an indicator of the attention that those movies will get in theaters. Some quantitative features showing the volume of people's interaction with the movie or qualitative features showing the general sentiment of people's comments can be used in the predictions.

There are different resources on the web that contain people's opinions about the movies. Some of them have reviews made by movie critics or some members of movie platforms. These sources mostly have a limited number of reviews but their content is more detailed and comprehensive. On the other hand, large-scale social media platforms can provide us thousands of user reviews. These reviews are shorter but they reflect user opinion more directly most of the time. As part of this study, we focused on the reviews on social media platforms and initially selected Twitter and YouTube as the viable sources of user feedback on movies. However, after working on extracting movie-related tweets from Twitter, we realized the challenges of extracting tweets written specifically about a particular movie. Many movie names share common words used in daily language and that makes it really hard to determine if the tweet is actually about a movie, even if the tweet text or a hash-tag contains the movie name. Therefore, we decided to leave twitter reviews out of this study's scope and focus only on YouTube data.

Before movies are released in theaters, movie trailers, which are short previews of movies, are made public by movie producers. They are used in advertisements, shown in theaters before demonstration of other movies, and made available on social media platforms. YouTube, as one of the biggest video streaming platforms, also has trailers of movies before their release. In a trailer video, one can see the positive and negative reactions that the trailer receives as well as the comments made by the platform users. Compared to Twitter, it is much easier to retrieve movie-related reviews since vast majority of comments in trailer is about that movie or the trailer. In general, movie reviews are made after a movie is released or the movie is actually watched, but trailers are mostly watched before the movie is released or watched by a person. Therefore, reactions to movie trailer can be valuable in predicting the box-office success of the movie in the pre-release period.

In this research, with the help of user reviews and reactions on YouTube, we propose a machine learning based method that predicts the box-office success of movies in terms of their profitability. Our motivation is to see if people's reactions to movie trailers is helpful in making more accurate predictions on movie box-office success.

1.1. Contributions of the thesis

The main contributions of this thesis are summarized as follows:

- A comprehensive open data set is created by crawling and integrating information from various sources. This data set has been made publicly available to the research community [9]. The data set contains various features about the movies ranging from basic information like actors, genre and run time to economic information like budget, revenue and number of screens that show the movie. Also, all YouTube user comments that were made on the movie trailer are added to the data set. Detailed information about the data set is given in the later chapters of this thesis.

- This study brings various movie features used in different studies together. By using various subsets of these features in different machine learning models, this study has produced results that can be taken as reference for alternative approaches in the prediction of box-office success.
- There was no prior work in this field that focus on YouTube user reviews and predict box-office success using features coming only from this review data. This study tries to extend the literature by investigating the predictive power of YouTube features on box-office success prediction.

1.2. Overview of the thesis

In the next chapter of this thesis, background information about the concepts and methods used in this study is given. Then, previous work in the field of box-office forecasting is reviewed. Data preparation chapter explains the sources of data used in the study and challenges we encountered in the data retrieval process. Later, in the methodology chapter, we present our approach, with details on how we process the data, how we analyze the sentiment of user reviews, and which features and machine learning methods are used in the forecasting. In the remaining chapters our test and experiment design are explained, the results are discussed, and the conclusion is presented with potential future work.

2. BACKGROUND

This chapter provides some background information about the field and, wherever applicable, highlights the related design choices in this study. Details of our implementation of the concepts mentioned here are explained in detail in further chapters. This chapter starts with some box-office statistics, then proceeds with a short introduction to sentiment analysis and cross validation, and finally, briefly explains each machine learning model used in this study.

2.1. Movie Industry and Box Office Numbers

In this chapter, we provide some numbers to help in understanding the economic size of the movie industry in the United States. According to data from 2018, presented in Table 2.1, there are six big movie studios in the USA with a total market share of 83.6% and a collective annual revenue of almost ten billion dollars.

Table 2.1. Box Office Shares by Studio in 2018 [6]

Rank	Distributor	Market Share	Total Gross (in millions)	Movies Tracked
1	Buena Vista	26.0%	$3,092.4	13
2	Warner Bros.	16.3%	$1,940.7	49
3	Universal	14.9%	$1,771.9	23
4	Sony / Columbia	10.9%	$1,304.3	28
5	20th Century Fox	9.1%	$1,082.3	17
6	Paramount	6.4%	$757.0	12

Movie production is a risky business; many movies fail to even meet their costs with their box-office revenues and end up losing large amounts of money. Table 2.2 and Table 2.3 show the twenty movies with largest profit, respectively largest loss, based on their worldwide gross revenues, based on data retrieved from The-Numbers.com.

Table 2.2. Twenty Movies with the Largest Profit based on Worldwide Revenues

	Release Date	Movie	Approx. Income	Approx. Expense	Profit
1	Dec 18, 2009	Avatar	$1,788,912,936	$516,262,000	$1,272,650,936
2	Dec 18, 2015	Star Wars Ep. VII: The Force Awakens	$1,217,124,229	$381,704,000	$835,420,229
3	Nov 22, 2013	Frozen	$1,036,301,740	$245,904,000	$790,397,740
4	Nov 18, 2005	Harry Potter and the Goblet of Fire	$997,451,307	$208,064,000	$789,387,307
5	Apr 27, 2018	Avengers: Infinity War	$1,140,852,599	$368,812,000	$772,040,599
6	May 4, 2012	The Avengers	$996,738,179	$300,290,000	$696,448,179
7	Jun 12, 2015	Jurassic World	$962,391,434	$287,756,000	$674,635,434
8	Jul 15, 2011	Harry Potter and the Deathly Hallows: Part II	$833,194,380	$177,306,000	$655,888,380
9	Jul 3, 2013	Despicable Me 2	$735,741,192	$143,056,000	$592,685,192
10	Apr 3, 2015	Furious 7	$831,555,147	$245,576,000	$585,979,147
11	Jul 7, 2006	Pirates of the Caribbean: Dead Man's Chest	$859,108,056	$293,790,000	$565,318,056
12	Dec 9, 2005	The Chronicles of Narnia: The Lion, the Witch and the Wardrobe	$808,001,598	$247,308,000	$560,693,598
13	Jul 10, 2015	Minions	$698,238,481	$141,424,000	$556,814,481
14	Jul 18, 2008	The Dark Knight	$817,957,770	$263,340,000	$554,617,770
15	Jul 11, 2007	Harry Potter and the Order of the Phoenix	$729,992,769	$202,594,000	$527,398,769
16	Jun 15, 1994	The Lion King	$718,371,807	$199,776,000	$518,595,807
17	Jun 18, 2010	Toy Story 3	$784,635,427	$267,398,000	$517,237,427
18	Feb 16, 2018	Black Panther	$768,445,609	$280,472,000	$487,973,609
19	Jun 22, 2018	Jurassic World: Fallen Kingdom	$716,567,691	$230,978,000	$485,589,691
20	Mar 17, 2017	Beauty and the Beast	$716,550,704	$231,846,000	$484,704,704

2.2. Sentiment Analysis

Natural Language Processing (NLP) is a field of Artificial Intelligence that aims to extract many attributes from texts like the polarity of expressed opinion, the subject that is talked about in the text, or the identity of the person who writes the text. Sentiment Analysis is a branch of NLP and it is also known as opinion mining or subjectivity analysis. Polarity of a text tells us whether a writer expresses a positive, neutral or negative opinion in the text. This information is highly valuable to understand the general approach of a group of people towards a topic in areas like politics, sports or social incidents. At this point, sentiment analysis techniques help to extract this information from text resources like social media, blogs, and forums. There are rule based, machine learning based and hybrid approaches for sentiment analysis problem, which can be challenging in some cases even for humans.

Table 2.3. Twenty Movies with the Largest Loss based on Worldwide Revenues

Rank	**Release Date**	**Movie**	**Approx. Income**	**Approx. Expense**	**Profit**
1	Mar 11, 2011	Mars Needs Moms	$26,740,081	$170,166,000	-$143,425,919
2	Mar 9, 2012	John Carter	$180,604,340	$307,124,000	-$126,519,660
3	Jul 2, 2013	The Lone Ranger	$177,111,998	$301,886,000	-$124,774,002
4	May 12, 2017	King Arthur: Legend of the Sword	$84,791,792	$199,580,000	-$114,788,208
5	Sep 30, 2016	Deepwater Horizon	$78,084,057	$189,348,000	-$111,263,943
6	Jan 13, 2017	Monster Trucks	$39,869,478	$149,450,000	-$109,580,522
7	Mar 1, 2013	Jack the Giant Slayer	$123,515,271	$228,504,000	-$104,988,729
8	Dec 17, 2010	How Do You Know?	$35,560,921	$140,454,000	-$104,893,079
9	Dec 25, 2013	47 Ronin	$93,151,465	$189,988,000	-$96,836,535
10	Feb 6, 2015	Jupiter Ascending	$110,709,770	$206,582,000	-$95,872,230
11	Apr 21, 2017	The Promise	$6,221,241	$99,886,000	-$93,664,759
12	Jul 19, 2013	R.I.P.D.	$55,135,155	$147,884,000	-$92,748,845
13	Nov 23, 2011	Hugo	$124,386,172	$215,796,000	-$91,409,828
14	May 22, 2015	Tomorrowland	$115,194,864	$205,644,000	-$90,449,136
15	Jun 22, 2007	Evan Almighty	$125,404,475	$212,964,000	-$87,559,525
16	Oct 9, 2015	Pan	$88,988,777	$174,726,000	-$85,737,223
17	Jul 21, 2017	Valerian and the City of a Thousand Planets	$119,752,122	$202,258,000	-$82,505,878
18	May 9, 2008	Speed Racer	$69,490,441	$149,420,000	-$79,929,559
19	Feb 26, 2016	Gods of Egypt	$85,645,390	$161,930,000	-$76,284,610
20	Feb 12, 2010	The Wolfman	$98,455,310	$174,496,000	-$76,040,690

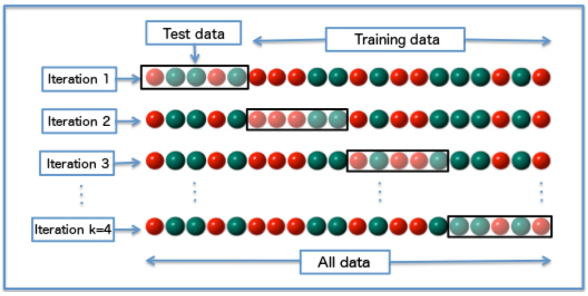

Figure 2.1. N-Fold Cross Validation [1]

In this study, we use two different sentiment analysis approaches to understand user opinions on YouTube movie trailers. Further details on these are given in the Methodology chapter.

2.3. Cross Validation

In machine learning studies, people conduct experiments to compare the performance of different algorithms or methods. In order to reach statistically confident results, it is critical to estimate the uncertainty of estimations in the experiments [10]. N-fold cross validation is a commonly used way of estimating algorithms' expected error and variance. N is a pre-determined number and the data set is randomly divided into N different groups, i.e., *folds*. One of these folds is used as a validation set, while the remaining folds are used to train the model, as depicted in Figure 2.1. This is repeated for each fold and then the performance of the model is evaluated by averaging the errors coming from the folds. In the figure, different colors represent different classes.

For one repetition of n-fold cross validation, only one random split is applied and iterations are not independent of each other. To lower the estimator bias, cross validation step is repeated; therefore the data set is randomly split multiple times. In our study, we apply 10-fold cross validation with 5 repetitions to evaluate different models and feature sets.

2.4. Supervised Machine Learning Methods

In the classification part of this study, in which we have a pre-labeled target variable, we use Decision Tree (DT), k-Nearest Neighbor (kNN), Support Vector Classifier (SVC), Extreme Gradient Boosting (XGB) and Artificial Neural Network (ANN) as supervised learning methods. This section provides a compact overview of these methods.

2.4.1. Decision Tree

Decision tree is a supervised learning algorithm that supports both numerical and categorical inputs. With its flowchart-like structure, a decision tree can be represented as a chain of if-then rules, as illustrated in Figure 2.2. That increases human readability and makes it easier to interpret the results.

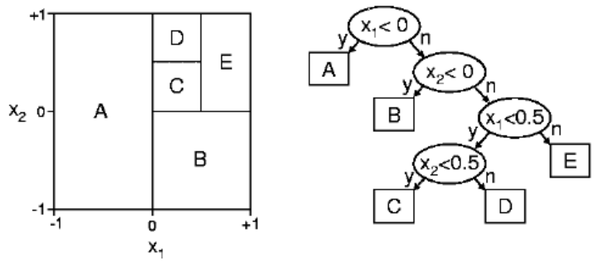

Figure 2.2. Schematic diagram of a decision tree [2]

In the construction of a tree, the *best attribute*, which classifies the training examples well, is selected as the root node. Setting the root node as parent, a child node is created for every possible value of the best attribute. This process is then repeated using the training examples associated with each child node to select the best attribute at that point in the tree. "Best attribute" is chosen according to a statistical property, known as information gain. It numerically measures how good an attribute is in grouping training instances according to their actual classes. While constructing the tree, this measure is used to compare all possible attributes and select the best of them [11].

2.4.2. Random Forest

Random Forest, which is an ensemble learning method, can be considered as a collection of Decision Trees. In the construction of a random forest model, random subsets of the features are created first. Then, using these subsets, smaller trees are built. This procedure helps to prevent over-fitting. After a large number of trees is generated, they vote for the most popular class [12]. This algorithm produces more robust results but it is slower and harder to understand compared to decision trees. Figure 2.3 illustrates a random forest structure. In the figure, each group of nodes represents a tree in the forest. A new instance is classified in each tree. A path of yellow nodes in the figure shows how an instance is assigned to a class by making decisions in each split. Then, classifications from all trees are assembled and final decision is made according to majority-voting method.

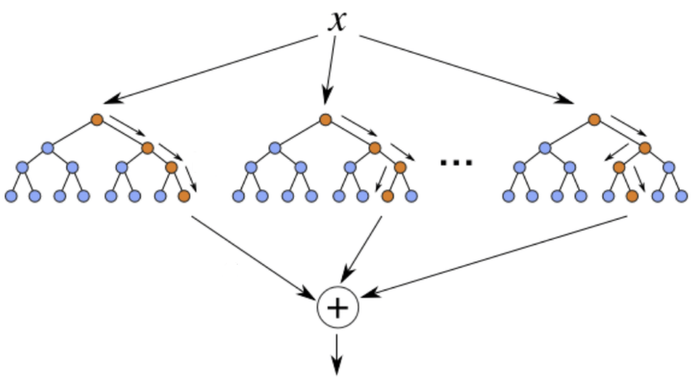

Figure 2.3. Schematic diagram of a random forest [3]

2.4.3. Extreme Gradient Boosting

Extreme gradient boosting(XGB) is an efficient and scalable implementation of gradient boosting framework [13]. Underlying principle of gradient boosting is to transform base learners, decision trees for XGB, into strong learners. During the iterations of this process, the algorithm increases the weights of the instances which were not correctly classified by the base learner. More specifically, at each iteration, a sample data is randomly selected without replacement from the full training set. This sample is to used to train the base learner and the model update for the current iteration is computed. This methodology improves the robustness of the learners.

XGB is capable of learning non-linear, complex decision boundaries and it is currently one of the most commonly used machine learning methods for regression and classification problems.

2.4.4. k-Nearest Neighbor

kNN is an instance-based learning algorithm and it is one of the simplest among machine learning methods. It assumes that all instances in the data belong to an n-dimensional space where n denotes the number of attributes. Using the standard Euclidean distance measure, k nearest neighbors of a data point are specified [11]. A new data point is classified by a plurality vote of its neighbors and assigned to the most common class among its nearest neighbors, as depicted with an example in Figure 2.4. k is pre-determined and an essential parameter of the algorithm. In general, if k is increased, the bias increases while the variance decreases.

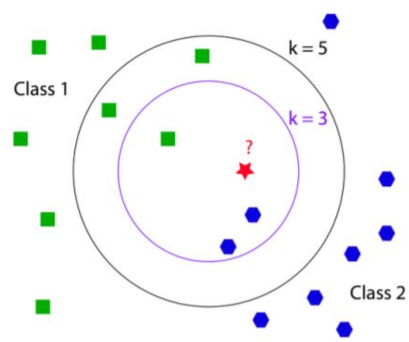

Figure 2.4. k-NN example for classifying the red star: Both 1-NN rule and 3-NN rule classify it as blue hexagon, while 5-NN rule classifies it as green square. [14]

2.4.5. Support Vector Classifier

Support Vector Classifier(SVC) method separates data points into one of the two corresponding classes through linear regression and hinge loss, as shown in Figure

2.5. SVC tries to maintain the largest possible margin in separating the two classes. Instances on the margin are called as support vectors. That structure makes it more applicable to binary classification problems.

A strong aspect of SVC is that they are not restricted to being linear. Using kernel trick, they can also distinguish different non-linear decision boundaries. They are effective in high dimensional spaces but they can perform poorly when the number of features for each sample exceeds the number of training data samples.

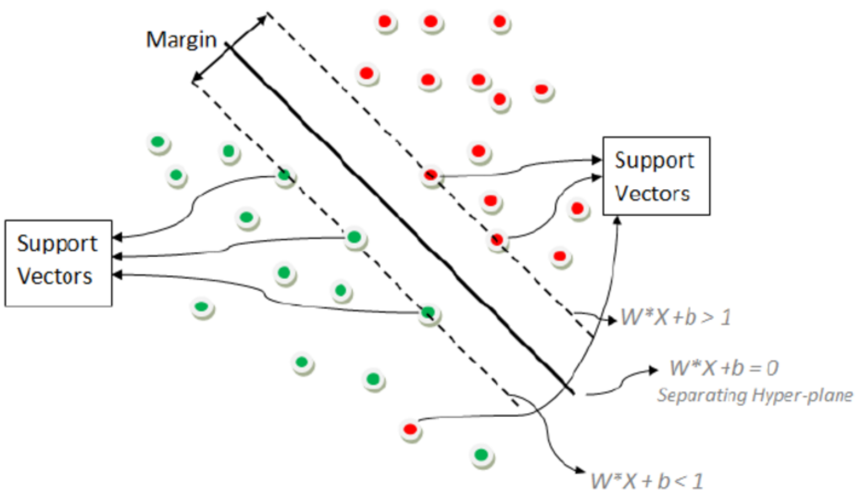

Figure 2.5. Support Vector Classifier [4]

2.4.6. Artificial Neural Network

Artificial neural networks (ANN), which have been inspired by the biological learning systems, provide a general method for learning real, discrete and vector valued functions from examples [11].

The parameters and weights of the network are calculated using back-propagation algorithm. It uses gradient descent to update parameter values after training instances and their corresponding classes are processed. Back-propagation algorithm minimizes the error between the actual answer and guessed answer. Figure 2.6 shows the general architecture of an ANN, which consists of an input layer, hidden (computation) layers and an output layer.

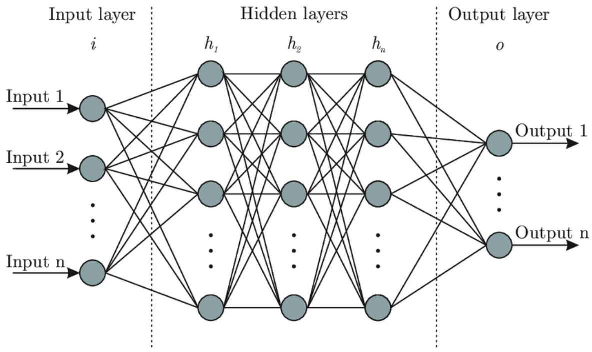

Figure 2.6. Artificial neural network architecture [5]

3. RELATED WORK

Social media platforms enable researchers to access opinions declared by thousands of people about a topic they are searching for. Therefore, the number of studies that use social media data has been increasing in the last decade. The data retrieved from social media can reflect the volume of people's interest or the sentiment of general opinion on a topic. An interesting and popular area among the studies is to use the data for predicting the future. There are many applications that use the social media data to make predictions in the fields of sports, politics, biology, economics, etc. An example research tries to discover patterns between public sentiment and real stock price movements [15]. Another one tries to analyze political sentiment and use this analysis to make predictions for the future elections [16].

Predicting Box Office revenue for movies is a popular area and there are many studies in the area that use applications of different prediction methods. In his study, Matt Vitelli used movie features like MPAA rating, release date, and movie genre for the predictive model. Actor-actor relationship graphs, actor-movie relationship graphs, and movie-movie relationship graphs are also added to the computational model for movie revenue prediction and were able to get more accurate results than using the meta-data only [17]. In one of the earliest studies, Jeffrey S. Simonoff and Ilana R. Sparrow tried to categorize movies as blockbusters and sleepers [18]. They used linear regression models that incorporate some or all of the variables like genre, MPAA rating, number of best actors, number of top dollar actors, and a Boolean feature showing if the movie was a summer release. They retrieved 73.5% R2 score for the movies opening on 10 or less screens and 96.6% R2 score for the movies opening on more than 10 screens.

As an interesting study, in 2017 Yao Zhou, Lie Zhang and Zhang Yi proposed a method to predict box-office revenues [19], which employed multi-modal deep neural networks on movie poster features. They used a Convolutional Neural Network (CNN) to extract features from movie posters. This approach achieved 52.2% accuracy among 3807 sample movies, which were categorized into six groups according to their revenue.

Sharda and Delen have trained a neural network to classify movies into nine categories according to their expected revenue, ranging from bad to good [20]. For test samples, the neural network classified 36.9% of the movies correctly, but 75.2% of the movies were one category away from correct.

Some studies had the idea of adding the user-generated data to the classical factors in their models and increase the success rate. To predict a movie's box office success, one example study used the activity level of editors and viewers of the corresponding entry to the movie in Wikipedia [21]. They used multivariate linear regression model with different set of variables and the model reached 94.0% R2 score a few days before the movie release.

P. Thomas Barthelemy, Devin Guillory, and Chip Mandal used twitter data for the prediction [22]. They hypothesized that an increased number of tweets about a movie before its release will result in increased box-office revenue. It was a challenge for them to identify tweets that are related to a movie, since the hash-tags on twitter did not yet exist. They used Naive Bayes classification for this task. Since hash-tags now exist, this task is easier but Naive Bayes classification can still be helpful to extract unrelated tweets in hash-tags. Then, they used regression for predicting the box-office revenue. Liu et al. focused on predicting movie box-office revenues by exploiting large-scale social media content [23]. They made use of Linear Regression and Support Vector Regression in predicting the box-office revenue of a movie before its release. Their experimentation has showed that a movie's box-office success is correlated with its content in the social media and this information can be useful to make more accurate predictions.

Another study used a multivariate linear regression model that combined metadata with text features from pre-release critiques; they did not use the reviews made after the movie is released [24]. Hollywood was not the only target market of the study. Dipak Damodar Gaikar and Bijith Marakarkandy worked on classifying Bollywood movies into three categories using twitter data: hit, flog, average [25]. A group of researchers from Princeton University examined if the comments on twitter and other social sites represent the public opinion [26]. Their conclusion was that scores computed from Twitter reviews and other on-line sites do not necessarily turn into predictable box-office figures. However, Asur and Huberman set up a linear regression model for the revenue of movies based on the volume of Twitter mentions [27]. The prediction model included parameters like rate of attention seeking, polarity of sentiments and reviews, and distribution parameter. For the first week revenue of 24 movies they used as sample, they obtained adjusted coefficient of determination of 0.97 on the night before the movie is released.

Another neural network approach was used by T. G. Rhee and F. Zulkernine [28]. They proposed a back-propagation neural network model for predicting the box-office success by classifying them as “bomb” or “flop”. They validated the approach using cross-entropy validation and the final results showed 91% accuracy in the prediction. User critics from movie databases are also included in their model but they did not consider the large-scale user reviews from twitter and the sentiments of the tweets. In a work applied on Korean movies [29], Minhoe Hur, Pilsung Kang and Sungzoon Cho divided the factors that affect movie audiences into three categories. Apart from motion picture factors like actor, director information, mpaa rating and external factors like screen ratio and seasonality; one category consisted of audience factors coming from the user reviews. In order to show variable importance in predicting box-office revenues, they applied CART, ANN and SVR based forecasting algorithms and an independent subspace method.

In 2014, M. Ghiassi, David Lio and Brian Moon presented a model to forecast movie revenues during the pre-production period based on a dynamic artificial neural network model, where the number of hidden layers is not priorly fixed [30]. In the first part of the study, they used a similar data set and features as used in Sharda and Delen’s study in their DAN2 model. Similarly, they grouped revenues into nine different ranges and trained their model to solve this multi-class classification problem. The results demonstrated the effectiveness of this approach by showing an accuracy improvement of 32.8%. In the second part, they added new features to their model like production budget and pre-release advertising expenditures. In the final version of their work, they used the data of 1758 movies and used the following features in the model: mpaa rating, binary information indicating if the movies is sequel, number of screens, production budget, run time and seasonality. Results of the study claim that they arrive accuracy values of 94.1%. Since the data set or the list of movies used to train/test their models are not available, it is hard to compare with other approaches but the reported accuracy of their study seem to be the highest in the literature.

In their work to examine consumer engagement behavior and movie box-office in social media [31], Chong Oh, Yaman Roumani, Joseph K. Nwankpa, Han-Fen Hu used user activities coming from Facebook, Twitter and YouTube to predict opening-weekend box-office gross revenue. They retrieved the following features from social media platforms: From Twitter, number of tweets posted for each movie and number of tweets posted about the movie. From Facebook, number of likes for the movie and number of profiles mentioning the movie profile. From YouTube, number of views of the trailer posted by the movie and number of comments posted by public. Their findings show that count of twitter public tweets are positively related with movie performance; but, this association weakens when Facebook and YouTube features are included in the model. On the other hand, this study also suggests that Facebook and YouTube based features are key indicators of movie's future economic performance. However, they did not use sentiments of YouTube comments which will be one of the contributions of our study.

4. DATA SET

Even though there are many existing data sets with various features on box-office forecasting, we were not able to identify any publicly available data set that fulfilled our needs, especially with the YouTube reviews. Therefore, we created our own comprehensive data set, which includes movie master data (such as director, cast, genre, IMDB rating), financial figures, and YouTube reviews. This has been shared as an open data set for the usage of other researchers [9]. In the remainder of this section, each of the main components of our data set will be explained in further detail.

4.1. Movie Master Data

IMDb (Internet Movie Database) is an online database of information which is mostly related to films and television programs [32]. It contains various information like movie cast, biographies, user reviews and ratings. It is a freely available and very rich source of content having a specific and detailed page for each movie. We started data preparation from searching movies using IMDb “Advanced Title Search” service, which enables users to look up movies according to wide variety of filters ranging from title, release date and genre to production company, country and language.

Among all available filters, we have chosen four different filters to apply. Figure 4.1 shows these filters and some of the available options for each them. As title type, we only selected “feature film”, also known as motion picture or movie, which is a sole film to fill a program with long enough running time [33]. Other than that, we were only interested in the English-spoken movies that were filmed in the USA, released between the beginning of 2000 and the end of 2018.

Figure 4.2 shows an overview of a search results page. By iterating over all pages, we obtained the list of movies to be used in the study. During this iteration, we retrieved the following information, which is available on the movie summary grid: Motion Picture Association of America (MPAA) film rating, genre, user ratings and

Figure 4.1. IMDb - Advanced Title Search Filters

number of votes, run time, director and top actor list. However, these summary grids did not include all the information we needed; therefore, we examined the movie-detail pages after another search for each movie. Figure 4.3 shows an example movie page. From those pages specific to each movie, we retrieved release date information and some box-office data like budget and gross revenue but this was not the only source we used to get the box-office statistics. Also, IMDb has a separate, public page showing the list of movies with sequel [34]. We utilized this page to retrieve a feature of movies indicating whether it is a sequel movie or not.

4.2. Financial Data

Since our motivation is to predict if a movie will make a positive profit after its release, it is crucial to have robust information on the financial facts about the movies. Again, IMDb provides some financial numbers but they are not available for all of the movies; that's why we looked for additional sources. We worked to retrieve four new features in this step: movie budget, movie domestic gross revenue, number of screens that will show the movie and general box-office seasonality information. The-Numbers.com tracks box-office numbers in a systematic way and then publishes these financial performance records [35]. As they note on their website, it is hard to find reliable budget data since film-makers might be reluctant to share this information or use some tricks to inflate or reduce numbers. Considering this, we have accepted this website as a reliable source of content since it is used in other studies as well [30], [19].

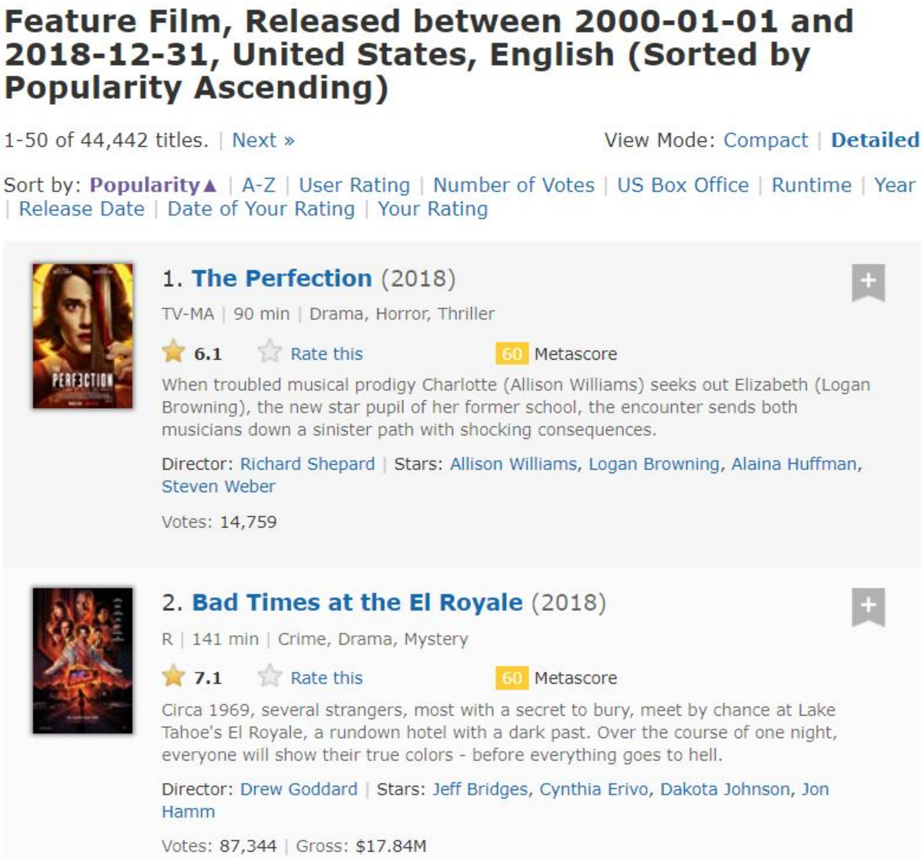

Figure 4.2. IMDb - Advanced Title Search Sample Results

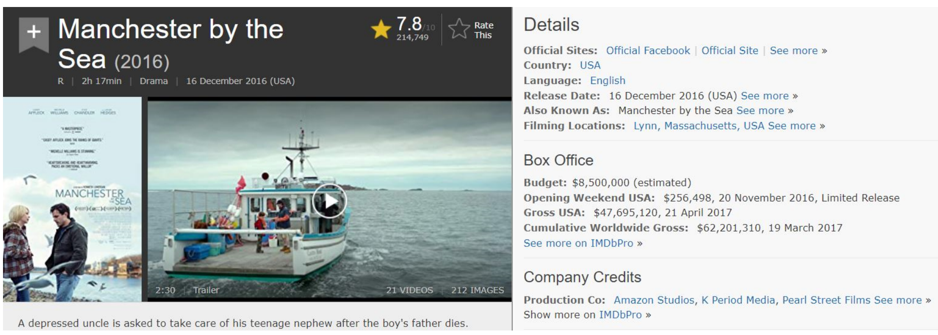

Figure 4.3. IMDb - Example Movie Page

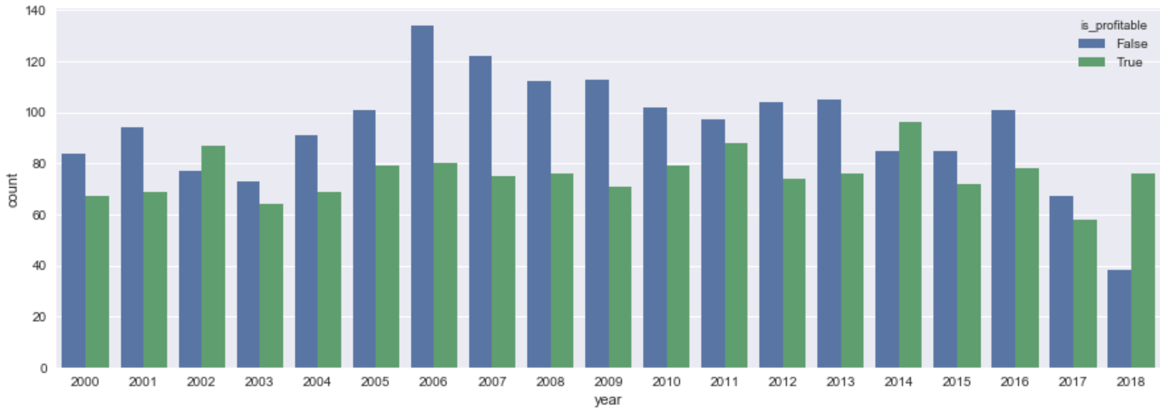

Figure 4.4. Yearly number of movies in the dataset which make profit or lose money

To extend the number of movies for which we have the financial data, we combined the numbers coming from IMDb and The-Numbers.com. We have chosen IMDb as the primary source, meaning that, if these sources have different values for a movie, we used the one coming from IMDb. There were some movies that exist in one and not in other; however, the numbers were close for the ones that exist in both. After retrieving the movies' budget and revenue data, we analyzed their profitability. Table 4.1 shows the number of movies in the data set making profit or losing money in each year between 2000 and 2018 and Figure 4.4 visualizes these values.

Table 4.1. Yearly number of movies in the dataset which make profit or lose money

Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	All
Profit (positive)	67	69	87	64	69	79	80	75	76	71	79	88	74	76	96	72	78	58	76	1434
Profit (negative)	84	94	77	73	91	101	134	122	112	113	102	97	104	105	85	85	101	67	38	1785
All	151	163	164	137	160	180	214	197	188	184	181	185	178	181	181	157	179	125	114	3219

Box Office Mojo is an online box-office reporting service, which is owned and operated by IMDb. It is accepted as one of the main data sources in the film industry. This data source provides the number of theaters that show a given movie [36]. We have retrieved this information, which we call screen counts, for all movies in our data set.

We also utilized seasonal box-office records provided by Box Office Mojo [37], having some information like number of days in a season, number of movies released in that period, total gross revenue and daily average ticket sales revenue. Details of how we use these season-by-season sale data in our models are explained in the Feature Analysis section.

4.3. YouTube Data

YouTube is a social video-sharing platform, which allows users to upload, view and share videos [38]. YouTube users can subscribe to other users and react to other videos by commenting on them or clicking on the like/dislike buttons. Music videos, short and documentary films, movie trailers, live streams are some examples of the available content on YouTube. Most of this content is provided by individuals, whereas some of it can be uploaded by companies or institutions.

In our study, we focus on extracting meaningful information from the YouTube users' reaction to movie trailers and using that information to see if we can have more accurate box-office forecasts; therefore, YouTube user reaction data is a key data for our work.

In order to retrieve YouTube data, we have benefited from YouTube Application Programming Interface (YouTube API), which allows developers to retrieve YouTube data and video statistics via REST calls. In order to use YouTube API, we had to have a developer id which must be attached to a YouTube account. This API has some quotas that limit the number of calls a developer can make.

To find movie trailer videos uploaded to YouTube, using the API, we searched for movie name with the word "trailer" for all of the movies and chose the one watched the most. Figure 4.5 shows an example movie trailer page we access after the search. Figure 4.6 shows some user comments added under a video. A comment consists of user information, comment text, replies, and the number of likes and dislikes it receives.

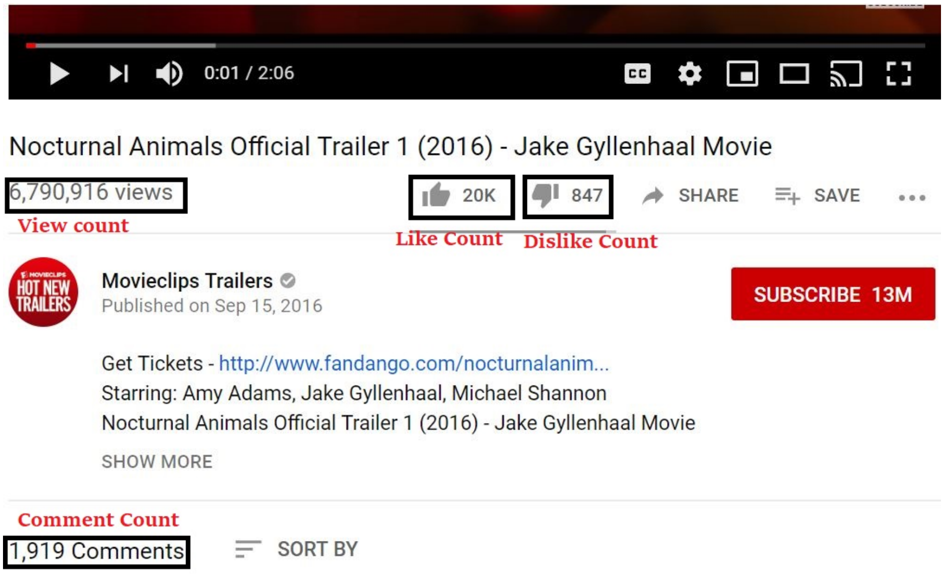

Figure 4.5. YouTube - Example Movie Trailer Page

Figure 4.6. YouTube - Example Movie Trailer Page - Sample Comments

For each YouTube movie trailer, we retrieved the following information:

- View Count
- Like Count
- Dislike Count
- Comment Count
- User Comments

For each comment on a video, we retrieved the following information:

- User Information
- Comment Text
- Publish Date & Time
- Number of likes the comment received

After YouTube data retrieval, data preparation step was complete. At the end, we had 3,219 movies and 5,846,083 YouTube user comments made on these movies' trailers. We have filtered out the movies for which some of the required information was not available; therefore, there is no missing values in our final data set.

5. METHODOLOGY

5.1. Sentiment Analysis on User Reviews

Before starting with the actual box-office forecasting, we need to extract new features from user reviews showing the polarity of their opinions, i.e., perform sentiment analysis. In this study, we do not focus on developing a new sentiment analysis approach, but we rather utilize two existing approaches from the literature and integrate them in our box-office prediction model.

5.1.1. Naive Bayes Approach

We start this sub-section with a brief explanation of Naive Bayes Classification method.

Conditional probability and Bayes Rule are the key concepts of Naive Bayes. Conditional probability can be defined as the probability that an event will happen given that another event has happened. In our study, example of this would be a text's probability of being negative given that a word has occurred in that text.

Let N denote the event that a text is negative and W denote the event that a given word occurs in a text. The probability of N and W occurring together is calculated as follows:

$$P(N\&W) = P(N) \times P(W \mid N) \tag{5.1}$$

As in the example above, we often know the frequency of observing an evidence only when the outcome is given. Bayes Rule helps us to reverse this and we can compute the probability of observing an outcome when the evidence is given with the following formula:

$$P(N \mid W) = \frac{P(W \mid N)\, P(N)}{P(W)} \tag{5.2}$$

So far, we have only considered one evidence to estimate the outcome. In a real scenario, multiple evidences should be taken into consideration. Similarly, in order to calculate the probability of a text having a positive or negative sentiment, we should consider the probability of the occurrence of multiple words. Naive Bayes approach assumes that all of these evidences are conditionally independent and we can combine their probability to calculate the probability of the outcome as follows:

$$P\left(N|W\right) = \frac{\prod_{i=1}^{n} P\left(W_i|N\right) P\left(N\right)}{P\left(W\right)} \tag{5.3}$$

During this study, we have utilized Naive Bayes Classification of *TextBlob*, which is a textual data processing library implemented in Python. Let us now see how this implementation applies Naive Bayes approach to sentiment analysis problem.

- Training
  - (i) Previously annotated positive and review files are retrieved. The Natural Language Toolkit (NLTK), which is a platform that brings programs and libraries for Natural Language Processing (NLP), provides annotated text corpora for the usage of NLP studies [39]. The Naive Bayes based sentiment analysis approach that we used utilizes a corpus that contains more than two thousand movies with sentiment polarity classification.
  - (ii) A dictionary that contains word occurrences is created by extracting each

word from each document in the corpus.

(iii) By counting how many times each word occurred in each negative and positive files, two probability distributions are calculated: $P(sentiment)$, $P(word \mid sentiment)$.

- Prediction
  (i) When a new text comes to be analyzed, it is tokenized into sentences and words respectively.
  (ii) For each word in the text, the conditional probabilities, which were calculated in the training part, are retrieved and then multiplied as in Equation (5.3).
  (iii) The probabilities of having negative and positive sentiment are calculated separately.
  (iv) Finally, using these probabilities, a value which is normalized between -1 and 1 is returned.

Naive Bayes performs very well in certain applications and text classification is considered to be one of those areas. As an another advantage of this approach, the model was trained using movie reviews which is directly related to our study.

5.1.2. Lexicon and Rule Based Approach

The Second approach we have used for sentiment analysis is a lexicon and rule based model, named as Valence Aware Dictionary and Sentiment Reasoner (VADER) [40], which is specifically adapted to texts in the social media. This method is sensitive to not only positive-negative polarity but also intensity of emotion in a text. One main difference of VADER with the previously mentioned Naive Bayes approach is that it uses lexicon which is a dictionary that maps words to their sentiments. While machine learning-based approaches require data of previously annotated sentences in order to train a model and predict sentiment of the new coming sentences, lexical approaches do not need this kind of data.

Construction of this dictionary is very crucial for lexical approaches. In VADER, this step starts with examining already existing, human-validated sentiment lexicons. In addition to the words in these lexicons, a new list of words that comes from emoticons, acronyms and slang is compiled. This is a distinguishing aspect of VADER's lexicon and it helps to understand expressed sentiments in micro texts in the social media as they are commonly used. It considers sentiment of emoticons, smiley faces and utf-8 encoded emojis; for example, ":)" indicates a positive sentiment and ":(" denotes negative. It examines acronyms and words with initials that can relate to sentiment, for example "LOL" stands for "Laughing Out Loud". Also, commonly used slang words like "nah" and "meh" are considered.

After the list of features is expanded, wisdom-of-crowd approach is used and all of these lexical features are scored by human raters in terms of polarity and intensity. In the range from -4 to 4, these human raters score the polarity and intensity of each token feature, where they also have a chance to say that it is neutral. Intensity score of the token features shows how strong the feature is to express its sentiment. For example, the word "fine" has a positive valence of 0.8, "good" is 1.9, and "awesome" is 3.1. Scores coming from different raters are averaged. At the end of human rating and validation step, features with non-zero mean score and standard deviation less than 2.5 are kept and VADER has a new lexicon with 7,500 lexical features in it.

Lexical features are not the only factors that determine the sentiment of a text. Considering this, VADER applies five heuristics to make a better sentiment analysis. The effect of each heuristic is also scored by human raters.

- Punctuation. When exclamation mark is used in a sentence, it increases the intensity of sentiment. For example, "This trailer is bad!!!" has stronger negative sentiment than "This trailer is bad". Similarly, when a question mark is used in sentence, it changes the emphasis of the sentiment text and the sentiment score is updated. VADER first calculates the sentiment score of a sentence, then modifies the score according to usage of exclamation and question marks. The algorithm adds 0.292 point for each exclamation mark (up to 4) and 0.18 point for a question mark (up to 3). These values are obtained empirically and for the negative sentences they are subtracted from the sentence score.

- Capitalization. If a lexical feature is used all capitalized, that increases the valence of the sentiment. For example, "The opening scene was OUTSTANDING" has a stronger positive sentiment than "The opening scene was outstanding".
- Degree modifiers. Some words, called as degree modifiers, increase or decrease the intensity of lexical features. For example, whereas the word "completely" increases the positive sentiment of the sentence "This movie looks completely better than the previous one", the word "slightly" decreases the positive sentiment of the sentence "This movie looks slightly better than the previous one". As it can be seen from the example, degree modifiers can affect the sentiment in different directions and their magnitude of effect also changes from one word to the other. VADER keeps a dictionary of modifiers that boost or dampen the sentiment. Moreover, the effect of a modifier varies according to the distance it has to the word it modifies; distant modifiers have less effect.
- It checks for the usage of an oppositional conjunction "but", which shifts the polarity of a sentence. VADER decreases the intensity score of the words before "but" conjunction and increases the ones after it. Therefore, the second part of the sentence dominates the sentiment.
- It examines the tri-gram before lexical features to catch polarity negations as in the example "The trailer isn't really all that great".

The algorithm returns a "compound" score by summing up the intensity scores of each word in the lexicon, modifying the sum according to heuristic rules, and finally normalizing it between -1 and 1 with the Equation (5.4), where x is the sum of the sentiment valence scores and α is the normalization parameter. Recall that the valence scores in the lexicon changes from -4 to 4.

$$\frac{x}{\sqrt{x^2 + \alpha}} \tag{5.4}$$

We have used this compound score, ranging between -1 (most negative) and 1 (most positive), in our forecasting models.

5.1.3. Example Reviews

In this section, we look at some example movie trailer comments from YouTube in detail and explore the results of the sentiment analysis applied on them.

An important point to mention here is that we did not spare comments according to the comment date and movie release date. We have two reasons for that: First, since YouTube is a relatively new platform and only the movies released in the recent years have pre-release user comments, keeping only the comments made before the movie release would cause us to lose most of the movies in the data set. Secondly, if a user makes a comment to a trailer after the movie is released, it does not necessarily mean that the user has watched the movie. On the contrary, even after the movie is released, people keep watching the trailer before they go to theaters; therefore, we believe that the relation between reaction to trailer and watching the movie afterwards remains.

The first example is from the trailer of the movie “Godzilla”:

> *I know this is just a trailer, but this probably the best looking Godzilla movie I've seen. Seriously, some parts can be paused to make a beautiful wallpaper for my desktop. Props to the cinematographer.*

This sentence's polarity score was calculated as +0.505 by the Naive Bayes approach and +0.9022 by VADER. This is clearly a positive review but the latter score might be a bit extreme.

The second example is a little controversial even for human readers. It comes from the movie “The Mummy”:

> *Trailer looks awful, but Tom Cruise doesn't usually star in trashy movies... I hope this doesn't disappoint. Seems like Hollywood is running out of ideas with all the reboots. Next thing you know there's gonna be a Forrest Gump movie with*

a trailer identical to this one.

Whereas NB approach says that it is a negative review with -0.41667 score, VADER claims that it is positive with 0.4717 polarity score. The comment starts with a really bad review about the trailer but then the user shares a hope about the movie since he thinks that the main actor shows up at good movies.

Here is a similarly controversial example comment on the movie "Mother's Day":

Looks like a good movie, but who ever designed the trailer should be fired.

For this one, NB approach scores as 0.7000 and VADER scores as -0.4939. As we mentioned in the details of VADER approach, its algorithm gives more weight to the clause after the "but" conjunction. This explains why VADER has given positive and negative sentiments to the last two sentences, respectively.

In Table 5.1, more example comments and their sentiment analysis coming from both approaches are shown. Like column shows the number of likes the comment gets, and pol-nb and pol-vader columns represent polarity scores of Naive Bayes and VADER approach, respectively. After manually controlling the results, outputs of both sentiment analysis approaches looked reasonable and we decided to use features coming from both of them in our experimentation.

Table 5.1 Example YouTube Comments and Their Scores Analysis

Movie Name	**Comment**	**Like**	**pol-nb**	**pol-vader**
Godzilla	I know this is just a trailer, but this probably the best looking Godzilla movie I've seen. Seriously, some parts can be paused to make a beautiful wallpaper for my desktop. Props to the cinematographer.	1981	0.505	0.902

Table 5.1 Example YouTube Comments and Their Sentiment Scores (cont.)

Movie Name	Comment	Like	pol-nb	pol-vader
The Mummy	Trailer looks awful, but Tom Cruise doesn't usually star in trashy movies... I hope this doesn't disappoint. Seems like Hollywood is running out ideas with all the reboots. Next thing you know there's gonna be a Forrest Gump movie with a trailer identical to this one.	24	-0.416	0.471
Mother's Day	Looks like a good movie, but who ever designed the trailer should be fired.	1672	0.700	-0.493
Logan	Good choice of music, Johnny Cash, the legend. Makes the trailer even better!	1	0.662	0.726
Jurassic World: Fallen Kingdom	Worst trailer ever. They chose the worst scenes. I thought it was fan-made.	584	-1.000	-0.848
Pacific Rim	Goddamn this is like the best trailer. Already own the movie and here I am watching the trailer just because the awesome.	693	0.866	0.827
Sherlock Holmes	This will get absolutely terrible ratings	350	-1.000	-0.525
Oblivion	Not a very good trailer but seems like a popcorn flick and I like Tom Cruise. I'll probably check it out.	0	-0.269	0.689
Spider-Man: Homecoming	Okay, this trailer is good and all, but nothing will ever top the trailer for the 1978 movie, Spider-Man Strikes Back.	423	0.425	-0.409

Table 5.1 Example YouTube Comments and Their Sentiment Scores (cont.)

Movie Name	**Comment**	**Like**	**pol-nb**	**pol-vader**
Black Swan	Amazing trailer! Love natalie... I heard that Vicent is a former dancer, can't wait to see him in this film.	525	1.000	0.509

5.2. Feature Analysis

In this section, we go over all of the features that we include in our study in order to increase the prediction accuracy. In the first part, we explore the features that are directly related to movies, which we call *movie master features.* These features have been previously used in different studies [30], [20], [29]. In the second part, we explain all of the features we generated from YouTube in detail. After conducting the experiments, we will see that only some of them have a positive effect on the forecasting results.

5.2.1. Movie Master Features

Even though the data set that we prepared has information about the director and top actors of the movies, we did not include these attributes in our study because M. Ghiassi, David Lio and Brian Moon's work [30], which reports the highest accuracy results in the literature, has showed that a movie's star power is highly correlated with its production budget and we already consider that information. Also, despite the fact that IMDb user vote numbers and movie rating information are correlated with a movie's profitability, we have excluded these features because this information is not actually available until the movie is released and watched by many people, while our focus in this study is to predict a movie's success before the release.

In the following we examine the movie features one by one.

(i) *MPAA Rating (mpaa)*

This is a categorical feature that indicates a rating given by Motion Picture Association of America to each movie. The movies in our data set have the following distinct ratings: 'PG-13', 'PG', 'R', 'G', 'NC-17'. Some of them did not have any rating so we labeled them as 'Not Rated'. Definitions of these rating codes are given below [41]:

- G: General audiences.
- PG: Parental guidance is suggested because might be inappropriate for children.
- PG-13: Parents are urged to be cautious because some material might not be suitable for children under 13.
- R: Restricted. Under 17 requires an adult as companion.
- NC-17: Adults only. No one 17 and under is admitted to watch.

Table 5.2 shows the number of movies that make profit or lose money for each MPAA rating. We can see from the table that 60.2% of the movies with rating R have negative profit, which might be an indicator of a correlation between MPAA ratings and the movie profitability.

Table 5.2. Number of movies in the dataset by their MPAA rating

MPAA	G	NC-17	Not Rated	PG	PG-13	R	All
Profit(Positive)	27	0	79	220	546	562	1434
Profit(Negative)	20	1	145	186	582	851	1785
All	47	1	208	406	1128	1413	3219

(ii) *Budget (budget)*

This feature shows the production budget of movies. It is an indispensable factor in the determination of whether a movie makes positive profit or not. Figure 5.1 shows the distribution of budget values for both profitability groups. We can observe from the figure that the two groups have different distributions. The movies with positive profit tend to have a higher budget, in all cases of comparison

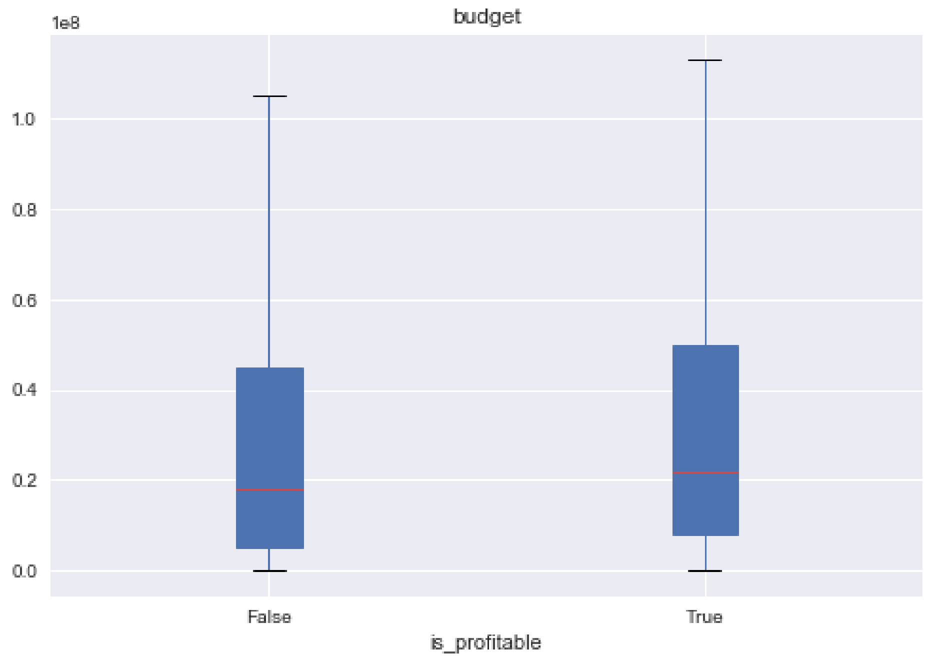

Figure 5.1. Box plots showing the distribution of budget values among movies

among the two, considering the minimum, maximum and median values.

(iii) *Seasonality (seasonality)*

A study of L. Einav on seasonality in the US motion picture industry shows that the number of people going to theaters change dramatically during the year [42]. Especially during the holiday weekends, movies with high budgets are released and that increases the box-office numbers during those times. As in the mentioned study, we consider major holidays in the USA and split the year into five different seasons as follows:

- Spring: A period that starts with the first Friday in March and ends with the Thursday before the first Friday in May.
- Fall: A period that starts with the day after Labor Day weekend and ends with the Thursday before the first Friday in November.
- Winter: A period that starts with the first day after New Year's week or weekend and ends with the Thursday before the first Friday in March.
- Summer: A period that starts with the first Friday in May and ends with Labor Day weekend.
- Holiday: A period that starts with the first Friday in November and ends

with New Year's week or weekend.

We have first determined the season of a movie by its release date and then calculated its seasonality by averaging the daily average gross revenues of movies in that season using the numbers from previous years. Table 5.3 shows season-by-season average gross revenues per movie in the years between 2010 and 2018. Using average revenues coming from all seasons, we scale each value between 0 and 1. Table 5.4 shows the final seasonality effect values we used in our study.

Table 5.3. Season-by-season average gross revenues per movie (dollars in millions) [7]

Year	**2010**	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**	**2018**
Holiday	29.0	25.8	27.6	25.7	25.5	27.4	27.4	30.7	18.7
Fall	10.1	9.0	8.7	7.8	9.6	9.6	7.5	7.5	8.3
Summer	22.7	21.3	19.2	20.9	17.6	17.6	17.9	15.2	14.4
Spring	17.5	14.2	12.7	10.7	11.9	12.4	12.2	14.7	12.3
Winter	13.7	12.1	14.1	11.9	10.7	10.9	13.3	11.6	12.6

Table 5.4. Seasonality Effects

Season	**Effect**
Holiday	0.33389
Fall	0.10966
Summer	0.23420
Spring	0.16652
Winter	0.15571

(iv) *Is Sequel (is-sequel)*

This is a Boolean value indicating whether a movie is continuation of a previous movie. In our data set, 380 out of 3219 were sequel movies.

(v) *Run Time (run-time)*

Run time is the duration of a movie in minutes. We do not expect this value to be really important but very large or small values of this feature might affect the decision of people to watch a movie or theaters to show it.

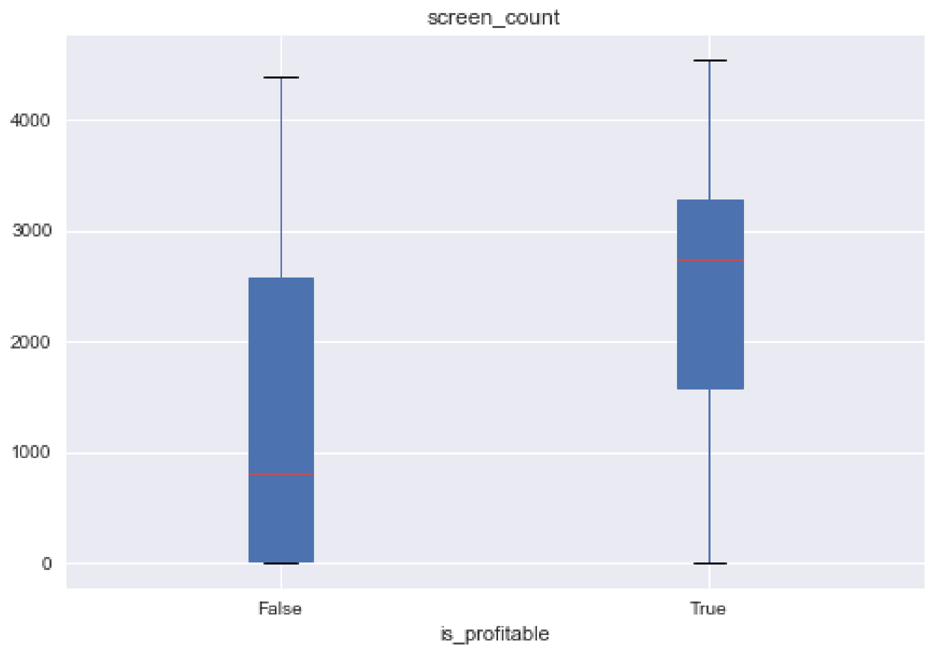

Figure 5.2. Box plots showing the distribution of screen count values among movies

(vi) *Screen Count (screen-count)*

This feature shows the total number of theaters showing a movie. Before the movie release, this is typically planned by the movie exhibitors, distributors and the theaters. It is reasonable to expect that the more theaters show a movie, the more money the movie makes, but the important question is about whether the screen count affects the movie's profitability. Figure 5.2 shows the huge difference between screen count distribution of the movies that make money and lose money. This looks like a promising feature for our forecasting model.

(vii) *Genre (genre)*

Genre information shows the category of a movie depending on its story, character, plot and setting. The list of movie genres in our data set is as follows: 'Action', 'Adventure', 'Animation', 'Biography', 'Comedy', 'Crime', 'Drama', 'Family', 'Fantasy', 'Horror', 'Music', 'Musical', 'Mystery', 'Romance, 'Sci-Fi', 'Thriller', 'War', 'Western'. A movie has typically more than one genre.

(viii) *Year (year)*

This feature tells which year a movie is released. Movie box-office revenue changes year by year and this information can be useful to capture this yearly trend in our forecasting model.

5.2.2. YouTube Features

We have used 19 different YouTube features in our experiments. While some features (like-count, dislike-count, view-count and comment-count) were readily available in the raw data, we had to do some calculations to generate the rest.

(i) *Like Count (like-count)*

Number of likes a movie trailer gets. It shows positive reactions towards the movie.

(ii) *Dislike Count (dislike-count)*

Number of dislikes a movie trailer gets. It shows negative reactions towards the movie.

(iii) *Like Ratio (like-ratio)*

Ratio of the number of likes and the number of dislikes a movie trailer gets. For like-count and dislike-count, it is normal to see higher values as the trailer is watched more. However, like-ratio does not directly depend on the trailer popularity and considers both positive and negative reactions together.

(iv) *View Count (view-count)*

Shows how many times a movie trailer is watched.

(v) *Comment Count (comment-count)*

Shows the number of comments made to a movie trailer.

(vi) *Polarity Score - Naive Bayes Approach (polarity-nb)*

Overall polarity score of a movie, calculated by averaging all YouTube comment polarity scores. Scores are retrieved with Naive Bayes sentiment analysis approach. Its values can change between -1 (most negative) and 1 (most positive).

(vii) *Weighted Polarity Score - Naive Bayes Approach (w-polarity-nb)*

In the YouTube data we retrieved, there is also the information of the number of likes each comment gets. Comments with a large number of likes show that it is endorsed by many people. Considering this, we have generated a weighted polarity score feature, which weigh the comments by their number of likes as in Equation (5.5). *w-polarity-nb* feature is based on Naive Bayes approach's polarity scores.

$$weighted\text{-}polarity = \frac{\sum_{i=1}^{n} polarity(c_i) \times (like(c_i) + 1)}{\sum_{i=1}^{n} like(c_i) + 1} \tag{5.5}$$

where c_i denotes i^{th} comment made on a movie.

(viii) *Positive Comment Count - Naive Bayes Approach (count-pos-nb)*
Represents the number of positive comments made to a movie trailer. For this feature, a comment is considered as positive if polarity score of Naive Bayes approach is greater than 0.

(ix) *Negative Comment Count - Naive Bayes Approach (count-neg-nb)*
Represents the number of negative comments made to a movie trailer. For this feature, a comment is considered as negative if polarity score of Naive Bayes approach is less than 0.

(x) *Polarity Score - Lexicon and Rule Approach (polarity-vader)*
Overall polarity score of a movie, which is calculated by averaging all YouTube comment polarity scores. Scores are retrieved with VADER sentiment analysis approach. Its values can change between -1 (most negative) and 1 (most positive).

(xi) *Weighted Polarity Score - Lexicon and Rule Approach (w-polarity-vader)*
This feature has the same idea with w-polarity-nb feature but this one is based on the VADER approach's polarity scores.

(xii) *Positive Comment Count - Lexicon and Rule Approach (count-pos-vader)*
Represents the number of positive comments made to a movie trailer. For this feature, a comment is considered as positive if polarity score of VADER approach is greater than 0.

(xiii) *Negative Comment Count - Lexicon and Rule Approach (count-neg-vader)*
Represents the number of negative comments made to a movie trailer. For this feature, a comment is considered as negative if polarity score of VADER approach is less than 0.

(xiv) *Average Polarity Score (polarity-avg)*

This value is the average of polarity score features polarity-nb and polarity-vader.

(xv) *Weighted Average Polarity Score (w-polarity-avg)*

This value is the average of weighted polarity score features w-polarity-nb and w-polarity-vader.

(xvi) *Positive Negative Ratio - Threshold:0.0 (pos-neg-ratio)*

It is the ratio of the number of positive comments and the number of negative comments made for a movie trailer. In this calculation, the average of polarity scores retrieved from both sentiment analysis approaches is used. For pos-neg-ratio feature, a comment has been considered as positive if its polarity score is greater than 0 and negative if the score is less than 0.

We have generated three more features that share the idea of using the ratio of positive and negative comment numbers. Their difference is that we use non-zero thresholds to consider a comment as positive or negative. In other words, we extend the neutral zone and use more extreme comments in the ratio calculation.

(xvii) *Positive Negative Ratio - Threshold:0.1 (pos-neg-ratio1)*

Comments with average polarity score greater than 0.1 are considered as positive, whereas the ones with polarity score less than -0.1 are considered as negative.

(xviii) *Positive Negative Ratio - Threshold:0.3 (pos-neg-ratio3)*

Comments with average polarity score greater than 0.3 are considered as positive, whereas the ones with polarity score less than -0.3 are considered as negative.

(xix) *Positive Negative Ratio - Threshold:0.5 (pos-neg-ratio5)*

Comments with average polarity score greater than 0.5 are considered as positive, whereas the ones with polarity score less than -0.5 are considered as negative.

5.3. Evaluation

In this study, we are dealing with a binary classification problem in which we try to predict if a movie will make a positive profit; in others words, if its total box-office gross revenue in the USA will exceed its total production cost or not. Therefore, we use the percentage of correctly classified movies as the evaluation metric of our models

(5.6):

$$\frac{M_p + M_l}{M} \tag{5.6}$$

where M_p denotes the number of correctly classified positive-profit movies, M_l denotes the number of correctly classified negative-profit movies, and M denotes the total number of movies in the data set.

To validate the results of experiments, we apply 10-fold cross validation approach with 5 repetitions.

5.4. Model Selection

In our experimentation, we include different sets of features and compare the prediction results of each set. Evaluating all of these feature sets on different forecasting models would be time-consuming. For this reason, we wanted to find the best-performing forecasting model and conduct experiments with that one.

Model selection aims to find the best forecasting model and the best set of hyper-parameters. In order to do this, we have applied grid search algorithm in which we try all possible combinations of hyper-parameters for each forecasting model. For each set of hyper-parameters, we have used cross validation and used the average accuracy of the folds as the score of a particular set. In order to limit the number of iterations, we applied 5-fold cross validation instead 5 $\times$ 10-fold cross validation for the model selection part. In this step, we have only included the movie master features and built the following forecasting models with the given hyper-parameters. The Performance metric was *accuracy* for all models.

For DT, kNN, SVC and RF algorithms, we have used the implementations of the scikit-learn library [43].

- *Decision Trees*
  (i) Max Depth: Determines the maximum depth of a decision tree and therefore controls the size of the tree in order to avoid over-fitting.
  Possible parameter values: [4, 5, 6, 7, 8]
  (ii) Max Features: Controls the maximum number of features to consider when finding the best split.
  Possible parameter values: [4, 5, 6, 7, 8]
- *k-Nearest Neighbor*
  (i) Number of neighbors: The number of neighbors to use in classification decisions.
  Possible parameter values: [4, 5, 6, 7, 8]
- *Support Vector Classifier*
  (i) Kernel: Specifies the type of kernel used in the algorithm.
  Possible parameter values: ['linear', 'rbf', 'poly']
  (ii) C: Penalty parameter of the error term. High values of C might result in over-fitting.
  Possible parameter values: [0.001, 0.01, 0.1, 1, 10]
  (iii) Gamma: As the gamma value increases, the model tries to fit the training set.
  Possible parameter values: [0.001, 0.01, 0.1, 1]
  (iv) Degree: Degree of the polynomial, not used for other types of kernels.
  Possible parameter values: [2, 3, 4]
  (v) Class Weight: Sets the C parameter according to class weights.
  Possible parameter values: ['none' (all classes have weight one), 'balanced' (gives more weight to less frequent classes)]
- *Random Forest*
  (i) Number of estimators: Number of trees in the forest
  Possible parameter values: [50, 200, 500]
  (ii) Max Features: Controls the maximum number of features to consider when finding the best split.
  Possible parameter values: ['auto', 'sqrt' (square root of the number of fea-

tures) 'log2'(log of the number of features)]

(iii) Max Depth: The maximum depth of each tree

Possible parameter values: [4, 5, 6, 7, 8]

(iv) Criterion: Name of the function to evaluate the quality of tree splits

Possible parameter values: ['gini', 'entropy']

- *Extreme Gradient Boosting*

We have used an open-source Python implementation of XGB algorithm [44].

(i) Min Child Weight: Represents the minimum sum of weights of all instances needed in a child. As this value increases, the algorithm gets more conservative and avoids over-fitting.

Possible parameter values: [1, 5, 10]

(ii) Gamma: It is a regularization parameter. Determines how much loss reduction is needed to make a split in the trees.

Possible parameter values: [0.5, 1, 1.5, 2, 5]

(iii) Sub-sample: Ratio of training instances to be used when building trees. Small values might prevent over-fitting.

Possible parameter values: [0.6, 0.8, 1.0]

(iv) Column Sub-sample by Tree: Ratio of columns to be used when building trees.

Possible parameter values: [0.6, 0.8, 1.0]

- *Artificial Neural Network*

To build ANN, we have used the scikit-learn wrapper of Keras, an open source deep learning library implemented in Python [45]. We built a feed-forward neural network with two hidden layers.

(i) Epochs: Determines how many times the network will iterate over the entire data set.

Possible parameter values: [5, 10, 50]

(ii) Batch Size: Determines the number of instances in the dataset to use before updating the weights.

Possible parameter values: [32, 64]

(iii) Learning rate: Determines how much to update the network weight at the end of each batch.

Possible parameter values: [0.01, 0.1, 1]

(iv) Momentum: Determines how much the previous update will affect the current update of the network weights.

Possible parameter values: [0.5, 0.7, 0.9]

(v) Optimizer: Optimization method.

Possible parameter values: ['rmsprop', 'adam']

Table 5.5 shows the best cross-validation scores of each forecasting model and the corresponding set of hyper-parameters. Considering these results, we have decided to continue with the Extreme Gradient Boosting (XGB) model and its best parameters.

Table 5.5. Grid Search Cross-Validation Scores of Forecasting Models

Model	**CV Accuracy**	**Best Parameters**
XGBClassifier	0.795	[{'colsample_bytree': 1.0, 'gamma': 1, 'min_child_weight': 5, 'subsample': 0.8}]
KerasClassifier	0.770	[{'epochs': 50, 'batch_size': 32, 'optimizer': 'adam', 'learning_rate':0.1, momentum: 0.7}]
RandomForestClassifier	0.773	[{'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'n_estimators': 500}]
SupportVectorClassifier	0.766	[{'C': 10, 'class_weight': 'balanced', 'degree': 2, 'gamma': 0.1, 'kernel': 'rbf'}]
DecisionTreeClassifier	0.756	[{'max_depth': 6, 'max_features': 4}]
KNeighborsClassifier	0.693	[{'n_neighbors': 4}]

6. EXPERIMENTS AND RESULTS

In this chapter, we explain our experiment design and then share our classification results. Since there was no open data set and we have created a completely new one, we could not directly compare our outcomes with the previous works in the literature. The goal of our experimentation is to see if YouTube features can make a statistically significant contribution to the predictive power of the models that forecast movies' box-office success. In our experiments, we compare the cross validation results of the models containing only the movie master features and the models containing movie master features and YouTube features together.

Ideally, we would like to apply an exhaustive feature selection method, in which we try all combinations of the features we have and compare the model accuracy for each feature combination. However, considering the fact that we have twenty-six different features, computation time of training models for all combinations of the features and evaluating results using 5 $\times$ 10-fold cross validation would be enormous. Instead, we have first applied one-feature-out and one-feature-add tests to determine the YouTube features that individually perform better. After that, we could apply exhaustive feature selection for a smaller set of features.

In one-feature-out method, we iterate over all features and build a model by extracting a feature at each step so that we can see the change in validation score of the model when that feature is removed. Therefore, a decrease in the score implies that the extracted feature positively affects the predictive power of the model. We can assume that, when the decrease is higher, the feature is more important. The idea of one-feature-add method is the opposite. We iterate over all features and build a model by adding a feature on top of the base features at each step so that we can see the change in validation score of the model when that feature is added. Therefore, an increase in the score implies that added feature positively affects the predictive power of the model. Similarly, we can assume that a higher increase implies that the feature is more important.

6.1. Classification Results

6.1.1. One-feature-out / One-feature-add Tests for YouTube Features

As mentioned in Section 1.1, one of the motivations of this study is to observe the predictive power of YouTube features on box-office forecasting. In Table 6.1, we present the cross validation results of one-feature-out and one-feature-add tests for YouTube features. As one of the outcomes of these tests, we show that, using only the reactions toward a movie's trailer on YouTube, without needing any other information, one can predict whether the movie will make a positive profit on box-office or not with 62.27% accuracy.

The individual feature performances are also presented in the same table. In one-feature-out test, while the accuracy was 62.06% when all YouTube features were used, the lowest accuracy of 61.75% was observed when polarity-vader feature was removed. In one-add test, there was no base features, meaning that we trained models using only one feature at each step. In this part, view-count has shown the best performance with 61.60% accuracy.

These two features have taken precedence over the others to be used in the exhaustive feature selection part.

6.1.2. One-feature-out / One-feature-add Tests for All Features

Table 6.2 shows the cross validation results of one-feature-out and one-feature-add tests for all features.

Mean cross validation score is 80.91% when all of the twenty-six features are used. In one-feature-out test, we have seen drastic decreases in the accuracy to 72.22% and 68.45% when budget and screen-count features were removed, respectively. This shows how important these features are. Hence, in one-feature-add test, we have used these two as the base features and added others on top of them. Even though the changes

Table 6.1. CV Results of One-out and One-add Tests for YouTube Features

	one-out	**one-add**
base	0.6206 (all)	- (none)
like-ratio	0.6196	0.5597
like-count	0.6227	0.5942
dislike-count	0.6212	0.5860
view-count	0.6202	0.6160
comment-count	0.6205	0.5780
polarity-nb	0.6226	0.5443
w-polarity-nb	0.6204	0.5375
count-pos-nb	0.6224	0.5923
count-neg-nb	0.6185	0.5788
polarity-vader	0.6175	0.5448
w-polarity-vader	0.6206	0.5411
count-pos-vader	0.6195	0.5816
count-neg-vader	0.6211	0.5824
polarity-avg	0.6217	0.5481
w-polarity-avg	0.6221	0.5370
pos-neg-ratio	0.6202	0.5481
pos-neg-ratio1	0.6209	0.5507
pos-neg-ratio3	0.6193	0.5606
pos-neg-ratio5	0.6204	0.5614

were very small, the highest decrease in the accuracy (80.71%) was observed when w-polarity-avg feature was removed.

Using only the base features we achieve 78.35% accuracy. As the best performing ones in the one-add test, from movie master features, year has increased the base accuracy to 79.15%. From the YouTube features, like-ratio has increased the base accuracy to 78.88%.

6.1.3. Exhaustive Feature Selection

As stated previously, the goal of our experimentation is to see if YouTube features can make a statistically significant contribution to the predictive power of the models. In order to be able to reach such conclusion we needed a base score without any YouTube features. We have achieved 80.60% accuracy when only the movie master features are used in the model.

On top of that, we have added some YouTube features which were selected based on their individual performances in the one-add and one-out tests. These YouTube features are: like ratio, view-count, w-polarity-avg, and polarity-vader. Additionally, we have added pos-neg-ratio5 as a different type of feature that might potentially increase the accuracy.

Then, we tried all combinations of these 12 features (7 movie master + 5 YouTube) that contains at least 7 features in the model. Therefore, we have generated cross validation results for 1586 different sets of features. In the appendix of this thesis, we share the mean cross validation scores and standard deviations for each feature combination. We have achieved the best accuracy of 81.61% when the following features were used: budget, seasonality, is-sequel, run-time, screen-count, year, like-ratio, view-count, pos-neg-ratio5. YouTube has increased the best accuracy by 1.01%. As one can observe, mpaa, w-polarity-avg and polarity-vader features were not included in the best performing model.

Table 6.2. CV Results of One-out and One-add Tests for All Features

	one-out	**one-add**
base	0.8091 (all)	0.7835 (budget + screen-count)
mpaa	0.8084	0.7833
budget	0.7222	-
seasonality	0.8075	0.7853
is-sequel	0.8077	0.7896
run-time	0.8039	0.7883
screen-count	0.6845	-
genre	0.8074	0.7845
year	0.7984	0.7915
like-ratio	0.8103	0.7888
like-count	0.8082	0.7830
dislike-count	0.8110	0.7851
view-count	0.8081	0.7854
comment-count	0.8093	0.7859
polarity-nb	0.8085	0.7876
w-polarity-nb	0.8093	0.7820
count-pos-nb	0.8101	0.7838
count-neg-nb	0.8080	0.7804
polarity-vader	0.8078	0.7869
w-polarity-vader	0.8093	0.7821
count-pos-vader	0.8089	0.7814
count-neg-vader	0.8086	0.7817
polarity-avg	0.8086	0.7862
w-polarity-avg	0.8071	0.7858
pos-neg-ratio	0.8091	0.7853
pos-neg-ratio1	0.8082	0.7870
pos-neg-ratio3	0.8086	0.7871
pos-neg-ratio5	0.8081	0.7878

In the next section, using paired t-Test method, we test if this increase in the forecasting accuracy is statistically significant or not.

6.2. Paired t-Test

In paired t-Test, using the base and the best performing models' accuracy results that correspond to each fold of each repetition of cross validation, we compare the means of two sets of observations. We have 50 different test results for each model. Table 6.3 shows the cross validation scores of each fold in the base model and Table 6.4 shows the cross validation scores of each fold in the best performing model. We apply the paired test on these values. Table 6.5 shows some statistics that describe the distribution of the scores in each group.

Table 6.3. CV scores of each fold in the base model (r: repetition, f:fold)

	f=1	**f=2**	**f=3**	**f=4**	**f=5**	**f=6**	**f=7**	**f=8**	**f=9**	**f=10**
r=1	0.8012	0.8106	0.795	0.8323	0.7609	0.8292	0.8261	0.7981	0.8323	0.8509
r=2	0.7888	0.8075	0.8354	0.8137	0.8012	0.7981	0.8137	0.8043	0.8106	0.8168
r=3	0.7857	0.8292	0.8323	0.7857	0.8385	0.8075	0.8261	0.8012	0.7826	0.7795
r=4	0.8634	0.7764	0.7888	0.8012	0.7888	0.7764	0.7453	0.823	0.8323	0.795
r=5	0.8043	0.7819	0.8043	0.7757	0.8137	0.838	0.8075	0.81	0.7888	0.7944

Table 6.4. CV scores of each fold in the best performing model (r: repetition, f:fold)

	f=1	**f=2**	**f=3**	**f=4**	**f=5**	**f=6**	**f=7**	**f=8**	**f=9**	**f=10**
r=1	0.7981	0.8447	0.8075	0.8354	0.7919	0.8075	0.823	0.823	0.8292	0.8571
r=2	0.8292	0.8323	0.8137	0.8199	0.8168	0.8292	0.8106	0.8075	0.7981	0.8447
r=3	0.8106	0.8416	0.8292	0.8106	0.8571	0.7981	0.8292	0.7888	0.7919	0.7981
r=4	0.8602	0.7733	0.7857	0.8075	0.8106	0.7795	0.795	0.8292	0.8199	0.8168
r=5	0.8261	0.7726	0.8292	0.8069	0.8199	0.8318	0.8261	0.8474	0.7857	0.8069

Table 6.5. Statistics of the results of the base and the best performing models

	base	**best**
count	50.000000	50.000000
mean	0.806087	0.816091
std	0.023196	0.021267
min	0.745342	0.772586
25%	0.788820	0.800316
50%	0.804348	0.816770
75%	0.825311	0.829193
max	0.863354	0.860248

The null hypothesis of this test is that the mean difference of the observed results of the two models, one including only the base movie features and the other also including some YouTube features, is zero. In other words, the null hypothesis claims that the results of these groups are equal and YouTube features have no significant effect on the prediction accuracy.

To be able to apply paired t-Test, the difference between the results of two groups should be normally distributed; so, we check this condition first. Figure 6.1 and Figure 6.2 show the distribution of the differences between the results of the two groups as histogram and q-q plot, respectively. It looks like there is some deviation from normality; but it is not much. To statistically check the normality, we have applied Shapiro-Wilk test and obtained the p-value of 0.3604. Since the p-value is high, we can say that the test was not significant and the difference between the two groups is normally distributed. After checking that the normality condition is satisfied, we have applied paired t-Test and obtained p-value of 9.7517e-05. Since the p-value is lower than the threshold (0.05), we reject the null hypothesis. Paired t-Test shows that cross validation scores are statistically significant and YouTube features have improved the prediction accuracy of the base movie features.

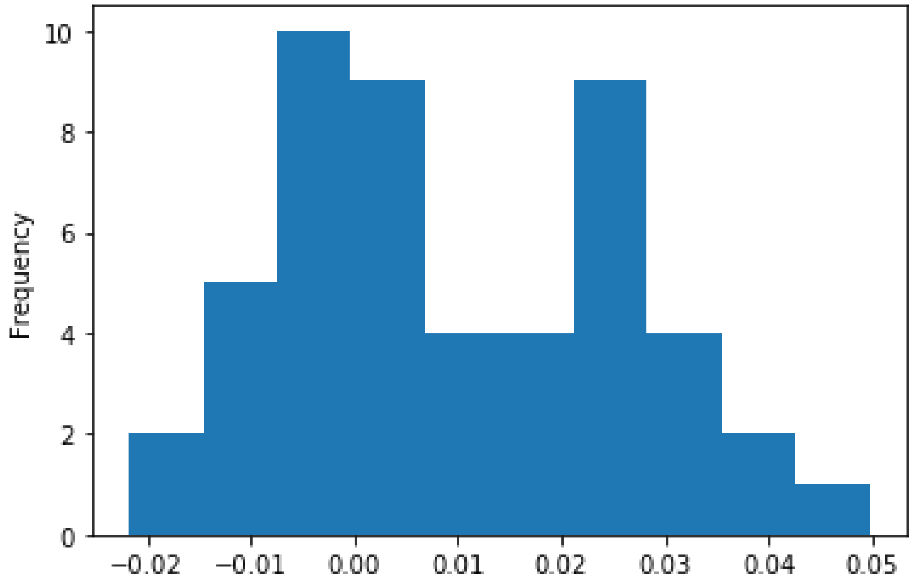

Figure 6.1. Histogram showing the differences between the results of the two groups

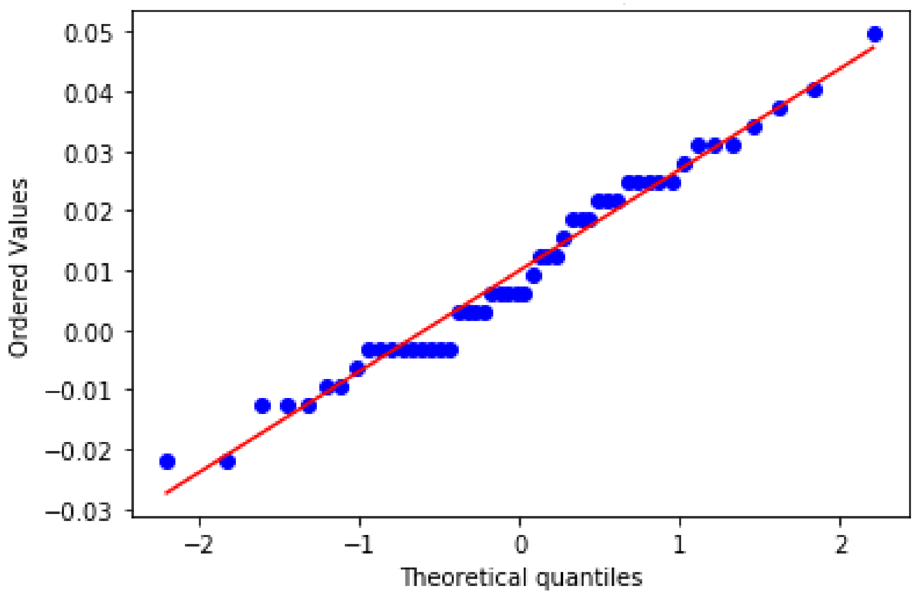

Figure 6.2. Q-Q Plot showing the differences between the results of the two groups

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

In this study, we have proposed a method to make better predictions on movies' box-office success using the reviews on the web, particularly using YouTube users' reactions to movie trailers.

We have generated nineteen different features from YouTube that might potentially help to increase the predictive power of our models. Some of these features were there to indicate the volume of the reactions, representing the popularity of a movie before its release. There were some other features showing positive, neutral or negative polarity of people's opinions about the movie. We have used different combinations of these YouTube features with some other movie features whose positive effects were proven in the previous works. After the experimentation, we have shown that YouTube reactions help to improve forecasting models and we could obtain 81.61% mean accuracy in predicting whether a movie's box-office revenue will exceed its production budget or not.

When we look at the features that positively affect our prediction, we can draw some conclusions. The first feature is *view-count* that shows how many times a movie trailer is watched. We observe from the results that a movie's profitability is correlated with the view count of the movie's trailer on YouTube. A movie trailer's popularity can be a clue about the people's tendency to the movie on theaters.

Other two features consider the polarity of the user reactions. *like-ratio* represents the ratio of the number of user clicks on the like button and the number of user clicks on the dislike button. High values of this ratio shows that the trailer is mostly appreciated by the users, which might possibly indicate that the movie will make a positive profit.

The last YouTube feature included in the best performing model is *pos-neg-ratio5*. While generating this feature, we have applied sentiment analysis on each comment made for movie trailers. We have retrieved the polarity score of each comment and then by setting a neutral zone between -0.5 and 0.5 we have counted the number of positive and negative user comments and finally retrieved the feature value by proportioning these counts. Seeing that this feature has been included in the best performing model, we understand that the sentiment of user comments on movie trailers is a helpful piece of information to predict movies box-office success.

Our study shows that these features on YouTube user reactions to movie trailers are helpful and can be used in further studies for box-office forecasting.

7.2. Future Work

As explained in the methodology chapter, it was not computationally feasible for us to try all combinations of YouTube features in a reasonable amount of time. Therefore, we have used a heuristic to make an initial selection among those features. We believe that further improvements in the accuracy could be achieved by trying all of the combinations. Future work on this computational exercise and further investigations on the topic might prove more significant contributions of YouTube reactions on box-office forecasting.

As an additional direction of future work, movie comments can be separated according to the comment date and the release date of the corresponding movie. This approach would decrease the number of movies with comprehensive data, but it would assure that the user comments on the trailer have been made before the user had actually seen the movie.

REFERENCES

1. Wikipedia contributors, *Cross-validation (statistics) — Wikipedia, The Free Encyclopedia*, 2019, `https://en.wikipedia.org/w/index.php?title=Cross-validation_(statistics)&oldid=901406671`, accessed in July 2019.

2. Potts, D. and C. Sammut, "Online Nonlinear System Identification in High Dimensional Environments", *Proc. of the Australasian Conf. on Robotics and Automation*, Citeseer, 2004.

3. Shoaran, M., B. A. Haghi, M. Taghavi, M. Farivar and A. Emami-Neyestanak, "Energy-efficient classification for resource-constrained biomedical applications", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 8, No. 4, pp. 693–707, 2018.

4. Ghafaryasl, B., F. van der Lijn, M. Poels, H. Vrooman, M. A. Ikram, W. J. Niessen, A. van der Lugt, M. Vernooij and M. de Bruijne, "A computer aided detection system for cerebral microbleeds in brain MRI", *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 138–141, IEEE, 2012.

5. Bre, F., J. M. Gimenez and V. D. Fachinotti, "Prediction of wind pressure coefficients on building surfaces using artificial neural networks", *Energy and Buildings*, Vol. 158, pp. 1429–1441, 2018.

6. *Box Office Shares by Studio in 2018*, 2018, `https://www.boxofficemojo.com/studio/?view=company&view2=yearly&yr=2018&p=.htm`, accessed in June 2019.

7. *Season-by-season average gross revenues per movie*, 2019, `https://www.boxofficemojo.com/seasonal/?chart=byyear&view=releasedate`, accessed in June 2019.

8. Wikipedia contributors, *Box office — Wikipedia, The Free Encyclopedia*,

2019, https://en.wikipedia.org/w/index.php?title=Box_office&oldid=899139854, accessed in June 2019.

9. Boğaziçi University Text Analytics and BioInformatics Lab, *Open Data Set for Box-Office Forecasting*, 2019, https://github.com/boun-tabi/box-office-data, accessed in July 2019.

10. Bengio, Y. and Y. Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation", *Journal of machine learning research*, Vol. 5, No. Sep, pp. 1089–1105, 2004.

11. Mitchell, T. M., *Machine Learning*, McGraw-Hill Science, 1997.

12. Breiman, L., "Random forests", *Machine learning*, Vol. 45, No. 1, pp. 5–32, 2001.

13. Friedman, J. H., "Stochastic gradient boosting", *Computational statistics & data analysis*, Vol. 38, No. 4, pp. 367–378, 2002.

14. Cover, T. M., P. E. Hart *et al.*, "Nearest neighbor pattern classification", *IEEE transactions on information theory*, Vol. 13, No. 1, pp. 21–27, 1967.

15. Bing, L., K. C. Chan and C. Ou, "Public sentiment analysis in Twitter data for prediction of a company's stock price movements", *2014 IEEE 11th International Conference on e-Business Engineering*, pp. 232–239, IEEE, 2014.

16. Andranik Tumasjan, P. G. S. I. M. W., Timm O. Sprenger, "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment", *Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

17. Vitelli, M., *Predicting Box Office Revenue for Movies*, 2015, https://pdfs.semanticscholar.org/ada7/df3ae3ca6a80bacabbe54cf7169287bef3b3.pdf, accessed in June 2019.

18. Simonoff, J. S. and I. R. Sparrow, "Predicting movie grosses: Winners and losers,

blockbusters and sleepers", *Chance*, Vol. 13, No. 3, pp. 15–24, 2000.

19. Zhou, Y., L. Zhang and Z. Yi, "Predicting movie box-office revenues using deep neural networks", *Neural Computing and Applications*, pp. 1–11, 2017.

20. Sharda, R. and D. Delen, "Predicting box-office success of motion pictures with neural networks", *Expert Systems with Applications*, Vol. 30, No. 2, pp. 243–254, 2006.

21. Mestyán, M., T. Yasseri and J. Kertész, "Early prediction of movie box office success based on Wikipedia activity big data", *PloS one*, Vol. 8, No. 8, p. e71226, 2013.

22. Barthelemy, P. T., D. Guillory and C. Mandal, "Using twitter data to predict box office revenues", *CA, United States: University of Stanford*, 2012.

23. Liu, T., X. Ding, Y. Chen, H. Chen and M. Guo, "Predicting movie Box-office revenues by exploiting large-scale social media content", *Multimedia Tools and Applications*, Vol. 75, No. 3, pp. 1509–1528, 2016.

24. Joshi, M., D. Das, K. Gimpel and N. A. Smith, "Movie reviews and revenues: An experiment in text regression", *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 293–296, Association for Computational Linguistics, 2010.

25. Gaikar, D. D., B. Marakarkandy and C. Dasgupta, "Using Twitter data to predict the performance of Bollywood movies", *Industrial Management & Data Systems*, Vol. 115, No. 9, pp. 1604–1621, 2015.

26. Wong, F. M. F., S. Sen and M. Chiang, "Why watching movie tweets won't tell the whole story?", *Proceedings of the 2012 ACM workshop on Workshop on online social networks*, pp. 61–66, ACM, 2012.

27. Asur, S. and B. A. Huberman, "Predicting the future with social media", *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pp. 492–499, IEEE Computer Society, 2010.

28. Rhee, T. G. and F. Zulkernine, "Predicting movie box office profitability: a neural network approach", *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 665–670, IEEE, 2016.

29. Hur, M., P. Kang and S. Cho, "Box-office forecasting based on sentiments of movie reviews and Independent subspace method", *Information Sciences*, Vol. 372, pp. 608–624, 2016.

30. Ghiassi, M., D. Lio and B. Moon, "Pre-production forecasting of movie revenues with a dynamic artificial neural network", *Expert Systems with Applications*, Vol. 42, No. 6, pp. 3176–3193, 2015.

31. Oh, C., Y. Roumani, J. K. Nwankpa and H.-F. Hu, "Beyond likes and tweets: Consumer engagement behavior and movie box office in social media", *Information & Management*, Vol. 54, No. 1, pp. 25–37, 2017.

32. Wikipedia contributors, *IMDb — Wikipedia, The Free Encyclopedia*, 2019, `https://en.wikipedia.org/w/index.php?title=IMDb&oldid=901740073`, accessed in June 2019.

33. Wikipedia contributors, *Feature film — Wikipedia, The Free Encyclopedia*, 2019, `https://en.wikipedia.org/w/index.php?title=Feature_film&oldid=899668480`, accessed in June 2019.

34. *IMDb movies with sequel*, `https://www.imdb.com/list/ls003495084`, accessed in June 2019.

35. *Movie Budget and Financial Performance Records*, `https://www.the-numbers.`

com/movie/budgets/all, accessed in June 2019.

36. *Number of theaters showing movies*, https://www.boxofficemojo.com/movies/alphabetical.htm, accessed in June 2019.

37. *Seasonal Box Office Records*, https://www.boxofficemojo.com/seasonal/?chart=byyear, accessed in June 2019.

38. Wikipedia contributors, *YouTube — Wikipedia, The Free Encyclopedia*, 2019, https://en.wikipedia.org/w/index.php?title=YouTube, accessed in June 2019.

39. *NLTK Annotated Text Corpora*, https://www.nltk.org/book/ch02.html, accessed in June 2019.

40. Hutto, C. J. and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text", *Eighth international AAAI conference on weblogs and social media*, 2014.

41. *MPAA Film Ratings*, 2019, https://www.filmratings.com/, accessed in June 2019.

42. Einav, L., "Seasonality in the US motion picture industry", *The Rand journal of economics*, Vol. 38, No. 1, pp. 127–145, 2007.

43. Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, 2011.

44. *XGB Documentation*, 2019, https://xgboost.readthedocs.io/en/latest/index.html, accessed in June 2019.

45. *Keras wrapper for the Scikit-learn API*, 2019, `https://keras.io/scikit-learn-api/`, accessed in June 2019.

APPENDIX A: TABLE OF EXHAUSTIVE FEATURE SELECTION RESULTS

We share the mean cross validation scores and standard deviations for each combination of 12 movie features that contains at least 7 features in Table 7.1. 1586 different sets of features were used to train XGB models and for each set the corresponding cross-validation scores are shared in the table. Explanations of the abbrevations in the table are as follows: b:budget, s:seasonality, g:genre, m:mpaa, is:is-sequel, rt:run-time, sc:screen-count, y:year, lr:like-ratio, vc:view-count, wpa:w-polarity-avg, pv:polarity-vader, pnr5:pos-neg-ratio5.

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection

feature_names	**avg_score**	**std_dev**
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8161	0.0211
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8153	0.0208
('b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.8152	0.0226
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8152	0.0205
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.815	0.0206
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8149	0.0224
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc')	0.8146	0.0215
('b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8145	0.0226
('m', 'b', 'is', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.8145	0.0225
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8145	0.0218
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8144	0.0208
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.8144	0.0227
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8142	0.0206
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8141	0.0236
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.814	0.0222
('b', 'is', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.814	0.0224
('b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.814	0.0223
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8138	0.0225

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'vc')	0.8137	0.0219
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8136	0.0209
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8136	0.0228
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.8135	0.0247
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc')	0.8135	0.0229
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.8135	0.0234
('b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8135	0.0243
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8134	0.023
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8133	0.0214
('b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8132	0.0221
('b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8132	0.0231
('b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8132	0.0221
('m', 'b', 's', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.8132	0.0234
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.8131	0.0234
('b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8131	0.0212
('b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8129	0.0226
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8129	0.0213
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8129	0.0208
('b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8129	0.0207
('m', 'b', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8128	0.0228
('b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'pv')	0.8128	0.025
('b', 's', 'rt', 'sc', 'y', 'lr', 'vc')	0.8127	0.0214
('b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8127	0.0223
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8127	0.0233
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8127	0.0213
('b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8125	0.0205
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8124	0.0219
('m', 'b', 'is', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8124	0.0235
('b', 'is', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8124	0.0232
('b', 's', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8122	0.0215

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8122	0.0229
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8121	0.0217
('b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8121	0.0228
('b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.8121	0.0217
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8121	0.0218
('m', 'b', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8121	0.0231
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.8121	0.0212
('m', 'b', 'rt', 'sc', 'y', 'lr', 'vc')	0.812	0.0202
('b', 'is', 'rt', 'sc', 'y', 'vc', 'pv')	0.812	0.0239
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.812	0.0232
('b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.8119	0.0233
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8119	0.0204
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8119	0.0248
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8119	0.0217
('m', 'b', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8119	0.0225
('b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8118	0.025
('b', 'is', 'rt', 'sc', 'y', 'lr', 'vc')	0.8117	0.0222
('m', 'b', 's', 'rt', 'sc', 'y', 'vc')	0.8116	0.0232
('b', 's', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.8116	0.0228
('b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8116	0.023
('b', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8116	0.0223
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8116	0.0215
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'pv')	0.8116	0.024
('b', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8115	0.0245
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8114	0.0227
('b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8114	0.0202
('b', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8114	0.0226
('b', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8112	0.0207
('m', 'b', 's', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8112	0.0233
('b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa')	0.8112	0.0232

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8112	0.0233
('b', 's', 'rt', 'sc', 'y', 'vc', 'wpa')	0.8112	0.0218
('b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8111	0.0213
('b', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8111	0.0232
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa')	0.8111	0.0228
('b', 's', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8111	0.0212
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.811	0.0207
('b', 's', 'is', 'rt', 'sc', 'y', 'vc')	0.811	0.0223
('b', 's', 'rt', 'sc', 'y', 'vc', 'pv')	0.811	0.0235
('b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8109	0.0214
('m', 'b', 's', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8109	0.0216
('m', 'b', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8109	0.0228
('b', 's', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8109	0.0218
('m', 'b', 's', 'rt', 'sc', 'y', 'vc', 'wpa')	0.8109	0.022
('m', 'b', 's', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8109	0.0228
('m', 'b', 'is', 'rt', 'sc', 'y', 'vc', 'pv')	0.8109	0.0239
('b', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8109	0.0217
('b', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8108	0.0231
('m', 'b', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8107	0.0222
('b', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8106	0.0208
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8106	0.0204
('b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8106	0.02
('b', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8106	0.0221
('m', 'b', 'rt', 'sc', 'y', 'vc', 'pv')	0.8106	0.0216
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8106	0.0212
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8106	0.0245
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'vc')	0.8105	0.0213
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8105	0.0231
('b', 'is', 'rt', 'sc', 'y', 'vc', 'wpa')	0.8105	0.023
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'vc')	0.8104	0.0229

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8104	0.0221
('b', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.8104	0.0213
('m', 'b', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8104	0.0233
('m', 'b', 's', 'sc', 'y', 'lr', 'vc', 'pv')	0.8103	0.0217
('m', 'b', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.8103	0.0206
('m', 'b', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8103	0.0206
('b', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8102	0.0229
('m', 'b', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.81	0.0213
('b', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8099	0.0216
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'pv')	0.8099	0.0197
('m', 'b', 's', 'rt', 'sc', 'y', 'vc', 'pv')	0.8099	0.0232
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.8099	0.0246
('m', 'b', 's', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8098	0.0216
('m', 'b', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.8098	0.0223
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8096	0.0217
('b', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8096	0.0202
('b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8096	0.021
('b', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8096	0.0197
('b', 'is', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8095	0.0213
('m', 'b', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8095	0.0209
('b', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8094	0.0204
('b', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8094	0.0196
('m', 'b', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8093	0.0219
('b', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8093	0.0207
('m', 'b', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8093	0.0206
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.8093	0.0246
('b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'pv')	0.8092	0.0207
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8092	0.0221
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8091	0.0234
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8091	0.0195

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8089	0.0194
('b', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8089	0.0204
('m', 'b', 'is', 'rt', 'sc', 'y', 'vc', 'wpa')	0.8089	0.0223
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8089	0.024
('b', 'is', 'sc', 'y', 'lr', 'vc', 'pv')	0.8088	0.0195
('b', 's', 'sc', 'y', 'lr', 'vc', 'pv')	0.8088	0.0204
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8088	0.0244
('m', 'b', 's', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8088	0.0202
('m', 'b', 'is', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8086	0.021
('m', 'b', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8086	0.0212
('b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8086	0.022
('m', 'b', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8086	0.0216
('b', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8086	0.0221
('m', 'b', 'is', 'sc', 'y', 'lr', 'vc', 'pv')	0.8086	0.0203
('b', 'is', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8085	0.02
('b', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8085	0.0229
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8084	0.0236
('m', 'b', 's', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8084	0.019
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8084	0.0238
('b', 'is', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8084	0.0226
('m', 'b', 'is', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8083	0.022
('b', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8083	0.0195
('b', 'is', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.8083	0.0213
('m', 'b', 's', 'is', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8083	0.0237
('b', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8083	0.0209
('b', 'is', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8083	0.0237
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8082	0.0227
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8082	0.0241
('m', 'b', 's', 'is', 'sc', 'y', 'vc', 'pv')	0.8081	0.023
('b', 's', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.8081	0.0194

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'is', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.8081	0.0222
('b', 's', 'is', 'sc', 'y', 'vc', 'pv')	0.8081	0.023
('b', 's', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.808	0.0204
('m', 'b', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.808	0.0216
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr')	0.808	0.0215
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'pv')	0.808	0.0232
('m', 'b', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.808	0.0198
('m', 'b', 's', 'sc', 'y', 'lr', 'vc', 'wpa')	0.808	0.022
('m', 'b', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8079	0.0197
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'pv')	0.8079	0.0219
('b', 'is', 'rt', 'sc', 'y', 'lr', 'pv')	0.8079	0.0237
('b', 's', 'is', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8079	0.0236
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr')	0.8078	0.0243
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.8078	0.0219
('b', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8078	0.0236
('b', 's', 'is', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8076	0.0226
('m', 'b', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8076	0.0188
('m', 'b', 's', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8076	0.0201
('m', 'b', 's', 'is', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8076	0.0244
('m', 'b', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8076	0.0211
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8075	0.0242
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'pv')	0.8075	0.0236
('m', 'b', 's', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8075	0.0197
('m', 'b', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8074	0.0185
('m', 'b', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.8073	0.0221
('b', 's', 'is', 'rt', 'sc', 'y', 'lr')	0.8073	0.0242
('b', 's', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8073	0.0194
('b', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8073	0.0193
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8072	0.0246
('b', 's', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.8072	0.0207

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'is', 'sc', 'y', 'lr', 'vc')	0.8071	0.0221
('m', 'b', 'is', 'sc', 'y', 'lr', 'vc')	0.8071	0.0223
('m', 'b', 's', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.8071	0.0196
('m', 'b', 'sc', 'y', 'lr', 'vc', 'pv')	0.8071	0.0214
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'pnr5')	0.8071	0.0231
('m', 'b', 's', 'is', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8071	0.0235
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa')	0.807	0.0205
('m', 'b', 'rt', 'sc', 'y', 'vc', 'wpa')	0.807	0.0233
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'vc')	0.807	0.0228
('m', 'b', 'is', 'rt', 'sc', 'y', 'vc')	0.807	0.0228
('b', 's', 'is', 'sc', 'y', 'vc', 'pnr5')	0.807	0.0233
('b', 's', 'is', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.807	0.0235
('b', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8068	0.0236
('m', 'b', 's', 'sc', 'y', 'lr', 'vc')	0.8068	0.0217
('m', 'b', 'is', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8067	0.0224
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.8067	0.0237
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'pv')	0.8066	0.0218
('b', 's', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8065	0.0228
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8064	0.0234
('b', 'is', 'rt', 'sc', 'y', 'lr', 'wpa')	0.8064	0.0228
('m', 'b', 's', 'is', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8063	0.0222
('b', 's', 'rt', 'sc', 'y', 'lr', 'pv')	0.8062	0.0215
('m', 'b', 's', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8062	0.0231
('b', 'is', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8062	0.0222
('m', 'b', 's', 'rt', 'sc', 'y', 'lr')	0.8062	0.0216
('m', 'b', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8061	0.0187
('m', 'b', 's', 'is', 'rt', 'sc', 'y')	0.8061	0.023
('b', 's', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.806	0.0232
('m', 'b', 'is', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.806	0.0234
('b', 's', 'sc', 'y', 'lr', 'vc', 'wpa')	0.8059	0.0203

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 'is', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8059	0.0242
('b', 's', 'is', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.8059	0.0229
('m', 'b', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.8058	0.0219
('m', 'b', 's', 'sc', 'y', 'vc', 'pv')	0.8058	0.023
('m', 'b', 'is', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8058	0.0243
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8058	0.0206
('b', 's', 'is', 'sc', 'y', 'vc', 'wpa')	0.8058	0.0236
('m', 'b', 's', 'sc', 'y', 'vc', 'pnr5')	0.8057	0.0217
('m', 'b', 's', 'is', 'sc', 'y', 'vc', 'pnr5')	0.8057	0.0252
('m', 'b', 'is', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8057	0.0252
('m', 'b', 'is', 'sc', 'y', 'vc', 'wpa')	0.8057	0.0221
('b', 's', 'is', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.8056	0.0231
('b', 's', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8055	0.0211
('m', 'b', 'is', 'sc', 'y', 'vc', 'pnr5')	0.8055	0.024
('b', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa')	0.8055	0.023
('b', 's', 'is', 'rt', 'sc', 'y', 'pnr5')	0.8055	0.0236
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.8055	0.024
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'wpa')	0.8055	0.021
('b', 'is', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8055	0.0232
('b', 's', 'is', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8055	0.023
('m', 'b', 's', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8054	0.0225
('b', 'is', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8054	0.0238
('m', 'b', 'is', 'sc', 'y', 'vc', 'pv')	0.8053	0.0235
('m', 'b', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.8053	0.0215
('m', 'b', 'is', 'rt', 'sc', 'y', 'lr', 'wpa')	0.8053	0.0223
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.8053	0.0235
('m', 'b', 'is', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8053	0.0236
('m', 'b', 'is', 'rt', 'sc', 'y', 'pnr5')	0.8053	0.0265
('m', 'b', 's', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8052	0.022
('b', 's', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.8052	0.0247

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'is', 'sc', 'y', 'vc', 'wpa')	0.8052	0.0231
('m', 'b', 's', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.8052	0.0233
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.805	0.0242
('b', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.805	0.023
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.805	0.0245
('b', 's', 'is', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.805	0.024
('b', 's', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.805	0.0237
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'pv')	0.8049	0.0208
('b', 'is', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.8049	0.0236
('b', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8048	0.0227
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.8048	0.0208
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'wpa')	0.8048	0.023
('m', 'b', 'is', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.8048	0.0229
('b', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8048	0.0216
('m', 'b', 's', 'rt', 'sc', 'y', 'pnr5')	0.8048	0.0231
('b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'pv')	0.8047	0.0211
('m', 'b', 'is', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.8047	0.0214
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8047	0.0204
('b', 's', 'is', 'rt', 'sc', 'y', 'wpa')	0.8047	0.0218
('m', 'b', 's', 'sc', 'y', 'vc', 'wpa')	0.8047	0.0223
('b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.8046	0.0214
('m', 'b', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8046	0.0216
('b', 'is', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.8045	0.0235
('b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.8045	0.0228
('b', 's', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8044	0.0224
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8043	0.0221
('m', 'b', 's', 'is', 'sc', 'y', 'vc')	0.8043	0.023
('m', 'b', 'is', 'rt', 'sc', 'y', 'wpa', 'pv')	0.8042	0.023
('b', 'is', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.8041	0.0218
('b', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8041	0.0219

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'is', 'sc', 'y', 'lr', 'pv')	0.8041	0.0195
('m', 'b', 's', 'is', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.804	0.0241
('m', 'b', 's', 'sc', 'y', 'vc', 'wpa', 'pv')	0.804	0.0213
('m', 'b', 'is', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.804	0.0255
('b', 's', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.804	0.0223
('b', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.8039	0.0235
('b', 's', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8039	0.0209
('b', 'is', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.8039	0.0211
('m', 'b', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.8039	0.0219
('b', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.8039	0.0215
('m', 'b', 's', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8039	0.0212
('b', 's', 'rt', 'sc', 'y', 'lr', 'wpa')	0.8039	0.021
('b', 's', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8038	0.0244
('b', 'is', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.8038	0.0235
('m', 'b', 'is', 'rt', 'sc', 'lr', 'vc', 'pv')	0.8037	0.0212
('m', 'b', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8037	0.0228
('m', 'b', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8037	0.0221
('m', 'b', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8037	0.0218
('m', 'b', 'is', 'rt', 'sc', 'vc', 'pv')	0.8037	0.0242
('m', 'b', 'rt', 'sc', 'y', 'lr', 'pv')	0.8037	0.0228
('b', 's', 'sc', 'y', 'vc', 'wpa', 'pv')	0.8037	0.0222
('b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.8037	0.0219
('b', 's', 'is', 'rt', 'sc', 'y', 'pv')	0.8037	0.0238
('b', 'is', 'rt', 'sc', 'y', 'wpa', 'pv')	0.8037	0.0237
('b', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8036	0.0217
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.8036	0.0229
('m', 'b', 'is', 'sc', 'y', 'lr', 'pnr5')	0.8035	0.0228
('m', 'b', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.8035	0.0223
('m', 'b', 'is', 'rt', 'sc', 'vc', 'pnr5')	0.8035	0.0236
('m', 'b', 's', 'is', 'rt', 'sc', 'vc', 'pnr5')	0.8035	0.0231

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 'is', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8034	0.0228
('b', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8034	0.0233
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8034	0.0224
('m', 'b', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.8033	0.0226
('b', 's', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.8033	0.0226
('b', 's', 'is', 'rt', 'sc', 'y', 'wpa', 'pv')	0.8033	0.0231
('m', 'b', 'is', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8032	0.0238
('m', 'b', 's', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.8032	0.0245
('m', 'b', 'rt', 'sc', 'y', 'lr', 'wpa')	0.8032	0.0206
('m', 'b', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8032	0.0219
('b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.8032	0.0216
('b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8032	0.0217
('m', 'b', 's', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.8032	0.0241
('m', 'b', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.8032	0.0238
('m', 'b', 'is', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.8031	0.0226
('b', 's', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.8031	0.0242
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.803	0.0217
('m', 'b', 'is', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.803	0.023
('m', 'b', 'is', 'rt', 'sc', 'y', 'pv')	0.803	0.0241
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'wpa', 'pv')	0.803	0.0232
('b', 'is', 'rt', 'sc', 'lr', 'vc', 'pv')	0.803	0.0229
('b', 's', 'is', 'rt', 'sc', 'vc', 'pnr5')	0.803	0.0232
('b', 's', 'is', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.803	0.0237
('m', 'b', 'is', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.803	0.0231
('m', 'b', 's', 'is', 'rt', 'sc', 'y', 'pv')	0.803	0.0235
('b', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8029	0.0201
('m', 'b', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8029	0.0221
('b', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.8029	0.022
('m', 'b', 'is', 'sc', 'y', 'lr', 'pv')	0.8028	0.0221
('m', 'b', 'is', 'rt', 'sc', 'y', 'wpa')	0.8028	0.0244

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.8027	0.0219
('m', 'b', 'is', 'rt', 'sc', 'lr', 'pnr5')	0.8027	0.0234
('b', 'is', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8027	0.0236
('m', 'b', 's', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.8027	0.0221
('b', 'is', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8026	0.022
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.8026	0.0228
('b', 'is', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.8026	0.0243
('b', 'is', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.8025	0.0221
('b', 's', 'is', 'rt', 'sc', 'pv', 'pnr5')	0.8025	0.0229
('b', 's', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.8025	0.0219
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8025	0.0221
('b', 's', 'is', 'rt', 'sc', 'vc', 'pv')	0.8024	0.0241
('b', 'is', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8024	0.0226
('b', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.8022	0.0211
('m', 'b', 's', 'is', 'rt', 'sc', 'vc', 'pv')	0.8022	0.0234
('b', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.8022	0.0225
('m', 'b', 's', 'rt', 'sc', 'y', 'pv')	0.8022	0.0228
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8022	0.0211
('m', 'b', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.8021	0.0234
('m', 'b', 'is', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8021	0.0231
('m', 'b', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.8021	0.0222
('b', 's', 'is', 'rt', 'sc', 'lr', 'pnr5')	0.8021	0.0248
('b', 'is', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.802	0.0235
('m', 'b', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.802	0.0207
('m', 'b', 's', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.802	0.0222
('b', 's', 'is', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8019	0.0212
('b', 'is', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.8018	0.0256
('m', 'b', 's', 'rt', 'sc', 'y', 'wpa', 'pv')	0.8018	0.022
('b', 's', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.8018	0.023
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'pnr5')	0.8017	0.0253

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'is', 'rt', 'sc', 'lr', 'pv')	0.8017	0.0229
('m', 'b', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.8017	0.0235
('b', 's', 'is', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.8017	0.0235
('b', 's', 'is', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.8017	0.0212
('b', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.8016	0.0239
('m', 'b', 'is', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.8016	0.0216
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8016	0.0229
('m', 'b', 's', 'is', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.8016	0.0238
('b', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.8016	0.0241
('b', 's', 'is', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.8016	0.0218
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'pnr5')	0.8016	0.0216
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'pv')	0.8016	0.0223
('m', 'b', 'is', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.8016	0.0225
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'pv')	0.8015	0.0203
('b', 's', 'rt', 'sc', 'y', 'wpa', 'pv')	0.8014	0.0223
('m', 'b', 's', 'rt', 'sc', 'y', 'wpa')	0.8014	0.0248
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.8014	0.0237
('b', 's', 'is', 'sc', 'y', 'lr', 'pnr5')	0.8014	0.0229
('m', 'b', 's', 'sc', 'y', 'lr', 'pv')	0.8014	0.0216
('b', 's', 'is', 'rt', 'sc', 'lr', 'wpa')	0.8014	0.0223
('b', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.8012	0.022
('b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.8011	0.0241
('b', 's', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8011	0.0214
('m', 'b', 'is', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.8011	0.0215
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.8011	0.0217
('m', 'b', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.8011	0.0211
('b', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.8011	0.0218
('m', 'b', 's', 'is', 'rt', 'sc', 'pnr5')	0.8011	0.0238
('b', 's', 'is', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.8011	0.0237
('m', 'b', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.8011	0.0216

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.8011	0.0239
('m', 'b', 's', 'is', 'rt', 'sc', 'pv', 'pnr5')	0.8011	0.0234
('m', 'b', 'is', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.801	0.0217
('m', 'b', 's', 'is', 'sc', 'y', 'lr', 'wpa')	0.8009	0.0209
('b', 's', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.8009	0.0227
('m', 'b', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.8009	0.0211
('m', 'b', 'is', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8009	0.0223
('m', 'b', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.8008	0.0242
('b', 's', 'sc', 'y', 'lr', 'wpa', 'pv')	0.8008	0.0213
('m', 'b', 's', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.8007	0.0208
('m', 'b', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.8007	0.022
('m', 'b', 'is', 'rt', 'sc', 'lr', 'pv')	0.8007	0.0221
('m', 'b', 's', 'is', 'sc', 'lr', 'vc', 'pv')	0.8006	0.0195
('b', 's', 'is', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8006	0.0233
('m', 'b', 's', 'sc', 'y', 'lr', 'pnr5')	0.8006	0.0212
('b', 's', 'is', 'sc', 'y', 'lr', 'wpa')	0.8006	0.0225
('m', 'b', 'is', 'rt', 'sc', 'lr', 'wpa')	0.8005	0.022
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.8004	0.0231
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.8004	0.0227
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'wpa')	0.8004	0.0232
('b', 's', 'is', 'sc', 'lr', 'vc', 'wpa')	0.8004	0.0214
('m', 'b', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.8003	0.0205
('b', 's', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.8002	0.0231
('b', 'is', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.8001	0.0216
('m', 'b', 's', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.8001	0.0201
('m', 'b', 's', 'is', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.8001	0.022
('m', 'b', 'is', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.8	0.0211
('m', 'b', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.8	0.021
('m', 'b', 'is', 'rt', 'sc', 'pv', 'pnr5')	0.8	0.0229
('m', 'b', 'rt', 'sc', 'y', 'wpa', 'pv')	0.7999	0.024

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.7999	0.025
('b', 's', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.7999	0.0207
('m', 'b', 's', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.7999	0.0204
('b', 's', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7999	0.0215
('b', 'is', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.7998	0.0226
('m', 'b', 's', 'sc', 'y', 'lr', 'wpa', 'pv')	0.7998	0.021
('m', 'b', 's', 'is', 'sc', 'lr', 'vc', 'pnr5')	0.7998	0.0203
('b', 's', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.7997	0.0206
('m', 'b', 'is', 'rt', 'sc', 'wpa', 'pnr5')	0.7996	0.0219
('m', 'b', 'is', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.7996	0.0221
('m', 'b', 's', 'is', 'sc', 'vc', 'pv')	0.7995	0.0258
('m', 'b', 'is', 'rt', 'sc', 'vc', 'wpa')	0.7994	0.0217
('m', 'b', 's', 'is', 'sc', 'y', 'lr')	0.7994	0.0198
('m', 'b', 'is', 'sc', 'y', 'lr', 'wpa')	0.7994	0.0227
('m', 'b', 's', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7994	0.0231
('b', 's', 'is', 'sc', 'lr', 'vc', 'pnr5')	0.7994	0.0212
('m', 'b', 's', 'is', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7994	0.0206
('m', 'b', 'is', 'sc', 'lr', 'vc', 'pv')	0.7993	0.0202
('m', 'b', 'is', 'sc', 'y', 'pv', 'pnr5')	0.7993	0.0212
('b', 's', 'is', 'rt', 'sc', 'wpa', 'pnr5')	0.7992	0.0226
('b', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.7992	0.0201
('m', 'b', 's', 'sc', 'y', 'lr', 'wpa')	0.7992	0.0196
('m', 'b', 's', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.7992	0.0209
('m', 'b', 'is', 'rt', 'sc', 'wpa', 'pv')	0.7992	0.0217
('m', 'b', 's', 'is', 'sc', 'vc', 'pv', 'pnr5')	0.7991	0.0225
('b', 's', 'is', 'sc', 'lr', 'vc', 'pv')	0.7991	0.0214
('m', 'b', 'is', 'sc', 'lr', 'vc', 'wpa')	0.799	0.0204
('b', 's', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.799	0.0224
('b', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7989	0.0218
('b', 's', 'is', 'sc', 'y', 'wpa', 'pnr5')	0.7989	0.0245

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.7988	0.0209
('b', 's', 'is', 'rt', 'sc', 'vc', 'wpa')	0.7988	0.024
('m', 'b', 's', 'is', 'sc', 'lr', 'vc', 'wpa')	0.7988	0.0229
('b', 's', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.7988	0.0207
('m', 'b', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7988	0.0228
('m', 'b', 's', 'is', 'sc', 'vc', 'pnr5')	0.7988	0.0232
('m', 'b', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.7986	0.0224
('m', 'b', 's', 'is', 'rt', 'sc', 'vc', 'wpa')	0.7986	0.0234
('m', 'b', 's', 'is', 'sc', 'vc', 'wpa', 'pnr5')	0.7986	0.0226
('b', 's', 'is', 'sc', 'vc', 'pv', 'pnr5')	0.7986	0.0236
('m', 'b', 'is', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7986	0.022
('m', 'b', 's', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7986	0.0215
('m', 'b', 'is', 'sc', 'lr', 'vc', 'pnr5')	0.7986	0.0213
('m', 'b', 's', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7986	0.0225
('b', 's', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.7986	0.0216
('m', 'b', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7986	0.0213
('m', 'b', 'is', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.7986	0.0237
('b', 's', 'is', 'rt', 'sc', 'lr', 'vc')	0.7985	0.0233
('m', 'b', 's', 'is', 'rt', 'sc', 'lr')	0.7985	0.0237
('m', 'b', 's', 'is', 'sc', 'y', 'pnr5')	0.7985	0.0222
('b', 's', 'is', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.7984	0.0224
('b', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7984	0.0222
('m', 'b', 's', 'is', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.7984	0.0224
('b', 's', 'is', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7984	0.0217
('m', 'b', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7983	0.0214
('b', 's', 'is', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7983	0.0215
('b', 's', 'is', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7983	0.0199
('m', 'b', 's', 'is', 'rt', 'sc', 'wpa', 'pv')	0.7983	0.0216
('m', 'b', 's', 'is', 'rt', 'sc', 'pv')	0.7983	0.022
('b', 's', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7983	0.0212

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.7982	0.0228
('b', 'is', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.7982	0.0217
('b', 's', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.7981	0.022
('m', 'b', 's', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.7981	0.0228
('m', 'b', 's', 'rt', 'sc', 'lr', 'pnr5')	0.7981	0.0223
('m', 'b', 's', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.7981	0.0217
('m', 'b', 's', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.7981	0.0212
('m', 'b', 's', 'is', 'rt', 'sc', 'wpa', 'pnr5')	0.7981	0.0227
('m', 'b', 'is', 'sc', 'y', 'wpa', 'pnr5')	0.798	0.0233
('m', 'b', 'is', 'sc', 'vc', 'wpa', 'pv')	0.798	0.0223
('m', 'b', 's', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.798	0.0217
('m', 'b', 's', 'is', 'rt', 'sc', 'lr', 'vc')	0.798	0.0226
('b', 'is', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7979	0.0223
('b', 's', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.7979	0.0226
('m', 'b', 's', 'is', 'sc', 'y', 'wpa')	0.7979	0.0221
('m', 'b', 's', 'rt', 'sc', 'lr', 'vc', 'pv')	0.7978	0.0236
('m', 'b', 's', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.7978	0.0232
('b', 's', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.7978	0.0215
('m', 'b', 'sc', 'y', 'lr', 'wpa', 'pv')	0.7978	0.0224
('b', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7978	0.0241
('b', 's', 'is', 'sc', 'vc', 'wpa', 'pnr5')	0.7978	0.0216
('m', 'b', 'is', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7978	0.0219
('m', 'b', 's', 'is', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7978	0.0238
('b', 's', 'is', 'sc', 'y', 'pv', 'pnr5')	0.7977	0.0243
('b', 's', 'rt', 'sc', 'lr', 'vc', 'pv')	0.7976	0.0228
('m', 'b', 's', 'rt', 'sc', 'vc', 'pnr5')	0.7976	0.0222
('b', 's', 'is', 'sc', 'lr', 'wpa', 'pv')	0.7976	0.0222
('m', 'b', 's', 'rt', 'sc', 'vc', 'pv')	0.7976	0.0208
('m', 'b', 'rt', 'sc', 'lr', 'vc', 'pv')	0.7975	0.0218
('b', 's', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7975	0.0223

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.7975	0.0218
('b', 's', 'is', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7975	0.0218
('m', 'b', 's', 'is', 'rt', 'sc', 'vc')	0.7974	0.0244
('m', 'b', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.7974	0.0241
('m', 'b', 's', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.7974	0.0212
('b', 's', 'is', 'rt', 'sc', 'wpa', 'pv')	0.7974	0.0215
('m', 'b', 's', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.7974	0.0213
('b', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.7973	0.0231
('m', 'b', 'is', 'sc', 'vc', 'pv', 'pnr5')	0.7973	0.0237
('b', 's', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7973	0.0185
('m', 'b', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7973	0.0207
('m', 'b', 's', 'is', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7973	0.0225
('m', 'b', 's', 'is', 'sc', 'y', 'wpa', 'pnr5')	0.7973	0.0227
('m', 'b', 's', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.7973	0.0217
('b', 'is', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7973	0.0224
('m', 'b', 's', 'is', 'sc', 'y', 'wpa', 'pv')	0.7973	0.0212
('b', 's', 'is', 'sc', 'vc', 'wpa', 'pv')	0.7972	0.0238
('m', 'b', 's', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7971	0.0205
('b', 's', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.7971	0.0223
('m', 'b', 's', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7971	0.0212
('b', 's', 'is', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.7971	0.0224
('m', 'b', 's', 'rt', 'sc', 'lr', 'wpa')	0.797	0.0219
('m', 'b', 's', 'is', 'sc', 'lr', 'wpa', 'pnr5')	0.797	0.0222
('m', 'b', 's', 'is', 'sc', 'lr', 'pv', 'pnr5')	0.797	0.0226
('m', 'b', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.797	0.0217
('m', 'b', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7969	0.0233
('b', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7969	0.0215
('b', 's', 'is', 'sc', 'lr', 'pv', 'pnr5')	0.7969	0.0222
('m', 'b', 's', 'is', 'sc', 'vc', 'wpa')	0.7969	0.0213
('m', 'b', 's', 'is', 'sc', 'y', 'pv', 'pnr5')	0.7968	0.0233

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'is', 'sc', 'lr', 'pv')	0.7968	0.0234
('m', 'b', 'is', 'sc', 'lr', 'pv', 'pnr5')	0.7966	0.0223
('m', 'b', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.7966	0.02
('m', 'b', 's', 'is', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.7966	0.0235
('m', 'b', 's', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.7965	0.0216
('b', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7965	0.0209
('m', 'b', 'is', 'sc', 'vc', 'wpa', 'pnr5')	0.7964	0.0232
('m', 'b', 's', 'rt', 'sc', 'lr', 'pv')	0.7963	0.0209
('m', 'b', 's', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.7963	0.0203
('m', 'b', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.7962	0.0231
('m', 'b', 's', 'sc', 'y', 'pv', 'pnr5')	0.7961	0.0212
('m', 'b', 's', 'is', 'sc', 'vc', 'wpa', 'pv')	0.7961	0.0246
('m', 'b', 'is', 'rt', 'sc', 'lr', 'vc')	0.7961	0.0231
('m', 'b', 'is', 'sc', 'lr', 'wpa', 'pnr5')	0.796	0.0203
('m', 'b', 'is', 'sc', 'y', 'wpa', 'pv')	0.796	0.0216
('m', 'b', 's', 'is', 'sc', 'lr', 'wpa', 'pv')	0.796	0.0217
('m', 'b', 's', 'is', 'sc', 'y', 'pv')	0.796	0.0214
('b', 's', 'is', 'sc', 'y', 'wpa', 'pv')	0.796	0.0223
('b', 's', 'is', 'sc', 'lr', 'wpa', 'pnr5')	0.796	0.0232
('b', 's', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.796	0.0221
('b', 's', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.7959	0.022
('b', 's', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.7958	0.0207
('m', 'b', 's', 'sc', 'lr', 'vc', 'pnr5')	0.7957	0.02
('m', 'b', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.7957	0.0235
('m', 'b', 's', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.7956	0.0217
('m', 'b', 's', 'is', 'rt', 'sc', 'wpa')	0.7955	0.0227
('m', 'b', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.7955	0.0214
('m', 'b', 's', 'rt', 'sc', 'lr', 'vc')	0.7955	0.0223
('m', 'b', 's', 'is', 'sc', 'lr', 'pnr5')	0.7954	0.0207
('b', 's', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.7953	0.0223

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	avg_score	std_dev
('b', 's', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.7953	0.0213
('m', 'b', 's', 'is', 'sc', 'lr', 'wpa')	0.7953	0.023
('b', 's', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7952	0.0209
('m', 'b', 's', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.7952	0.0225
('m', 'b', 's', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7952	0.0211
('m', 'b', 's', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.795	0.0203
('m', 'b', 's', 'sc', 'y', 'wpa', 'pnr5')	0.795	0.0235
('m', 'b', 'is', 'sc', 'wpa', 'pv', 'pnr5')	0.7948	0.021
('b', 'is', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7948	0.023
('m', 'b', 's', 'is', 'sc', 'pv', 'pnr5')	0.7948	0.0219
('m', 'b', 'is', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7947	0.0223
('m', 'b', 's', 'rt', 'sc', 'pv', 'pnr5')	0.7947	0.0222
('m', 'b', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.7947	0.0221
('b', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7945	0.0202
('b', 's', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7944	0.0208
('m', 'b', 's', 'rt', 'sc', 'vc', 'wpa')	0.7943	0.0219
('b', 's', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7943	0.0204
('b', 's', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.7942	0.0203
('m', 'b', 'is', 'sc', 'lr', 'wpa', 'pv')	0.7942	0.0213
('m', 'b', 's', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.7942	0.0188
('m', 'b', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.7942	0.0219
('m', 'b', 's', 'sc', 'lr', 'vc', 'pv')	0.7941	0.0213
('m', 'b', 's', 'sc', 'y', 'wpa', 'pv')	0.7941	0.0218
('m', 'b', 's', 'is', 'sc', 'wpa', 'pnr5')	0.7941	0.0212
('m', 'b', 's', 'sc', 'vc', 'pv', 'pnr5')	0.794	0.0232
('m', 'b', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.7939	0.0213
('b', 's', 'is', 'sc', 'wpa', 'pv', 'pnr5')	0.7939	0.0223
('m', 'b', 's', 'is', 'sc', 'wpa', 'pv', 'pnr5')	0.7938	0.0221
('m', 'b', 's', 'is', 'sc', 'lr', 'vc')	0.7937	0.0224
('m', 'b', 's', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7936	0.0207

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.7936	0.0212
('m', 'b', 's', 'rt', 'sc', 'wpa', 'pv')	0.7935	0.0198
('m', 'b', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7934	0.021
('m', 'b', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7934	0.0212
('m', 'b', 's', 'rt', 'sc', 'wpa', 'pnr5')	0.7934	0.0202
('m', 'b', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7933	0.0188
('m', 'b', 's', 'sc', 'lr', 'wpa', 'pv')	0.7927	0.0233
('m', 'b', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7924	0.0203
('m', 'b', 's', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7922	0.0209
('m', 'b', 's', 'is', 'sc', 'wpa', 'pv')	0.7922	0.0229
('m', 'b', 's', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.7921	0.0206
('b', 's', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.792	0.0223
('m', 'b', 's', 'sc', 'lr', 'pv', 'pnr5')	0.7919	0.0223
('m', 'b', 's', 'sc', 'lr', 'wpa', 'pnr5')	0.7919	0.0216
('m', 'b', 's', 'sc', 'lr', 'vc', 'wpa')	0.7918	0.0216
('m', 'b', 's', 'sc', 'vc', 'wpa', 'pv')	0.7917	0.0217
('m', 'b', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7916	0.0235
('m', 'b', 's', 'sc', 'vc', 'wpa', 'pnr5')	0.7915	0.0211
('b', 's', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.7909	0.0209
('m', 'b', 's', 'sc', 'wpa', 'pv', 'pnr5')	0.7881	0.0219
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc')	0.7048	0.0199
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.7039	0.0209
('m', 'is', 'rt', 'sc', 'y', 'lr', 'vc')	0.7039	0.0188
('m', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.7036	0.0214
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.7034	0.0228
('s', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.7034	0.0212
('is', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.7028	0.0189
('is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.7028	0.0229
('is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.7027	0.0186
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.7026	0.0217

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.7025	0.0206
('m', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.7024	0.0223
('m', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.702	0.0241
('m', 's', 'is', 'sc', 'lr', 'vc', 'pnr5')	0.702	0.0229
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.7018	0.0233
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7016	0.023
('s', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.7016	0.0253
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.7015	0.022
('m', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.7014	0.0221
('s', 'is', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.7014	0.0174
('m', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.7013	0.0199
('m', 'is', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.7013	0.0209
('m', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.7013	0.0216
('m', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7012	0.024
('is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.7012	0.0235
('m', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.7011	0.0222
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.7011	0.0236
('m', 's', 'is', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.7011	0.0234
('m', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.701	0.0232
('m', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.7009	0.0214
('is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7007	0.0218
('m', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.7007	0.0224
('m', 's', 'rt', 'sc', 'y', 'lr', 'vc')	0.7007	0.0217
('is', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7007	0.0207
('s', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.7006	0.0222
('s', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.7005	0.0222
('m', 'is', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.7005	0.0215
('m', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7005	0.0225
('m', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.7005	0.0247
('m', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.7005	0.0192

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7005	0.0211
('m', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.7003	0.0214
('m', 'is', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7002	0.0199
('s', 'is', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.7002	0.0187
('m', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.7002	0.0229
('is', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.7002	0.0229
('m', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.7	0.022
('m', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.7	0.0193
('m', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.7	0.022
('m', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6998	0.0251
('m', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6998	0.0229
('m', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6998	0.0221
('rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6997	0.0252
('s', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.6997	0.0229
('s', 'is', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.6996	0.0213
('s', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6995	0.0217
('m', 'is', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.6994	0.0225
('is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6994	0.0216
('m', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6993	0.022
('rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6993	0.024
('m', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6993	0.0228
('s', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6993	0.0222
('m', 's', 'rt', 'sc', 'lr', 'vc', 'pnr5')	0.6992	0.0231
('m', 's', 'is', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.6992	0.0224
('s', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.6992	0.0206
('rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.699	0.0212
('s', 'is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.699	0.0223
('s', 'rt', 'sc', 'y', 'lr', 'vc', 'pv')	0.6989	0.0242
('m', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6989	0.0232
('m', 's', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6988	0.0229

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('s', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa')	0.6987	0.0206
('m', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6987	0.0236
('is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6987	0.0231
('m', 's', 'is', 'rt', 'sc', 'lr', 'vc')	0.6987	0.0221
('is', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6986	0.0217
('m', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6985	0.0238
('m', 's', 'is', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6985	0.0219
('m', 'is', 'rt', 'sc', 'lr', 'vc', 'pv')	0.6985	0.0209
('m', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6983	0.0201
('s', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6983	0.023
('s', 'is', 'rt', 'sc', 'y', 'lr', 'vc')	0.6981	0.0198
('m', 'is', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6981	0.0211
('m', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6981	0.0242
('s', 'rt', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.698	0.0229
('m', 's', 'is', 'rt', 'sc', 'y', 'vc')	0.6977	0.0232
('m', 's', 'is', 'rt', 'sc', 'vc', 'pnr5')	0.6977	0.0249
('m', 's', 'is', 'sc', 'y', 'vc', 'pnr5')	0.6974	0.0238
('s', 'rt', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.6973	0.0218
('s', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6971	0.0217
('rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6971	0.0226
('m', 's', 'rt', 'sc', 'y', 'vc', 'wpa')	0.6969	0.0224
('m', 's', 'is', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.6969	0.0239
('m', 's', 'is', 'rt', 'sc', 'lr', 'vc', 'pv')	0.6968	0.0227
('m', 'is', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.6967	0.0208
('m', 's', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6967	0.0229
('is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6966	0.022
('m', 's', 'is', 'sc', 'lr', 'vc', 'pv')	0.6966	0.0236
('m', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6966	0.0216
('m', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.6965	0.0216
('m', 's', 'is', 'rt', 'sc', 'y', 'vc', 'pv')	0.6964	0.0252

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 's', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.6964	0.0248
('rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6962	0.0203
('s', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.6962	0.0204
('m', 's', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6962	0.0262
('s', 'is', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.6962	0.0226
('m', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6961	0.0233
('m', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6961	0.0237
('m', 's', 'is', 'sc', 'y', 'lr', 'vc')	0.6961	0.0225
('is', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6961	0.0215
('s', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.696	0.024
('m', 's', 'is', 'sc', 'y', 'lr', 'vc', 'pv')	0.696	0.0237
('s', 'rt', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6959	0.0234
('s', 'is', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6959	0.0246
('m', 's', 'rt', 'sc', 'lr', 'vc', 'pv')	0.6959	0.0227
('m', 'is', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.6959	0.0234
('m', 'is', 'rt', 'sc', 'y', 'vc', 'pv')	0.6959	0.0247
('m', 's', 'is', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.6959	0.0201
('m', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa')	0.6959	0.0201
('s', 'is', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.6958	0.0234
('m', 's', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.6957	0.0206
('s', 'is', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6957	0.0219
('m', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6957	0.0234
('s', 'is', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.6956	0.0221
('m', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6955	0.0236
('s', 'is', 'rt', 'sc', 'lr', 'vc', 'pv')	0.6955	0.0193
('m', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6954	0.0234
('m', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6954	0.0249
('m', 's', 'is', 'sc', 'y', 'vc', 'pv')	0.6954	0.0248
('m', 's', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.6954	0.0246
('m', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6953	0.0213

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	avg_score	std_dev
('m', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.6953	0.0225
('s', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6952	0.0256
('m', 's', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6952	0.0235
('m', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6951	0.0215
('m', 's', 'rt', 'sc', 'lr', 'vc', 'wpa')	0.695	0.0215
('m', 'is', 'sc', 'y', 'lr', 'vc', 'pv')	0.695	0.0247
('s', 'is', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.695	0.0246
('m', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6949	0.0242
('m', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6949	0.0237
('is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6949	0.025
('m', 's', 'is', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.6949	0.0267
('s', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.6948	0.0197
('m', 's', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6948	0.0219
('m', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6948	0.0216
('m', 's', 'is', 'rt', 'sc', 'vc', 'pv')	0.6948	0.0223
('is', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.6948	0.0261
('s', 'rt', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.6947	0.0215
('m', 's', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.6947	0.0209
('m', 's', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6947	0.0244
('s', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6947	0.0224
('s', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6947	0.0217
('m', 's', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6946	0.0244
('m', 's', 'sc', 'y', 'lr', 'vc', 'pv')	0.6946	0.0237
('m', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6945	0.0237
('m', 's', 'rt', 'sc', 'y', 'vc', 'pv')	0.6945	0.0256
('m', 's', 'sc', 'lr', 'vc', 'pv', 'pnr5')	0.6945	0.0217
('m', 's', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6945	0.022
('s', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6944	0.0233
('m', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa')	0.6944	0.0222
('m', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6944	0.0209

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 's', 'is', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.6943	0.0231
('is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6943	0.0246
('is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6943	0.0209
('m', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6943	0.0212
('m', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6942	0.0228
('m', 's', 'sc', 'y', 'lr', 'vc', 'pnr5')	0.6942	0.0209
('s', 'is', 'rt', 'sc', 'y', 'vc', 'wpa')	0.6942	0.0212
('m', 's', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6941	0.0222
('m', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.6941	0.0234
('m', 's', 'sc', 'lr', 'vc', 'wpa', 'pnr5')	0.694	0.019
('m', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6939	0.0216
('m', 's', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6939	0.0216
('m', 's', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6938	0.022
('m', 'is', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.6938	0.0255
('m', 's', 'is', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6937	0.0264
('m', 's', 'is', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.6937	0.0233
('m', 's', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.6937	0.0261
('m', 's', 'is', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.6937	0.0234
('m', 's', 'is', 'rt', 'sc', 'vc', 'wpa')	0.6937	0.0218
('m', 's', 'is', 'sc', 'lr', 'vc', 'wpa')	0.6935	0.0216
('s', 'is', 'rt', 'sc', 'y', 'vc', 'pnr5')	0.6935	0.024
('s', 'is', 'sc', 'y', 'lr', 'vc', 'pv')	0.6935	0.0221
('m', 'is', 'rt', 'sc', 'y', 'vc', 'wpa')	0.6935	0.0233
('is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6933	0.0224
('is', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6933	0.0222
('m', 's', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6933	0.0248
('m', 's', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6933	0.0209
('m', 's', 'rt', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6932	0.0204
('m', 's', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6932	0.0237
('m', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6932	0.0249

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6931	0.024
('s', 'is', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.6931	0.0227
('s', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.6931	0.023
('m', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.693	0.0215
('m', 's', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6929	0.0238
('m', 's', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6929	0.0243
('s', 'sc', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6928	0.0242
('m', 's', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.6928	0.0203
('s', 'is', 'rt', 'sc', 'y', 'vc', 'pv')	0.6928	0.0265
('s', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6926	0.0261
('m', 's', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6925	0.0229
('s', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6925	0.0233
('m', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6925	0.0236
('s', 'is', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.6925	0.0256
('s', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6924	0.0236
('m', 'is', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.6924	0.0247
('s', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6924	0.0253
('is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6924	0.0227
('sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6923	0.0251
('s', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6923	0.0236
('s', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6923	0.023
('s', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv')	0.6923	0.0234
('s', 'is', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6922	0.0263
('m', 'rt', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.692	0.0269
('s', 'sc', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.692	0.022
('m', 's', 'sc', 'y', 'lr', 'vc', 'wpa')	0.692	0.0231
('m', 'is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.692	0.023
('m', 's', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.692	0.0256
('m', 'is', 'sc', 'y', 'vc', 'pv', 'pnr5')	0.6919	0.0232
('m', 's', 'is', 'sc', 'vc', 'pv', 'pnr5')	0.6918	0.0228

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'is', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6917	0.0252
('m', 's', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.6916	0.0236
('rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6916	0.0248
('s', 'is', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.6916	0.0213
('m', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6916	0.0248
('m', 's', 'is', 'sc', 'y', 'vc', 'wpa')	0.6916	0.0241
('is', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6915	0.0236
('m', 'is', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.6915	0.0211
('s', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6914	0.0249
('s', 'sc', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6913	0.0221
('s', 'rt', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6912	0.0257
('m', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.691	0.0219
('m', 's', 'is', 'sc', 'vc', 'wpa', 'pnr5')	0.691	0.0249
('is', 'sc', 'y', 'lr', 'vc', 'wpa', 'pv')	0.691	0.0241
('m', 's', 'is', 'sc', 'vc', 'wpa', 'pv')	0.6909	0.0252
('m', 's', 'is', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6909	0.0236
('s', 'is', 'sc', 'y', 'lr', 'vc', 'wpa')	0.6909	0.0217
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.6908	0.0229
('m', 's', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6908	0.0236
('m', 'is', 'sc', 'y', 'lr', 'vc', 'wpa')	0.6908	0.022
('s', 'is', 'sc', 'lr', 'vc', 'wpa', 'pv')	0.6906	0.0233
('m', 'is', 'rt', 'sc', 'vc', 'pv', 'pnr5')	0.6906	0.0232
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6906	0.0224
('m', 's', 'rt', 'sc', 'vc', 'wpa', 'pnr5')	0.6905	0.0265
('m', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6905	0.0233
('m', 's', 'is', 'sc', 'y', 'lr', 'pnr5')	0.6905	0.0257
('m', 'is', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6904	0.022
('m', 's', 'is', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6904	0.0234
('s', 'is', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6903	0.0238
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'pv')	0.6903	0.0241

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('s', 'is', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6903	0.0232
('s', 'is', 'sc', 'y', 'vc', 'wpa', 'pv')	0.6901	0.0233
('is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.69	0.023
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.69	0.0238
('m', 's', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6899	0.0242
('s', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6897	0.023
('m', 'is', 'rt', 'sc', 'y', 'lr', 'pv')	0.6896	0.0228
('s', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6896	0.0228
('s', 'is', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6896	0.0216
('is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6895	0.0241
('m', 's', 'is', 'rt', 'sc', 'y', 'lr')	0.6893	0.0206
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6893	0.0229
('m', 's', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6891	0.0249
('is', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6891	0.0217
('m', 's', 'is', 'sc', 'lr', 'wpa', 'pnr5')	0.6891	0.0247
('m', 'is', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.689	0.0237
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6889	0.023
('m', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6889	0.0247
('m', 'is', 'sc', 'y', 'vc', 'wpa', 'pnr5')	0.6888	0.0222
('m', 'rt', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6888	0.0226
('s', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6888	0.0214
('m', 's', 'is', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.6888	0.0234
('s', 'is', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6887	0.0232
('m', 's', 'is', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6885	0.0247
('m', 's', 'is', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6885	0.0253
('m', 's', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6884	0.0249
('m', 'is', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.6884	0.0238
('m', 'is', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.688	0.0233
('s', 'is', 'rt', 'sc', 'y', 'lr', 'wpa')	0.6877	0.0202
('m', 'is', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6877	0.0249

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'is', 'rt', 'sc', 'y', 'lr', 'wpa')	0.6877	0.0223
('m', 'is', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6876	0.0247
('s', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6874	0.0226
('m', 's', 'is', 'rt', 'sc', 'y', 'lr', 'wpa')	0.6874	0.0239
('s', 'is', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6874	0.0252
('is', 'sc', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6872	0.0236
('m', 's', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6871	0.0225
('s', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6871	0.0203
('m', 's', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6871	0.0226
('m', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6869	0.0223
('s', 'is', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.6869	0.0193
('s', 'is', 'rt', 'sc', 'y', 'lr', 'pv')	0.6869	0.0228
('m', 's', 'rt', 'sc', 'y', 'lr', 'pv')	0.6869	0.0237
('m', 's', 'is', 'rt', 'sc', 'lr', 'pv')	0.6869	0.0241
('m', 's', 'is', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6868	0.0262
('m', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6867	0.022
('m', 'is', 'sc', 'vc', 'wpa', 'pv', 'pnr5')	0.6867	0.0254
('m', 's', 'rt', 'sc', 'y', 'lr', 'pnr5')	0.6867	0.0215
('m', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6866	0.0227
('m', 's', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6865	0.0245
('m', 'is', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6864	0.0263
('m', 's', 'is', 'sc', 'lr', 'pv', 'pnr5')	0.6864	0.0262
('rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6861	0.0212
('m', 'is', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6861	0.0237
('m', 's', 'is', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.686	0.0226
('m', 's', 'is', 'rt', 'sc', 'lr', 'pnr5')	0.6859	0.024
('is', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6859	0.0247
('m', 's', 'is', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6858	0.0246
('is', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6857	0.0223
('m', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6852	0.026

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('s', 'is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6852	0.0234
('m', 's', 'is', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6851	0.0229
('m', 'rt', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6849	0.0228
('m', 's', 'is', 'sc', 'lr', 'wpa', 'pv')	0.6847	0.0265
('m', 's', 'is', 'sc', 'y', 'lr', 'pv')	0.6847	0.0256
('s', 'rt', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6847	0.0203
('s', 'is', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6847	0.0255
('m', 's', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6846	0.0224
('s', 'is', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6846	0.0242
('is', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6845	0.0254
('m', 's', 'is', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6844	0.0265
('m', 's', 'is', 'rt', 'sc', 'y', 'pnr5')	0.6842	0.0244
('s', 'is', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6841	0.0253
('m', 's', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.6841	0.026
('m', 'is', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6839	0.0248
('s', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6838	0.0196
('s', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6838	0.0213
('m', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.6835	0.0233
('s', 'is', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.6834	0.0204
('m', 's', 'sc', 'y', 'lr', 'pv', 'pnr5')	0.6834	0.0254
('m', 's', 'rt', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6833	0.0242
('s', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6832	0.0225
('m', 's', 'is', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.683	0.0288
('m', 's', 'is', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.6828	0.026
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'vc')	0.6826	0.0225
('m', 's', 'rt', 'sc', 'lr', 'wpa', 'pnr5')	0.6826	0.0235
('s', 'is', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6825	0.0249
('m', 's', 'sc', 'y', 'lr', 'wpa', 'pnr5')	0.6824	0.0235
('m', 'is', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.6822	0.0246
('m', 's', 'is', 'rt', 'sc', 'lr', 'wpa')	0.6822	0.0236

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 's', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6821	0.0263
('m', 's', 'is', 'sc', 'y', 'lr', 'wpa')	0.682	0.0263
('s', 'is', 'sc', 'y', 'lr', 'wpa', 'pv')	0.6817	0.0262
('m', 'b', 's', 'is', 'rt', 'y', 'vc', 'pnr5')	0.6817	0.0282
('m', 'is', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.6816	0.0244
('m', 'is', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6814	0.0258
('m', 's', 'rt', 'sc', 'lr', 'pv', 'pnr5')	0.6813	0.0213
('m', 's', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6812	0.0241
('m', 'b', 'is', 'y', 'lr', 'vc', 'pnr5')	0.6812	0.026
('m', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6811	0.0258
('m', 's', 'rt', 'sc', 'y', 'lr', 'wpa')	0.6811	0.0223
('m', 's', 'is', 'rt', 'sc', 'y', 'pv')	0.681	0.0267
('m', 's', 'is', 'rt', 'sc', 'pv', 'pnr5')	0.681	0.0252
('m', 'b', 'is', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6808	0.0228
('m', 'b', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6808	0.0282
('m', 's', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6807	0.0244
('m', 's', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.6805	0.0243
('s', 'is', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6804	0.0243
('m', 'b', 'is', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.6804	0.0234
('m', 'b', 'is', 'rt', 'lr', 'vc', 'pnr5')	0.6803	0.0222
('m', 'b', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6803	0.025
('m', 'b', 's', 'is', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6803	0.0274
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.6802	0.0271
('m', 's', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.6802	0.0237
('m', 'b', 'is', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6801	0.025
('m', 'b', 'is', 'rt', 'y', 'lr', 'vc', 'pv')	0.68	0.0232
('m', 'b', 's', 'is', 'y', 'lr', 'vc', 'wpa')	0.68	0.0274
('m', 's', 'is', 'rt', 'sc', 'wpa', 'pnr5')	0.68	0.0276
('m', 'b', 's', 'is', 'y', 'lr', 'vc')	0.68	0.0249
('s', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6799	0.0225

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	avg_score	std_dev
('m', 's', 'is', 'sc', 'y', 'pv', 'pnr5')	0.6799	0.026
('m', 'b', 'is', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6799	0.028
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6798	0.028
('m', 'b', 'is', 'y', 'lr', 'vc', 'pv')	0.6798	0.026
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6798	0.0258
('m', 'sc', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6797	0.0244
('m', 'b', 's', 'is', 'rt', 'y', 'vc')	0.6796	0.0265
('m', 'b', 's', 'is', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6796	0.0257
('m', 'b', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6795	0.026
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'pv')	0.6795	0.025
('m', 'b', 'is', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6794	0.027
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6793	0.0245
('m', 's', 'is', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6793	0.0234
('s', 'is', 'rt', 'sc', 'lr', 'wpa', 'pv')	0.6793	0.0205
('m', 'b', 'is', 'rt', 'y', 'vc', 'pnr5')	0.6792	0.0263
('is', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.679	0.0255
('m', 'b', 'is', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.679	0.0237
('m', 's', 'is', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.679	0.025
('m', 's', 'is', 'rt', 'sc', 'y', 'wpa', 'pv')	0.679	0.0253
('m', 'b', 'is', 'y', 'lr', 'vc', 'wpa')	0.679	0.0284
('m', 'b', 's', 'is', 'y', 'lr', 'vc', 'wpa', 'pv')	0.679	0.0258
('m', 's', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.6788	0.0252
('m', 'is', 'rt', 'sc', 'y', 'wpa', 'pv')	0.6787	0.0275
('m', 'b', 's', 'is', 'y', 'lr', 'vc', 'pv')	0.6787	0.0259
('s', 'is', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.6786	0.0249
('m', 'is', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.6785	0.0266
('m', 'b', 's', 'is', 'rt', 'lr', 'vc', 'pnr5')	0.6785	0.0203
('m', 's', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6784	0.0247
('m', 'b', 's', 'is', 'y', 'lr', 'vc', 'pnr5')	0.6784	0.0249
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6784	0.0251

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'is', 'rt', 'y', 'vc', 'wpa')	0.6783	0.0256
('m', 'b', 's', 'is', 'rt', 'lr', 'vc', 'pv')	0.6782	0.02
('m', 'b', 'is', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6782	0.0247
('m', 'b', 'is', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6782	0.0264
('m', 'b', 'is', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6781	0.0245
('s', 'rt', 'sc', 'lr', 'wpa', 'pv', 'pnr5')	0.678	0.0192
('s', 'is', 'rt', 'sc', 'y', 'wpa', 'pnr5')	0.678	0.0239
('m', 's', 'rt', 'sc', 'y', 'pv', 'pnr5')	0.6779	0.0257
('m', 'b', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.6777	0.0241
('m', 'b', 'is', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6775	0.0228
('m', 'b', 'is', 'rt', 'lr', 'vc', 'pv')	0.6775	0.0217
('m', 'b', 'is', 'rt', 'y', 'lr', 'vc')	0.6775	0.0239
('m', 'b', 'is', 'y', 'vc', 'pv', 'pnr5')	0.6775	0.0256
('m', 'b', 's', 'is', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6775	0.0289
('m', 's', 'rt', 'sc', 'y', 'wpa', 'pv')	0.6774	0.0238
('m', 'b', 's', 'is', 'lr', 'vc', 'pv', 'pnr5')	0.6774	0.025
('m', 'b', 's', 'is', 'rt', 'lr', 'vc')	0.6774	0.0213
('m', 'b', 'is', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6773	0.0273
('m', 'is', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6772	0.0256
('m', 'b', 's', 'is', 'y', 'vc', 'pnr5')	0.6772	0.0246
('m', 'b', 'is', 'rt', 'y', 'vc', 'pv')	0.6772	0.0242
('m', 'b', 's', 'is', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6771	0.0265
('m', 'b', 'is', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.677	0.025
('m', 'b', 's', 'is', 'rt', 'lr', 'vc', 'wpa')	0.677	0.0259
('m', 'b', 's', 'is', 'rt', 'y', 'vc', 'pv')	0.6769	0.0248
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6769	0.0247
('m', 'b', 's', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.6769	0.023
('m', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6768	0.0271
('m', 'b', 'is', 'lr', 'vc', 'wpa', 'pnr5')	0.6768	0.0243
('m', 'b', 'is', 'y', 'vc', 'wpa', 'pnr5')	0.6768	0.0246

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'is', 'y', 'vc', 'wpa')	0.6767	0.0272
('m', 'b', 's', 'is', 'lr', 'vc', 'pnr5')	0.6767	0.0253
('m', 'b', 's', 'is', 'rt', 'y', 'vc', 'wpa', 'pv')	0.6764	0.0276
('m', 's', 'is', 'sc', 'y', 'wpa', 'pnr5')	0.6764	0.0252
('m', 'b', 's', 'is', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.6764	0.0241
('m', 's', 'is', 'rt', 'sc', 'y', 'wpa')	0.6763	0.0217
('m', 'b', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6763	0.0222
('m', 'b', 's', 'is', 'lr', 'vc', 'pv')	0.6761	0.0226
('b', 's', 'is', 'rt', 'y', 'vc', 'pnr5')	0.6761	0.0245
('m', 's', 'is', 'sc', 'y', 'wpa', 'pv')	0.676	0.024
('m', 'b', 'is', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.6759	0.025
('m', 'b', 's', 'is', 'rt', 'vc', 'pv')	0.6759	0.0238
('m', 'b', 's', 'is', 'y', 'vc', 'pv')	0.6759	0.0275
('m', 'b', 'is', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6758	0.0224
('m', 'b', 's', 'is', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6758	0.0281
('m', 's', 'is', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6757	0.025
('m', 'b', 's', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6757	0.0242
('m', 'b', 's', 'is', 'lr', 'vc', 'wpa')	0.6757	0.0274
('m', 'b', 'is', 'lr', 'vc', 'wpa', 'pv')	0.6757	0.0231
('s', 'is', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6756	0.0254
('s', 'rt', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6756	0.0236
('m', 'b', 's', 'is', 'lr', 'vc', 'wpa', 'pnr5')	0.6755	0.0247
('b', 'is', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6755	0.0273
('m', 'b', 's', 'is', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6755	0.0263
('m', 'b', 'is', 'rt', 'y', 'vc', 'wpa', 'pv')	0.6755	0.0253
('m', 'b', 's', 'is', 'y', 'vc', 'pv', 'pnr5')	0.6754	0.0258
('m', 'b', 's', 'is', 'lr', 'vc', 'wpa', 'pv')	0.6754	0.0266
('s', 'is', 'rt', 'sc', 'y', 'wpa', 'pv')	0.6752	0.0263
('m', 'b', 'is', 'rt', 'vc', 'pv', 'pnr5')	0.6751	0.0249
('b', 's', 'is', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6751	0.0281

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'is', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.675	0.027
('m', 'b', 's', 'is', 'vc', 'pv', 'pnr5')	0.6749	0.0245
('m', 'b', 's', 'is', 'y', 'vc', 'wpa', 'pnr5')	0.6749	0.0273
('m', 'b', 's', 'is', 'rt', 'vc', 'pnr5')	0.6747	0.0214
('s', 'is', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.6746	0.0252
('m', 'b', 'is', 'lr', 'vc', 'pv', 'pnr5')	0.6745	0.0232
('b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6744	0.0265
('m', 'b', 'is', 'rt', 'y', 'vc', 'wpa')	0.6744	0.0284
('m', 's', 'is', 'sc', 'wpa', 'pv', 'pnr5')	0.6743	0.0233
('m', 'b', 'is', 'rt', 'lr', 'vc', 'wpa')	0.6742	0.0246
('m', 's', 'is', 'rt', 'sc', 'wpa', 'pv')	0.6742	0.0253
('m', 'b', 's', 'is', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6742	0.0245
('m', 'b', 's', 'is', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6742	0.0265
('m', 'b', 's', 'is', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.6741	0.0248
('b', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6741	0.0275
('b', 'is', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.6741	0.0264
('m', 's', 'rt', 'sc', 'wpa', 'pv', 'pnr5')	0.6741	0.0262
('m', 'b', 'is', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.6739	0.0229
('b', 'is', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6737	0.0266
('b', 's', 'is', 'rt', 'y', 'lr', 'vc')	0.6737	0.0234
('m', 'b', 's', 'is', 'y', 'vc', 'wpa', 'pv')	0.6737	0.0274
('m', 'b', 's', 'is', 'rt', 'vc', 'wpa')	0.6736	0.0245
('b', 's', 'is', 'y', 'vc', 'pv', 'pnr5')	0.6736	0.0282
('b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.6736	0.0246
('b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6734	0.0257
('m', 'b', 'is', 'y', 'vc', 'wpa', 'pv')	0.6731	0.026
('b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6731	0.0236
('m', 'b', 's', 'is', 'vc', 'wpa', 'pv')	0.6731	0.0264
('m', 'b', 'is', 'rt', 'vc', 'wpa', 'pv')	0.673	0.0212
('b', 's', 'is', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6729	0.0283

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'is', 'rt', 'vc', 'pv', 'pnr5')	0.6726	0.0254
('b', 'is', 'rt', 'y', 'lr', 'vc', 'pv')	0.6724	0.0265
('m', 'b', 'is', 'rt', 'vc', 'wpa', 'pnr5')	0.6724	0.0226
('b', 'is', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6723	0.0272
('m', 'b', 's', 'is', 'rt', 'vc', 'wpa', 'pnr5')	0.6721	0.0235
('b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'pv')	0.6721	0.0281
('b', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.672	0.0246
('b', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.672	0.0244
('m', 'b', 's', 'is', 'rt', 'vc', 'wpa', 'pv')	0.6719	0.0224
('b', 'is', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6718	0.028
('b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6718	0.0251
('b', 's', 'is', 'y', 'lr', 'vc', 'wpa')	0.6718	0.0268
('b', 's', 'is', 'y', 'lr', 'vc', 'pnr5')	0.6718	0.0278
('b', 's', 'is', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6716	0.0288
('b', 'is', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6713	0.0288
('b', 's', 'is', 'rt', 'y', 'vc', 'wpa')	0.6711	0.0283
('b', 's', 'is', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6711	0.0271
('b', 's', 'is', 'rt', 'y', 'vc', 'wpa', 'pv')	0.6711	0.0297
('b', 's', 'is', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.671	0.028
('b', 's', 'is', 'y', 'vc', 'wpa', 'pnr5')	0.6709	0.0287
('b', 's', 'is', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6709	0.0286
('m', 'b', 'is', 'vc', 'wpa', 'pv', 'pnr5')	0.6707	0.0256
('b', 'is', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6707	0.0269
('b', 's', 'is', 'rt', 'y', 'vc', 'pv')	0.6706	0.0285
('b', 'is', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6705	0.0275
('b', 'is', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.6705	0.0246
('m', 'b', 's', 'is', 'vc', 'wpa', 'pnr5')	0.6704	0.0226
('m', 'b', 's', 'is', 'vc', 'wpa', 'pv', 'pnr5')	0.6704	0.0262
('b', 's', 'is', 'lr', 'vc', 'pv', 'pnr5')	0.6703	0.0261
('b', 's', 'is', 'y', 'vc', 'wpa', 'pv')	0.6703	0.0291

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'is', 'y', 'lr', 'vc', 'pv')	0.6701	0.0295
('b', 'is', 'rt', 'y', 'vc', 'wpa', 'pv')	0.6698	0.0271
('b', 'is', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6695	0.0258
('b', 's', 'is', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6694	0.0255
('m', 's', 'sc', 'y', 'wpa', 'pv', 'pnr5')	0.6692	0.0277
('b', 'is', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6692	0.0266
('b', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6689	0.0222
('b', 'is', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6688	0.0275
('b', 's', 'is', 'rt', 'lr', 'vc', 'pnr5')	0.6688	0.0236
('b', 's', 'is', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6687	0.0281
('b', 's', 'is', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6685	0.0273
('b', 's', 'is', 'rt', 'vc', 'pv', 'pnr5')	0.6685	0.0277
('b', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6683	0.0261
('b', 's', 'is', 'rt', 'lr', 'vc', 'pv')	0.6683	0.0251
('b', 's', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6675	0.0243
('b', 's', 'is', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.6674	0.0229
('b', 's', 'is', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.6674	0.0277
('b', 's', 'is', 'rt', 'vc', 'wpa', 'pv')	0.667	0.0246
('b', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.6665	0.0259
('b', 'is', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6661	0.0246
('b', 's', 'is', 'lr', 'vc', 'wpa', 'pv')	0.6652	0.0274
('b', 's', 'is', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.665	0.026
('b', 's', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.6649	0.023
('b', 's', 'is', 'lr', 'vc', 'wpa', 'pnr5')	0.664	0.0275
('b', 's', 'is', 'rt', 'vc', 'wpa', 'pnr5')	0.6638	0.0273
('m', 'b', 's', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.6636	0.0239
('b', 's', 'is', 'vc', 'wpa', 'pv', 'pnr5')	0.6636	0.0283
('m', 'b', 's', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6631	0.0258
('m', 'b', 's', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6629	0.0258
('m', 'b', 's', 'y', 'lr', 'vc', 'pnr5')	0.6629	0.0237

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'is', 'rt', 'lr', 'vc', 'wpa')	0.6625	0.0259
('b', 'is', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6624	0.0265
('b', 'is', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.6623	0.0276
('m', 'b', 's', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6616	0.0253
('m', 'b', 's', 'y', 'lr', 'vc', 'wpa')	0.6607	0.0287
('m', 'b', 's', 'rt', 'y', 'lr', 'vc', 'pv')	0.6605	0.0226
('m', 'b', 's', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6602	0.0236
('m', 'b', 's', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6601	0.0209
('m', 'b', 's', 'y', 'lr', 'vc', 'pv')	0.6598	0.0241
('m', 'b', 's', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6595	0.0233
('m', 'b', 's', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6592	0.024
('m', 'b', 's', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6588	0.0258
('m', 'b', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.6587	0.0253
('m', 's', 'is', 'rt', 'y', 'vc', 'pnr5')	0.6584	0.0257
('m', 'b', 's', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6584	0.026
('m', 'b', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6579	0.0263
('m', 'b', 's', 'rt', 'y', 'lr', 'vc')	0.6579	0.0242
('m', 'b', 's', 'rt', 'y', 'vc', 'pv')	0.6575	0.024
('m', 's', 'is', 'y', 'lr', 'vc', 'pv')	0.6574	0.0266
('m', 'b', 's', 'rt', 'y', 'vc', 'pnr5')	0.6573	0.0259
('m', 'b', 's', 'lr', 'vc', 'wpa', 'pnr5')	0.657	0.0236
('m', 'b', 's', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6569	0.0223
('m', 'b', 's', 'y', 'vc', 'pv', 'pnr5')	0.6568	0.0269
('m', 'b', 'rt', 'y', 'lr', 'vc', 'pv')	0.6568	0.0238
('m', 'b', 's', 'y', 'vc', 'wpa', 'pv')	0.6568	0.0255
('m', 'b', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6566	0.0249
('m', 'b', 's', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6565	0.0229
('m', 'b', 's', 'lr', 'vc', 'wpa', 'pv')	0.6564	0.0253
('m', 's', 'is', 'rt', 'y', 'vc', 'pv')	0.6563	0.0266
('m', 'is', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6562	0.024

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6562	0.0264
('m', 'b', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6561	0.0259
('m', 'is', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.656	0.0256
('m', 'b', 's', 'rt', 'lr', 'vc', 'wpa')	0.6559	0.0269
('m', 's', 'is', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6558	0.0232
('m', 'is', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6558	0.0239
('m', 'b', 's', 'rt', 'lr', 'vc', 'pv')	0.6555	0.0215
('m', 'b', 's', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6553	0.0254
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6553	0.0254
('m', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6553	0.0257
('m', 'b', 's', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6553	0.0254
('m', 'b', 's', 'rt', 'lr', 'vc', 'pnr5')	0.6552	0.0227
('m', 'is', 'rt', 'y', 'lr', 'vc', 'pv')	0.6552	0.0237
('m', 'is', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6552	0.0257
('m', 'b', 's', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.6551	0.0213
('m', 's', 'is', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.6551	0.0253
('m', 's', 'is', 'rt', 'y', 'lr', 'vc')	0.6551	0.024
('m', 'b', 's', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6551	0.0265
('m', 'b', 's', 'rt', 'y', 'vc', 'wpa', 'pv')	0.655	0.0255
('m', 's', 'is', 'rt', 'y', 'lr', 'vc', 'pv')	0.655	0.0245
('m', 'b', 's', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6549	0.0241
('m', 'b', 's', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6546	0.0258
('m', 'b', 's', 'is', 'rt', 'y', 'pnr5')	0.6546	0.0292
('m', 'b', 's', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.6546	0.0241
('m', 'b', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6545	0.0255
('m', 'b', 's', 'y', 'vc', 'wpa', 'pnr5')	0.6545	0.0261
('m', 'b', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6545	0.0255
('b', 's', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6545	0.0286
('b', 's', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6544	0.0267
('m', 'b', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6542	0.0257

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'rt', 'y', 'vc', 'wpa')	0.6541	0.0241
('m', 's', 'is', 'y', 'lr', 'vc', 'pnr5')	0.6541	0.0268
('m', 'b', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6541	0.0239
('m', 'b', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.654	0.0279
('m', 'b', 's', 'lr', 'vc', 'pv', 'pnr5')	0.654	0.0217
('m', 'b', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.654	0.0267
('m', 'b', 's', 'is', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6537	0.0278
('m', 'b', 'is', 'y', 'lr', 'wpa', 'pnr5')	0.6537	0.0288
('m', 's', 'is', 'rt', 'y', 'vc', 'wpa', 'pv')	0.6537	0.0238
('m', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6536	0.0243
('m', 's', 'is', 'y', 'vc', 'pv', 'pnr5')	0.6536	0.0296
('m', 's', 'is', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6534	0.0256
('m', 's', 'is', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6533	0.0258
('m', 'is', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6533	0.0258
('m', 'b', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6532	0.024
('b', 's', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.6532	0.0258
('s', 'is', 'rt', 'y', 'lr', 'vc', 'pv')	0.6531	0.0274
('b', 's', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6531	0.0267
('m', 'b', 's', 'is', 'rt', 'y', 'wpa', 'pnr5')	0.6531	0.0288
('m', 'b', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.653	0.023
('m', 'is', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.653	0.0249
('is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6528	0.024
('m', 's', 'is', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6527	0.026
('m', 's', 'is', 'rt', 'lr', 'vc', 'pnr5')	0.6527	0.021
('m', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6526	0.0242
('m', 'b', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6526	0.0233
('b', 's', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6524	0.0247
('b', 's', 'rt', 'y', 'lr', 'vc', 'pv')	0.6523	0.0266
('m', 'b', 's', 'is', 'y', 'lr', 'pv', 'pnr5')	0.6523	0.027
('is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6523	0.026

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 's', 'is', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6523	0.025
('m', 's', 'is', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6522	0.0271
('m', 'b', 's', 'is', 'rt', 'y', 'pv', 'pnr5')	0.6522	0.0266
('b', 's', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6519	0.025
('m', 's', 'is', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6519	0.0254
('m', 'b', 'is', 'rt', 'y', 'pv', 'pnr5')	0.6519	0.0271
('b', 's', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6518	0.0229
('m', 's', 'is', 'rt', 'lr', 'vc', 'pv')	0.6517	0.0223
('m', 'b', 'is', 'rt', 'y', 'lr', 'pnr5')	0.6516	0.0276
('is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6516	0.0279
('b', 's', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6516	0.0268
('m', 'b', 'is', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6516	0.028
('m', 'is', 'rt', 'y', 'vc', 'wpa', 'pv')	0.6516	0.0228
('m', 'is', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6514	0.0255
('m', 'b', 'is', 'rt', 'y', 'wpa', 'pnr5')	0.6514	0.0266
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6513	0.0263
('m', 'b', 'is', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6513	0.0248
('m', 'b', 's', 'is', 'y', 'lr', 'wpa', 'pnr5')	0.6513	0.0274
('m', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6513	0.0248
('m', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6513	0.0244
('b', 's', 'rt', 'y', 'vc', 'wpa', 'pv')	0.6512	0.0264
('m', 'b', 'is', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6511	0.0281
('m', 'is', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6511	0.0293
('m', 'b', 'is', 'y', 'lr', 'pv', 'pnr5')	0.651	0.0265
('m', 'b', 'rt', 'y', 'vc', 'wpa', 'pv')	0.651	0.0249
('m', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.651	0.0273
('s', 'is', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.651	0.0257
('is', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6508	0.0276
('s', 'is', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6508	0.0249
('m', 's', 'is', 'y', 'lr', 'vc', 'wpa')	0.6507	0.0226

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'is', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6507	0.0243
('m', 's', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6506	0.0258
('s', 'is', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6506	0.025
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'pnr5')	0.6505	0.0276
('is', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6504	0.0268
('s', 'is', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6504	0.0286
('m', 'is', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6503	0.0267
('s', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6503	0.0272
('m', 's', 'is', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.6503	0.0239
('m', 'b', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6503	0.0252
('m', 's', 'is', 'lr', 'vc', 'pv', 'pnr5')	0.6503	0.0247
('m', 'b', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6502	0.025
('m', 'is', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6502	0.0269
('m', 's', 'is', 'y', 'vc', 'wpa', 'pnr5')	0.6502	0.0277
('b', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6501	0.0253
('s', 'is', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6501	0.0263
('m', 'b', 'is', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.65	0.0242
('m', 's', 'is', 'rt', 'y', 'vc', 'wpa')	0.6499	0.0236
('m', 'b', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.6499	0.0254
('b', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6499	0.0282
('m', 'b', 's', 'rt', 'vc', 'pv', 'pnr5')	0.6498	0.0256
('m', 'b', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6498	0.0261
('m', 'b', 's', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.6497	0.0222
('m', 'is', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6497	0.0252
('m', 'is', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.6497	0.0225
('m', 'b', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6496	0.0243
('m', 'b', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.6496	0.0249
('is', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6496	0.0281
('b', 's', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6495	0.025
('m', 's', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.6494	0.0259

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.6493	0.0254
('m', 'b', 's', 'is', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6493	0.0283
('b', 's', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6492	0.0252
('b', 's', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6492	0.0256
('m', 's', 'is', 'y', 'vc', 'wpa', 'pv')	0.649	0.0252
('m', 's', 'is', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.649	0.0262
('s', 'is', 'rt', 'y', 'vc', 'wpa', 'pv')	0.6489	0.0265
('m', 'b', 's', 'rt', 'vc', 'wpa', 'pnr5')	0.6488	0.0228
('m', 's', 'is', 'lr', 'vc', 'wpa', 'pv')	0.6488	0.0265
('m', 'b', 'is', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6487	0.0275
('s', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6487	0.0239
('m', 'b', 's', 'is', 'y', 'wpa', 'pnr5')	0.6487	0.0287
('m', 'b', 's', 'is', 'y', 'pv', 'pnr5')	0.6486	0.0295
('m', 'b', 's', 'rt', 'vc', 'wpa', 'pv')	0.6485	0.0246
('m', 'b', 's', 'is', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.6485	0.0244
('m', 'b', 's', 'is', 'y', 'lr', 'pnr5')	0.6485	0.0279
('m', 's', 'is', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6484	0.0267
('m', 's', 'is', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6484	0.0288
('m', 'b', 'is', 'y', 'wpa', 'pv', 'pnr5')	0.6483	0.0272
('b', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6483	0.0264
('s', 'is', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6483	0.0275
('s', 'is', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6482	0.027
('m', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.6481	0.0234
('b', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.648	0.0254
('m', 's', 'is', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6479	0.0242
('s', 'is', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6479	0.0265
('m', 'b', 's', 'is', 'rt', 'lr', 'pv', 'pnr5')	0.6478	0.0242
('s', 'is', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6478	0.0269
('b', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6476	0.0239
('b', 's', 'is', 'rt', 'y', 'wpa', 'pnr5')	0.6476	0.0266

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 's', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6475	0.0244
('m', 'b', 's', 'is', 'y', 'lr', 'pv')	0.6475	0.0266
('s', 'is', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6474	0.0255
('m', 'is', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6473	0.0267
('m', 's', 'is', 'rt', 'vc', 'pv', 'pnr5')	0.6473	0.0255
('m', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.647	0.022
('b', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6469	0.0255
('b', 's', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6467	0.0233
('m', 'b', 's', 'is', 'y', 'wpa', 'pv', 'pnr5')	0.6467	0.0282
('m', 'b', 'is', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.6467	0.0241
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'pv')	0.6467	0.0263
('m', 'b', 's', 'is', 'y', 'lr', 'wpa')	0.6466	0.024
('m', 's', 'is', 'lr', 'vc', 'wpa', 'pnr5')	0.6465	0.0277
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'wpa')	0.6465	0.0253
('b', 's', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6464	0.0237
('s', 'is', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6463	0.0291
('b', 'is', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6462	0.0261
('m', 'b', 'is', 'rt', 'lr', 'pv', 'pnr5')	0.6462	0.0223
('m', 'b', 'is', 'rt', 'lr', 'wpa', 'pnr5')	0.6462	0.0234
('s', 'is', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6462	0.023
('m', 'b', 's', 'is', 'rt', 'y', 'pv')	0.646	0.0252
('b', 's', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.646	0.02
('m', 's', 'is', 'rt', 'vc', 'wpa', 'pv')	0.646	0.0241
('b', 's', 'is', 'rt', 'y', 'pv', 'pnr5')	0.646	0.0303
('b', 's', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6458	0.0234
('m', 'b', 's', 'is', 'y', 'lr', 'wpa', 'pv')	0.6457	0.0265
('m', 'b', 's', 'is', 'rt', 'pv', 'pnr5')	0.6454	0.0254
('b', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6453	0.0211
('b', 's', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6453	0.025
('m', 'b', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.6452	0.0214

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'is', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.6452	0.0239
('m', 'b', 'is', 'y', 'lr', 'wpa', 'pv')	0.6452	0.0253
('m', 'b', 's', 'is', 'rt', 'lr', 'pnr5')	0.6451	0.0251
('m', 's', 'is', 'rt', 'vc', 'wpa', 'pnr5')	0.6451	0.0228
('b', 's', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.645	0.0251
('m', 's', 'is', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.645	0.024
('b', 's', 'is', 'y', 'lr', 'pv', 'pnr5')	0.6449	0.0261
('m', 's', 'is', 'vc', 'wpa', 'pv', 'pnr5')	0.6449	0.027
('s', 'is', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6449	0.0239
('m', 'b', 's', 'is', 'rt', 'y', 'wpa', 'pv')	0.6447	0.0261
('m', 's', 'is', 'rt', 'lr', 'vc', 'wpa')	0.6447	0.0238
('m', 'b', 's', 'is', 'rt', 'y', 'lr')	0.6442	0.0248
('m', 'b', 'is', 'rt', 'y', 'lr', 'pv')	0.6442	0.0252
('m', 'b', 's', 'is', 'rt', 'y', 'lr', 'wpa', 'pv')	0.6441	0.0259
('m', 'b', 's', 'vc', 'wpa', 'pv', 'pnr5')	0.6441	0.0234
('m', 'b', 's', 'is', 'lr', 'pv', 'pnr5')	0.6441	0.024
('b', 'is', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.644	0.0263
('s', 'is', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.644	0.0243
('m', 'b', 'is', 'rt', 'y', 'lr', 'wpa')	0.6439	0.0272
('b', 's', 'is', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6438	0.0254
('b', 'is', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6434	0.0247
('m', 'b', 's', 'is', 'lr', 'wpa', 'pv', 'pnr5')	0.6434	0.0235
('b', 's', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6434	0.0269
('b', 's', 'is', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6432	0.0246
('m', 'b', 's', 'is', 'rt', 'lr', 'wpa', 'pnr5')	0.6432	0.0231
('b', 's', 'is', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6432	0.027
('b', 's', 'is', 'y', 'lr', 'wpa', 'pnr5')	0.6432	0.0254
('m', 'b', 'is', 'rt', 'y', 'lr', 'wpa', 'pv')	0.6431	0.023
('b', 's', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.643	0.0241
('b', 's', 'is', 'rt', 'y', 'lr', 'pnr5')	0.6429	0.0259

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'is', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.6426	0.0248
('b', 'is', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6426	0.0264
('b', 's', 'is', 'rt', 'y', 'lr', 'wpa')	0.6426	0.0219
('m', 's', 'rt', 'y', 'lr', 'vc', 'pnr5')	0.6424	0.0282
('m', 'b', 'is', 'rt', 'y', 'wpa', 'pv')	0.6424	0.0267
('m', 'b', 's', 'is', 'lr', 'wpa', 'pnr5')	0.6424	0.0272
('is', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6424	0.0262
('s', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6422	0.0266
('b', 's', 'is', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6421	0.0299
('s', 'is', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6416	0.0279
('m', 'b', 's', 'is', 'rt', 'wpa', 'pnr5')	0.6415	0.0274
('s', 'is', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.6415	0.0249
('m', 's', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6413	0.0288
('s', 'is', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.6409	0.0274
('b', 's', 'is', 'rt', 'lr', 'pv', 'pnr5')	0.6406	0.0225
('m', 'b', 's', 'is', 'y', 'wpa', 'pv')	0.6406	0.0261
('m', 'b', 'is', 'rt', 'wpa', 'pv', 'pnr5')	0.6405	0.028
('m', 's', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6403	0.0286
('m', 's', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6403	0.0287
('b', 'is', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6403	0.0299
('m', 'b', 'is', 'lr', 'wpa', 'pv', 'pnr5')	0.6402	0.0259
('s', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6401	0.0251
('m', 's', 'rt', 'y', 'vc', 'pv', 'pnr5')	0.6401	0.026
('b', 's', 'is', 'rt', 'y', 'lr', 'pv')	0.6399	0.0239
('m', 's', 'rt', 'y', 'lr', 'vc', 'pv')	0.6399	0.0259
('m', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6398	0.0279
('b', 's', 'is', 'rt', 'lr', 'wpa', 'pnr5')	0.6398	0.0212
('m', 'b', 's', 'is', 'rt', 'y', 'wpa')	0.6398	0.0251
('b', 's', 'is', 'y', 'wpa', 'pv', 'pnr5')	0.6397	0.0295
('s', 'rt', 'y', 'lr', 'vc', 'pv', 'pnr5')	0.6396	0.0286

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('b', 's', 'is', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.6396	0.0221
('m', 'b', 's', 'is', 'rt', 'lr', 'wpa')	0.6395	0.0256
('b', 's', 'is', 'y', 'lr', 'wpa', 'pv')	0.6393	0.0232
('m', 's', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6391	0.0267
('s', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6391	0.0254
('m', 's', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6389	0.0299
('rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6388	0.0255
('m', 'b', 's', 'is', 'rt', 'lr', 'pv')	0.6388	0.0236
('m', 's', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6385	0.0254
('s', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6385	0.0296
('m', 'b', 's', 'is', 'rt', 'wpa', 'pv', 'pnr5')	0.6384	0.0292
('m', 's', 'rt', 'y', 'vc', 'wpa', 'pnr5')	0.6383	0.0285
('m', 'b', 's', 'is', 'lr', 'wpa', 'pv')	0.6381	0.0233
('m', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6377	0.0268
('b', 's', 'is', 'rt', 'y', 'lr', 'wpa', 'pv')	0.6377	0.0224
('m', 's', 'rt', 'y', 'lr', 'vc', 'wpa')	0.6371	0.0247
('b', 'is', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.6368	0.0266
('m', 'b', 's', 'is', 'wpa', 'pv', 'pnr5')	0.6368	0.025
('b', 'is', 'rt', 'y', 'lr', 'wpa', 'pv')	0.6367	0.0241
('m', 'rt', 'y', 'lr', 'vc', 'wpa', 'pnr5')	0.6365	0.026
('m', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6363	0.0287
('m', 's', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6362	0.0281
('m', 's', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6362	0.0275
('m', 's', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.636	0.0278
('m', 's', 'rt', 'y', 'vc', 'wpa', 'pv')	0.636	0.0283
('m', 's', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.636	0.0271
('s', 'y', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6355	0.0277
('m', 'b', 's', 'is', 'rt', 'lr', 'wpa', 'pv')	0.6354	0.0233
('b', 's', 'is', 'rt', 'y', 'wpa', 'pv')	0.6353	0.0243
('m', 'rt', 'y', 'lr', 'vc', 'wpa', 'pv')	0.6351	0.0258

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'is', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6346	0.0244
('m', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6345	0.0257
('b', 's', 'is', 'lr', 'wpa', 'pv', 'pnr5')	0.6344	0.0219
('s', 'rt', 'y', 'vc', 'wpa', 'pv', 'pnr5')	0.6342	0.0293
('b', 's', 'is', 'rt', 'wpa', 'pv', 'pnr5')	0.6342	0.0273
('m', 'is', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6337	0.0245
('m', 's', 'is', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6334	0.0264
('m', 's', 'is', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.6327	0.027
('m', 's', 'is', 'rt', 'y', 'lr', 'pnr5')	0.6326	0.0284
('m', 'b', 'is', 'rt', 'lr', 'wpa', 'pv')	0.6321	0.0249
('m', 's', 'is', 'rt', 'y', 'pv', 'pnr5')	0.6319	0.0258
('m', 's', 'rt', 'lr', 'vc', 'pv', 'pnr5')	0.6319	0.0237
('m', 'is', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.6316	0.0263
('b', 's', 'is', 'rt', 'lr', 'wpa', 'pv')	0.6314	0.0226
('m', 's', 'is', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6312	0.0266
('m', 's', 'is', 'y', 'lr', 'pv', 'pnr5')	0.6309	0.0278
('m', 's', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6309	0.0274
('s', 'is', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.6307	0.0264
('m', 's', 'is', 'rt', 'y', 'lr', 'pv')	0.6306	0.0263
('m', 's', 'is', 'rt', 'y', 'lr', 'wpa')	0.6303	0.0263
('m', 's', 'rt', 'lr', 'vc', 'wpa', 'pv')	0.63	0.0237
('m', 's', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6296	0.0232
('m', 's', 'is', 'rt', 'y', 'wpa', 'pnr5')	0.6295	0.0293
('is', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6294	0.0274
('m', 's', 'rt', 'lr', 'vc', 'wpa', 'pnr5')	0.6293	0.024
('m', 's', 'is', 'rt', 'y', 'lr', 'wpa', 'pv')	0.629	0.0266
('m', 'is', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6289	0.0288
('m', 'b', 's', 'is', 'rt', 'wpa', 'pv')	0.6276	0.027
('s', 'is', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6276	0.0267
('m', 's', 'is', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6268	0.0281

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 's', 'is', 'y', 'lr', 'wpa', 'pnr5')	0.6267	0.0251
('m', 's', 'is', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6263	0.026
('m', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6263	0.0229
('m', 's', 'is', 'rt', 'lr', 'wpa', 'pnr5')	0.626	0.0281
('m', 'is', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6251	0.0278
('m', 'is', 'rt', 'y', 'lr', 'wpa', 'pv')	0.625	0.0274
('m', 's', 'rt', 'vc', 'wpa', 'pv', 'pnr5')	0.625	0.0233
('m', 's', 'is', 'rt', 'lr', 'pv', 'pnr5')	0.625	0.0258
('m', 'is', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.6249	0.0257
('s', 'is', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6245	0.0273
('s', 'is', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6242	0.0268
('m', 's', 'is', 'y', 'lr', 'wpa', 'pv')	0.6233	0.0283
('m', 's', 'is', 'lr', 'wpa', 'pv', 'pnr5')	0.6233	0.0263
('s', 'is', 'rt', 'y', 'lr', 'wpa', 'pv')	0.6229	0.0252
('m', 's', 'is', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.6229	0.026
('m', 's', 'is', 'y', 'wpa', 'pv', 'pnr5')	0.6227	0.0275
('s', 'rt', 'lr', 'vc', 'wpa', 'pv', 'pnr5')	0.6219	0.0246
('m', 'b', 's', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6201	0.0256
('s', 'is', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.62	0.0306
('m', 'b', 's', 'y', 'lr', 'pv', 'pnr5')	0.6197	0.0273
('m', 's', 'is', 'rt', 'y', 'wpa', 'pv')	0.6193	0.0257
('m', 's', 'is', 'rt', 'lr', 'wpa', 'pv')	0.6183	0.0246
('m', 's', 'is', 'rt', 'wpa', 'pv', 'pnr5')	0.618	0.0268
('m', 'b', 's', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6175	0.029
('m', 'b', 's', 'rt', 'y', 'lr', 'pnr5')	0.6175	0.0288
('s', 'is', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.6175	0.0242
('m', 'b', 's', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.6174	0.0274
('m', 'b', 's', 'rt', 'lr', 'pv', 'pnr5')	0.6172	0.027
('m', 'b', 's', 'rt', 'y', 'lr', 'pv')	0.6167	0.0287
('m', 'b', 's', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6165	0.0276

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 'b', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6157	0.0284
('m', 'b', 's', 'y', 'lr', 'wpa', 'pnr5')	0.6155	0.0283
('m', 'b', 's', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.6153	0.0304
('m', 'b', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.6152	0.03
('b', 's', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6144	0.0245
('m', 'b', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6142	0.0288
('m', 'b', 's', 'rt', 'y', 'lr', 'wpa', 'pv')	0.6139	0.0284
('m', 'b', 's', 'y', 'lr', 'wpa', 'pv')	0.6135	0.026
('m', 'b', 's', 'rt', 'lr', 'wpa', 'pnr5')	0.6133	0.0287
('m', 'b', 's', 'lr', 'wpa', 'pv', 'pnr5')	0.6129	0.0261
('m', 'b', 's', 'rt', 'y', 'lr', 'wpa')	0.6121	0.03
('m', 'b', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.6118	0.0269
('m', 'b', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.6111	0.0268
('b', 's', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6101	0.025
('b', 's', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6099	0.0231
('b', 's', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.6094	0.0242
('m', 'b', 's', 'rt', 'y', 'wpa', 'pnr5')	0.6093	0.0231
('b', 's', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.608	0.0241
('m', 'b', 's', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6073	0.0296
('m', 'b', 's', 'rt', 'wpa', 'pv', 'pnr5')	0.6063	0.0276
('m', 'b', 'rt', 'y', 'lr', 'wpa', 'pv')	0.6062	0.027
('m', 'b', 's', 'rt', 'lr', 'wpa', 'pv')	0.6058	0.0266
('m', 'b', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6052	0.0284
('b', 's', 'rt', 'y', 'lr', 'wpa', 'pv')	0.6052	0.026
('m', 'b', 's', 'rt', 'y', 'pv', 'pnr5')	0.6051	0.0287
('m', 'b', 's', 'y', 'wpa', 'pv', 'pnr5')	0.6039	0.0268
('b', 's', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.6021	0.0255
('b', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6016	0.0235
('m', 's', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.6011	0.0255
('m', 's', 'rt', 'y', 'lr', 'pv', 'pnr5')	0.5996	0.0247

Table 7.1 Results of Each Feature Set in Exhaustive Feature Selection (cont.)

feature_names	**avg_score**	**std_dev**
('m', 's', 'rt', 'y', 'lr', 'wpa', 'pnr5')	0.5991	0.0231
('m', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.5988	0.0271
('m', 'b', 's', 'rt', 'y', 'wpa', 'pv')	0.5986	0.0269
('m', 's', 'rt', 'y', 'wpa', 'pv', 'pnr5')	0.5975	0.0304
('m', 's', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.5965	0.0263
('m', 's', 'rt', 'lr', 'wpa', 'pv', 'pnr5')	0.5947	0.0287
('m', 's', 'rt', 'y', 'lr', 'wpa', 'pv')	0.5935	0.0295
('s', 'rt', 'y', 'lr', 'wpa', 'pv', 'pnr5')	0.5923	0.0246