

AN ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM
FOR A LOCATION-BASED SOCIAL NETWORK

AYCAN AYYILMAZ

BOĞAZIÇI UNIVERSITY
2019

AN ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM
FOR A LOCATION-BASED SOCIAL NETWORK

Thesis submitted to the
Institute for Graduate Studies in Social Sciences
in partial fulfillment of the requirements for the degree of

Master of Arts
in
Management Information System

by
Aycan Ayyılmaz

Boğaziçi University

2019

DECLARATION OF ORIGINALITY

I, Aycan Ayyılmaz, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature.....

Date20.06.2019.....

ABSTRACT

An Adaptive Neuro-Fuzzy Inference System for a Location Based Social Network

Location Based Social Networks (LSBNs) has started with broadcasting the location via text messages long before the mobile applications and mobile data. Due to the technological developments, a notable number of technologies we use today are location-aware, and they are available to more people. When one of the most popular LSBN, Foursquare, has brought together both motivation of gameplay and social networks, and it introduced a new business model to the market. They have used points and badges to motivate users for “checking in” to locations. After that point, LSBNs started to be a useful tool that effected purchase decision and became vital for information research, evaluation of alternatives and post-purchase evaluation. Typically, the score of the venues is a significant decision-making parameter for most of the users for a purchase decision. Due to the complex and undisclosed score calculation method of Foursquare, it has been a wonder to users and venue owners. Purpose of the research is to build a model that can predict venue scores based on variables such as check-in counts, review, tip and photo counts of venues.

ÖZET

Konum Bazlı Sosyal Ağ için

Adaptif Sinirsel Tabanlı Bulanık Mantık Çıkarım Sistemi

Günümüzde kullanılan mobil data ve uygulamalardan çok daha önce Konum Bazlı Sosyal Ağlar (KBSA), kısa mesaj servisleri ve toplu yayın özelliklerinin de yardımıyla kullanılmaya başlanmıştır. Teknolojinin ilerlemesi nedeniyle, bugün kullandığımız araçlar konum bilgisine sahip ve bu araçlar birçok insanın erişimine açık. KBSA'ların en çok kullanılanlarından biri olan Foursquare, oyun oynama motivasyonu ile sosyal ağ kavramlarını bir araya getirdi ve yeni bir iş modeli oluşturdu. Bu noktadan sonra KBSA'lar, satın alma kararını etkileyen, özellikle araştırma, alternatiflerin karşılaştırılması ve satın alma sonrası değerlendirme başlıklarında önemli araçlar haline geldiler. Özellikle, mekanların müşterileri tarafından yapılan değerlendirme sonuçları satın alma kararı için önemli bir karar verme parametresi olmaya devam ediyor. Ancak Foursquare'in karmaşık ve gizli değerlendirme sayısal değer hesaplama algoritması, kullanıcılar ve özellikle de mekan sahipleri için gizemini koruyor. Bu çalışmanın amacı, “check in”, yorum, puan, fotoğraf, fiyatlama gibi parametreleri kullanarak, mekan skorunu tahminleyen bir model önermektir.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	1
1.1 Why we use LBSN	2
1.2 Why marketers/businesses use LBSNs.....	4
1.3 Purpose of the research.....	5
CHAPTER 2: LITERATURE REVIEW	6
2.1 Fuzzy logic.....	7
2.2 Artificial and adaptive neural networks.....	15
2.3 Neuro-fuzzy systems	17
2.4 Fuzzy versus statistical approaches	20
2.5 Other predictive analysis models	23
CHAPTER 3: DATA AND METHODOLOGY	25
3.1 Workflow of model.....	25
3.2 Data gathering.....	26
3.3 Insights of data set	28
3.4 Data cleaning	31
3.5 Data transformation	32
3.6 Feature selection	35
CHAPTER 4: BUILDING A MODEL.....	38
4.1 Model details of the anfisHybrid model	40
4.2 Model details of the anfisBckProp model	45
4.3 Other ANFIS variations	47
4.4 Other regression models	48
CHAPTER 5: RESULTS AND DISCUSSION.....	49
REFERENCES	52
APPENDIX A: DATA GATHERING CODES	58

APPENDIX B: VENUE DATA RETRIEVING CODES	60
APPENDIX C: VENUE DETAIL DATA RETRIEVING CODES.....	61

LIST OF TABLES

Table 1. Summary of Variables in Data Set.....	28
Table 2. Number of Venues in Cities.....	29
Table 3. Distribution of Venues in Categories.....	29
Table 4. Calculated Feature Weights for All Variables	36
Table 5. Final Results of anfisHybrid Model with 100 Epochs.....	42
Table 6. Variations of ANFIS Model.....	47
Table 7. Comparison of ANFIS Models	51
Table 8. Comparison of All Models.....	51

LIST OF FIGURES

Figure 1. Fuzzy inference system (FIS) (Jang, 1993)	10
Figure 2. First-order Sugeno fuzzy model (Sugeno & Takagi, 1985)	12
Figure 3. (a) The Sugeno fuzzy model with two-input and rules (b) Equivalent ANFIS architecture (Jang, 1997)	19
Figure 4. (a) ANFIS architecture for a two-input Sugeno Fuzzy Model with nine rules (b) The input space that is partitioned into nine fuzzy regions (Jang, 1997) .	19
Figure 5. The workflow of the model followed in this study	26
Figure 6. Histogram and q-q plot of the variable vrating.....	30
Figure 7. Histogram of original vcheckinscount variable	34
Figure 8. Histogram of vcheckinscount variable after log transformation	35
Figure 9. Feature index of feature weights of original data compared to transformed data	36
Figure 10. Sample of the dataset after pre-processing	37
Figure 11. A high-level diagram of the Fuzzy Inference System (FIS) for anfisHybrid model.....	40
Figure 12. Examples to membership functions considered in this model (gbellmf is used)	41
Figure 13. Train error for hybrid optimization method (anfisHybrid).....	41
Figure 14. ANFIS model structure for anfisHybrid	42
Figure 15. Fuzzy rules and membership functions on Rule Viewer with input for anfisHybrid model.....	43
Figure 16. Training data results compared to estimated results for anfisHybrid	44

Figure 17. Test data set results compared to estimated results for anfisHybrid	44
Figure 18. Surface graphic for vrating (output) compared to vcheckinscount (input1) and vuserscount (input2)	45
Figure 19. Back-Propagation Gradient Descent Optimization method train error ...	45
Figure 20. Training data results compared to estimated results for anfisBckProp ...	46
Figure 21. Test data set results compared to estimated results for anfisBckProp.....	46

CHAPTER 1

INTRODUCTION

Nowadays a notable number of technologies we use today are location aware. The data has been produced by users of social media platforms have geographical information such as latitude/longitude of the user in addition to other valuable information. Also, it is known that we have an increasing amount of data every day since the location-aware devices are cheaper and available to more people.

Mobile devices are collecting location information, and that information has been used in different areas of web, starting from Google searches, traffic alerts, advertising. As the user of location-based technologies increasing, it also brings important questions to the table related to personal privacy, freedoms and collateral surveillance.

According to Leighton and Saker (2017), increasing connectivity of localities strengthened by location-aware mobile technologies in public spaces has enabled a new kind of social connection.

It all began with Dodgeball, the predecessor to Foursquare. It was similar to today's social media. Users were "checking in" to a location by sending text messages to Dodgeball. Then Dodgeball was broadcasting their location to people in their network via text messages (Humphreys, 2007). In the same study, Humphreys (2007) highlighted that the purpose of the check-ins is not only to find friends nearby but also showing-off. According to the research, people have different reasons to check-in such as finding nearby friends, meeting up with available friends, showing off and social cataloging of people's lives. The author concluded the study by

suggesting that combining social networks with physical spaces via mobile technology can change their urban public space experience.

1.1 Why we use LBSN

When it was first launched, Foursquare has brought together both motivation of gameplay and social networks. They have used points and badges to motivate users for “checking in” to locations. In addition to badges, points and competition between friends, a chance of becoming mayor hold a significant role for the user. Users are competing to hold this desired virtual position. All the location-based services such as Yelp, Foursquare, and Google Latitude are free for users. So, they are monetizing their services through advertising. For instance, Foursquare users are offered coupons and deals when they check in certain locations and benefits can vary according to the user’s status. Motivation to use this example of Location-based Social Network (LBSN) is coming from not only internal rewards and competition but also external rewards such as coupons and free stuff. These features make Foursquare a great example of gamification (McKenzie, 2011), which has been defined as “use of game design elements in non-game contexts” by Deterding, Dixon, Khaled, and Nacke (2011, p. 9).

According to the work of Frith (2013), participants of the study keen to go to places where they usually would not because of the satisfaction comes from collecting badges and performing tasks. Some of the participants are considering branded badge campaigns for discovering cities to collect badges. Branded badges are badges sponsored by companies and institutions. It shows that the application created a game people see the city as a game board and move according to the necessity of winning the game and earn the badge.

Saker and Leighton (2016) presented a new concept named “playeur” in their study. They have built their concept over the study of Luke’s (2006) “phoneur” notion. The definition of ‘playeur’ is “an engaged actor who develops relationships with space and place through intentional playful activities“ (Saker & Leighton, 2016, p. 1181).

This type of business model brings inevitable privacy concerns to one’s mind. Even though LBSN providers are obliged to obtain the consent of the users, it does not mean that users understand what they are signing in the fine print of user agreements (Gordon & Silva, 2011).

Sutko and Silva (2011) suggested that the assumption has been made about LBSN that its potential to increase spatial awareness and meet new people in public places might be flawed. They added the following:

They [LBSNs] do challenge traditional sociability, particularly in how we coordinate and communicate in public spaces. There may thus be a tradeoff in which LBSNs’ users engage in more direct coordination but less direct communication. However, this tradeoff can also lead to unexpected and perhaps pleasant social and spatial experiences (p. 819).

Schwartz and Halegoua (2014) stated in their study that Foursquare check-ins do not necessarily mean that those places matter to users. They highlighted that not only the motivation for producing and sharing such data but also the meaning of those to users and their audiences should be considered since they have been used as “a form of self-presentation as well as (re)productive practices of experience and reception of urban space” (p. 1657).

Lindqvist, Cranshaw, Wiese, Hong, and Zimmerman (2011) concluded in their studies that there are reasons for Foursquare users not checking in certain places (e.g., fast food restaurants) because they would be embarrassed to be seen or they

think it may threaten their privacy. This study also supports the idea of checking in as a way of self-presentation.

1.2 Why marketers/businesses use LBSNs

According to Salt (2011), LBSN users check in at locations that they wanted to be associated with, and they may also intend to show their network that they can afford to be in certain places. After this point, it is not only about the location or venue but also what that location or venue represents. At other times users want to promote local businesses that they have received excellent service. By doing this, they believe they are showing their gratitude and help them to reach out to more people and stay in business. All of these motivators can be leveraged by marketers/businesses and can be used to change the purchase decision of customers.

When it comes to purchasing decision there are different steps to market businesses, one can summarize it such as “Problem Definition”, “Information Research”, “Evaluation of Alternatives”, “Purchase Decision”, “Purchase” and “Post-Purchase Evaluation”. LBSNs can be useful tools for information research, evaluation or comparison of alternatives and post-purchase evaluation (Salt, 2011).

Especially for small businesses having no or little advertising budget can make it harder to compete. When they have the budget, they tend to spend their budget on mass advertising such as local newspapers, radio or TV. However, conventionally word of mouth much more powerful for small business to help them grow. It is also important for businesses and marketers to understand so that they can ensure they include the right type of offers, targeted at the right type of user, on the right platform (Salt, 2011).

Salt (2011) also highlighted the difference between social media and LBSN with these words:

Although having lots of Twitter followers is nice, it is ultimately just that, a number. The number of check-ins at a venue has a direct effect on the money at the checkout. These are more than just followers or fans—they are real customers who are at the location. They are spending money with the business, and they are telling others about their activity. It is a huge opportunity to reward and take advantage of that behavior. Reinforcing this behavior with positive rewards for customers will encourage them to do it more often (p. 17).

One of the most prominent LBSN is Foursquare, and it promotes itself to businesses with qualities such as user reward feature, attracting local customers and analytics to learn more about the customers (Foursquare, 2017).

1.3 Purpose of the research

Purpose of the research is to build a model that can predict venue scores based on variables such as check-in counts, review, tip and photo counts of venues. Due to the complex and undisclosed score calculation method of Foursquare, it has been a wonder to users and venue owners. This work aims to find a model that predicts venue scores with parameters in hand.

CHAPTER 2

LITERATURE REVIEW

Throughout the history of industrial and technological advancements, the human has sought a chance to automate various tasks. However, it has always been a challenge due to the difference in the way humans' reason and the way computers have been programmed. Until Zadeh's (1984) work on this very subject of Fuzzy Logic, only 1/0 or yes/no questions were answerable by computers. Zadeh (1984) stated in his work that human reasoning is not as precise as computers and it is much more complicated than them. He proposed Fuzzy Sets to bridge the gap between the way of reasoning of the machines and humans by differentiating Fuzzy sets from conventional "Crisp" sets in terms of boundaries.

Hard computing underlines exactness, certainty, and rigor. However, soft computing requires computation, reasoning, and decision making to benefit from the tolerance for imprecision and uncertainty wherever possible. Fuzzy logic is one of the primary components of soft computing accompanied by neural network theory that learning is the core of it; and probabilistic theory, which deals with uncertainty. Zadeh (1994) also noted that even though there are significant overlaps between Fuzzy Logic, Neural Network, and Probabilistic Theory, in general, they are complementary rather than competitive. For this reason, using Fuzzy Logic, Neural Network, and Probabilistic Theory in combination rather than exclusively can be an advantage depending on the case.

First industrial application of fuzzy logic was the F.L. Smidth Cement kiln, and it laid the groundwork for the use of fuzzy logic. Authors concluded the study by

stating that the fuzzy control system in their research has run properly, steady, effectively and better than human operators (Sheridan & Skjoth, 1984).

After his first publication on Fuzzy Sets in 1965, Fuzzy logic has gained ground by time and its applications have grown in terms of number and variety. Zadeh (1996) proposed that Fuzzy logic serves many purposes including computing with words.

Laviolette, Seaman, Barrett, and Woodall (1995) highlighted the fact that Fuzzy Set Theory publications have been increased dramatically in less than 30 years by 15.000 publication since Zadeh's first paper on this subject.

A fuzzy set is an assemblage of objects with graded membership. Zadeh (1978) suggested that, if the membership function indicated the possible occurrence of an event or outcome, the membership function could be considered as a generalization of classical probability. This generalization was called "possibility" by Zadeh.

2.1 Fuzzy logic

Fuzzy logic is the theory of fuzzy sets, and fuzzy sets were introduced by Lotfi Zadeh (1965). The aim of Zadeh (1965) was bringing human behavior in engineering and mathematical modeling together. Since then researchers have worked on this area to support and enlarge its area of use or to prove he was wrong with his theory. Applications of this concept can be found in many different contexts from device control to linguistics, from consumer products to finance, from medicine to image processing and so forth.

The need for fuzzy sets raised from the complexity of reproducing biological or humanistic systems. Zadeh (1965) stated in his famous article that imprecisely

defined classes play an important role in pattern recognition, communication information, and abstraction.

Fuzzy approach shares with Artificial Intelligence the idea of the information provided by humans is useful, and it should be used as an algorithm to enhance our capability of dealing with problems. Zadeh (1968) used a fuzzy algorithm for parking a car in his paper. He highlighted the fact that the question is very complicated even with large scale computers if you evaluate it as a precise problem rather than fuzzy. However, it is solvable if one can formulate a fuzzy algorithm for solving it.

Fuzzy logic is about the relative importance of precision. Is it matters to be precisely right when imprecise result works?

Zadeh (1973) worked on these two concepts such as precision and significance. He published his idea as the now-famous Principle of Incompatibility in his paper with these words; “As the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics.” (p. 28)

The use of the algorithm took the attention of the engineering world the early 80s, some of the prime examples were F.L. Smidth cement kiln and the Sendai subway system designed by Hitachi. These applications of fuzzy logic laid the preliminary work for the use of it in a broad sense such as consumer products. The first product was a fuzzy logic-controlled shower head produced in 1987, and the first fuzzy logic-based washing machine followed this in 1989, Matsushita designed both. In 90s product range employing fuzzy logic had increased. In addition to that, neural network techniques have been combined with fuzzy logic in these products, so products have gained the ability to adapt and learn from experience. The author

foresaw the ubiquitousness of neuro-fuzzy products from the 90s in consumer products, robotics, industrial systems, and process control (Zadeh, 1994).

The reasons for using fuzzy sets varies since it is flexible, easy to understand, and able to model non-linear functions. Dubois and Prade (2000) listed contributions of fuzzy sets to information modeling and processing.

- Connecting numerical data with linguistic labels, by representing gradual attributes or classes with imprecise boundaries;
- Summing-up numerical data in a soft, semantic and comprehensive way;
- Enabling modeling of complex subjective evaluations;
- Representing imprecise information with qualitative scales;
- Building the base for artificial systems; • Approximating imprecise statistical information;
- Being a framework for commonsense reasoning.

Now, basic definitions of fuzzy set theory will be provided. Here you can find how Zadeh (1965) defined fuzzy sets in his first paper on this.

Let X be a space of points noted by x .

$$X = \{x\}$$

A fuzzy set is a class A in X characterized by a membership function with the value of $f_A(x)$ at x representing the grade of membership of x in A .

$$f_A(x) \in [0,1]$$

2.1.1 Fuzzy Inference Systems

Jang (1993) defined the Fuzzy inference system includes five main blocks which can be seen in Figure 1.

1. Rule Block; contains fuzzy if-then rules,
2. Database Block; defines the membership functions of the fuzzy sets used in the fuzzy rules,
3. Decision-making unit; performs the inference operations on fuzzy rules,
4. Fuzzification unit; maps crisp inputs to fuzzy inputs,
5. Defuzzification unit; transforms the fuzzy results into crisp results.

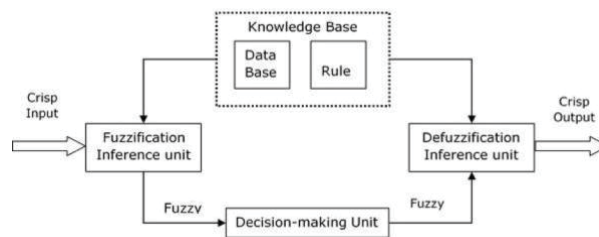


Figure 1. Fuzzy inference system (FIS) (Jang, 1993)

As Jang (1993) highlighted in his paper, fuzzy if-then rules regardless of its type can mimic the use of linguistic labels and “rule of thumb” used by humans. The author also stated that from certain angle fuzzy if-then rules, which are the description of the system modeled, is the backbone of the fuzzy inference system.

Method for constructing a fuzzy inference system is called fuzzy modeling which has features such as;

1. Rule structure based on domain knowledge. This feature is where fuzzy modeling is robust compared to other modeling approaches.
2. Use of numerical data also an essential element of fuzzy modeling which is similar to alternative mathematical approaches.

In some cases, there may be a collection of input/output data that one would like to use for modeling and apply a fuzzy inference system. Also, it may not be possible to figure out what membership functions look like by looking at the data. Instead of choosing the parameters randomly, choosing these parameters that are suitable for membership functions that allow the fuzzy inference system to track given data (Kulla, Elmazi, & Barolli, 2016).

Sugeno and Mamdani methods are popular methods and fuzzy inference process, fuzzifying the inputs and applying fuzzy operators are the same in both. However, some differences make that make the Sugeno method much more convenient and computationally proficient. Singla (2015) stated that Sugeno performs better with optimization techniques, especially with non-linear systems. The author highlighted another difference between two methods as follows; Mamdani method employs defuzzification process of the fuzzy outcome, while Sugeno calculates crisp result using a weighted average, and this improves process time and interpretability in favor of Sugeno.

Sugeno fuzzy model is also known as TSK (Takagi-Sugeno-Kang) model is proposed in the 1980s to build a structured approach to fuzzy modeling (Kang & Sugeno, 1988; Sugeno & Takagi, 1983, 1985; Sugeno, 1985).

Sugeno fuzzy model can be generalized as follows;

If x is A , y is B then,

$$z = f(x, y)$$

where A and B are fuzzy sets in the antecedent, $z = f(x, y)$ is a crisp function in the consequent. In general, this function is polynomial, when it is the first-order polynomial, the resulting FIS system is called first-order Sugeno fuzzy model. When

f is constant, then it is called zero-order Sugeno fuzzy model. In Figure 2, a fuzzy reasoning procedure is shown for first-order Sugeno fuzzy model. In this model each rule has a crisp output; therefore the overall output is calculated by weighted average or weighted sum for some cases to avoid the computational cost. The reason behind the computational advantage of Sugeno over Mamdani lays on avoiding the defuzzification process that is required for Mamdani Method (Jang, 1997).

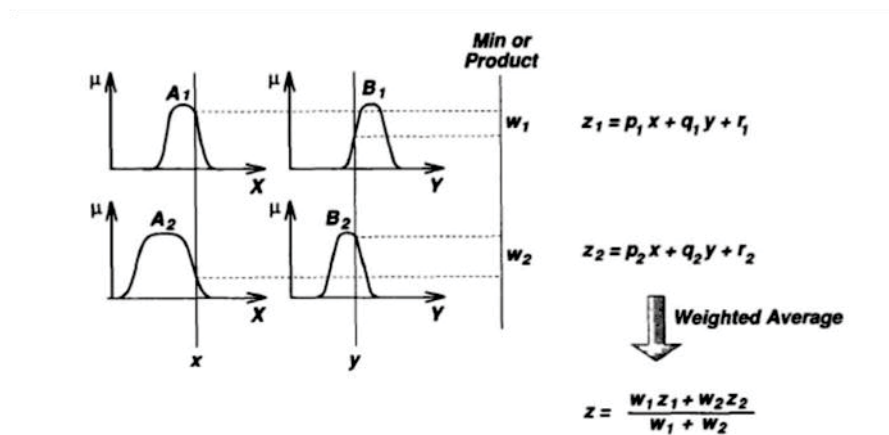


Figure 2. First-order Sugeno fuzzy model (Sugeno & Takagi, 1985)

2.1.2 Membership Functions

Since Zadeh (1965) introduced the fuzzy sets, the meaning and measurement of membership functions have been the most debated part of the theory. Fuzzy sets are characterized by their membership functions, and there are various interpretations and practical methods for determining membership functions.

A basic definition of membership function (MF) is a curve that maps each point in input space to a membership value. Membership value can also be named as a degree of membership between 0 and 1. The only rule a membership function must meet is that it must get values between 0 and 1. The function can be a random curve

whose shape we can define as a function that is appropriate for us in terms of simplicity, convenience, speed, and efficiency.

As it is mentioned before, there are various approaches and point of views on membership functions. Hisdal (1988) advocated a likelihood view of membership grades. Mabuchi (1992) also shared the likelihood view; however, in the paper, the degree of membership is interpreted as conditional probability conditioned by the given element. On the other hand, Zysno (1981) used the similarity view of the membership function. The author took a human being as the measurement device and defined membership function as a distance of the object from the ideal. Giles (1988) offered a pragmatic, semantic interpretation for membership function and came up with the concept of a graded truth by describing a fuzzy sentence as a sentence to which we attribute a degree of belief. Then the author defined fuzzy sets and grades of membership in terms of assertions.

In the experimental research concerning membership functions, one needs ways of extracting membership functions. Below, methods to obtain membership functions with subjective interpretations are summarized; (Chameau & Santamarina, 1987; Norwich & Turksen, 1982; Turksen, 1991)

- Pairwise comparison: it consists of comparing the strength of the quality of two objects. “Which color, A or B, has the characteristic of darkness more strongly, and how much more?”
- Point estimation (direct rating): Assessors select one element from a list that best answers the question. “Classify color A relative to its degree of darkness.”
- Membership function exemplification: Assessors are required to provide the object membership to several discrete points on the reference axis. "Give the degree of belongingness of color A to the set of very dark colors."

- Interval estimation: Assessors select a suitable range of possible values that answers the question well. "Give an interval in which you estimate the color lies."
- Reverse rating: Assessor is given a membership degree and asked to identify the object for which degree corresponds to the fuzzy term in question. "Identify the person who is tall to the degree 0.6."
- Polling: Assessors have been asked a Yes/no question and the average of the answers is taken. "Do you agree that A is a dark color? (Yes or No)"

Moreover, there is a significant concern about membership functions when it comes to constructing them from a given set of data. When sufficient data available to use and the aim is to use model-based fuzzy sets to analyze the problem, the membership function should be constructed from the data in hand (Sugeno & Yasukawa, 1993).

- Fuzzy Clustering: Sugeno, Turksen, and Nakanishil (1993) analyzed the procedure as follows; Output data is subjected to clustering, and it is projected to the input data. The variables that are relevant from generated clusters are selected. Then, membership functions are structured for the variables chosen. After that the data with selected variables are divided into three groups; two groups are used in the model building and cross-validation while the third group is used as the test data.
- Neural Fuzzy Techniques: Artificial Neural Networks (ANN) and fuzzy systems are two separate but overlapping concepts to come near to human brain functionality in an imprecise situation. There has been an important interest in combining these two (Kosko, 1992). Takagi & Hayashi (1991) formulated the determination of fuzzy inference rules using Neural Network (NN) models

which provided an automatic partition of a fuzzy rule and adjustment of membership function in case of a change in environment by using non-linearity of NN learning function.

2.2 Artificial and adaptive neural networks

Artificial Neural Networks inspired by biological neural networks. The simplest definition to understand Artificial Neural Networks (ANN) is from Dr. Robert Hecht Nielsen. Nielsen defined a neural network as: "...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs." (As cited in Caudill, 1987, p. 91)

Caudill and Butler (1990) stated that Neural Networks or neuro-computers were designed to mimic the brain and biological neural architecture. Continues to present biological neural systems; neuron is a cell and can transmit and process chemical or electrical signals. The neuron is connected with other neurons to create a network. Every neuron has a "dendrite" which acts as an input source of the cell, a "cell body" which processes the input dendrite has sent to and an "Axon" which can be mapped as the output of the system. Outputs connect inputs of other neurons with a special connection called "synapse" and that creates the network. Moreover, synapses can have positive or negative effects on activities of their neurons, and they pass signals in only one direction.

Bell (2014) highlighted the fact that understanding the artificial neural network starts with recognizing that if one is applying this model this means that one is not exactly sure the connection of the input and output nodes.

Perceptron is base of a neural network, and it is responsible for getting an input signal and passing the value through the “activation function” and outputting the result of this function. The node can process any number of inputs and takes a weighted sum of all inputs. The activation function is the processing part that starts after input is passed into the neuron and it decides on if the value will be passed to output axon, therefore the next neuron or not. Moreover, multilayer perceptrons defined as perceptrons which have one or more layers between the input node and the output nodes (Bell, 2014).

Adaptive Neural Network (ANN) concept comes from the idea of environmental changes that occur where learning and training take place. When a neural network trains in a still environment, the network can also learn the environment. The learning keeps continue until the synaptic weights remain unchanged and there is nothing to learn after a certain point. However, in real life, the environment is so dynamic and cannot be learned since it is not possible to foresee the future state. ANN can adapt the changes in the environment (Haykin, 2008). Palnitkar and Cannady (2004) defined ANNs as networks that can learn as they are performing in the real world such as online learning and they continue to learn.

Widrow and Winter (1988) gave successful examples of ANN to provide some insight;

1. System modeling: modeling systems that have known input and outputs but an unknown system.
2. Statistical Prediction: present and past samples can be used to estimate future values of digital signals.

3. Noise canceling: ANN can be used to solve a signal processing problem which is removing noise from the signal.
4. Echo canceling: Echo canceling is also similar to noise canceling; it is eliminating echo from telephone circuitry.

2.3 Neuro-fuzzy systems

Neural Networks has been seen as “Black Boxes” since it is hard to look into the network and comprehend what it has precisely learned and interpreted its internal parameters. Benitez, Castro, and Requena (1997) opposed this argument and provides an interpretation of neural networks built with fuzzy rules using a new fuzzy logic operator. Nonetheless, a neural network has learning and generalization abilities. On the other hand, Fuzzy Systems are highly interpretable and convenient tools for modeling complex systems (Mencar, 2013). Notwithstanding, they don't have learning or generalization abilities.

A substantial endowment of adding fuzziness to a neural network is that the nodes are open to linguistic interpretation after the training. This may allow gaining some insight into how features are related when it comes to making a class decision. The second contribution is that networks of fuzzy neurons train in fewer epochs (Bezdek, Mikhail, Keller, & Krisnapuram, 1999).

In his editorial piece, Bezdek (1993a) discussed “the marriage” of Fuzzy Logic with Computational Neural Networks (CNN). The author stated that CNNs are low-level compared to fuzzy methods and they complement each other. Both technologies can be used as a tool in the framework of the model. Bezdek reminded us of the ability to represent functions of CNN and how this is useful for a fuzzy model which builds its foundation on membership functions.

The author categorized a combination of these two technologies as follows;

1. Fuzzification of traditional CNN architectures and models,
2. The use of CNN's as an instrument in fuzzy models.

2.3.1 Adaptive Neuro-Fuzzy Inference Systems (ANFIS)

As mentioned before, conventional methods based on mathematical tools perform imperfectly for complex and uncertain systems. In contrast, a fuzzy inference system can bring the qualitative aspect of human reasoning to the table. Nevertheless, there are also problems with this method as well such as not having formal ways to transfer expert knowledge to a fuzzy inference system and lack of learning algorithms to optimize membership functions by minimizing error measure. At this point, the Adaptive Neuro-Fuzzy Inference System (ANFIS) is proposed by Jang (1991) using gradient descent algorithm and Kalman filter algorithm.

This proposed algorithm was known for being time consuming and proneness to be trapped in local minima. Therefore, Jang (1993) came up with a hybrid rule which can speed the whole process dramatically in his original work on ANFIS.

In general, the ANFIS model building starts by feeding with initial fuzzy model and rules are extracted from input and output data. After that neural network is used to tune the fuzzy rules to create final ANFIS model (Mandal, 2018). In Figure 3, the Sugeno Fuzzy Model and corresponding ANFIS architecture can be observed. Moreover, the relation between inputs of the ANFIS model and results are visualized for two-input Sugeno fuzzy model in Figure 4.

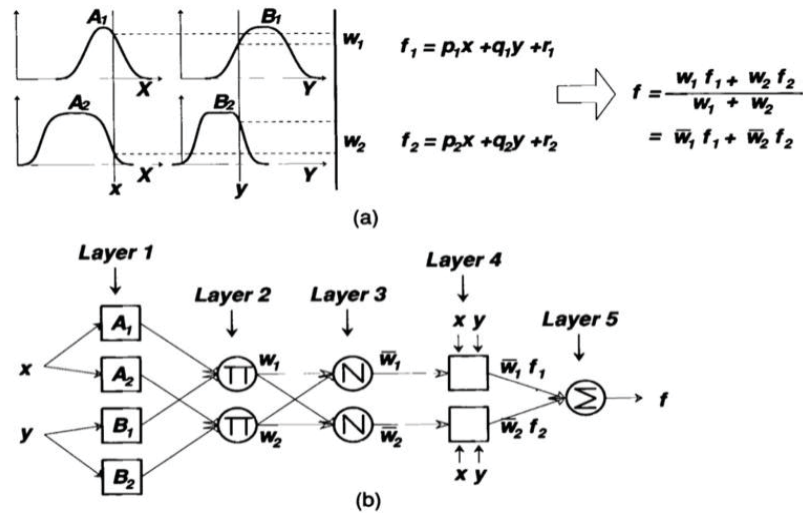


Figure 3. (a) The Sugeno fuzzy model with two-input and rules (b) Equivalent ANFIS architecture (Jang, 1997)

In a Sugeno model, every rule must lead to a crisp output; this output is calculated by function in figure 3(a). The overall output is calculated by weighted average; hence the process avoids defuzzification part of the Mamdani method.

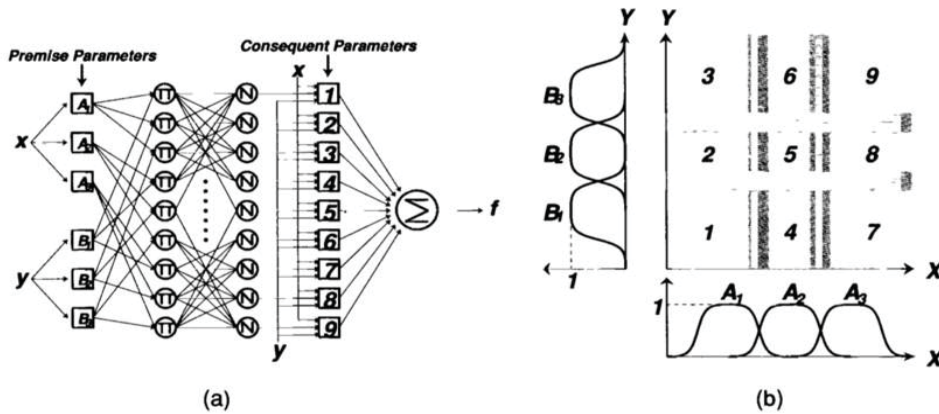


Figure 4. (a) ANFIS architecture for a two-input Sugeno Fuzzy Model with nine rules (b) The input space that is partitioned into nine fuzzy regions (Jang, 1997)

In Figure 4 (a), ANFIS architecture for two-input Fuzzy Sugeno Model with nine rules is modeled. Each input has three membership functions. Figure 4(b) is a

demonstration of nine regions managed by fuzzy rules applied on two-input dimensional space.

2.4 Fuzzy versus statistical approaches

Even though Fuzzy Set theory initially associated with control theory and uncertainty in artificial intelligence, its methods have been used as a substitute to traditional statistical methods since then (Laviolette, Seaman, Barrett, & Woodall, 1995).

Coppia, Gilb, and Kiersc (2006) highlighted the fundamental problem of the Information Society as dealing with cognitive science considering basic features of uncertainty, imprecision, and vagueness. The authors looked at this problem from both theoretical and practical aspects. Theoretical aspect says information and uncertainty are very relevant. In General Information Theory (GIT) uncertainty can be explained by a lack of information while information can be explained as a reduction of uncertainty. Therefore, GIT can be broadened by including fuzzy sets to classical set theory to include more expressive non-additive measures (Klir, 2005).

The fuzzy set theory provides valuable tools for overcoming uncertainties of traditional probability theory in its standard form (Coppia, Gilb, & Kiersc, 2006).

Controversially, some scholars such as Lindley (1982) declares that probability theory is the only adequate way to quantify uncertainty at least in the Bayesian version. This issue has been discussed in statistical literature occasionally.

Laviolette et al. (1995) compared Fuzzy Logic Controller and Probabilistic Controllers both theoretically and by applying both for the same data and model. The authors claimed that the conditional probabilities in the probabilistic controller are much easier to interpret than the membership functions of the Fuzzy Logic Controller. They concluded their research on this matter by voicing their skeptical

thoughts on Fuzzy Logic and stated that they had not found any instance in which Fuzzy Set Theory is uniquely useful and anything can be done with Fuzzy Set Theory can be done using probability and statistics.

Researchers such as Laviolette et al. (1995) and Coletti and Scozzafava (2006) argued or compared fuzzy set applications to probabilistic and statistical applications mostly by mapping conditional probability to membership function and applied their proposed method to a data set and draw their conclusion as detailed above.

In response to their article, Kandel, Martins, and Pacheco (1995) stated that approach of Laviolette et al. wasn't adequate to draw the conclusion they have due to lack of application of different membership functions and using the same data for different quality classifications.

However, there is a difference between those two concepts, and it has been demonstrated with an example in Bezdek's (1993b) paper. As a case in point, a person whose life on the edge due to thirst in a desert is given two bottles. First bottle (A) has a label on it and says that it has 0.9 membership in the class of fluids known as drinkable, non-poisonous water. Second bottle's label (B) states that it contains 90% probability of being pure drinking water and 10% probability of being a poisonous drink. Bezdek questions which bottle would one choose? Bottle B (probability bottle) involves crisp set theory, and it does not allow partial membership in a class. So, there is 1 in 10 chance of the bottle being poisonous and deadly. However, bottle A (fuzzy bottle) contains swamp water, "fairly similar" to perfectly potable liquids, it belongs to the class of non-poisonous fluids.

On the other hand, assume that both membership and probability values were 0.5, then one would choose bottle B since it offers 50% chance of being drinkable

while membership value shows that it is not suitable for drinking. To conclude, the fuzzy set theory deals with similarities of the elements while Probability is an indicator of the likelihood that element in the class. Both are valid approaches in different use cases, and both approaches portray different types of information philosophically.

In his editorial piece Bezdek (1993a) also highlighted a valued opinion by most of the researchers that fuzzy and statistical approaches don't need to compete, they can be complementary to each other and fuzzy approach can provide additional value to statistical methods and broaden their foundations and applications.

Final words on this subject will belong to Zadeh (1995), father of fuzzy set theory. In his discussion on this matter, he explained the connection between probability theory and fuzzy logic. He pointed out shortages of probability theory such as not supporting the concept of a fuzzy event. Firstly, the author highlighted that Bayesian theory does not answer the question of the independence of fuzzy events. Secondly, he claimed that probability theory does not have a method to deal with quantifiers such as many, most, few. Thirdly, he stated that probability theory falls short of computing fuzzy probabilities expressed as likely, unlikely, not likely, and so forth. The author continued to list the limitations of probability theory in his paper and concluded his argument with stating that probability theory is not much effective in areas where the dependencies between variables are not well defined, probabilities are imprecise or incomplete, systems that require human reasoning and emotions are vital. Expert systems, pattern recognition, group decision analysis, weather and earthquake forecasting, handwriting and speech recognition are some of the examples. Complementing probability theory with fuzzy logic can enhance the results when dealing with these systems according to the author.

2.5 Other predictive analysis models

The predictive analysis aims to make use of historical data to predict the best probable and optimal outcome by building a mathematical model. It follows, using an up to date or current data to predict the result. Predictive analytic covers both statistical and machine learning techniques to predict value or estimate the probability of an event to occur. There are two problem categories in supervised learning as regression and classification where the outputs of classification are discrete, and the outputs of regression are continuous.

Regression (Linear, Exponential, Logistic...etc.) is one of the most popular algorithms used to model the relationship between two or more variables. Also, decision tree algorithms are used for both classification and regression problems.

A regression tree ensemble algorithm is a combination of multiple decision tree models to produce better performance than a single tree model. There are two ensemble method called Bagging (Bootstrap Aggregating) and Boosting. Bagging aims to reduce the variance of the prediction by creating training data from the original data set using different combinations and repetitions. Building data sets do not provide improvement for the prediction always, but it can decrease the variance and tune the forecast to the desired outcome. Boosting is a two-step approach, starts with creating a subset of the original data to produce a bunch of models without any consideration on their performance. At the next step, these models are combined using cost functions to boost their performance.

Support vector machines are supervised learning models that analyze the data and finds patterns to be used in classification or regression.

The Gaussian process is a generalized version of the Gaussian probability distribution. This model aims to determine model parameters for which the probability of the data is maximized (Rasmussen & Williams, 2006).

CHAPTER 3

DATA AND METHODOLOGY

This chapter is dedicated to model building and data pre-processing. Data gathering and pre-processing methods used in this work have been summarized. Furthermore, the ANFIS Model is introduced, and its infrastructure is described. An adaptive neuro-fuzzy inference system or adaptive network-based fuzzy inference system (ANFIS) is an artificial neural network that is based on the Takagi–Sugeno fuzzy inference system. To do so, the following steps have been taken;

- Data Gathering
- Data Cleaning
- Data Standardization
- Feature Selection
- Building a Model

3.1 Workflow of model

The workflow of the model is visualized as in Figure 5 and detailed in further sections.

For data gathering process, Google API, Foursquare API, and Postal Services' web sites are used as data sources. Python programming language is used to make API calls and parsing JSON response of the requests.

Next step of the process was creating a database with the data gathered in the previous step. Postgre database is used to store, clean and transform data. DataGrip IDE is used for SQL.

Finally, for statistical and modeling purposes, MATLAB R2018a has been used intensively.

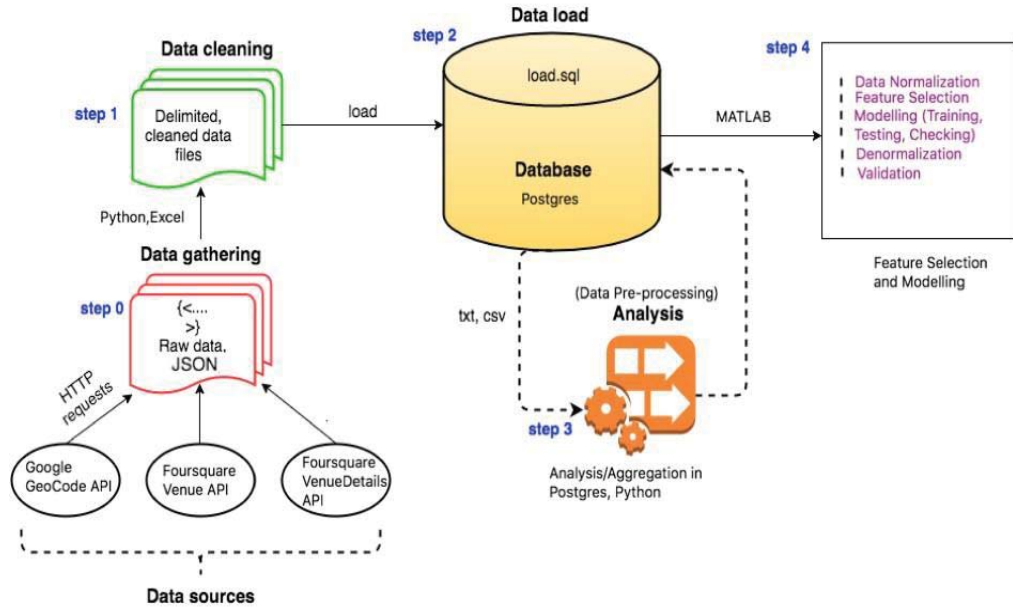


Figure 5. The workflow of the model followed in this study

3.2 Data gathering

Data used in this research gathered from Google Maps Geocoding API (Google Developers, 2018) and Foursquare APIs (Foursquare Developers, 2018) which are available publicly.

3.2.1 Gathering Administrative Area Names

Google Maps Geocoding API requires specific keywords to search, and there is no database for Turkey that can be used for this purpose. The website of Turkish Postal Services has been used to overcome the keyword problem. List of data that includes city, district, neighborhood, town, suburb area is retrieved (8.536 rows) (Postal Services, 2018).

3.2.2 Gathering coordinates

Google Maps Geocoding API is used to gather geographic coordinates which will be used in the request of Foursquare APIs. This API requires the name of the place such as “Sarıyer, İstanbul”, “Adana” or “Atatürk Mah., Yenice, Mersin”. It works just like Google Maps, it is a process of converting addresses into geographic coordinates, one can enter the keyword as a request to this API and it returns geographic coordinates and/or boundary coordinates of that place if it is an administrative area rather than a specific address such as neighborhoods, towns, cities.

Database on Postgre is created to store input and output data and used Python language to make API calls. In the end, it has been possible to retrieve 23.008 unique coordinates to use in Foursquare API requests.

3.2.3 Gathering Venues and Venue Details

Foursquare has various APIs that serves for different needs such as recommending a place nearby, searching for a specific place, and getting tips or photos of the specific place. In this research, Venue Search and Venue Explore APIs have been used to gather venue list. The request of this API includes geographic coordinates of a particular location, and the API returns a list of venues near the current location. Difference between those two is Search returns nearest venues while Explore returns top rated venues nearby. Since the coordinates include data from administrative areas, venue rating output of Venue Search API was missing 99% of calls. To be able to analyze venue ratings and components that effects score, more data which has rating info were required. Venue Explore API served for this purpose. 145.946 venue

gathered from API calls, and 68.356 of them were unique and had a rating of the venue.

At this stage, Python language has been used to make API calls and parse JSON response along Postgre database to store API request and response. Moreover, PyCharm and DataGrip have been used as an IDE for Python and Postgre accordingly.

3.3 Insights of data set

The total venue data set consists of 68346 rows of venue details. Variable summary of the data set is in Table 1.

Table 1. Summary of Variables in Data Set

Variable Name	Maximum Value	Minimum Value	Average Value
vrating	9	4	7.39
vlisted	2179	0	12.94
vtipcount	7184	0	40.31
vlikes	32648	0	176.49
vprice	4	0	1
vuserscount	2110361	1	7251.03
vvisitscount	10756287	1	40.31
vphotos	206718	0	482.71
vcheckinscount	7369516	0	12585.04

The oldest venue has creation the date of “2009-10-16” and newest has the date data retrieved, “2017-08-18”. Verified listings are only 8890 out of 68346. Dataset consists of eight cities in Turkey and the number of venues listed in these cities are listed in Table 2.

Table 2. Number of Venues in Cities

City Name	Number of Venues
Adana	2658
Ankara	6669
Antalya	7291
Bursa	4978
Gaziantep	1909
Istanbul	31896
Izmir	10114
Konya	2831

Grouped categories and number of records in these categories are listed in Table 3.

Table 3. Distribution of Venues in Categories

Category Name	Number of Records
Academic	136
Accommodation	3079
Art	835
Bar	3252
Business	1618
Cafe	7797
Entertainment	2210
Food	5932
Health	179
International Food	5234
Local Food	9624
Other	1065
Park	2142
Poi	4219
Religious	80
Restaurant	4293
Shop	10766
Spa	272
Sports	5288
Transportation	325
Grand Total	68346

All of the variables have been tested to see if they are coming from normal distribution first with plotting histograms then q-q (quantile-quantile) plots. Since

these plots are visual, it can be somewhat subjective and not airtight. For example, one can say vrating is normally distributed from the histogram below, in Figure 6, however, after considering the q-q plot of the variable, the need to run more test is evident. q-q plot is a scatterplot created by plotting two sets of quantiles, one is sample data which is vrating in this case, and the other one is assumed data that is normally distributed. If sample data is normally distributed, the line in the q-q plot must be linear. However, in this case, the data plot is not exactly linear.

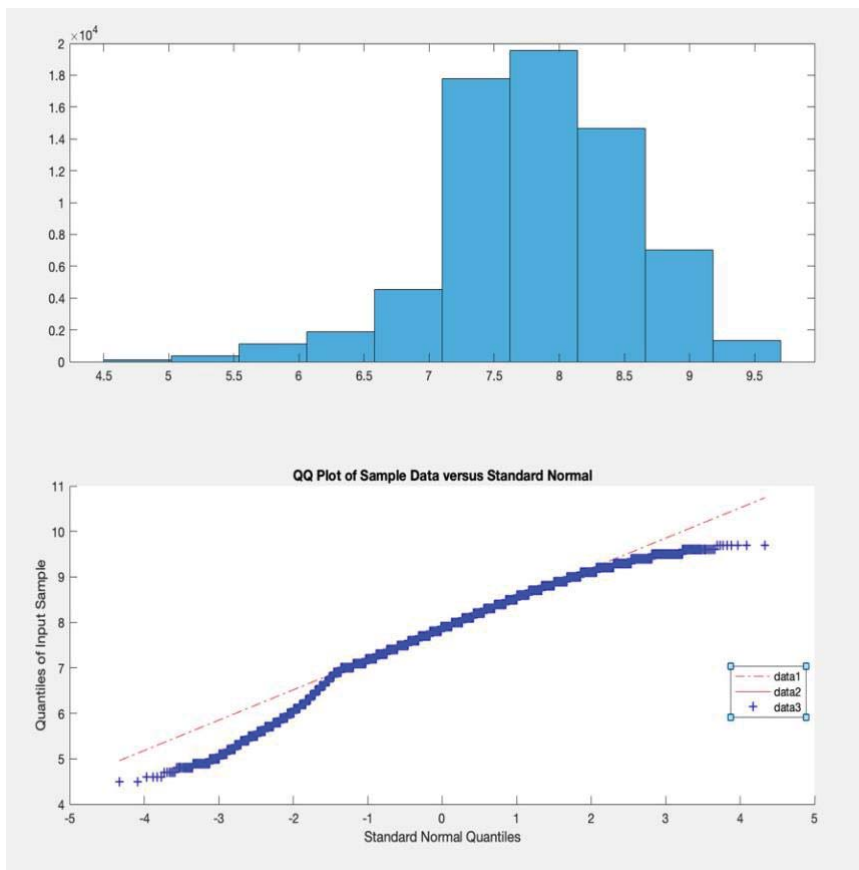


Figure 6. Histogram and q-q plot of the variable vrating

Along with visual tests, statistical tests have been run on these types of variables such as the Lilliefors test, which is based on the Kolmogorov-Smirnov test. Lilliefors tests the null hypothesis that the data come from a normally distributed population (Lilliefors, 1967).

In addition to Lilliefors and Kolmogorov-Smirnov tests, the Anderson–Darling test is used since Sim and Yan (2011) concluded in their comparison research that *AD* test performs better on asymmetric, skewed distributions. All of the tests resulted as $h = 1$, means that the null hypothesis must be rejected at the 5% significance level. Hence variables are not normally distributed.

3.4 Data cleaning

The data cleaning process has been conducted to detect and correct corrupt data from Venue Detail records. Incomplete, duplicate, missing data has been detected and removed from the dataset. After the cleaning process, 68.356 out of 145.946 venues has made the final data set. Variables of the set are defined below; (Foursquare Developers, 2018)

- *venueid*: A unique string identifier for the venue,
- *vname*: The best-known name for the venue,
- *vcat*: An array, possibly empty, of categories that have been applied to the venue,
- *vrating*: Numerical rating of the venue (0 through 10). Not all venues will have a rating,
- *vcheckinscount*: Total check-ins for the venue,
- *vuserscount*: Total users who have ever checked in the venue,
- *vtipcount*: Number of tips for the venue,
- *vvisitscount*: Total visits ever for the venue,
- *vlikes*: The count of users who have liked the venue, and groups containing any friends and others who have liked it,
- *vphotos*: A count and groups of photos for the venue,

- vcreatedat: Seconds since epoch when the venue was created. Converted to passed years since opening,
- vlisted: Number of lists that contain the venue,
- vprice: An object containing the price tier from 1 (least pricey) - 4 (most pricey),
- vverified: Boolean indicating whether the owner of the business has claimed it and verified the information,
- city: City where venue located in.

3.5 Data transformation

In data standardization phase categorical variables were converted to numerical ranges. For example; City variable had values of Adana, Ankara, Antalya, Bursa, Gaziantep, Istanbul, Izmir, and Konya. Those values converted to a numeric range from one to eight. However, due to the need of the data transformation process, categorical variables mapped to dummy variables. City variable has been converted to variable isistanbul, which is one if the city is Istanbul, zero if other.

Moreover, category variables for venues were multiple in most of the cases. Therefore, the first category has been selected as a primary category of venue. After retrieving distinct category number as 544, there has been a need for mapping those categories to main categories that are pre-defined by the researcher. Categories mapped to 20 main categories as followed; Academic Venue, Accommodation, Art, Bar, Business, Cafe, Entertainment, Food, Health, International Food Venue, Local Food Venue, Other, Park, POI (Point of Interest), Religious, Restaurant, Shop, Spa, Sports, Transportation. Then those categories converted to numerical values corresponding to them in alphabetical order from 1 to 20. Due to the need of the data

transformation process, this categorical variable is also converted to dummy variables. For this process, the number of venues in these categories has been listed and ordered by the count of them. Then they have been converted to dummy variables as “isFood, isPOI, isSports, isOther, isShop, isEntertainment”.

Also, another categorical variable named *vverified*, which represents the owner acclamation of venue was valued as False and True. This variable is also converted to dummy variable as *isverified* with values of one for True and zero for False.

Besides, the only date type variable *vcreatedat*, which shows the date venue has been created on Foursquare, has been converted to the number of past years since the venue has been created, then it is converted to dummy variable as *isnew* since 90% of the venues were created more than three years ago. This variable is one if the venue is created in last the two years, zero if else.

One last conversion has been applied to the *vprice* variable which is also categorical with the values varies between zero and four. The *isexpensive* variable has been created and mapped as one if *vprice* is three or four, zero if else.

Data transformation is necessary when using algorithms including neural networks, nearest neighborhood, and clustering. Especially for the neural network, normalizing input values in training samples will shorten the learning process. There are many methods for data normalization such as min-max normalization, z-score normalization, and normalization by decimal scaling (Kasasbeh, Shaaban, & Shalabi, 2006).

Since data is quite skewed continuous variables, which are variables other than dummy variables should be log transformed. Log transformation is a very debatable topic in literature, and it should be taken very seriously. In this research, all

of the variables that are transformed are counts of that action such as check-in, visits, likes. However, before applying a logarithmic transformation, all of the variables should be checked for zero values since the log of zero is undefined. In this case, there are two main approaches to be applied to variables; one of them is adding one to all values and then applying log transformation the other one is finding the minimum value and using half of that value for zero values for log transformation. In this study, the latter approach is chosen and applied. In Figure 7, the histogram of the original values of the vcheckinscount variable can be observed along with the same plot with transformed values in Figure 8.

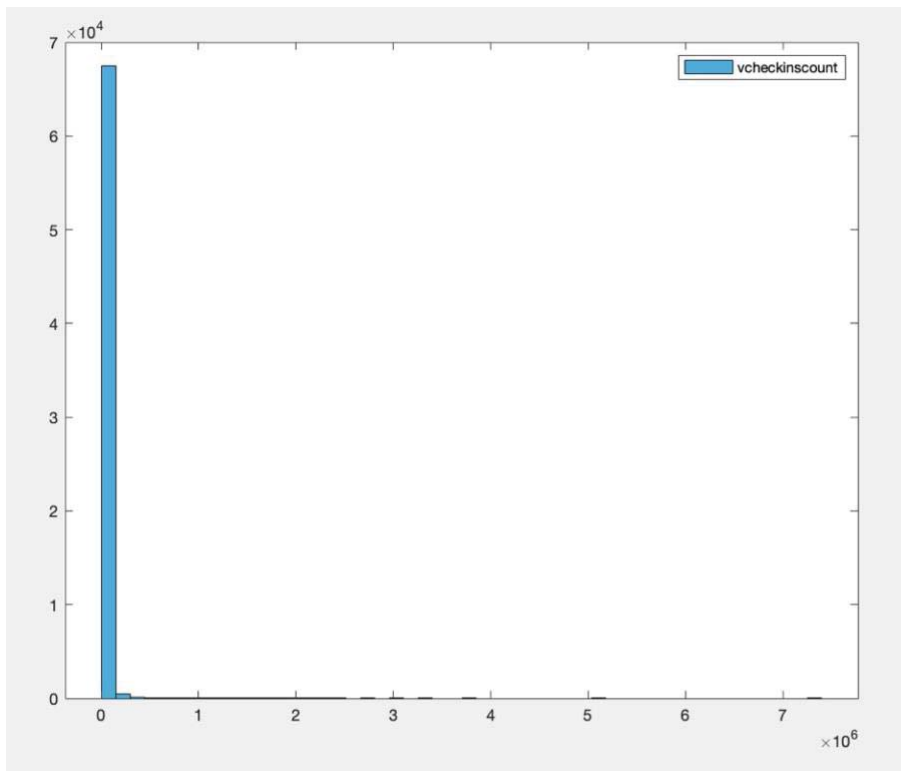


Figure 7. Histogram of original vcheckinscount variable

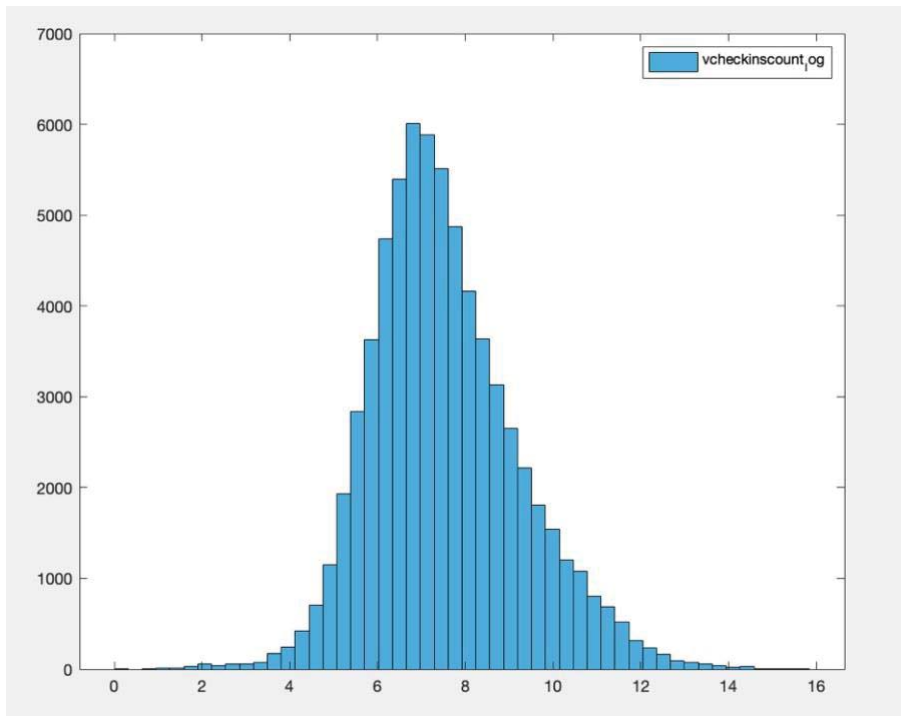


Figure 8. Histogram of vcheckinscount variable after log transformation

3.6 Feature selection

Correlation analysis on independent and dependent variables have been executed, it showed significant correlation on certain variables. Also, Principal Component Analysis (PCA) has been executed on eight independent and continues variables, and as a conclusion, one principal component was enough to explain most of the components. When there is a high degree of correlation among the original predictive variables, only the first few of the principal components are likely to capture the majority of the variance of the original predictive variables. Hence, PCA also supports the correlation analysis executed on this dataset (Maitra & Yan, 2008).

Feature selection is preferable to feature transformation when the original units and the meaning of features are essential, and the modeling goal is to identify an influential subset. When categorical features are present, and numerical

transformations are inappropriate, feature selection becomes the primary means of dimension reduction.

For this purpose, feature selection using neighborhood component analysis for regression model has been chosen by using `fsrnca` function in MATLAB (The MathWorks, Inc., 2018). The weights of the features have been observed, and irrelevant ones were close to zero. According to the result of this selection in Table 4, seven features were chosen to explain the data well enough. Those features are `vcheckinscount`, `vuserscount`, `vvisitscount`, `vliked`, `vlisted`, `isistanbul`, `isnew`. Calculated feature weights for all the variables are below in Table 4 and Figure 9;

Table 4. Calculated Feature Weights for All Variables

Variable	Feature	Weight	Variable	Feature	Weight
Var1:	<code>vcheckinscount</code>	55.418	Var10:	<code>isistanbul</code>	43.073
Var2:	<code>vuserscount</code>	47.799	Var11:	<code>isfood</code>	0
Var3:	<code>vtipcount</code>	0	Var12:	<code>ispoi</code>	0
Var4:	<code>vvisitscount</code>	32.802	Var13:	<code>issports</code>	0
Var5:	<code>vlikes</code>	72.136	Var14:	<code>isother</code>	0
Var6:	<code>vphoto</code>	0	Var15:	<code>isshop</code>	0
Var7:	<code>vlisted</code>	34.586	Var16:	<code>isent</code>	0
Var8:	<code>vverified</code>	0	Var17:	<code>isnew</code>	13.452
Var9:	<code>isexpensive</code>	0			

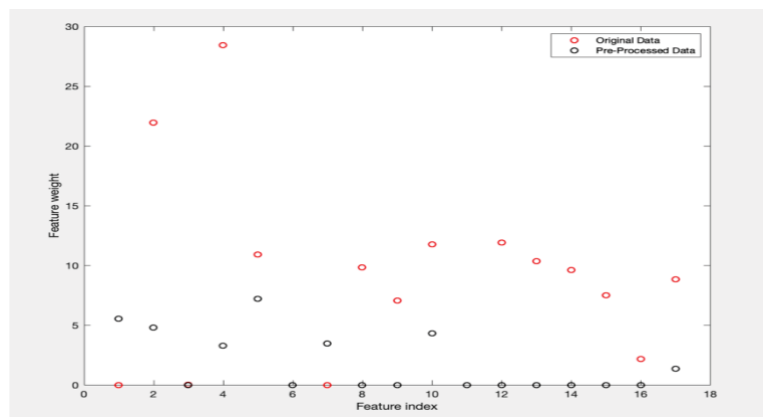


Figure 9. Feature index of feature weights of original data compared to transformed data

In Figure 10, there is a sample of the pre-processed data with the selected features that will be used in modeling.

68346x8 table

	1	2	3	4	5	6	7	8
	vcheckinscount	vuserscount	vvisitscount	vlikes	vlisted	vistanbul	vnew	vrating
1	7.8853	7.9862	8.6170	4.6634	0	0	0	2.0412
2	11.5685	10.7167	12.0093	7.3658	3.2581	0	0	2.0149
3	8.5090	8.2028	8.9219	5.3706	3.1781	0	0	2.1163
4	12.5893	11.5453	12.8870	8.0478	4.3041	0	0	2.1163
5	9.4469	8.9269	9.5503	6.0403	3.2189	0	0	2.0669
6	12.0183	11.8230	12.2202	8.8853	6.7696	0	0	2.1748
7	12.5428	11.7067	12.8439	8.4301	5.7301	0	0	2.1972
8	8.9782	9.3601	9.9048	5.9026	3.9318	0	0	1.9879
9	11.2152	10.5324	11.2908	7.4378	5.0752	0	0	2.1748
10	8.5867	8.0150	8.9430	3.5264	0	0	0	2.1041
11	11.6033	10.7360	11.7069	7.7205	3.2189	0	0	2.0669
12	14.9038	13.8023	15.1639	9.3482	5.5452	0	0	2.1401
13	10.4516	10.2785	10.8516	7.1907	4.9200	0	0	2.1163
14	14.5710	12.9251	14.8529	9.6035	5.3706	0	0	2.2192
15	14.7238	13.6008	15.0618	10.0842	5.6937	0	0	2.1282
16	8.1365	7.7519	8.1859	4.9767	3.0445	0	0	2.0794
17	12.2813	11.5054	12.6550	6.8167	3.6889	0	0	2.1282
18	10.5593	9.9895	11.0553	6.5396	3.6889	0	0	2.1861
19	14.6572	13.5952	14.9604	9.0348	5.4889	0	0	2.1633
20	11.2495	10.7696	11.4406	7.4838	4.8978	0	0	2.1401

Figure 10. Sample of the dataset after pre-processing

CHAPTER 4

BUILDING A MODEL

At this section, cleaned data and selected features have been used to build the ANFIS model. Data is partitioned with the `cvpartition` function of MATLAB which creates a random non-stratified partition for holdout validation on observations by dividing the observations into a training set and a test set (The MathWorks Inc., 2018). 30% of the data (number of samples: 20503) is partitioned as test data, and 70% of it is used for model training (Number of samples: 47843).

MATLAB Fuzzy Logic Toolbox is used in this present work. The number of training epochs, membership functions and the number of fuzzy rules plays an important role in modeling to avoid over-fitting or under-fitting. ANFIS function of MATLAB allows editing these options. In this work different epochs, membership functions and optimization methods have been used. ANFIS function tunes Fuzzy Inference System (FIS) using either a back-propagation algorithm alone or in combination with a least squares type of method. This adjustment allows fuzzy systems to learn from the data they are modeling. In this research, both optimization methods are used, and results are compared. The parameters related to membership functions are subject to change during the learning process of the network. Gradient vector enables the adjustment of these parameters with computing the rate of change of the function to measure the fitness of the fuzzy inference system model for a given set of parameters. Optimization and adjustment of the parameters can be applied after gradient vector is obtained with the goal to reduce error measure which is usually a sum of the squared difference between actual and predicted outputs (The MathWorks Inc., 2018).

ANFIS function tunes Sugeno type fuzzy inference system using training data. Model validation is the process followed after model building, which data set that is not used in training process of FIS, is presented to the trained FIS model to see how the FIS model predicts with distinct data set. Training data is used for modeling with options below;

- epoch = [10:10:100];
- opt = anfisOptions;
- opt.InitialStepSize = 10;
- opt.ErrorGoal = 0.05;
- opt.StepSizeIncreaseRate = 3;
- opt.StepSizeDecreaseRate = 0.3;
- opt.OptimizationMethod = 0;

Optimization Method defines the method that will be used in membership function training. There are two options; one (1) is a hybrid method which uses a combination of backpropagation to evaluate input membership function parameters, and least squares estimation to evaluate output membership function parameters and zero (0) which uses backpropagation gradient descent to evaluate all parameters. In this research, both optimization methods have been used, and results are compared.

The step size option is defined to determine the magnitude of the gradient transitions in the parameter space since ANFIS training algorithm tunes FIS parameters using gradient descent optimization methods.

ANFIS function is run on seven input and one output with training data set. Epoch value is set as 10 and increased till 100 for hybrid optimization method (from now on will be called anfisHybrid) and 500 for backpropagation gradient descent optimization method (from now on will be called anfisBckprop). The reason behind

the different epochs is the computational cost of running the hybrid optimization method.

4.1 Model details of the anfishybrid model

In this model, the optimization parameter (OptimizationMethod) is defined as one while other parameters are set as defined in the previous section. In Figure 11, a high-level diagram of the fuzzy inference system (FIS) can be viewed for the anfishybrid model. There are two membership functions for each input and 128 fuzzy rules for one output.

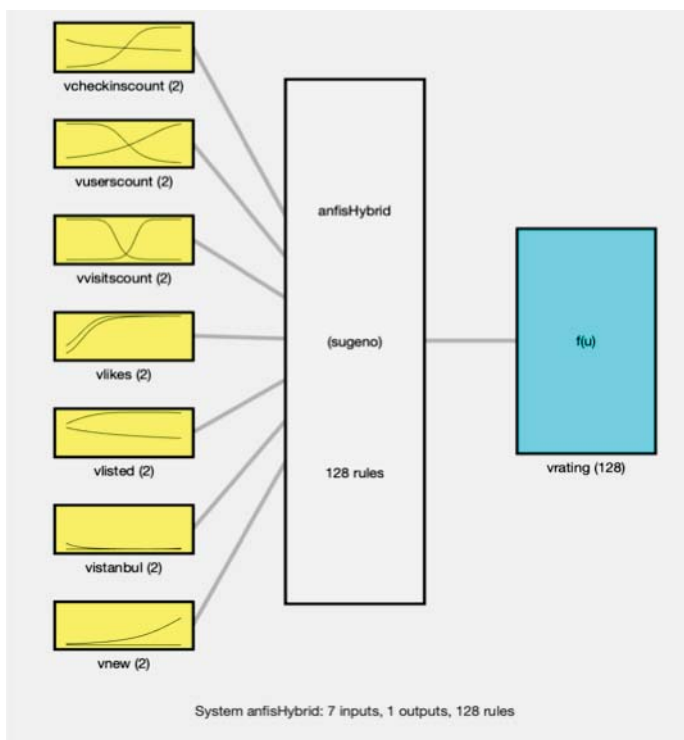


Figure 11. A high-level diagram of the Fuzzy Inference System (FIS) for anfishybrid model

Input membership functions in this model are generalized bell membership function, which is specified by three parameters, named as gbellmf in MATLAB. This membership function has one more parameter than the Gaussian, so it can approach a

non-fuzzy set if the free parameter is tuned. Gaussian and bell membership functions are the most used methods for specifying fuzzy sets. Both of these curves have the advantage of being smooth and nonzero at all points. Also, the output membership function in the model is linear.

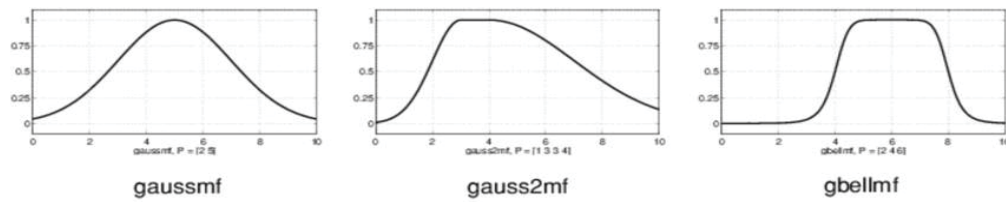


Figure 12. Examples to membership functions considered in this model (gbellmf is used)

After training FIS with different epochs starting from 10 to 100, the error graphic is as in Figure 13.

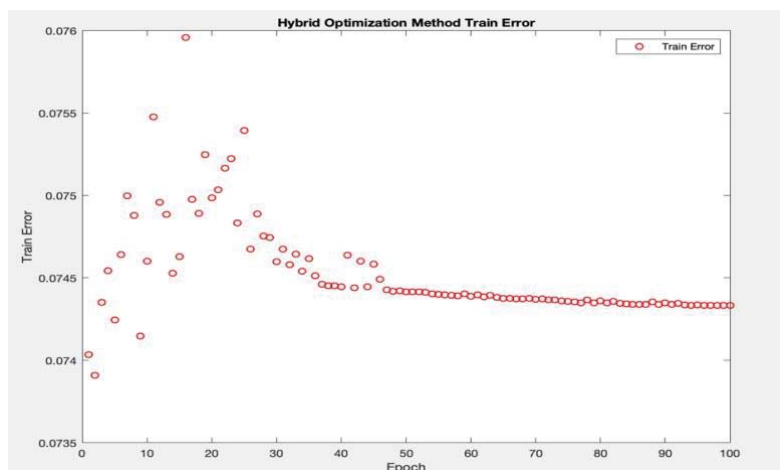


Figure 13. Train error for hybrid optimization method (anfisHybrid)

The result of the model, number of nodes, training data pairs and fuzzy rules and minimal training *RMSE* can be seen in Table 5. In addition to that complex model structure of the method is shown in Figure 14 below.

Table 5. Final Results of anfisHybrid Model with 100 Epochs

Final Result	Value
Number of nodes	294
Number of linear parameters	1024
Number of nonlinear parameters	42
Total number of parameters	1066
Number of training data pairs	47843
Number of checking data pairs	0
Number of fuzzy rules	128
Minimal training <i>RMSE</i>	0.07391

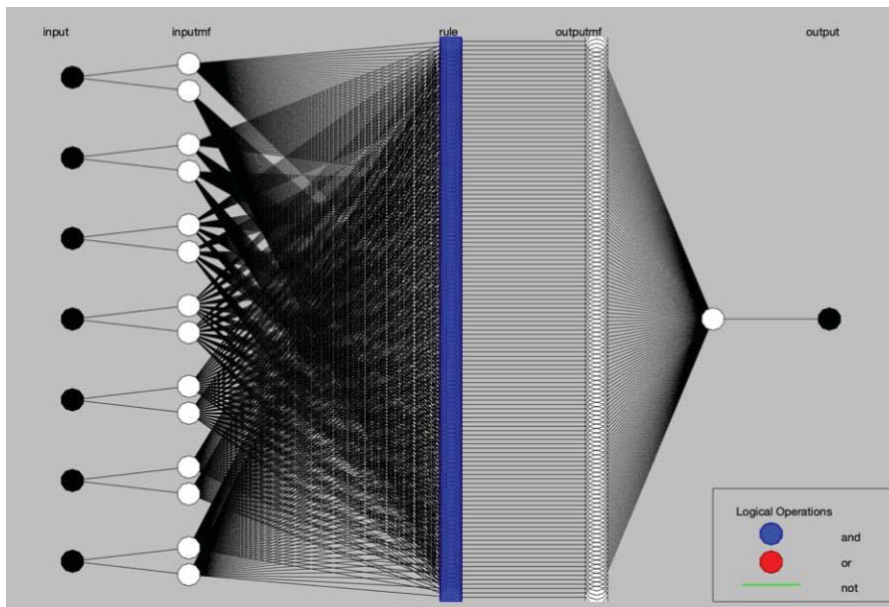


Figure 14. ANFIS model structure for anfisHybrid

Rule viewer for a certain input ([7.9065; 7.7786; 8.1617;5.2040;2.7726 ;1;1]) is captured in Figure 15 as an example. The output obtained with these input values is 2.12 whereas the real value in the data set is 2.17. When we get the exponential of this value, we get the result as 8.31 which is the predicted rating for the venue.

After these insights about the model, test data that has been obtained by using cvpartition function is used to see how this model performs when it comes to prediction. Although Rule Viewer interface is useful to calculate the output for given inputs using the

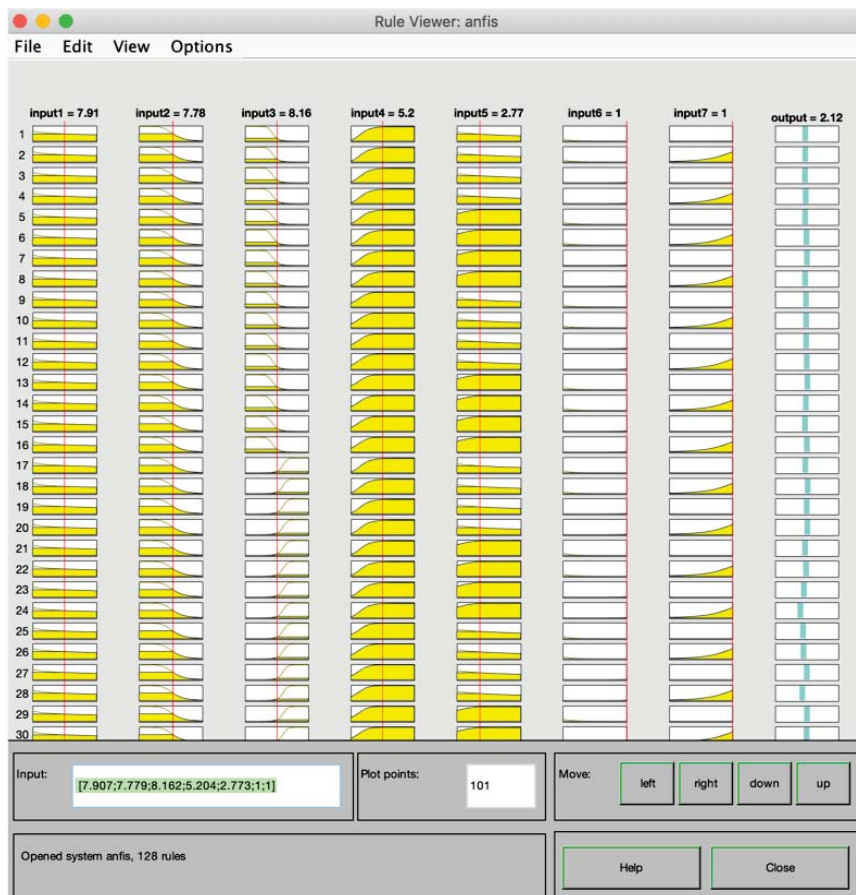


Figure 15. Fuzzy rules and membership functions on Rule Viewer with input for the anfisHybrid model

FIS model, MATLAB tool has evalfis function to evaluate the prediction for the given test data set (The MathWorks Inc., 2018). In addition to that, min $RMSE$ has been calculated for test data and prediction. In Figure 16 and 17, the comparison of predicted results for training and testing data sets can be found.

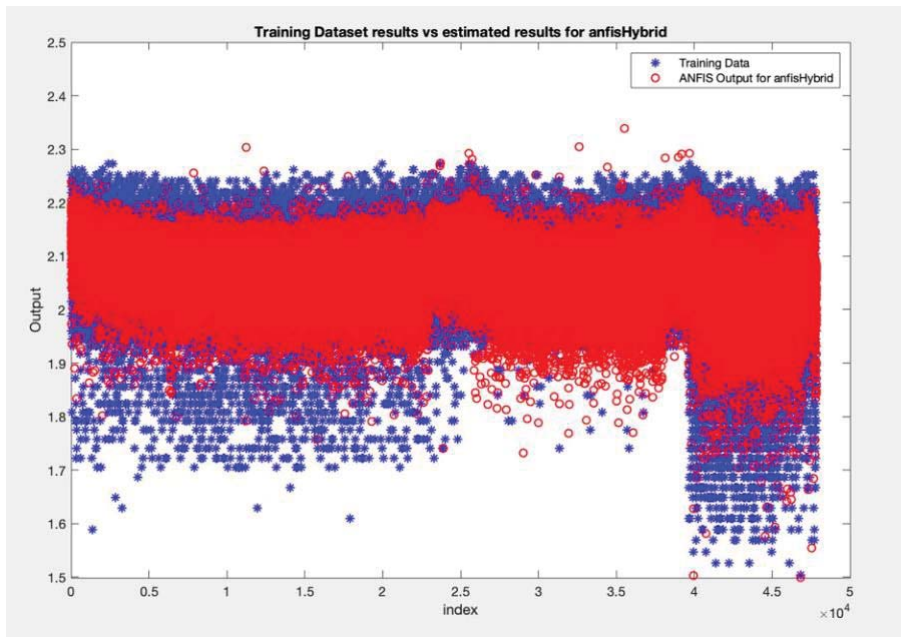


Figure 16. Training data results compared to estimated results for anfisHybrid

Surface graphic of the model for two selected variables is placed in Figure 18, and other attributes have to be fixed to obtain this surface. Values of the fixed attributes are as follow; [NaN NaN 8.096 5.197 3.843 0 1]

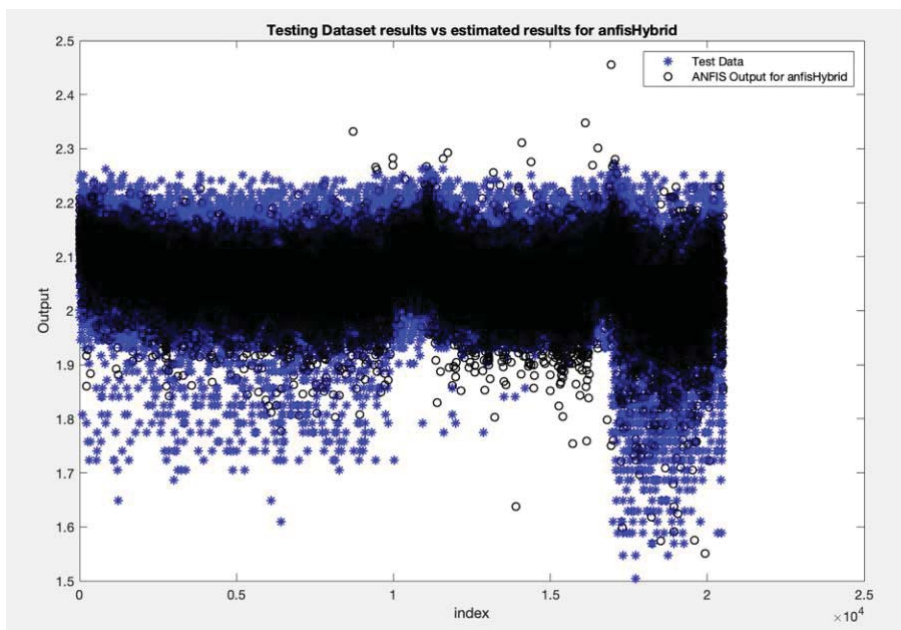


Figure 17. Test data set results compared to estimated results for anfisHybrid

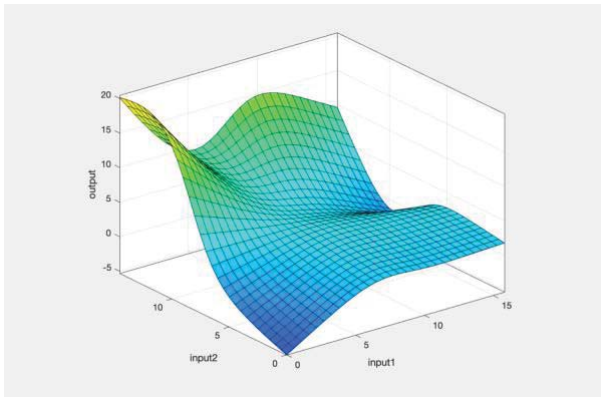


Figure 18. Surface graphic for vrating (output) compared to vcheckinscount (input1) and vuserscount (input2)

4.2 Model details of the anfisBckProp model

In this model, the optimization parameter (OptimizationMethod) is defined as zero while other parameters are set as defined in the previous section. In Figure 19, a high-level diagram of the fuzzy inference system (FIS) can be viewed for the anfisBckProp model. There are two membership functions for each input and 128 fuzzy rules for one output.

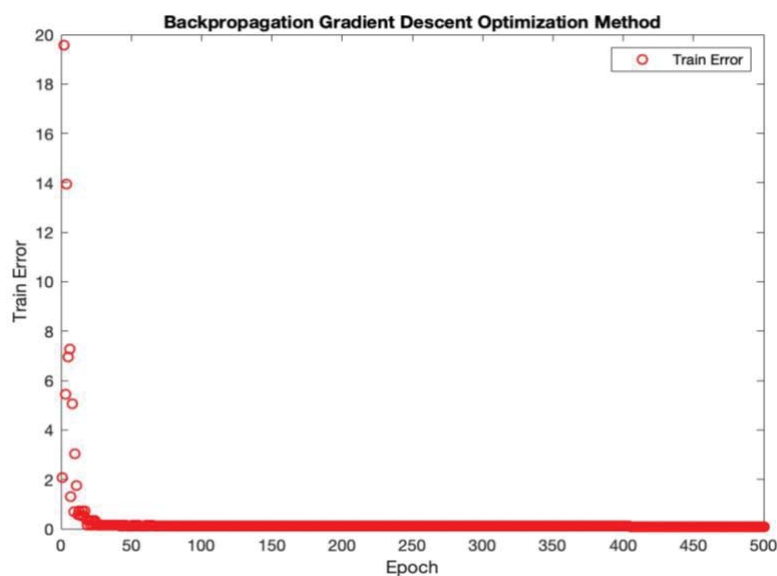


Figure 19. Back-Propagation Gradient Descent Optimization method train error

Comparison of estimated results of data set and training data set can be seen in Figure 20. The figure shows that predicted results are not aligned with the real results of the training data.

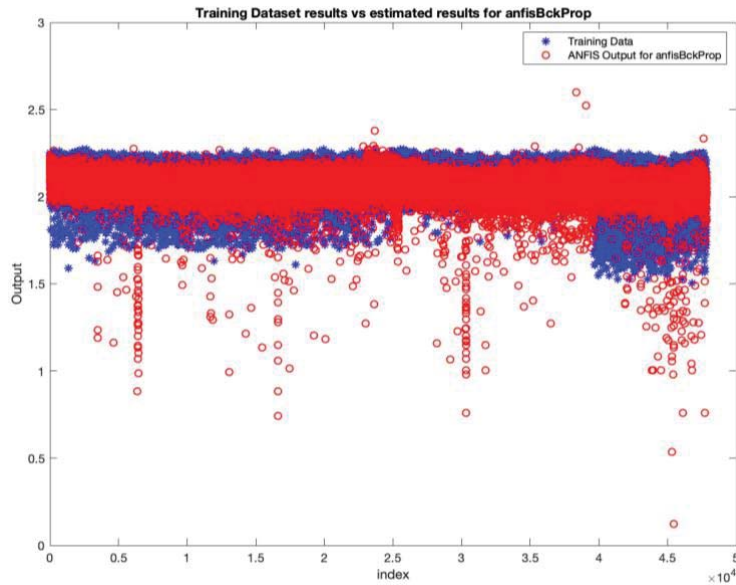


Figure 20. Training data results compared to estimated results for anfisBckProp

Testing data comparison can be observed in Figure 21. Black dots are the estimated results, and it shows that the result is not viable.

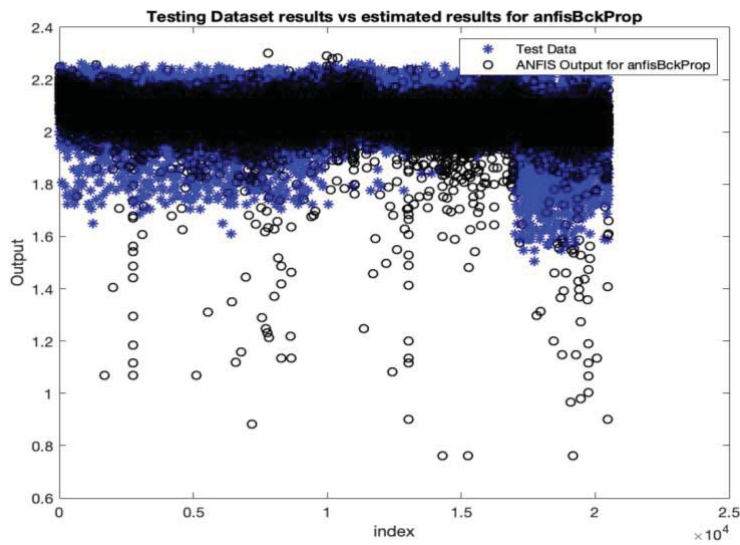


Figure 21. Test data set results compared to estimated results for anfisBckProp

4.3 Other ANFIS variations

In addition to these two models, different variations of the ANFIS model are considered in this work such as tuning ANFIS by creating initial FIS model, increasing epochs and modifying membership functions. Details of the models are listed in Table 6.

Table 6. Variations of ANFIS Model

Model Name	Initial FIS Type	Membership Function	Epoch
anfisBckProp	NA	Generalized Bell	500
anfisHybrid	NA	Generalized Bell	100
anfisGpGa100	Grid Partition	Gaussian	100
anfisGpGa50	Grid Partition	Gaussian	50
anfisGpGb100	Grid Partition	Generalized Bell	100
anfisScGa100	Sub-clustering	Gaussian	100
anfisScGa50	Sub-clustering	Gaussian	50

After tuning to find the best fitting ANFIS model, results are compared. Two common metrics to measure the accuracy of continuous variables, Mean Absolute Error (*MAE*) and Root Mean Square Error (*RMSE*) have been used for this purpose.

The *MAE* measures the distance between predicted and actual results without concerning their direction. The formula can be written as;

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i|$$

where f_i is the prediction and y_i the actual value. The score calculated with *MAE* is linear since errors in this formula have the same weight.

RMSE can be used along with *MAE* to compare the models to comprehend the errors. Firstly, the difference between predicted and actual values are calculated and

squared. Then, the errors are averaged which leads to the result of assigning large weights to large errors (Mathura, Gleska, & Buisb, 2016).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - y_i)^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2}$$

Both of these metrics are calculated for Training Data and Test Data for all ANFIS models. Results are compared in the next section.

4.4 Other regression models

Regression Learner App on MATLAB can run a search for the best regression model type, including linear regression models, regression trees, Gaussian process regression models, support vector machines, and ensembles of regression trees (The MathWorks Inc., 2018).

RMSE of these models is calculated and listed for all models selected. Results are compared in the next section.

CHAPTER 5

RESULTS AND DISCUSSION

Neuro-fuzzy regression is a field of research that took attention of both academic and industrial researchers in recent decades. The neural network has learning and generalization abilities, and Fuzzy Systems are highly interpretable and convenient tools for modeling complex systems, bringing these powerful tools together created a robust alternative model.

Usage of ANFIS methodology can vary such as;

- Estimating demand quantity for manufacturing companies,
- Predicting outages of cell edge users,
- Predicting heavy metal concentration in rivers, quality monitoring,
- Estimating regional rainfall amount,
- Predicting ERP user satisfaction.

ANFIS has proven itself as a prominent and successful alternative to traditional approaches. The practical importance of this project resides in the prediction of a score. This work shows that in encountering problems which deal with uncertainty, traditional approaches may not be the only choice because of the complexity of the problem.

This study showed that even undisclosed scoring systems could be mimicked by the ANFIS models with very notable success. One drawback of ANFIS may be the computational cost comparing to traditional regression methods. In researches on ANFIS, promising results recorded by the proposed method however, the optimal sets of hidden layer nodes and their respective weighted links still are a challenge.

Also, there are few research studies that compare data mining and ANFIS methods used in forecasting. This work tries to supplement this research field.

Data set is partitioned, and 30% of the data (20503) is used for testing, and 70% of it for training purposes (47843). Regression models are used to predict the rating of the venues on Foursquare based on specific parameters. After applying data pre-processing techniques, some of the features were able to explain the model well enough. Seven features out of 18 were chosen since they were able to explain the data well enough. Those features are vcheckinscount, vuserscount, vvisitscount, vliked, vlisted, isistanbul, isnew. Selected features have been used in ANFIS models with different optimization methods. The ANFIS function used in this study tuned Fuzzy Inference System (FIS) using both a back-propagation algorithm alone and in combination with a least squares type of method. Also, other machine learning algorithms have been used to compare the results with the ANFIS model.

For both *RMSE* and *MAE* metrics, smaller values mean a better fit. Therefore, anfisHybrid and anfisGpGa100 perform slightly better according to the results. These two models are selected as ANFIS models to be compared with other regression models.

Regression Learner App on MATLAB is used to search for the best regression model type, including linear regression models, regression trees, Gaussian process regression models, support vector machines, and ensembles of regression trees (The MathWorks, Inc., 2018). Results of these models and selected ANFIS models are compared. According to Table 7 and Table 8, the anfisHybrid model has a minimum *RMSE* value.

Table 7. Comparison of ANFIS Models

Model Name	<i>RMSE</i> Training	<i>RMSE</i> Test	<i>MAE</i> Training	<i>MAE</i> Test
anfisBckProp	0.09537	0.09649	0.06662	0.06733
anfisHybrid	0.07391	0.08164	0.0571	0.05615
anfisGpGa100	0.07395	0.07597	0.05612	0.05705
anfisGpGa50	0.07399	0.07606	0.05614	0.05714
anfisGpGb100	0.4859	0.47672	0.30012	0.29963
anfisScGa100	0.07762	0.07787	0.05891	0.05908
anfisScGa50	0.07762	0.07787	0.05891	0.05908

Table 8. Comparison of All Models

Model Name	<i>RMSE</i>
Linear Regression	0.081418
Tree - Coarse Tree	0.079923
SVM - Medium Gaussian SVM	0.076397
Ensemble - Bagged Trees	0.075767
Exp. Gaussian Process Regression	0.074558
anfisBckProp	0.095202
anfisHybrid	0.073909
anfisGpGa100	0,073950

Results have been compared based on *RMSE* results and graphics of prediction results. *RMSE* values of all of the models can be seen in Table 8. Generally speaking, most of the models present similar and acceptable results, however best results are obtained from ANFIS-hybrid method.

Generally speaking, application of this algorithm was about engineering problems, and with smaller data sets, however, this study shows that application of the algorithm can be used for data mining of other systems with big data as well.

The future work can be through applying various ANFIS architecture with other regression algorithms together. Another important point will be improving and automating the tuning process of the algorithm. Also, removing the obstacle of the computational cost will give an advantage for those want to work on this model. Using a modified algorithm (combined with other optimization algorithms) may provide better convergence during the training process.

REFERENCES

- Bell, J. (2014). *Machine learning: hands-on for developers and technical professionals*. Indianapolis, IN: John Wiley & Sons.
- Benitez, J. M., Castro, J. L., & Requena, I. (1997). Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks*, 8(5), 1156-1164.
- Bezdek, J. C. (1993a). Fuzzy models-What are they, and why? [Editorial]. *IEEE Transactions on Fuzzy Systems*, 1(1), 1 - 6.
- Bezdek, J. C. (1993b). A review of probabilistic, fuzzy, and neural models for pattern recognition. *Journal of Intelligent and Fuzzy Systems*, 1(1), 1-25.
- Bezdek, J. C., Mikhil, P. R., Keller, J., & Krisnapuram, R. (1999). *Fuzzy models and algorithms for pattern recognition and image processing*. Norwell, MA: Kluwer Academic Publishers.
- Caudill, M. (1987). Neural network primer: Part I. *AI Expert*, 2(12), 46-57.
- Caudill, M., & Butler, C. (1990). *Naturally intelligent systems*. Cambridge, MA: MIT Press.
- Chameau, J. L., & Santamarina, J. C. (1987). Membership functions part I: Comparing method of measurement. *International Journal of Approximate Reasoning*, 1(3), 287-301.
- Colettia, G., & Scozzafavab, R. (2006). Conditional probability and fuzzy information. *Computational Statistics & Data Analysis*, 51(1), 115 – 132.
- Coppia, R., Gilb, M. A., & Kiersc, H. A. (2006). The fuzzy approach to statistical analysis. *Computational Statistics & Data Analysis*, 51(1), 1 – 14.
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining “Gamification”. *MindTrek '11 Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (pp. 9-15). New York, NY: Association for Computing Machinery.
- Dubois, D., & Prade, H. (2000). *Fundamentals of Fuzzy Sets (The handbooks of Fuzzy Sets)*. Toulouse, France: Springer.
- Foursquare for Business. (2017). Foursquare for business. Retrieved from <http://business.foursquare.com>
- Foursquare Developers. (2018). Foursquare developers API. Retrieved from <https://developer.foursquare.com/>

- Foursquare Developers. (2018). Get details of a venue. Retrieved from <https://developer.foursquare.com/docs/api/venues/details>
- Frith, J. (2013). Turning life into a game: Foursquare, gamification, and personal mobility. *Mobile Media & Communication*, 1(2), 248-262.
- Giles, R. (1988). The concept of grade of membership. *Fuzzy Sets and Systems*, 25(3), 297-323.
- Google Developers. (2018). *Google Maps APIs*. Retrieved from <https://developers.google.com/maps/documentation/geocoding>
- Gordon, E., & Silva, A. (2011). *Net locality: Why location matters in a networked world*. West Sussex, England: Wiley Publishing.
- Haykin, S. S. (1998). *Neural networks: A comprehensive foundation* (2nd Ed.). Upper Saddle River, NJ: Prentice Hall.
- Hisdal, E. (1988). Are grades of membership probabilities? *Fuzzy Sets and Systems*, 25(3), 325 - 348.
- Humphreys, L. (2007). Mobile social networks and social practice: A case study of Dodgeball. *Journal of Computer-Mediated Communication*, 13(1), 341-360.
- Jang, J.-S. R. (1991). Fuzzy modeling using generalized neural networks and Kalman Filter algorithm. *AAAI'91 Proceedings of the Ninth National Conference on Artificial Intelligence* (pp. 762-767). Anaheim, CA: AAAI Press.
- Jang, J.-S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665 - 685.
- Jang, J.-S. R. (1997). Fuzzy inference system. In E. Mizutani, C.-T. Sun, & J.-S. R. Jang, *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence* (pp. 73-90). Upper Saddle River, NJ: Prentice Hall.
- Kandel, A., Martins, A., & Pacheco, R. (1995). Discussion: On the very real distinction between fuzzy and statistical methods. *Technometrics*, 37(3), 276-281.
- Kang, G. T., & Sugeno, M. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1), 15-33.
- Kasasbeh, B., Shaaban, Z., & Shalabi, L. (2006). Data mining: A preprocessing engine. *Journal of Computer Science*, 2(9), 735-739.
- Klir, G. J. (2005). *Uncertainty and information: Foundations of generalized information theory*. Hoboken, NJ: Wiley.
- Kosko, B. (1992). *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence*. Englewood Cliffs, NJ: Prentice Hall.

- Kulla, E., Elmazi, D., & Barolli, L. (2016). Neuro-adaptive learning fuzzy-based system for actor selection in wireless sensor and actor networks. *10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)* (pp. 488-493). Fukuoka, Japan: IEEE.
- Laviolette, M., Seaman, J.W., Barrett, J.D., & Woodall, W.H. (1995). A Probabilistic and Statistical View of Fuzzy Methods. *Technometrics*, 37(3), 249-292.
- Leighton, E., & Saker, M. (2017). *Location-based social media: space, time and identity*. Cham, Switzerland: Palgrave Pivot.
- Lilliefors, H. W. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62(318), 399-402.
- Lindley, D. V. (1982). Scoring rules and the inevitability of probability. *International Statistical Review*, 50(1), 1-11.
- Lindqvist, J., Cranshaw, J., Wiese, J., Hong, J., & Zimmerman, J. (2011). I'm the mayor of my house: Examining why people use foursquare - a social-driven location sharing application. *In Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (CHI '11)* (pp. 2409-2418). New York, NY: Association for Computing Machinery.
- Luke, R. (2006). The phoneur: mobile commerce and the digital pedagogies of the wireless web. In: Trifonas P.P. (Ed.) *Communities of Difference* (pp. 185-204). London, England: Palgrave Macmillan.
- Mabuchi, S. (1992). An interpretation of membership functions and the properties of general probabilistic operators as fuzzy set operators - Part I: Case of type 1 fuzzy sets. *Fuzzy Sets and Systems*, 49(3), 271-283.
- Maitra, S., & Yan, J. (2008). Principle Component Analysis and Partial Least Squares: Two Dimension Reduction Techniques for Regression. *2008 Discussion Paper Program* (pp. 79-90). Arlington, VA: Casualty Actuarial Society.
- Mandal, D. (2018). Adaptive neuro-fuzzy inference system based grading of basmati rice grains using image processing technique. *Applied System Innovation*, 1(12), 1-19.
- Mathura, N., Gleska, I., & Buisb, A. (2016). Comparison of adaptive neuro-fuzzy inference system (ANFIS) and Gaussian processes for machine learning (GPML) algorithms for the prediction of skin temperature in lower limb prostheses. *Medical Engineering and Physics*, 38(10), 1083-1089.
- McKenzie, G. D. (2011). *Gamification and location-based services*. Presented as a Vision Statement for the Cognitive Engineering for Mobile GIS Workshop at COSIT 2011. Belfast, ME.

- Mencar, C. (2013). Interpretability of fuzzy systems. In: Masulli F., Pasi G., Yager R. (Eds.) *Fuzzy Logic and Applications. WILF 2013. Lecture Notes in Computer Science* (Vol. 8256, pp. 22-34), Genoa, Italy: Springer.
- Norwich, A. M., & Turksen, I. B. (1982). The construction of membership functions. In R. R. Yager, *Fuzzy sets and possibility theory: recent developments* (pp. 6167). New York, NY: Pergamon Press.
- Palnitkar, R. M., & Cannady, J. (2004). A review of adaptive neural networks. *IEEE SoutheastCon 2004 Proceedings* (pp. 3-47). Greensboro, NC: IEEE.
- Postal Services. (2018). Posta Kodu Öğrenme Ekranı. Retrieved from <http://postakodu.ptt.gov.tr>
- Rasmussen, C. E., Williams, C. K. (2006). *Gaussian processes for machine learning*. Cambridge, MA: The MIT Press.
- Saker, M., & Leighton, E. (2016). Everyday life and locative play: An exploration of Foursquare and playful engagements with space and place. *Media, Culture & Society*, 38(8), 1-15.
- Salt, S. (2011). *Social location marketing: outshining your competitors on Foursquare, Gowalla, Yelp & other location sharing sites*. Indianapolis, IN: Que.
- Schwartz, R., & Halegoua, G. R. (2014). The spatial self: Location-based identity performance on social media. *New Media & Society*, 17(10), 1643-1660.
- Sheridan, S. E., & Skjoth, P. (1984). Automatic kiln control at Oregon Portland Cement Company's Durkee plant utilizing fuzzy logic. *IEEE Transactions on Industry Applications*, IA-20(3), 562-568.
- Sim, C. H., & Yap, B. W. (2011). Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation*, 81(12), 2141-2155.
- Singla, J. (2015). Comparative study of Mamdani-type and Sugeno-type fuzzy inference systems for diagnosis of diabetes. *2015 International Conference on Advances in Computer Engineering and Applications (ICACEA)* (pp. 517-522). Ghaziabad, India: IMS Engineering College.
- Sugeno, M. (1985). *Industrial applications of fuzzy control*. New York, NY: Elsevier Science Inc.
- Sugeno, M., & Takagi, T. (1983). Derivation of fuzzy control rules from human operator's control actions. *IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis* (pp. 55-60). Marseille, France: IFAC Proceedings Volumes

- Sugeno, M., & Takagi, T. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-15*(1), 116-132.
- Sugeno, M., & Yasukawa, T. (1993). A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems, 1*(1) 7-31.
- Sugeno, M., Turksen, B., & Nakanishil, H. (1993). A review and comparison of six reasoning methods. *Fuzzy Sets and Systems, 57*(3), 257-294.
- Sutko, D. M., & Silva, A. D. (2011). Location-aware mobile media and urban sociability. *New Media & Society, 13*(5), 807-823.
- Takagi, H., & Hayashi, I. (1991). NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning, 5*(3), 191-212.
- The MathWorks, Inc. (2018, December 01). Create cross-validation partition for data. Retrieved from <https://www.mathworks.com/help/stats/cvpartition.html>.
- The MathWorks, Inc. (2018, December 23). Evaluate fuzzy inference system. Retrieved from <https://www.mathworks.com/help/fuzzy/evalfis.html>.
- The MathWorks, Inc. (2018, December 01). Feature selection using neighborhood component analysis for regression. Retrieved from <https://www.mathworks.com/help/stats/fsrnca.html>.
- The MathWorks, Inc. (2018, December 07). Neuro-adaptive learning and ANFIS. Retrieved from <https://www.mathworks.com/help/fuzzy/neuroadaptive-learning-and-anfis.html>.
- The MathWorks, Inc. (2018, December 20). Train regression Models in regression learner app. Retrieved from <https://www.mathworks.com/help/stats/train-regression-models-in-regression-learner-app.html>
- Turksen, I. B. (1991). Measurement of membership functions and their acquisition. *Fuzzy Sets and Systems, 40*(1), 5-38.
- Widrow, B., & Winter, R. (1988). Neural nets for adaptive filtering and adaptive pattern recognition. *Computer, 21*(3), 25 - 39.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*(3), 338-353.
- Zadeh, L. A. (1968). Fuzzy algorithms. *Information and Control, 12*(2), 94-102.
- Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-3*(1), 28-44.

- Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 100(1), 3-28.
- Zadeh, L. A. (1984). Making computers think like people [Fuzzy Set Theory]. *IEEE Spectrum*, 21(8), 26 - 32.
- Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, 37(3), 77-84.
- Zadeh, L. A. (1995). Probability theory and fuzzy logic are complementary rather than competitive. *Technometrics*, 37(3), 271-276.
- Zadeh, L. A. (1996). Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2), 103-111.
- Zysno, P. V. (1981). Modeling membership functions. In R. B. Burghard, & P. V. Zysno (Eds.), *Empirical semantics* (pp. 350-375). Bochum: Studienverlag Brockmeyer.

APPENDIX A

DATA GATHERING CODES

```
Step-1- importing relevant
libraries import googlemaps
import psycopg2
Step-2- connecting to local database conn = psycopg2.connect("dbname = 'XXX'
user = 'XXX' host = 'localhost' password = ''") cur = conn.cursor()
cur.execute("""SELECT geocodeinput.veri from geocodeinput""") rows =
cur.fetchall()
gmaps = googlemaps.Client(key='XXX')
Step-3 parsing Google API call response and inserting results to local DB for
row in rows:
i = -1
while i < len(rows):
    i = i+1
    geocoderesult =
    gmaps.geocode(rows[i])
    if
    len(geocoderesult) > 0:
        lat =
        geocoderesult[0]['geometry']['location']['lat']
        lng =
        geocoderesult[0]['geometry']['location']['lng']
        geocode = "%s,%s" % (lat, lng)
        cur2 =
        conn.cursor()
        query1 = "INSERT INTO geocodeoutput(coord, input) VALUES (%s, %s)
RETURNING id"
        data1 = (geocode, rows[i])
        cur2.execute(query1, data1)
        conn.commit()
        bounds = geocoderesult[0]['geometry'].get('bounds', None)
        viewport = geocoderesult[0]['geometry'].get('viewport',
None)
        if bounds:
            latbn =
            bounds['northeast']['lat']
            lngbn =
            bounds['northeast']['lng']
            geocodebn = " try:
            cur3 = conn.cursor()
            query2 = "INSERT INTO geocodeoutput(coord,input) VALUES (%s, %s)
RETURNING id"
            data2 = (geocodebn,rows[i])
            cur3.execute(query2, data2)
            conn.commit()
        except:
            pass
            latbs =
            bounds['southwest']['lat']
            lngbs =
            bounds['southwest']['lng']
            geocodebs =
            "%s,%s" % (latbs, lngbs)
            try:
            cur4 =
            conn.cursor
            ()
```

```
query3 = "INSERT INTO geocodeoutput(coord,input) VALUES (%s, %s)
RETURNING id" data3 = (geocodebs,row[i]) cur4.execute(query3, data3)
conn.commit() except: pass if viewport:
latwn = viewport['northeast']['lat']
lngwn = viewport['northeast']['lng']
geocodewn = "%s,%s" % (latwn,
lngwn) try:
cur5 = conn.cursor()
query4 = "INSERT INTO geocodeoutput(coord,input) VALUES (%s, %s)
RETURNING id" data4 = (geocodewn, row[i]) cur5.execute(query4, data4)
conn.commit() except: pass latws = viewport['southwest']['lat'] lngws =
viewport['southwest']['lng'] geocodews = "%s,%s" % (latws, lngws) try:
cur6 = conn.cursor()
query5 = "INSERT INTO geocodeoutput(coord,input) VALUES (%s, %s)
RETURNING id"
data5 = (geocodews,
row[i])
cur6.execute(query5,
data5) conn.commit()
except: pass
```

APPENDIX B

VENUE DATA RETRIEVING CODES

Step-1- Used libraries are

```
imported import foursquare
import psycopg2
```

Step-2- local DB has been created to store coordinates. this block connects it to DB

```
conn = psycopg2.connect("dbname = 'XXX' user = 'XX' host = 'localhost'
password = 'XX') cur = conn.cursor() cur.execute("""SELECT coord from
geocodeOutput order by id ASC """) rows = cur.fetchall()
```

Step-3- this block provides authorization with unique client id and secret that is assigned to my developer account

```
CLIENT_ID = 'XXX'
```

```
CLIENT_SECRET = 'XXX'
```

```
client = foursquare.Foursquare(client_id = CLIENT_ID, client_secret =
CLIENT_SECRET)
```

Step-4- this block makes API calls to Foursquare API server for given coordinates, parse the response and writes to DB resultnum = 0 for row in rows:

```
i = -
```

```
1 i =
```

```
i + 1
```

```
if ((resultnum+1) time.sleep(10) vsearch =
client.venues.search(params = 'll': row[i], 'limit': 1) liste
= vsearch['venues'] for item in liste:
```

```
resultnum = resultnum
```

```
+ 1 venueid =
```

```
item.get('id') try:
```

```
query1 = "INSERT INTO venueSearchU(coordinate, venueid)
VALUES (%s, %s) RETURNING id"
```

```
data1 = (row[i],
```

```
venueid,)
```

```
cur2.execute(query1,
```

```
data1) conn.commit()
```

```
except: pass else:
```

```
vsearch = client.venues.search(params='ll': row[i],
'limit': 1) liste = vsearch['venues'] for item in liste:
```

```
resultnum = resultnum + 1
```

```
venueid = item.get('id') try:
```

```
query1 = "INSERT INTO venueSearchU(coordinate, venueid)
VALUES (%s, %s) RETURNING id"
```

```
data1 = (row[i],
```

```
venueid,) except:
```

```
pass
```

APPENDIX C

VENUE DETAIL DATA RETRIEVING CODES

```
Step-1- Used libraries are
imported import foursquare
import psycopg2
Step-2- connecting to local database and API server conn =
psycopg2.connect("dbname = 'XXX' user = 'XXX' host = 'localhost' password =
'") cur = conn.cursor() cur.execute("""SELECT venueexpuniq.venueid from
venueexpuniq""") rows = cur.fetchall() resultnum=0
CLIENTID = 'XXX'
CLIENTSECRET = 'XXX'
client = foursquare.Foursquare(clientid=CLIENTID,
clientsecret=CLIENTSECRET) Step-3- retrieving results from API calls and
parse the response, then inserts to the DB for row in rows:
i = -1 i = i + 1 if ((resultnum
+ 1) % 4500 == 0):
time.sleep(3600) vdet =
client.venues(row[i]) if vdet
is not None:
vitem = vdet.get('venue', None)
vname = vitem.get('name', None)
vcat = [c['shortName'] for c in vitem.get('categories')] vrating =
vitem.get('rating', None) vratingSignals = vitem.get('ratingSignals', None)
vverified = vitem.get('verified', None) vcheckinsCount =
vitem.get('stats')['checkinsCount'] vusersCount =
vitem.get('stats')['usersCount'] vvisitsCount =
vitem.get('stats')['visitsCount'] vtipCount = vitem.get('stats')['tipCount']
vlikes = vitem.get('likes', 'x': None).get('count') vphotos =
vitem.get('photos', 'x': None).get('count') vcreatedAt = time.strftime('%Y-
%m-%d', time.localtime(vitem.get('createdAt'))) vlisted = vitem.get('listed',
'x': None).get('count') vprice = vitem.get('price', 'x': None).get('tier') cur3
= conn.cursor() try:
query = "INSERT INTO venuedetails (venueID, vname, vcat, vrating,
vratingSignals, " " vcheckinsCount,vusersCount,vtipCount, vvisitsCount, vlikes,
vphotos, vcreatedAt, "
"vlisted, vprice,vverified,endpoint) VALUES(%s, %s,%s, %s,%s, %s,%s, %s,%s,
%s,%s, %s,%s,%s,%s,'search') returning id;"
data = (row[i], vname, vcat,vrating,vratingSignals,vcheckinsCount, vusersCount,
vtipCount, vvisitsCount, vlikes, vphotos, vcreatedAt, vlisted, vprice,vverified)
cur3.execute(query, data) conn.commit() except: pass resultnum = resultnum + 1 d
```