COMPARISON OF CLASSIFICATION ALGORITHMS IN PITCH TYPE PREDICTION PROBLEM

A Thesis Submitted to
the Graduate School of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Fatih TÜRKMEN

July 2020
İZMİR

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my thesis supervisor Assoc. Prof. Dr. Belgin Ergenç Bostanoğlu for guiding me into the correct direction and helping me throughout this thesis. It was a big pleasure to work with her invaluable support that moves me one step further.

Secondly , I am thankful to Assoc. Prof. Dr. Serap Şahin who has supported and encouraged me. I have always felt her encouraging and motivating guidance with me.

Furthermore, my special thanks go to my friends Ali Nehrani, Emre Karakış for their support and valuable advices.

Additionally, I must express my gratitude to my family who are always there for me.

The last but not least, i thank my sister Merve Türkmen for her invaluable help for my whole life.

ABSTRACT

COMPARISON OF CLASSIFICATION ALGORITHMS IN PITCH TYPE PREDICTION PROBLEM

The dramatic increase in the use of IoT devices has been leading to a huge amount of valuable data to be discovered. The knowledge extraction from such a huge amount of data requires an organized scientific set of processes. This requirement has pointed out the importance of the data mining applications. As a major data mining application, classification is a supervised learning technique that requires a feature set and target class through the training process. For the training process, the key point is determining the appropriate feature set for the classification algorithm. The improvements in cutting-edge technologies such as high resolution camera systems have made extracting the insights about next pitch available. Consequently, pitch type prediction has been standing out as an important research topic. In order to predict next pitch type, existing researches mostly focus on pitcher profile, batter profile and previous pitch data in feature set. There is no study analyzing the effect of the zone information in the prediction of the next pitch type. Therefore, this study has analyzed the contribution of zone information in pitch type prediction. Our approach is that, we aimed to reveal the contribution of zones with the high strike low bat rates for pitch type decision in pitcher and batter player match up. This aim directed us to analyze the pitch type prediction problem for both zone-based and non-zone-based approaches so that we can exhibit how much zone information contributes to the problem through different classification algorithms.

ÖZET

ATIŞ TİPİ TAHMİNLEME PROBLEMİNDE SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI

İnternet bağlantılı cihaz kullanımındaki çarpıcı artış, devasa miktarda keşfedilecek kıymetli verinin oluşmasına neden olmaktadır. Bu kadar büyük miktarlardaki veriden anlamlı bilgi çıkarmak organize edilmiş bir dizi bilimsel işlem gerektirmektedir. Bu gereklilik veri madenciliği uygulamalarının önemine işaret etmektedir. Temel bir veri madenciliği uygulaması olarak sınıflandırma, eğitim süresince özellik kümesi ve hedef sınıfı gerektiren denetimli bir öğrenme tekniğidir. Eğitim işlemi için önemli nokta sınıflandırma algoritması için uygun özellik dizisine karar vermektir. Yüksek çözünürlüklü kamera sistemleri gibi son gelişen teknolojiler bir sonraki atış hakkında çıkarım yapmaya imkan sağlamıştır. Bunun sonucunda atış tipi tahminlemesi önemli bir araştırma konusu olarak öne çıkmaktadır. Bir sonraki atış tipini tahminlemek için mevcut çalışmalar özellik kümesinde çoğunlukla atıcı profili, vurucu profili ve önceki atış bilgilerini kullanmıştır. Bölge bilgisinin bir sonraki atış tipini tahminlemedeki etkisini analiz eden bir çalışma olmadığından dolayı bu çalışma bölge bilgisinin atış tipi tahminlemesindeki katkısını analiz etmiştir. Yaklaşımımız atışı vurucu eşleşmelerinde, yüksek atış ve düşük vuruş değerli bölgelerin atış tipi kararına katkısını ortaya çıkarmak şeklindedir. Bu amaç bizi atış tipi tahminleme probleminde bölge bilgisinin katkısını ortaya koyabilmek amacıyla bölge temelli ve bölgesiz olarak sınıflandırma algoritmalarını incelemeye yöneltmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS

IoT . . . . . Internet of Things
mlb . . . . . Major League Baseball
FF . . . . . Fourseam Fastball
CU . . . . . Curveball
FC . . . . . Cutter
SI . . . . . Sinker
CH . . . . . Changeup
FT . . . . . Twoseam Fastball
SL . . . . . Slider
KC . . . . . Knuckle-Curve
FF . . . . . Fourseam Fastball
EP . . . . . Ephesus
FS . . . . . Splitter
KN . . . . . Knuckle Ball
UN . . . . . Unidentified
SC . . . . . Screwball
FA . . . . . Fastball

CHAPTER 1

INTRODUCTION

The recent technological improvements have lead to a dramatic increase in the use of data sources such as the world wide web, mobile networks and IoT devices. Consequently, the amount of data streaming through different sources has increased. As the amount of data increases, understanding the huge amount of data has been a challenge. It is difficult to analyze and understand such a big amount of data with usual techniques. Due that reason, we need a group of scientific and well-organized set of processes to understand the knowledge in the data. Data mining contains three major approaches to reveal undiscovered useful information from raw data that are classification, clustering and association (Prasad, 2011).

Data mining is the process of extracting undiscovered patterns or useful information from large volume of raw data (Jawad et al., 2015). As a major data mining application, classification has been used in a variety of problems related to scientific research and business. It includes different classification methods with different characteristics. These characteristics define the effectiveness and appropriateness of the method for the problem. Determining the feature set and classification algorithm is the key factor that affects the efficiency and performance metrics of the solution. Classification result metrics are sensitive to the feature set and must be analyzed for different feature set scenarios.

Classification algorithms are supervised learning techniques that require dataset with class labels in the training phase. They are trained by the training set and tested with test set which is a predetermined portion of the dataset. The training set must include class labels so that the classification algorithm can be trained. After the training phase, the classifier model is tested with the test set to determine the performance of the classifier model. To determine the performance of the classifier model, it is tested with the test set but training set performance metrics must also be taken into consideration to analyze bias and variance error. If the classifier model achieves well with the training set but not with the test set, we define this situation as variance error. When the classifier model is exposed to high bias, classifier model performance is not acceptable and we call this problem

underfitting. When the classifier model is exposed to high variance between training and test set, the classifier model can generalize enough beyond the training dataset and we call this problem overfitting. Classifier model performance must be in a balance point between bias and variance errors so that overall model performance can be at an acceptable level.

In baseball games, pitchers are the players who throw the ball to batter players with various types that are called pitch type. Pitch types differ in various metrics mainly launch angle, horizontal and vertical break, speed etc (Li et al., 2010). Pitch type tendency of a pitcher differs by many factors such as player handedness, strength. Batter performance changes against different pitch types. Strike zone is the imaginary field that is about half meter above from the ground. It is about 0.6 meter width and 0.7 meter height. This imaginary field is just in front of batter player. A pitcher usually wants to throw a strike that must go though the strike zone and can not be hit by a batter. A batter wants to hit the ball and earn score for the batting team. As the pitcher performance changes, batter performance also changes against the different pitch types. A batter may perform well against a specific pitch type while performing bad against another. Another situation is that pitcher and batter players may perform differently in the strike zone for the same pitch types. A pitcher may throw fastballs with a higher success rate into specific strike zones than another. A batter may hit a specific pitch type with a higher success rate in a specific strike zone.

The current studies focus on predicting the next pitch type with pitcher and batter profile informations and previous pitch metrics. In 2012, Ganeshappilai and Guttag made a binary classification by using a static feature set mainly including fundamental pitch metrics, game and player information (Ganeshapillai and Guttag, 2012). In 2014, M.Hamilton, et al. extended the study of Ganeshappilai and Guttag by implementing the adaptive feature set selection(Hamilton et al., 2014). This research focused on revealing how adaptive feature set selection contributes to pitch type prediction. In 2018, Sidle and Tran made a multi-class classification by using historical player tendencies, pitcher and batter informations(Tran, 2017). When we analyze these 3 major studies, we realize they focus on the correlation between pitch type and player performance metrics such as pitcher and batter profile informations, individual pitch metrics. However pitch zone information is neglected as a mutual information between pitcher and batter player. The term pitch-zone uncertainity also points out the importance of zone information on player

performances (Kim and JuUNG, 2018). As we review the discussions about the effect of zone information on player performances, we decided to study on extracting the effect of zone information in pitch type prediction problem.

In the proposed solution, we have implemented a zone-based and non-zone-based pitch type prediction approach. We calculated the strike and batting counts in each zone. When predicting the pitch type for a pitcher batter pair, we have calculated the pitch type that pitcher throws with high strike rates and batter hits low bat rates. To determine this pitch type, we counted strike and batting values. Furthermore, we have implemented an equation that boosts pitch type with high strike rate for pitcher and low strike rate for batter. In the normalization function part we have explained these processes. In order to observe the contribution of zone information for pitch type prediction problem, we trained classifier models for two feature set separately. The feature set that contains zone information is explained in implementation chapter as zone based pitch type prediction. We also implemented the version without zone information and explained it as a subsection in implementation. For both approaches, we explained and compared the results. To sum up, we observed whether zone is a significant feature and important decision maker for baseball players.

The aim of this thesis is to reveal the importance of zone information in pitch type prediction problem. In this manner both zone-based and non-zone-based approach has been implemented. We considered understanding the importance of zone information as a feature. Additionally, we also aimed to understand how often pitcher players take zone information into consideration or how significant the zone information as a decision maker. We also wanted to understand how classification algorithms perform for zone-based and non-zone-based approaches. Another conclusion that we wanted to reveal is whether the zone is a contributive feature or an additional cost in pitch type prediction problem.

To roughly introduce the design of this thesis, chapter 2 composes of the background information about the data mining and major data mining applications, classification algorithms and detailed information about pitch type prediction problems. We explained the definition of data mining and major data mining applications. To reinforce the meaning of the data mining applications, we used graphics and visual materials. As this thesis context is related to classification we highlighted the classification as one of

the data mining applications. We explained the differences and structures of the classification algorithms we used in the implementation chapter. We have also introduced what the pitch type prediction problem is in this chapter. Chapter 3 composes of the results of the literature search that we made to clarify how we can contribute as either theoretical manner or problem domain. Chapter 4 composes of the theoretical framework that we implement including calculations, matrices, formulations and diagrams were explained in this chapter. We explained the processes starting from the operations through dataset ending with training schema with metrics and calculations. We also explained the aim of this thesis by illustrating with diagram and charts in this chapter. Chapter 5 explains the results of implementation which define the consequence of our aim in this study. We explained how the implementation resulted and evaluate the results by referring to the implementation chapter. In chapter 6, we concluded the test results of pitch type prediction for both zone-based and non-zone-based approaches. For both approaches, we discussed and evaluated the contribution of zone information.

CHAPTER 2

BACKGROUND

In this chapter we explained the fundamental concepts of data mining and pitch type prediction problem. In the data mining section, we have explained major data mining applications those are association, clustering and classification. Since classification is the main focus of this thesis, we have explained the classification in detail as the last subsection. We have also explained the characteristics of the classification algorithms that we have implemented. The characteristics of each algorithm have been discussed with visual materials and plots. The explanations of classification algorithms have been referred by implementation chapter. We also explained the concepts and definitions related to the pitch type prediction problem that are required to understand the implementation of this thesis.

2.1. Data Mining

With the advancement of technological research studies on information technologies, the amount of unprocessed data has been increasing in various areas (Bharati and Ramageri, 2010). Due to that reason, the importance of data becomes more valuable. The main factor for such a condition is intensive usage of IoT devices and world wide web. This situation leads to an increase in the amount of raw data to be processed. These are the inevitable things for our daily life and they can not be processed via traditional methods. The issue of such big amount of data leads to the reveal of big data concepts meaning that it handles the evaluation and analysis of big data with new approaches. Data mining is introduced as one example of major approaches to analyze the data effectively and to be able to make it meaningful for different aspects.

Data mining is the organized set of processes to extract hidden information in raw dataset. The extracted information from raw datasets are mostly assessed as patterns which represents implicit useful information through data. Data mining is necessary to understand the future trends and strategies. Various data mining techniques are shaping

the kinds and patterns of data. The most popular data mining applications can be exemplified as classification, association and clustering methods as we have mentioned before.

2.2. Association

Association analysis is generally performed to extract the relationships between attributes of datasets. It is used to discover the interesting patterns and rules between the attributes in datasets. Association looks for the association rules and these rules aim at anticipating the existence of an item based on the existence of other items inside database records. One of the most typical scenarios is positioning products inside stores aims to increase to sale records.

2.2.1. Support

Support shows the percentage of a transaction among the dataset (Prajapati et al., 2017). It can be calculated for a single item or a group of items which we call itemset. Support is calculated by the frequency of an itemset with respect to whole dataset.

$$support(x) = \frac{|x|}{N} \tag{2.1}$$

Equation 2.1 states that, support is calculated by dividing the quantity of the itemset by the whole dataset. N is the total number of records in dataset.

Table 2.1. Invoice List

Invoice	Items
1	Tea,Coke,Bread
2	Bread, Cheese, Olive, Macaroni
3	Sugar, Cheese, Detergent,Bread, Macaroni
4	Bread, Cheese, Tea, Macaroni
5	Cheese, Macaroni, Beer,Coke

In the example dataset in Table 2.1, there is a dataset composes of invoices. In order to find Support(Tea) we divide the frequency of Tea by dataset count which is 2/5.

Similarly, to find the support of an itemset we do the same calculation. For example, support(Cheese, Macaroni) is 4/5 which means, Cheese and Macaroni exist together in the 4 records out of 5 total records. So support is 75%.

2.2.2. Confidence

Confidence shows the ratio of an itemset which exists in a specified portion of the dataset. Confidence(x,y -> z) means the ratio of the records in which z exists out of records in which x and y already exists.

$$confidence(x \rightarrow y) = \frac{Supp(xUy)}{Supp(x)} \tag{2.2}$$

Equation 2.2 (Ait-Mlouk et al., 2017) states that, division of the number of records in which x,y and z exists together by the number of records in which x and y exist together gives the confidence(x,y $\rightarrow$ z)

Table 2.1 states that; cheese, macaroni and bread exist 3 times in the records where Cheese and Macaroni exists. So confidence is 75%. The idea which might be revealed through such a scenario is that, store owner may send bread advertisements to customers who already bought cheese and macaroni.

2.3. Clustering

Clustering is an unsupervised learning technique which aims to find any group of data in a dataset by using similarity metrics.(Omran et al., 2007). These groups are constructed according to the specific similarity metrics. Additionally, the similarity between different clusters should be minimum. This similarity is called as between-group similarity. And also, each group has inner similarity value to indicate the similarities between group objects that should be maximum. It is known as within-group similarity.Figure 2.1 shows that two data groups have separate characteristics those have triangle and square shapes. Shapes are different from each other. This means that the datapoints form a high within group similarity and low between group similarity. Ideally, within-group similarity

should be high and between-group similarity should be low. In Figure 2.1, y is a small distance since data-points are inside same cluster. In contrast x is a large distance as the points reside in different cluster.

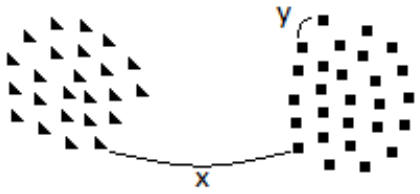

Figure 2.1. Clusters

2.3.1. Euclidean Distance

Euclidean distance measures the similarity by calculating the distance between data-points. For a 2-dimensional input space those are x and y, we can calculate the similarity metric with cardesian coordinates of datapoints. Let us consider that x_1, y_1, x_2 and y_2 are the x and y coordinates of samples as unit of distance.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2.3}$$

Equation 2.3 shows the calculation of distance between two points in 2-d input space. Any new datapoint must be appointed to the cluster which is closer to the new datapoint. For example, k-means clustering algorithm appoints a new node with euclidean distance similarity

2.3.2. Jaccard Distance

Jaccard distance measures the ratio of intersection of data groups out of the union of data groups. X and Y corresponds to data classes.

$$J(X, Y) = |X \cap Y|/|X \cup Y| \tag{2.4}$$

Equation 2.4 (Vorontsov et al., 2013) shows the formulation of jaccard distance similarity.

2.3.3. K-Means Clustering

K-Means is one of the major clustering algorithms which requires initial clustering counts. K represents the number of clusters. K-means algorithm randomly locates the initial cluster centroids (Goyal and Kumar, 2014). In each iteration, K-Means algorithm updates the centroids with new attending nodes. Figure 2.2 shows us that, distance

Figure 2.2. New Node

between new node and cluster centroids are d_1 and d_2. In this iteration new node is attended to the cluster whose centroid is closer to new node. After attending, cluster centroid is updated.

2.4. Classification

Classification is a supervised data mining application for classifying data by using classifier models. Classifier models are generated by training from the training set with classification algorithms. The aim of a training a classifier model is the generalization of the data. Common classification algorithms are decision tree, naive bayes classifier, artificial neural networks, support vector machines and boosting classifiers.

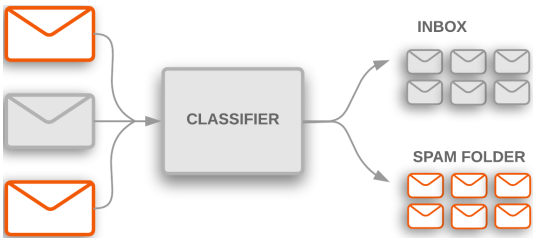

Figure 2.3. Classification
(Source: Google, 2020)

2.4.1. Decision Trees

A decision tree is a classifier which generalizes the data by partitioning recursively. Decision trees starts with a root node to construct the tree (Xiaohu et al., 2012). The other nodes have incoming edges. Any node except root which has outgoing edges is an internal node. The remaining nodes are called leaves. Leaves are the terminal points and decisions are performed by terminal nodes. Internal node splits the input space into subspaces. Classification starts with the root node and ends with the leave nodes by classifying recursively. We say recursively because the decision tree splits the input space into subspaces until leaves. In Figure 2.4, play tennis is classified by leaf nodes and other attributes are located in tree according to splitting criteria. Entropy and gini index are the fundamental metrics to determine splitting attributes for decision trees. These attributes are used to determine best splitting attibute for creating the decision tree.

$$gini\,index = 1 - \sum_{i=1}^{n} p^2(x_i) \tag{2.5}$$

$$entropy = \sum_{i=1}^{n} -p(x_i)\log_2 p(x_i) \tag{2.6}$$

The x values are the splitting attributes to be determined in Equation 2.5 nd 2.6. p probability represents the ratio of each class after separation. During the construction of tree, in each level gini index and entropy is calculated to determine splitting attribute.

Day	Outlook	Temperature	Humidity	Wind	Play tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Mild	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Figure 2.4. Classification
(Source: Ao et al., 2008)

ID3 is the first decision tree algorithm which was developed by Ross Quinlann in 1986. It was developed to classify categorical features. In order to construct the tree, ID3 algorithm uses the maximum information gain. ID3 algorithm prunes the tree to increase generalization rate.

C4.5 is the extended version of ID3 algorithm. In addition to ID3, C4.5 algorithm has the ability to work on continuous data. It can dynamically partition the data into discrete set of intervals.

2.4.2. Naive Bayesian Classifier

Naive Bayesian classifiers are statistical classifiers based on Bayes Theorem and calculates the conditional probability of all target classes by feature vector. Naive Bayesian classifier selects the highest conditional probability to classify the feature vector. Naive Bayesian classifiers require features to be independent from each other. Naive Bayesian classifiers have been used in practical applications such as text mining, diagnostics support systems.

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{2.7}$$

As we see in Equation 2.7 (Kaviani and Dhotre, 2017), Y composes of the vector set and X is the target class, To calculate the conditional probability of X given Y, bayesian equation is calculated. Naive bayesin classifier calculates X values with all possible Y conditions and classifies to the highest probability class.

2.4.3. Support Vector Machines

Support vector machines are supervised learning algorithms based on statistical learning theory (Evgeniou and Pontil, 2001). The support vector classifier finds a separating hyperplane to generalize input space. To decide the hyperplane, support vector machines use the support vectors. Support vector machines have been using in various problems such as speech analysis and face recognition.

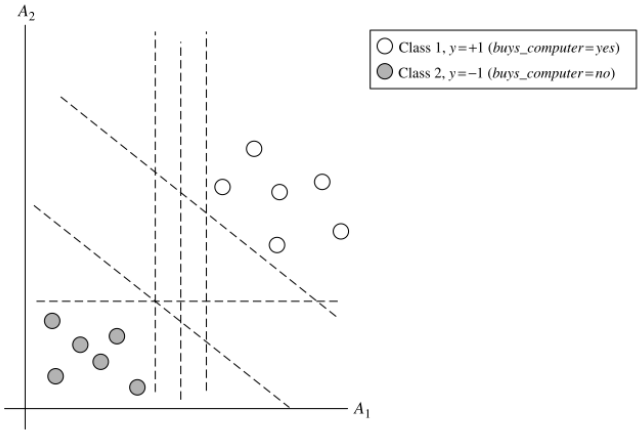

Figure 2.5. Support Vectors on 2D Data
(Source: Han et al., 2011)

Figure 2.5 shows that, there are multiple support vectors in a dataset. To determine the hyperplane which generalize the data, most efficient support vectors must be selected by the classification algorithm.

As larger margin separates the classes better, the hyperplane generalizes the data better and achieves better accuracy. In Figure 2.6 a large margin finds a separating hyperplane which is wider than others. This means that this support vector machine generalizes

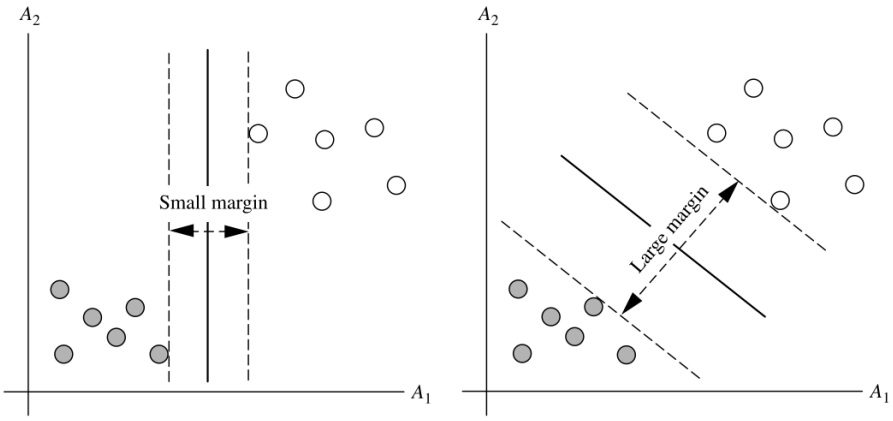

Figure 2.6. Separating Hyperplanes with Different Sizes
(Source: Han et al., 2011)

the dataset with better accuracy. Support vector classifiers focus on the seperating hyperplane to determine the classes.

2.4.4. Artificial Neural Network

The concepts of artificial neural networks are explained by human neurons. Roughly explaining, a neural network is a set of connected perceptron nodes. Each connection has a weight which is computed after iterative backward and forward propagation steps(Han et al., 2011). These weights are adjusted in repetitive training iterations to classify classes with a sufficient accuracy level.

A typical neural network consists of an input layer, one or multiple multiple hidden layers and output layer. They are fully connected neural networks because every node in a layer is connected to all nodes in the next layer. In figure 2.7 we see an artificial neural network 3 layers that are input layer, hidden layers and output layer. Each node in each layer corresponds to a human neurons. Input layers forward the inputs taken from output nodes of the previous output layer. We see w symbols. They are the weights through each connection. During the training phase any input coming to a neural network is multiplied by these weights and forwarded to the next node which is called forward propagation. If the accuracy level is not acceptable, it is iteratively trained again that is called backward propagation.

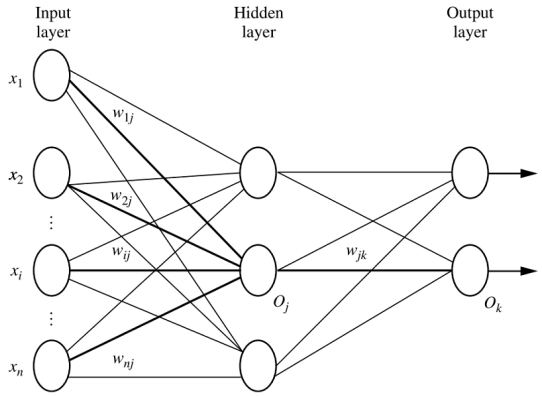

Figure 2.7. Artificial Neural Networks
(Source: Han et al., 2011)

2.4.5. Ensemble Methods

Emsemble methods are specialized classification methods that aims increasing accuracy by using combination of multiple models instead of using single model (Buhlmann, 2012).

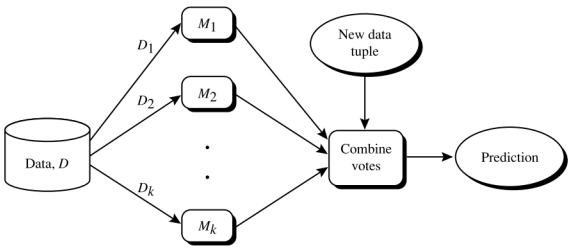

Figure 2.8. Ensemble Methods
(Source: Han et al., 2011)

Bagging and boosting classifiers are the major ensemble methods. Bagging classifiers perform sampling with replacement. Each classifier is trained by samples and final classification considers the voting of there multiple classifiers. Similarly boosting classifiers contains multiple classifiers each of which is trained through the dataset. Each classifier may achieve different accuracy results. In Figure 2.8 we see that, ensemble classifier is composed of multiple sampling weights and new data is classified with ensemble classifier.

2.5. Performance Evaluation Metrics

As data mining is organized set of processes, to evaluate the performance of a classification, we need performance evaluation metrics. Selecting the appropriate performance metrics is one of the key factors to evaluate classifier performance (Liu et al., 2014). In order to evaluate classifier performance, we need to know which performance metrics should we use. In this chapter we explain the common performance evaluation metrics used to evaluate classification results.

2.5.1. Positive and Negative Classes

In a binary classification, the class that we need to investigate is the positive class. For example if we try to detect heart disease patients with classification, the hearth disease patients can be stated with positive classes and healthy patients can be stated as negative classes. Class labels depend on how to interpret the classes.

2.5.2. TP, FP, TN and FN

As performance evaluation metrics TP (True Positive), FP (False Positive), TN (True Negative) and FN (False Negative) are the fundamental metrics that we need to know. In typical metrics the letter on the right shows the actual class and the letter on the left shows the classification. For instance if we state TP, this states a sample in positive class which is classified as correctly positive.

2.5.3. Accuracy and Error Rate

Accuracy shows the ratio of the data which is classified by the classification algorithm correctly.

$$accuracy = \frac{TP + TN}{P + N} \tag{2.8}$$

It is calculated by dividing the sum of correctly classified positive and negative values by positive and negative classes.

Error rate shows the ratio of the data which is classified by the classification algorithm incorrectly.

$$errorrate = \frac{FP + FN}{P + N} \tag{2.9}$$

It is calculated by dividing the sum of incorrectly classified positive and negative values by sum of positive and negative data.

2.5.4. Sensitivity and Specificity

Sensitivity shows the ratio of the true positive classifications out of true classifications.

$$sensitivity = \frac{TP}{TP + TN} \tag{2.10}$$

It is calculated by dividing the amount of correctly classified positive values by the sum of correctly classified positive values and correctly classified negative values.

Specificity shows the ratio of the data which is classified as negative out of negative data. It is calculated by dividing the amount of correctly classified negative values by sum of correctly classified negative values and incorrectly classified positive values(Van Stralen et al., 2009).

$$specificity = \frac{TN}{TN + FP} \tag{2.11}$$

Specifity is calculated by divding true negative classifications by sum of true negative classifications and false positive classfications.

2.6. Pitch Type Prediction

The pitch type prediction problem is predicting the next pitch type to be thrown by the pitcher. For the baseball game, estimating the type of pitch is very strategic to the opponent team. Because if the opponent team can estimate the pitch type, they can be in a better condition to hit the ball. The common attributes are pitch type metrics that are collected until the specific time (Hoang et al., 2014). In this manner, the type of pitch affects the difficulty of the pitch.

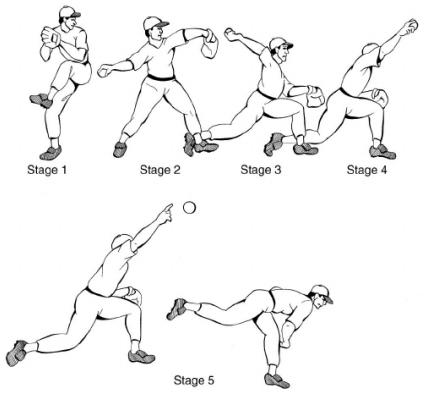

Figure 2.9. Illustration of a Typical Pitch
(Source: Williams Jr and Kelley, 2000)

In the Figure 2.9 pitcher player throws the ball with certain body movements. This depends on the various metrics such as speed, angle, and location of the ball. To sum up all the metrics determine the type of the pitch. Pitch type prediction is the problem of predicting the next pitch type to be thrown by pitcher player to hitter player. As wee see in Figure 2.10 hitter player can take the position against pitch type.

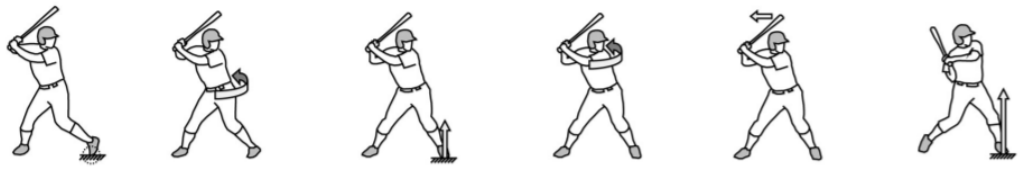

Figure 2.10. Illustration of a Typical Hit
(Source: Kidokoro et al., 2020)

2.6.1. Pitch Type

Pitch type is a combination of multiple metrics and factors that mostly speed, angle, direction. These metrics distinguish the type of a pitch as illustrated in Figure 2.11 for fourseam fastball and two seam fastball.

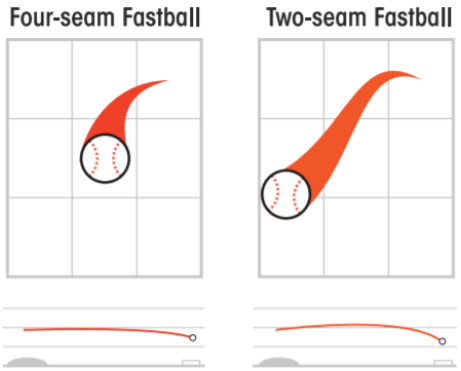

Figure 2.11. Fourseam Fastball and Twoseam Fastball
(Source: Dhakar, 2020)

2.6.2. Strike Zone

Strike zone is the zone where the ball thrown by the pitcher player must arrive. To throw a valid pitch, the pitcher player must throw the ball inside the strike zone. Strike zone is a virtual field that is about half meters above from the ground, 0.6-meter in width and 0.7-meter in height next to hitter player. As we see in Figure 2.12, the hitter player is waiting for a pitcher player to throw the ball. Pitcher player aims to locate the ball inside the strike zone so that pitch is valid.

In the baseball game, strike zone is a critical factor to win the game. The points are determined whether strikes can reach to the strike zone or not. So, a pitcher player tries to throw the ball inside of strike zone. Similarly batter player defends the strike zone and tries to prevent ball from reaching to the strike zone. In order to throw the ball, there are few pitch types for pitcher players. These pitch types defines how the ball flies against strike zone. Launch angle, trajectory are the one of the metrics that are effected

by pitch types. So, pitch type is a critical factor and each pitcher may set some strategies for determining the pitch type.

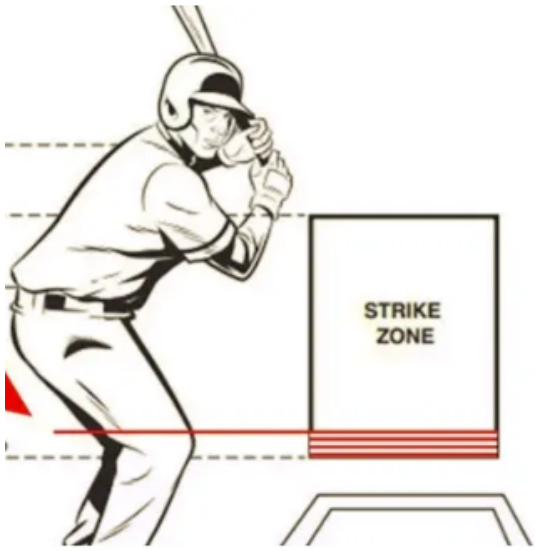

Figure 2.12. Strikezone
(Source: Bible, 2020)

CHAPTER 3

RELATED WORK

In this chapter, we introduced the previous studies that we analyzed during the literature search process. We have explained how the previous studies handled the pitch type prediction problem. In a theoretical manner, we analyzed the classification algorithm they have used. We have also analyzed how their approach against the problem. We have analyzed 3 papers that aim to predict the next pitch type with classification algorithms.

3.1. Predicting The Next Pitch

If a batter can estimate the next pitch type to be thrown by the opponent pitcher player, he can be in a better position to hit the ball (Ganeshapillai and Guttag, 2012). The motivation of the research is that, estimating the type of pitch provides a better condition for the hitter player. To hit such a fast-moving ball, a hitter player may focus on the correct location on the strike zone. In this study Ganeshappilai and Guttag used a linear support vector machine classifier with a soft margin. The most useful features were pitcher/batter prior, pitcher/count prior, the previous pitch, and the score of the game. This study is a binary classification to predict whether the next pitch type is a fastball or not.

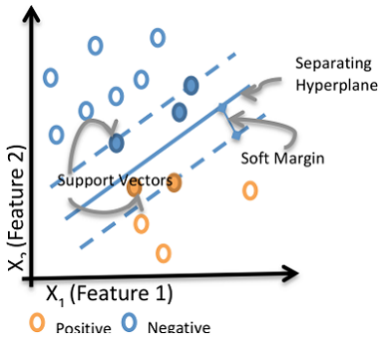

Figure 3.1. Support Vector Machine and Soft Margin (Source: Ganeshapillai and Guttag, 2012)

As we see in Figure 3.1, the support vector machine classifier classifies the pitch types with a separating hyperplane.

Game performance	Balls	Strikes	Outs	Score differential	Bases loaded
Game state	Inning	Handedness	Number of Pitches thrown		
Prior probability	Home team	Batting team	Count	Batter	Defense formation
Support of the priors	Home team	Batting team	Count	Batter	Defense formation
Batter Profile	Slugging Percentage	Runs	For each pitch class		
			Runs	Slugging percentage	
Previous Pitch	Pitch Type	Pitch Result	Velocity	Vertical Zone	Horizontal Zone
Gradient over last 3 pitches	Velocity	Vertical Zone	Horizontal Zone		

Figure 3.2. Feature Vector
(Source: Ganeshapillai and Guttag, 2012)

As we see in Figure 3.2, the feature vector consists of features related to the game, player, previous pitches. They considered 359 pitchers who threw at least 300 pitches in 2008 and 2009. The average accuracy of their model is 70%.

3.2. Applying Machine Learning Techniques to Baseball Pitch Prediction

The key difference between this research and previous research(Ganeshapillai and Guttag, 2012) is the feature selection method. Rather than using a static set of features, a different optimal set of features is used for each pitcher/count pair. They aimed to increase the accuracy with an adaptive feature selection algorithm. They considered data from 236 pitchers and their mode achieved 77.45%. They used k nearest neighborhood and support vector machine classifier to predict the next pitch. Another significant advantage of this model is that, a feature selection algorithm provides the feature independence. Naive Bayesian classifiers require the feature independence which is mostly not satisfied. However, as they select the features ultimately with dynamic feature selection, the state features are highly independent. They stated their model can be improved with batting averages and multi-class classification. This study is a binary classification which predicts whether the next pitch type will be fastball or not.

3.3. Using Multi-Class Classification Methods to Predict Baseball Pitch Types

This study aims the make a multi-class classification (Tran, 2017). They used 3 classification algorithms to predict the next pitch type that are linear discriminant analysis, multi-class support vector machines and classification trees. To reduce the model variance between different models, they used voting techniques with ten of each model. For the support vector machine classifier, they used a one-vs-one approach rather than one-vs-all. Because one-vs-all leaves a gap where the classification algorithm may fail. They used 5-fold cross-validation to find the optimal classification parameters.

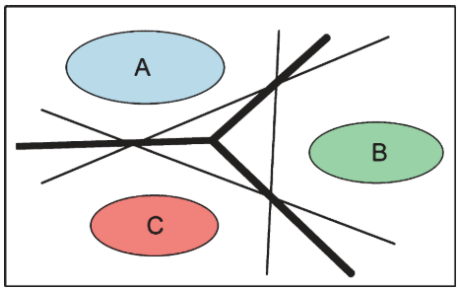

Figure 3.3. One-vs-one and one-vs-all Approaches
(Source: Tran, 2017)

Figure 3.3 explains the difference between one-vs-one and one-vs-all approach. Bold lines show the class separation by using support vectors with a one-vs-one approach. Since this approach considers the support vectors between class as pairwise, it does not cause to gap. Thin lines show the class separation by using a one-vs-all approach. Since this approach considers the support vectors between one class to remaining classes, one-vs-all approach may cause a generalization gap through dataset. This gap may lead to wrong classification results.

CHAPTER 4

ZONE BASED PITCH TYPE PREDICTION

This chapter mainly focuses on explaining the steps that we followed to evaluate to reveal the contribution of zone information in pitch type prediction. In order to understand the effect of zone information, we have analyzed the classification algorithms for predicting the next pitch type for baseball pitchers against batters. As classification algorithms we selected 5 common classification methods those are decision trees, naive bayesian classifier, support vector machines, neural networks(perceptron) and boosting classifier. We have compared characteristics of these classification algorithms and explained the whole procedure in 4 sections those are problem evaluation, strike and batting stats, probability distribution matrices, training schema.

In the problem evaluation part, we have the whole process that we followed by starting from the dataset to the training part. We explained why we calculated strike and batting stats for pitcher and batter players, structure of probability distribution matrices. We also explained the pitcher and batter probability distribution matrices and the normalization equation that we applied in this section. Algorithm 4 and Algorithm 5 explain the transformation operations that we applied to data.

Strike and batting stats section explains how we calculated the zone-based and non-zone-based strike and batting stats for pitcher and batter players. We have also explained the dataset attributes that we picked up and their meaning.

The probability distribution matrix section explains the calculation of a single probability distribution matrix for a pitcher batter match up. We have explained the calculation steps of pitcher and batter probability matrices as separate subsections and the transformations that we applied to each. We have also explained dot product matrix multiplication operation for pitcher and batter probability matrices to calculate the probability distribution matrix.

In the final step, we explained the training schema that we set up. We have mentioned about the feature set and target here to figure out the training schema. Training schema explains the the structure of feature set and how we trained the classifier model.

4.1. Problem Evaluation

As we explained in the background chapter, characteristics of pitch types differ in different parameters like speed, horizontal and vertical breaks, etc. In this study, our approach is detecting the pitch type that pitcher throws with high success and batter bats with low success for a pitcher batter match up.

In order to analyze the player pitch and bat informations, we used master league baseball data of 2015-2018 years(Schale, 2019). This dataset composes of pitch by pitch data. Every pitch is recorded with pitcher id, batter id and other metrics such as speed, angle, type etc. We used pitcher id, batter id, type, pitch type, zone fields. Pitcher id and batter id are unique player ids corresponding to players that throws the ball and hits the ball. Type represents whether the pitch is a strike or ball or is in play. Strike is represented as S, the ball as B, in play as X. A pitcher normally aims to make strike(S) and batter aims to hit the ball which means(X). We have explained the definitions of strike(S), ball(B) or ball-in-play(X)

To predict the next pitch type, we analyzed the player performances and calculated the most successful pitch types that pitcher throws and most unsuccessful pitch types that batter hits.

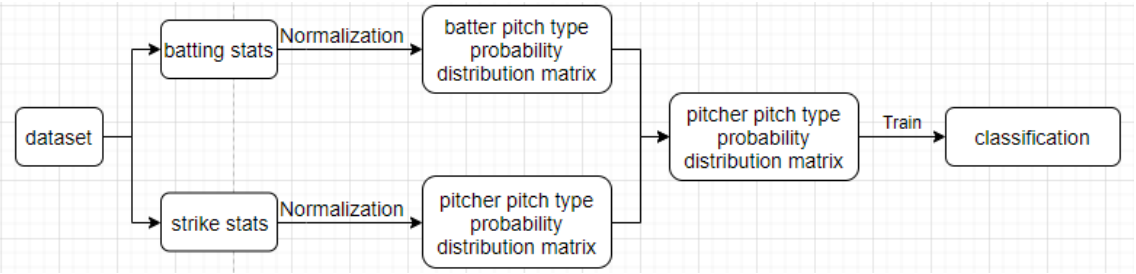

Figure 4.1. Dataset Transformation

In Figure 4.1 we observe data transformation steps. In the first level, we calculated the strike and batting stats from historical data of each pitcher and batter players. In the strike and batting stats section we explained the calculations in detail. Then we grouped the pitches and hits into zones and pitch types. In other words, we have had strike and batting stats of each pitch type in 14 zones of the strike zone. By using these zone-based strike and batting information, we have extracted the probability distributions of pitcher

and batter players in match up. Then, we calculated the dot product of pitcher and batter probability matrices to calculate the overall pitch type probability matrix. This matrix represents the joint probability of pitch types in each zone.

4.2. Feature Vector Calculation

The feature vector composes of an 18x14 matrix for zone-based approach and 1x18 matrix for non-zone-based approach. The difference is that, we have taken the zone information into consideration for zone based approach. Therefore, we have calculated 18 joint probability for 18 pitch types for a zone.

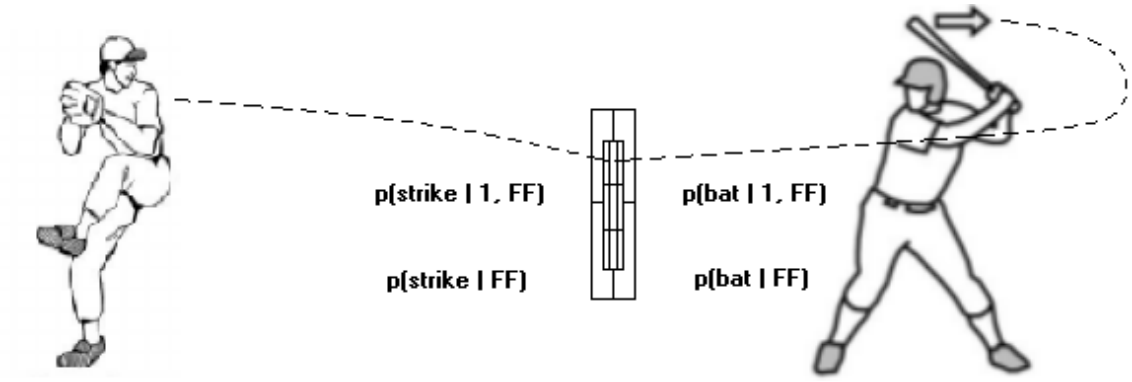

Figure 4.2. Strike and Batting Probabilities for Both Approaches (Source:Williams Jr and Kelley, 2000; Kidokoro et al., 2020)

As we analyze the 4.2, we observe that, the pitcher player may throw the ball against 14 different zones in strike zone. There are 14 different zones for a pitcher player to throw the ball inside strike zone. The figure 4.2, the pitcher player throws the pitch type FF which means fourseam fastball against zone 1.

In this study we calculated strike and batting stats for pitcher and batter players. This process is the first level of data transformation in figure 4.1. This profile contains the successful strike and batting averages for each strike zone. In this study we calculated strike and batting averages in 14 zones for each pitch type.

$$avg\_strike_{zone} = \frac{strike_{zone}}{total_{zone}} \tag{4.1}$$

$$avg\_strike = \frac{strike}{total} \tag{4.2}$$

$$avg\_bat_{zone} = \frac{bat_{zone}}{total_{zone}} \tag{4.3}$$

$$avg\_bat = \frac{bat}{total} \tag{4.4}$$

Equation 4.1 shows zone-based strike average for a pitch type. We calculate strike average for 14 zones explicitly however for non-zone-based approach, we only calculated 1 strike average since we do not take the zone information into consideration. Similarly we have calculated the batting averages and there is no difference between strike and batting average calculation

$$weighted\_expected\_strike\_average = \frac{e^{(avg\_strike_{zone})} * total_{zone}}{total} \tag{4.5}$$

$$weighted\_expected\_batting\_average = \frac{e^{1-(avg\_bat_{zone})} * total_{zone}}{total} \tag{4.6}$$

Equation 4.5 and Equation 4.6 shows weighted expected strike and batting count. We applied exponential function because we wanted to encourage high strike and low batting averages more. The important point is that we subtracted the avg_{bat} from 1 because we want to find less successful bat zones. For non-zone-based approach, there is no zone distinction and we calculated single weighted strike and batting averages for a pitch type. To sum the Equation 4.5 and Equation 4.6 again, they actually tell how we have evaluated the pitch type prediction problem. In order to find successful strike average zone and unsuccessful batting zones we applied exponential. In order to find unsuccessful batting zones we subtracted batting averages from 1 before applying exponential function.

$$P(t|s,b) = \sum_{i=1}^{14} P(s|p)P(b|b) \tag{4.7}$$

We formulated the probability of next pitch by the equation 4.7. $P(s|p)$ corresponds to the strike probability for pitcher with pitcher profile p. We calculated the $s|p$ value with $weighted\_expected\_strike\_average$. Similarly $P(b|b)$ corresponds to the batting probability for batter with batter profile $weighted\_expected\_batting\_average$. Pitcher profile composes of 14 $weighted\_expected\_strike\_average$ values for a single pitch type and contains 18 pitch type. So, $P(s|p)$ composes of 252 values and similarly $P(b|b)$ contains 252 values as well. We explained $P(s|p)$ as strike probability distribution matrix in section 4.6.2. Strike probability distribution matrix represents the characteristics of a pitcher player in strike performance manner through the 14 zones. We have called it as probability distribution matrix in section 4.11

$P(b|b)$ corresponds to the batter proability distribution matrix which composes of 14 $weighted\_expected\_batting\_average$ values for a single pitch type. Similar to $P(s|p)$, $P(b|b)$ composes of 18 pitch types and 252 probability values.

We have explained the batting/pitching stats, Normalization and pitcher/batter probability distribution matrix parts in Figure 4.1.

The last but not least, we have calculated the probability distribution matrix by multiplying the strike and batting probability distribution matrices. Consequently we have a 18x14 matrix which contains joint probabilities of pitch types to be happen together.

	1	2	3
FF	.3108	.2387	.1102
CU	.0	.0	.0
FC	.0209	.0569	.0418

×

	1	2	3
FF	.0108	.0387	.0102
CU	.0	.0	.0
FC	.0209	.0569	.0418

Figure 4.3. Joint Probability of $s|p$ and $b|b$ matrices

As we notice in Figure 4.3, the probability of throwing strike against zone 1 is 0.3 which

is slightly higher than other values. We know that, 31% of the fourseam fastball has been successfully thrown as strike against zone 1. The batter player has been able to hit the 1% of the fourseam fastballs. This means that, if the pitcher throws a fourseam fastball agains zone 1, it will more likely be a strike. In the feature vector applied exponential function and inverted batting average to warn the classification algorithm against this values. We send the message to classification algorithm and say that .3108 and .0108 are high strike and low batting values. So this will probably a strike. In order to help the classification algorithm realize it, we have increased the strike and batting probabilities with exponential function. The important key is we subtracted the batting probability from 1 before applying exponential function

4.3. Preprocessing

For this study, we used the mlb pitch dataset of 2015-2018 years(Schale, 2019). Dataset composes of 5 different csv files those are atbats.csv, ejections.csv, games.csv, pitches.csv and player_names.csv. Since we need to the pitch and bat information with player names, we merged atbats.csv, pitches.csv and player_names.csv files. We merged pitches.csv and atbats.csv with ab_id column which corresponds to a pitcher and batter matchup. Additionally, in order to merge player_names.csv with pitches.csv and atbats.csv, we used pitcher_id and batter_id fields. Dataset contains imbalanced pitch type distributions.

$$imbalanced\,rate = \frac{|pitchtype_{max}| - |pitchtype_{min}|}{|pitchtype_{max}|} < 0.5 \tag{4.8}$$

$$|pitchtype_{min}| > 50 \tag{4.9}$$

To eliminate imbalanced players, we grouped pitch types under players and used the Equation 4.8 and Equation 4.9. In Equation 4.8 we calculated the ratio of difference difference between maximum pitch type and minimum pitch type counts. Thus, we fil-

tered the most imbalanced players as 0.5. Additionally we selected the players who have at least 50 pitches.

Dataset selection performed by sorting the players with imbalanced rate metrics. Lower imbalanced rate value gives more balanced class distributions. However, imbalanced rate metric is not sufficient. Because we observed players with zero pitch types and they pretend to have low imbalanced metrics but they are useless data. So we filtered them with minimum 50 pitch criteria.

4.4. Non-Zone-Based Pitch Type Prediction

Non-zone based pitch type prediction approach differs in feature vector from zone-based approach. Zone based approach separates the pitches into 14 zones and then separates into pitch types in each zone.

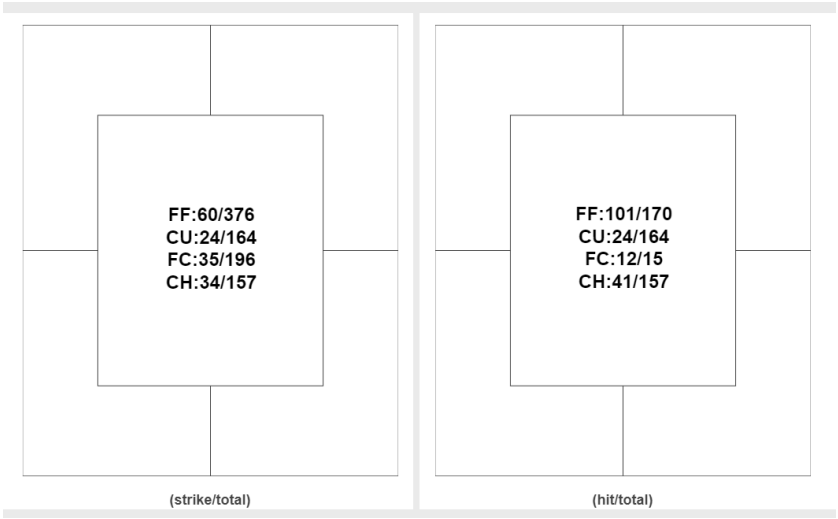

Figure 4.4. Strike and Batting Stats

As we see in Figure 4.4, average strike and batting values are calculated as a single zone for pitcher and batter player for non-zone-based approach. This provides the weighted success rates for each pitch type. In contrast, non-zone-based feature set does not contain the zone information. It calculates the overall success rates through pitch types.

$$p(t|p,b) = p(s|t).p(b|t) \tag{4.10}$$

As we analyze the Equation 4.10, probability of a pitch type is represented by multiplication of strike probability and bat probability for that pitch type. For non-zone-based approach we do not provide zone information, instead we provide 18 strike probability. We let the classification algorithm to interpret the 18 pitch type probability to generalize the problem

4.5. Strike And Batting Stats

Strike and batting stats are calculated by iterating through the pitches thrown by pitcher and batters batted by batter. Difference is for pitchers, we calculated type value S and batters type value X. S corresponds to a successful strike and X corresponds to a successful bat.

Algorithm 1 Calculating Zone Based Strike Stats

```
1:  procedure CALCULATE ZONE BASED STRIKE STATS(p_1,p_2,...,p_n :P set of pitches)
2:      n = |P|
3:      m = array of 18x14
4:      for i =1 to n do
5:          type = P[i][type]
6:          pitchtype = P[i][pitchtype]
7:          zone = P[i][zone]
8:          if type is S then
9:              incr m[zone][pitchtype][strike]
10:         end if
11:         incr m[zone][pitchtype][total]
12:     end for
13:     return strike stats array m
14: end procedure
```

In Algorithm 1 and Algorithm 2, zone-based and non-zone-based strike stats calculations are explained. Similarly, Algorithm 3 and 4 explain the batting stats. Algorithm 1 iterates through the n pitches which means the all pitches of a player. The successful strike counts for 18 pitch type for each 14 zone are calculated. The line

Algorithm 2 Calculating Non-Zone-Based Strike Stats

```
1: procedure CALCULATE NON-ZONE-BASED STRIKE STATS(p_1,p_2,...,p_n :P set of pitches)
2:     n = |P|
3:     m = array of length 18
4:     for i =1 to n do
5:         type = P[i][type]
6:         pitchtype = P[i][pitchtype]
7:         if type is S then
8:             incr m[pitchtype][strike]
9:         end if
10:        incr m[pitchtype][total]
11:    end for
12:    return strike stats array m
13: end procedure
```

m[zone][pitchtype] states pitches are firstly grouped as zone-based and then as pitch type based which is 14x18=252 sub groups. The only difference between zone-based and non-zone-based strike calculations is that, we did not group pitch types with zone for non-zone-based approach.

Algorithm 3 Calculating Zone-Based Batting Stats

```
1: procedure CALCULATE ZONE-BASED BATTING STATS(p_1,p_2,...,p_n :P set of pitches)
2:     n = |P|
3:     m = array of 18x14
4:     for i =1 to n do
5:         type = P[i][type]
6:         pitchtype = P[i][pitchtype]
7:         zone = P[i][zone]
8:         if type is X then
9:             incr m[zone][pitchtype][bat]
10:        end if
11:        incr m[zone][pitchtype][total]
12:    end for
13:    return batting stats array m
14: end procedure
```

Algorithm 3 and 4 similarly calculate the batting stats for each type however, we have calculated the batting counts with X type value. X corresponds to the pitch which could be hit by the hitter successfully. We used these stats to calculate the pitcher and batter probability distribution matrices and then pitch type probability distribution

matrix. The probability distribution matrix section explains how we transformed pitcher and batter probability distribution matrices.

Algorithm 4 Calculating Non-Zone-Based Batting Stats

procedure CALCULATE NON-ZONE-BASED BATTING STATS(p_1,p_2,...,p_n :P set of pitches)
    $n = |P|$
    m = array of length 18
    for i =1 to n **do**
        $type = P[i][type]$
        $pitchtype = P[i][pitchtype]$
        $zone = P[i][zone]$
        if type is X **then**
            incr m[pitchtype][bat]
        end if
        incr m[pitchtype][total]
    end for
    return batting stats array m
end procedure

Figure 4.5 illustrates the structure of the 14x18 strike and batting stats arrays. The left strike zone shows the strike and total values and the right one shows the batting and total values. This match up actually states the overall strike and batting stats of pitcher and batter player.

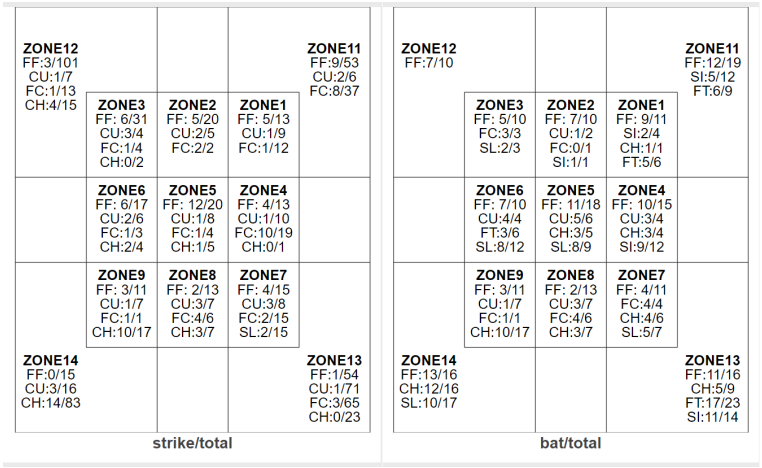

Figure 4.5. Zone Based Strike and Batting Stats

4.6. Probability Distribution Matrix

Pitch type probability matrix is the dot product of pitcher player pitch type probability matrix and batter player probability matrices.

$$p(t|p,b) = \sum_{i=1}^{14} p(s|t,z_i).p(b|t,z_i) \tag{4.11}$$

$$p(t|p,b) = \sum_{i=1}^{14} p(s|t).p(b|t) \tag{4.12}$$

In Equation 4.11 and 4.12 we have calculated the strike probability of a pitcher and batting probability of a batter. $p(s|t,z_i)$ shows the conditional probability of strike for pitch type t in zone z_i. $p(b|t,z_i)$ shows the corresponding conditional batting probability for the same pitch type in the same zone. They represent pitcher and batter probability distributions that we explained in section 4.3.1 and 4.3.2. Equation 4.12 is the non-zone-based approach and does not contain the zone information. We multiplied these two values to calculate the joint probability.

The key point is that, our aim is calculating the joint probability of two events to happen together. For example the probability of pitching fourseam fastball of a player is calculated by sum of strike probability in 14 zones. However for non-zone-base approach, feature set does not contain zone information which seems considering the strike zone as whole. For each 14 zones, we calculate the strike probability of pitcher and batting probability of hitter. In order to strongly claim it would be a strike, we expect a higher strike probability success for pitcher and lower batting probability for batter for that pitch type in each zone. However, for the feature set, we want lower strike probability be an inverted value so that, lower batting probability be a boosting effect for classification algorithm. We expect to catch the critical match up by preparing these metrics in feature set. High strike average and low batting average values are expected to result in a successful strike. Furthermore, by implementing exponential function, we notify the classification algorithm for high strike low batting averages.

4.6.1. Batting Probability Distribution Matrix

Higher $p(s|t, z_i)$ value and lower $p(b|t, z_i)$ value means for the specific pitch type, it is more likely to be a strike which hitter player can not hit. This means that, we must increase the probability of batting in low success rates. In order to do that, we have implemented an exponential function in batter probability distribution matrix calculation that we explained in Algorithm 4.1 $-$ $success$ calculates the prior probability of batting in the current zone for current pitch. By subtracting from 1, we inverted and gave higher probability for not hitting case.

$$zonebasedexpectedbats = totalpitch * \sum_{i=1}^{14} bat_{type} e^{1-success} \tag{4.13}$$

$$expectedbats = totalpitch * bat_{type} e^{1-success} \tag{4.14}$$

Algorithm 5 Zone Based Batting Probability Distribution Matrix

procedure ZONE BASED BATTING PROBABILITY DISTRIBUTION MATRIX(P: zonebasedexpectedbats)
    n = $|P|$
    m = array of 18x14
    for i =1 to 14 **do**
        for j =1 to 18 **do**
            $batcount$: number of bat
            $totalpitch$: number of pitch
            $p(successrate)$ = $batcount/total$
            $p(type)$ = $totalpitch * \frac{e^{1-success}}{expectedbats[j]}$
        end for
    end for
    return zone-based batter probability distribution matrix
end procedure

In the Equation 4.13, we calculated the expected bat count for each pitch type in 14 zone. Then, we divided the expected bat amount by total expected bat for 14 zone. This means that, how much probability does batter player successfully hits a specific pitch type

in a specific zone. Actually, we found the most successful zones for batter. Algorithm 5 is zone based approach. For non zone based approach, we just removed the inner for loop.

Table 4.1. Zone-Based Batter Strike Probability Distribution Matrix

	1	2	3	4	5	6	7	8	9	11	12	13	14
FF	.0047	.0968	.0394	.0636	.1593	.0863	.0816	.114	.1054	.0155	.0484	.0943	.09
CU	.0	.0	.0	.08	.0971	.0294	.172	.1233	.0294	.1720	.0	.2669	.0294
FC	.0209	.0569	.0418	.138	.1223	.0569	.0876	.1753	.1223	.0	.0209	.0876	.069
SI	.0295	.0295	.0243	.0634	.1042	.059	.2142	.0667	.0667	.0243	.0243	.1516	.1417
CH	.0148	.0	.0	.1256	.0866	.0403	.1382	.0489	.1351	.0762	.0	.319	.0148
FT	.0497	.0383	.0183	.0604	.1738	.0383	.1812	.0649	.0604	.0183	.0091	.1499	.1369
IN	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714
SL	.0284	.0209	.0	.0537	.0104	.069	.1469	.3022	.0537	.0209	.0104	.195	.0876
KC	.0	.0	.0	.2976	.1488	.0	.0	.4046	.0	.0	.0	.0	.1488
EP	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714
FS	.0	.0	.0	.0	.4387	.0	.3616	.133	.0	.0	.0	.0665	.0
FO	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714
PO	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714
KN	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714
UN	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714
SC	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714
FA	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714
AB	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714	.0714
	4/8	17/24	8/11	22/35	28/54	19/29	33/62	36/59	21/37	9/14	9/12	54/78	26/43

Table 4.1 shows the probability of hitting to a specific pitch type in each zone. For non zone based approach, we have an array composes of overall probabilities of 18 pitch type. The highlighed cells in Table 4.1 shows the unsuccessful batting probabilities. For example 0.1593 of fourseam fastball in zone 5 means, this batter fails to hit the coming pitches with this probability. If these values are relatively higher, we expect the opponent batter player to fail hitting the ball.

4.6.2. Strike Probability Distribution Matrix

Pitcher probability distribution matrix is similar to batter except exponential part. For pitcher, we did not do any inversion because we expect to higher success values for strike case. However, for batting, we inverted batting averages to encourage bad batting averages.

$$zonebasedexpectedstrike = totalpitch * \sum_{i=1}^{14} strike_{type} e^{success} \tag{4.15}$$

$$expectedstrike = totalpitch * strike_{type} e^{success} \tag{4.16}$$

Algorithm 6 Zone Based Strike Probability Distribution Matrix

procedure CALCULATE(zonebasedexpectedstrikes)
    $n = |P|$
    m = array of 18x14
    for i =1 to 14 **do**
        for j =1 to 18 **do**
            $strikecount$: number of strike
            $totalpitch$: number of pitch
            $p(success) = strikecount/totalpitch$
            $p(type) = totalpitch * \frac{e^{p(success)}}{expectedstrikes[j]}$
        end for
    end for
    return zone based pitcher probability distribution matrix
end procedure

For zone based and non zone based pitcher probability distribution matrices, calculation is similar with batting probability distribution matrix. The only difference is strike success calculation in each zone. As we want high success rate for pitchers, we did not apply inversion.

Algorithm 7 Non-Zone-Based Strike Probability Distribution Matrix

procedure CALCULATE(expectedstrikes)
    m = array of length 18
    for j =1 to 18 **do**
        $strikecount$: number of strike
        $totalpitch$: number of pitch
        $p(success) = strikecount/totalpitch$
        $p(type) = totalpitch * \frac{e^{p(uccess)}}{expectedstrikes[j]}$
    end for
    return zone-based pitcher strike probability distribution matrix
end procedure

Non-zone-based approach calculates 18 pitchtype probabilities for expected strike values. In the Algorithm 7, we calculated the exponential of pitch type success probability to boost higher values. Multiplication of pitch type probability with total pitch gives us the expected strike count. By dividing it to expected strike probability, we calculated the pitch type probability for the current zone. For non-zone-based approach we directly calculated the pitch type probability. Table 4.2 shows the 18x14 array which is the zone-based strike probability distribution matrix. Non-zone-based matrix is going to be a 1x14 matrix which is actually an array of length 14. The highlighted cells indicates that, the

Table 4.2. Zone-Based Pitcher Strike Probability Distribution Matrix

	1	2	3	4	5	6	7	8	9	11	12	13	14
FF	.025	.017	.013	.031	.044	.031	.03	.023	.019	.039	.052	.146	.072
CU	.001	.0032	.0068	.007	.018	.018	.044	.0319	.022	.0021	.0262	.307	.0687
FC	.0149	.0163	.0057	.0577	.0265	.0163	.0609	.0344	.0180	.0428	.0145	.3216	.0478
SI	.012	.0059	.0075	.0249	.0218	.0246	.0313	.0787	.1063	.0075	.0556	.1403	.1998
CH	.0	.0	.0	.0121	.0121	.0273	.0084	.0349	.0664	.0	.0403	.0892	.4304
FT	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
IN	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
SL	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
KC	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
EP	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
FS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
FO	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
PO	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.3678	.0	.0
KN	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
UN	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
SC	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
FA	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
AB	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	18/61	16/45	9/38	41/100	42/96	31/84	43/133	43/106	31/111	18/126	5/203	78/821	37/402

pitcher throws successful pitches for selected pitch types in selected zones. For example, 0.307 means his player pitches strikes with curveballs into zone 13 with 0.307 probability. If these values are relatively higher, we expect the pitcher player to strike the ball.

4.6.3. Probability Distribution Matrix

Probability distribution matrix is the dot product of pitcher and batter probability matrices. The line $matrix_{pitcher}[i][j]xmatrix_{batter}[i][j]$ multiplies each probability of

each pitch type in each zone. For non-zone-based approach, 18 pitch type probabilities were directly multiplied. In Algorithm 8, we calculated the probability of two events to

Algorithm 8 Calculating Zone-based Probability Distribution Matrix

procedure CALCULATE_MATRIX($matrix_{pitcher}, matrix_{stats}$:pitcher and batter matrices)
    n = $|P|$
    m = array of 18x14
    for i =1 to 14 **do**
        for j =1 to 18 **do**
            $matrix_{probability}$ = $matrix_{pitcher}[i][j] x matrix_{batter}[i][j]$
        end for
    end for
    return probability distribution matrix
end procedure

be happen together. The probability of successful strike and successful batting events are calculated. So, the result matrix shows the pitch type to be thrown with high strike ratio and low bat ratio. Because we boosted the successful strike and unsuccessful bat events by using exponential function. This matrix is the feature set to be trained with target pitch types in training section.

4.7. Training

In this section we have explained how we set up the training schema. As we illustrated in Figure 4.6, training set composes of probability distribution matrix which is calculated by multiplying pitcher and batter probability matrices. We have multiplied the pitcher and batter matrices to calculate the probability of two independent events. Our purpose here is that, what is the probability of throwing a pitch type for pitcher and hitting for batter. In the Equation 4.11 and 4.12 we have calculated the conditional probability of $P(t|p,b)$.

$$P(t|s,b) = \sum_{i=1}^{14} P(s|p)P(b|b) \tag{4.17}$$

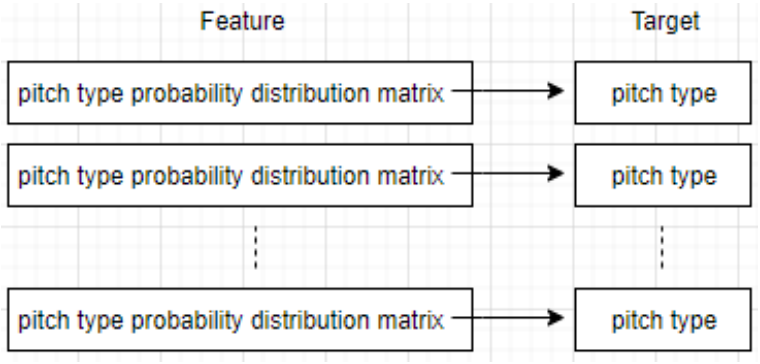

Figure 4.6. Feature Set and Target

The feature set contains probability distribution matrices which composes of pitcher and batter player match ups. According to the historical data, these match ups are forms to a matrix by multiplying each of them. They are pitch and batter player matrices that contains strike and batting aveare values in 14 zones.

CHAPTER 5

EXPERIMENT AND RESULTS

In this chapter, we showed the results of comparison schema that used to compare classification algorithms. In the implementation chapter, we have explained the workflow that we followed by starting from dataset transformation to training plan. The classification results for zone-based and non-zone-based approaches with different number of classes have been evaluated. We have also analyzed correlation matrix to observe the correlation between pitch type with Figure 5.1.

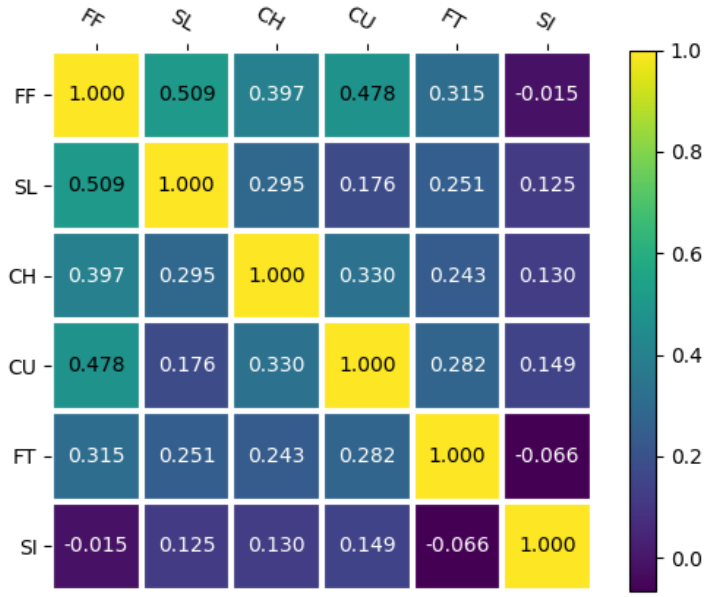

Figure 5.1. Pitch Type Correlation Matrix

5.1. Binary Classification

For the binary classification, we selected 3 pitchers who threw 820 fourseam fastballs and 300 sliders total. Class distributions are imbalanced and we observed the the results under this imbalanced condition as in Figure 5.2.a. According to Figure 5.2.b naive

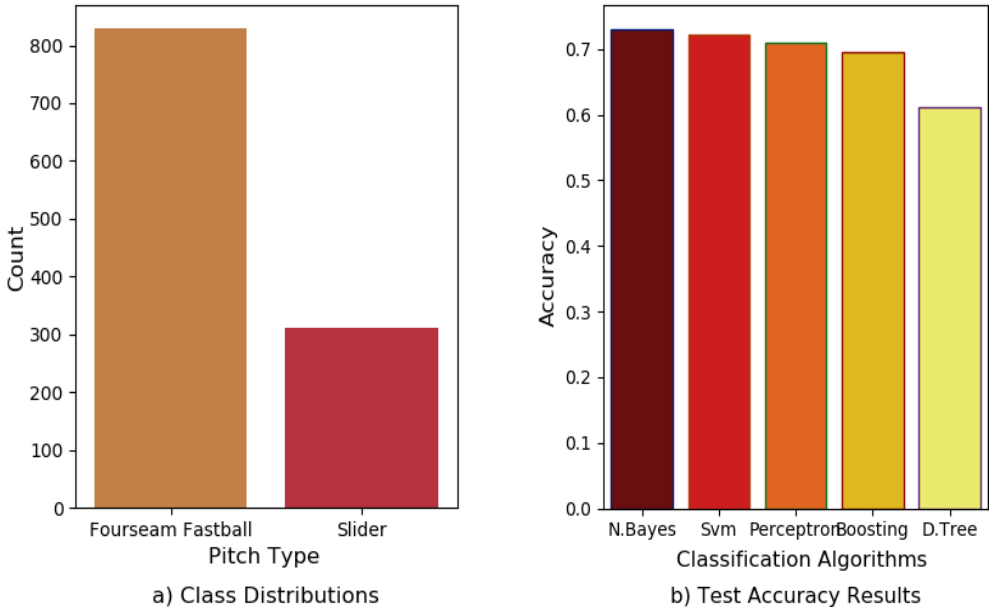

Figure 5.2. Zone Based Pitch Type Classification with Imbalanced Data

bayes classifier achieved best about 72%. But this impression is misleading because, since data is highly imbalanced, classifier model biased to classify sliders as fourseam fastball which is the majority of data. The confusion matrix in Figure 5.3 seems supporting this idea as 75% of the sliders classified as fourseam fastball.

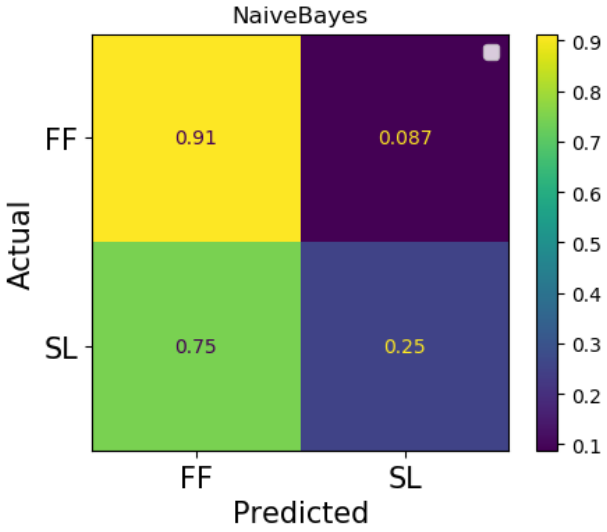

Figure 5.3. Naive Bayesian Classification Confusion Matrix

In order to fix the imbalanced classes we downsampled the fourseam fastballs to sliders as illustrated in Figure 5.4.a and observed the improvements in the results Figure 5.4.b. Naive bayes, boosting and decision tree classifiers improved accuracy about 10% however, support vector machine and percepton remained same. As svm classifier cares about the seperating hyperplane downsampling did not increase svm performance. Similarly, as the perceptrons are the smallest unit of neural networks, we should have designed a neural network to increase perceptron performance. Furthermore, neural networks require relatively more data.

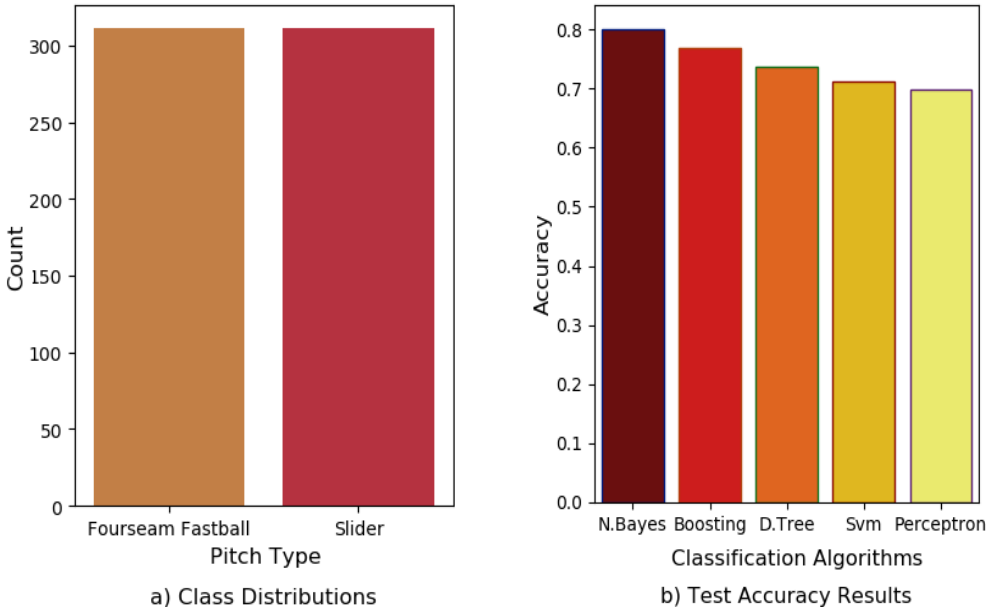

Figure 5.4. Zone Based Classification with Balanced Data

For the non zone based approach, we can say that, naive bayesian classifier was effected most because the zone based approach is based on bayesian theory. We give the probability distributions of 18 pitch types for each zone and let the classifier interpret each. Another important point is that, svm performance improved because, the classes are linearly seperable as we removed the zones from feature set.

Naive Bayesian classifier decreased the performance as illustrated in Figure 5.5 the feature set is based on bayesian theory. Since we removed the zones from feature set, we had probability distributions of pitch types in which dependency rate is more than the zone-based approach.

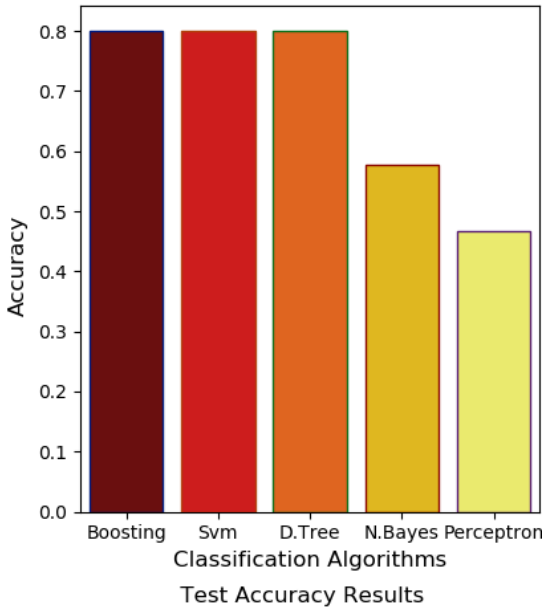

Figure 5.5. Non Zone Based Classification Test Accuracy Results

5.2. Multi Class Classification

In this section we introduced the experiments with 3,4 and 5 classes to observe the classification results for both approaches.For 3-class classification, we used 3929 pitches that composes of 2716 fourseam fastballs, 786 sliders and 426 curveballs.

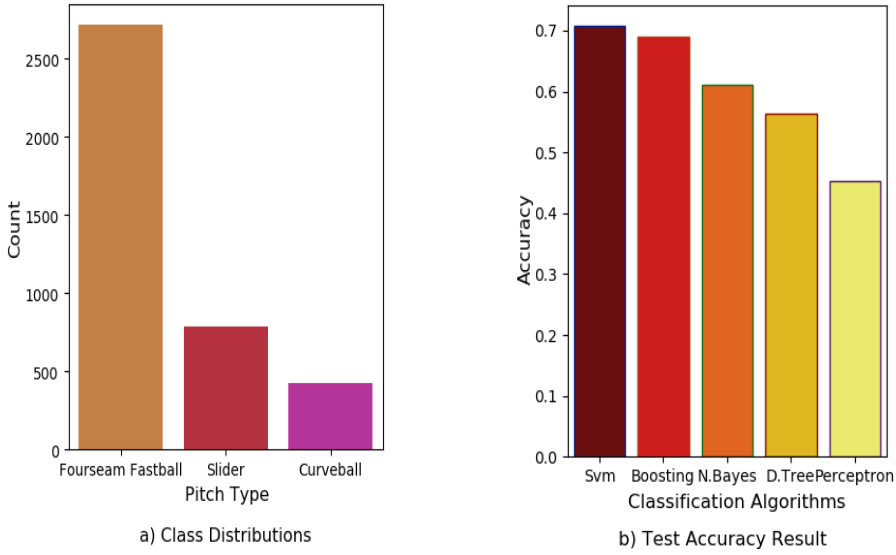

Figure 5.6. Zone Based Multi Class Classification with Imbalanced Data

Figure 5.6.b shows that perceptron achieved really bad results. The reason is that, since the perceptrons are the smallest unit of neural networks, they should be trained with backward and forward propagation. The accuracy for binary imbalanced set was about 69% and we observed 45% for perceptron. As the feature vector complexity and data size increases, perceptron performance decreases, since it requires forward and backward propagation. Svm achieved 70% but we see that it classified 62% of the curveballs and 93% of the sliders as fourseam fastballs.Since the fourseam fastball data size is dominant, svm seemed successfull. But this is not the case and we observed it from confusion matrix in Figure 5.7.To analyse the confusion matrix of results, we can say that, support vector classifier achieved well with fourseam fastball which is but it confused with 93% of sliders by evaluating as fourseam fastball. Naive Bayesian classifier and boosting classifier achieved close to svm classifier which is around 65%.

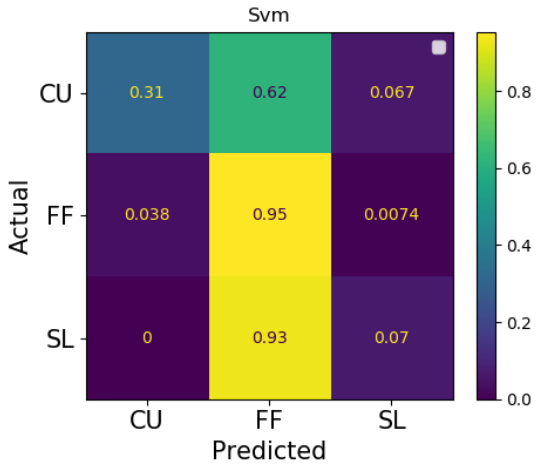

Figure 5.7. Svm Classifier Confusion Matrix

Similarly, they also performed well with fourseam fastballs but performed bad with sliders. Decision tree classifier and perceptron performed bad and we can say that current data is not convenient for them. We need to analyze the data again for decision trees. Perceptrons which are the smallest unit of neural networks requires multi layer neural network design. They should be trained with forward and backward propagation. Consequently we observed worst results from perceptron. To sum up we can say that, the data illustrated in Figure 5.6.a is not convenient for naive bayes, decision tree and

perceptron classifier. Due that reason, we applied undersampling to fourseam fastball and observed again. We observed better results with balanced which is in Figure 5.8.a

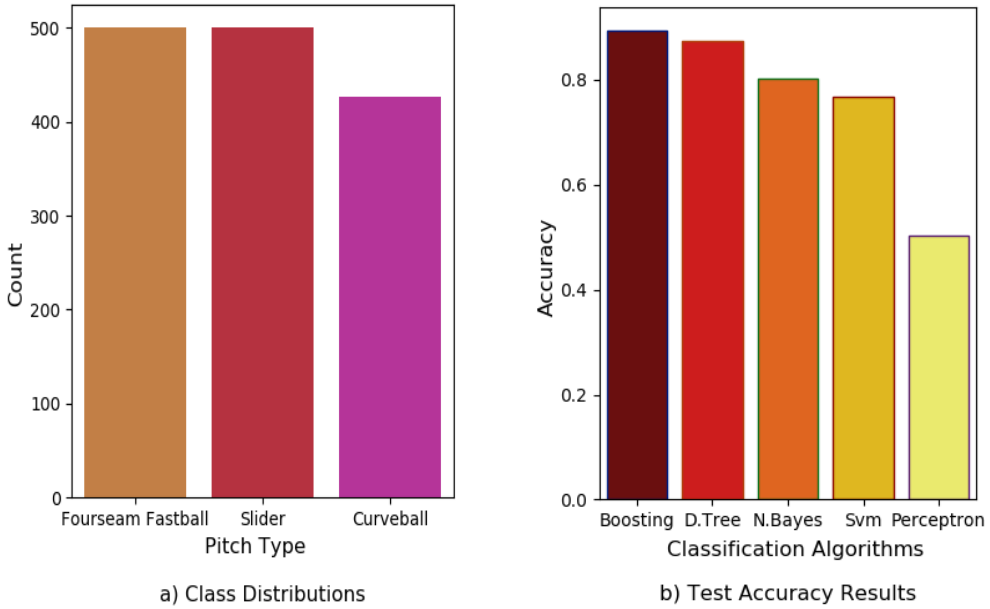

Figure 5.8. Zone Based Multi Class Classification with Balanced Data

data except perceptron. Boosting performed 89%, decision tree 87%, naive bayes 79%, svm 76% and perceptron achieved 50%. Classifers increased their performance except perceptron as illustrated in Figure 5.8.b.Figure 5.9 shows us that, boosting classifer classified 19% of the sliders as fourseam fastball. Decision tree classifier classified 20% of the sliders as fourseam fastball. It means that decision tree and boosting classifier could not disinguish 20% of the sliders from fourseam fastballs.

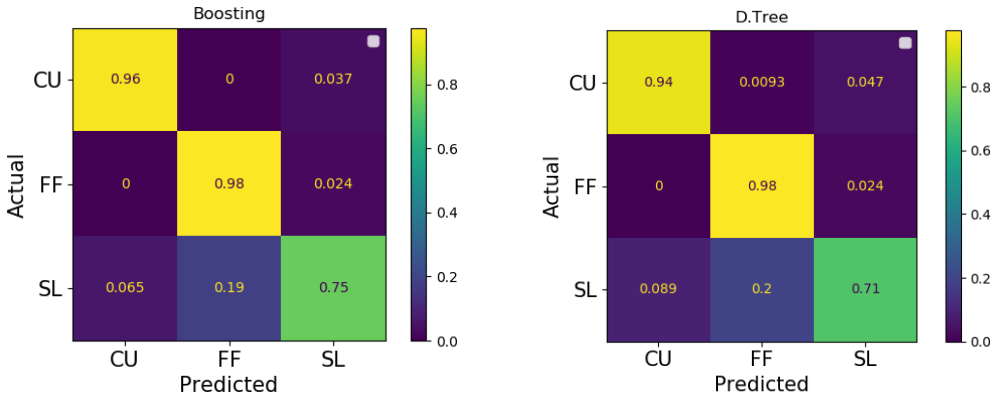

Figure 5.9. Confusion Matrix for Boosting and Decision Tree

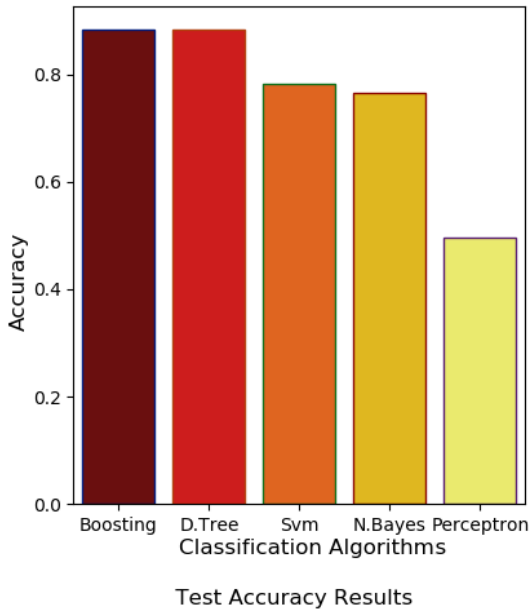

Figure 5.10. Non Zone Based Classification Test Accuracy Results

Table 5.1. Test Accuracy Results for Zone/Non Zone Based Approaches

	Binary (%)	3-Class (%)	4-Class (%)	5-Class (%)
Boosting	76/79	89/87	84/85	77/77
Svm	71/79	76/77	76/78	71/71
D.Tree	73/79	87/87	83/84	75/77
N.Bayes	80/57	79/76	79/75	71/69
Perceptron	70/46	50/49	36/39	29/32

The non-zone-based classification results with balanced data is illustrated in Figure 5.10. The comparison of zone based and non zone based classification test accuracy results are illustrated in Table 5.1. In order to evaluate the classification results from zone based to non zone based, we can say that, decision tree tend to achieve better results. It seems decision tree was not able to find sufficient splitting attributes in performance evaluation matrix which corresponds to the feature vector. In contrast, naive bayes classifer performed better with zone based classification. Since our probability matrix calculation is based on bayesian theory, we can state that naive classifier achieves better with zone based approach. The decrease from 80% to 57% seems supporting our idea. Perceptron as the smallest unit of neural networks performed insufficient in this study. When we ob-

tain the test accuracy metrics in Table 5.1, perceptron achived worse results in non zone based approach. It seems decreasing the complexity and dimensionality from zone-based to non zone based did not fit for perceptron.

Boosting classifier kept its performance and performed statefull results. We can claim that, as boosting classifiers uses multiple classification algorithms or samples during classification, boosting classifier generally performed more smooth performance lines relative to other classifiers.

Support vector classifier is another classifier which did not perform a dramatic decrease from zone based to non zone based approach. Svm classifiers care about the separating hyper plane, switching to non zone based approach did not effect. The plot in Figure 5.11 contains the zone based and non zone based results which are the blue and red bars respectively. Boosting, support vector machine classifier and decision tree classifier achieved better results with non zone based classification. Binary classification classified fourseam fastballs and sliders. It seems zone information is not vital to distinguish fourseam fastballs from sliders for boosting, svm and decision tree classifier. However, naive bayes and perceptron decreased accuracy rate. We consider that, since our feature vector based on bayesian theory and we removed the zone information from conditional probabilities, naive bayes classifier performance effected.

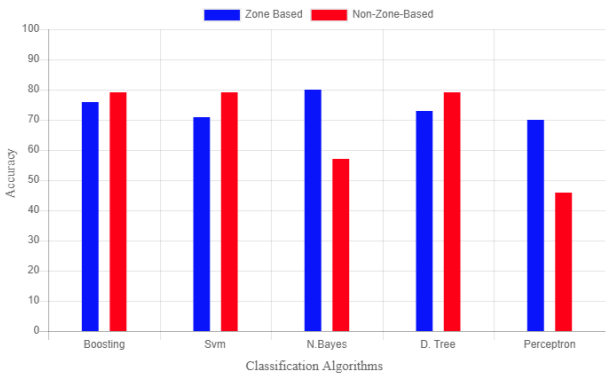

Figure 5.11. Comparison of 2 Approaches for Binary Classification in Test Set

Perceptron performance also decreased because perceptron is not applicable for the problem. We should design a neural network instead of a single perceptron.

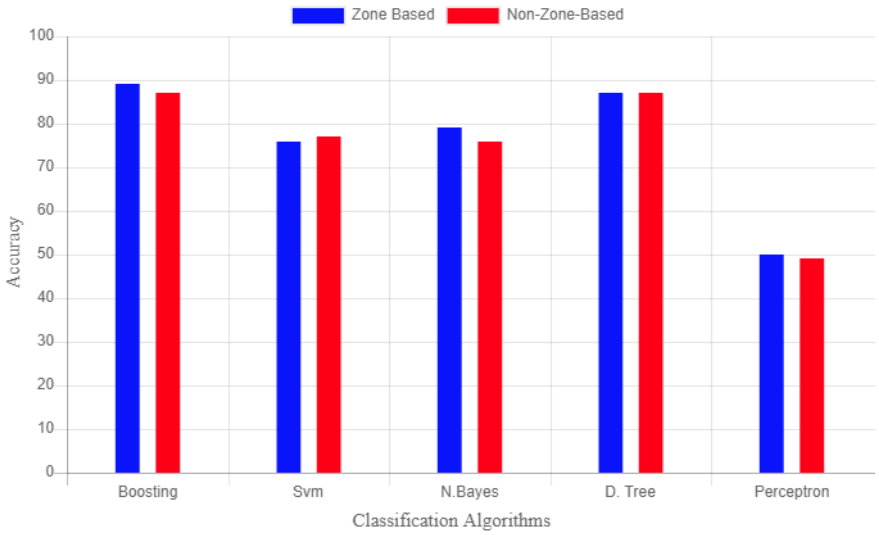

Figure 5.12. Comparison of 2 Approaches for 3 Class Classification in Test Set

For the 3 class classification we classified for fourseam fastball, slider and curveball. As we move from binary to 3 class, boosting and decision tree classifier performance increased. This gives us the idea that, the data size and class count are the significant factors in test accuracy rates. Support vector machine classifier and naive bayesian classifier performance shows a similar accuracy rates. Another point is that, decision tree performance shows almost same between zone-based and non-zone-based approach. This gives us the idea that, decion tree did not get significant knowledge from zone information.

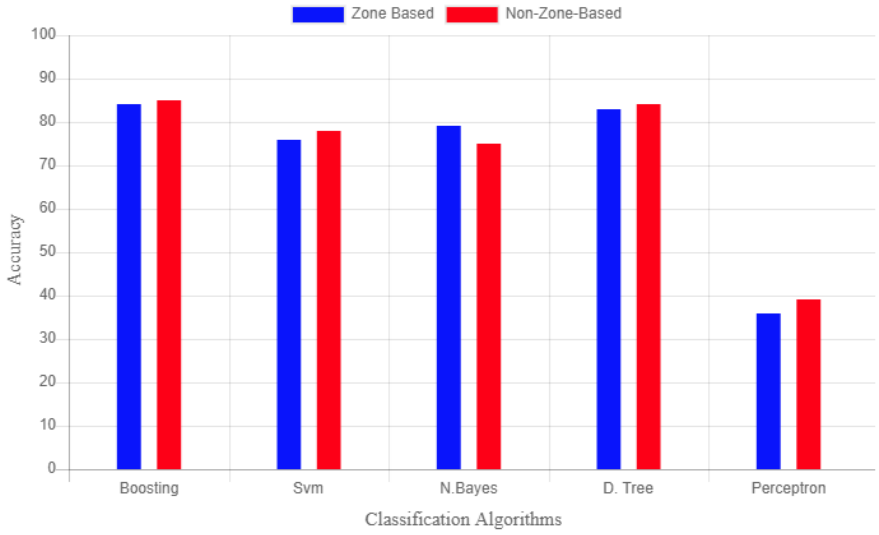

Figure 5.13. Comparison of 2 Approaches for 4 Class Classification in Test Set

For analyzing the results from 5.13 to Figure 5.14, we observe an overall decrease in accuracy rates of all classifiers.The current training schema seems insufficient to generalize the problem as the class counts increase.

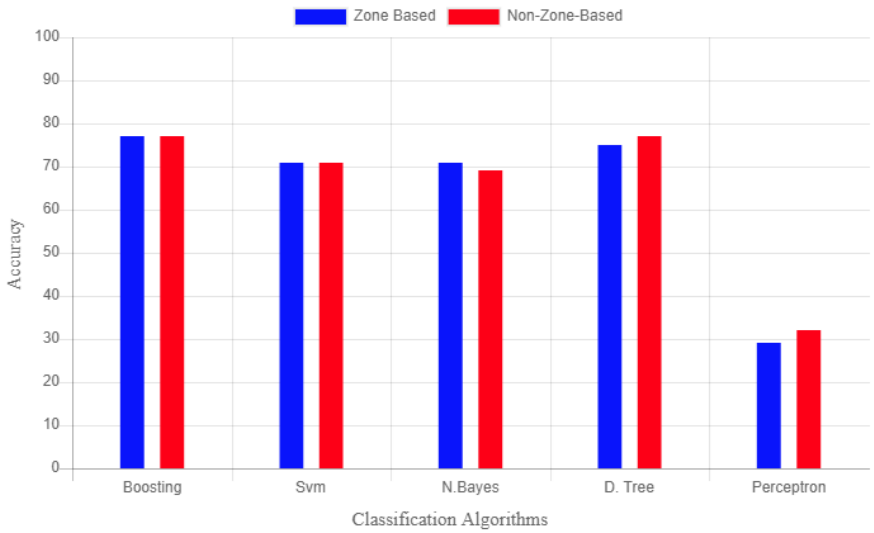

Figure 5.14. Comparison of 2 Approaches for 5 Class Classification in Test Set

The reason could be related to number of data or feature set. So 5-class classification must be trained again with different data size and feature set.

CHAPTER 6

CONCLUSION

The context of this thesis is revealing the contribution of zone information in pitch type prediction problem by implementing major 5 classification algorithms for zone-based and non-zone-based approaches. We implemented binary and multi-class classification with support vector machine, decision tree, naive bayes, boosting and perceptron classifiers for both approach and observed the results. Furthermore, for both approach we repeated the training and test processes with different data size and class distributions to observe the effect of data conditions to the results.

Pitch type prediction problem aims to estimate the next pitch type between pitcher and batter player in baseball games. Pitcher and batter player matchup forms a pairwise condition and pitcher players make strategical decisions against batter players to pitch most successful pitch type. In order to predict the pitcher player next pitch type decision, existing researches has focused on evaluating it as a classification problem. Average team, player and pitch based metrics has been mostly used for classification processes.

In order to determine the direction of this study, we analyzed 3 previous studies of Ganeshappilai and Guttag, M. Hamilton et al., Sidle and Tran. Ganeshappilai and Guttag implemented a binary classification by evaluating the problem as fastball and non-fastball prediction. M. Hamilton et al. extended the study of Ganeshappilai and Guttag by implementing an adaptive feature set selection algorithm. Sidle and Tran implemented multi-class classification for pitch type prediction with 3 classification methods.

As the theory of the this study, we based our study on bayesian theory in which we set pitcher and batter player matchup as conditions. As we mentioned in probability distribution matrix section, we studied to catch most successful pitch type stats for pitcher player. In contrast, for batter player, we studied to catch most unsuccessful pitch types by reverting strike success rates for batter player. We calculated the joint probabilities of pitcher and batter player matchup as probability distribution matrix and trained. We implemented this training schema for both zone-based and non-zone-based approach to observe whether zone is a significant attribute in pitch type prediction problem.

To review the classification algorithms in each approach, naive bayes classifier mostly achieved better with zone information for binary, 3-class, 4-class and 5-class classifications. The largest decrease in accuracy is in binary classification. This decrease points out that zone significantly contributed to distinguish fourseam fastballs from other pitch types. Another important point is that, fourseam fastballs are intented to be thrown against certain strike zones with respect to other pitch types. Because naive bayes classifier performance dramatically decreased for fourseam fastball and other seperation. For 3-class, 4-class and 5-class we observed a smooth decrease in accuracy values. As the probability distribution matrix is based on bayes theory, zone information seems fitting to our training schema and contributed to the results.

Decision tree classifier performed robust performance trends as we analyze from zone-based to non-zone-based test accuracy results in Table 5.1. Accuracy values were higher in non-zone-based approach and it seems decision tree successfully partitioned the data in leaf nodes as target pitch types. It seems zone information is not vital for decision tree however decision tree results were highly affected by the imbalanced class distributions as we see in Figure 5.10. This points out that, decision tree classifier is sensitive to the class distributions. To roughly speaking, we can say that decision tree classifier can be used for non-zone-based approach efficiently.

Support vector machine classifier performed typical characterics of separating hyperplane structure because support vector classifier performed better test accuracy values with non-zone-based approach. Since support vector classifiers focus on the separating hyper planes, the feature set of strike and batting success rates were enough to distinguish pitch types. Performance values were also satisfying for imbalanced cases and this is another typical support vector machine behavior as it does not care about number of samples in class distributions. It seems support vector machine classifier is a suitable method for both approaches.

Ensemble method classifier achieved one of highest performance values and robust trends agains non-zone-based approach and imbalanced data conditions. Since we used boosting classifier, it executed the combination of multiple models. Ensemble clasifer performed well against imbalanced class distributions and non-zone-based feature set. We can say that, ensemble classifier is a suitable method for both approaches.

Perceptron classifier showed one of the most interesting results. For both ap-

proach, it performed worst test accuracy values and inacceptable results. The reason is that, implementing a single perceptron is not sufficient for this problem. We should have designed a multi-layer perceptron which called artificial neural network. Consequently it performed inacceptable results for both zone-based and non-zone-based approaches.

To sum up the overall view of results for zone-based and non-zone-based approaches, we can say that zone information is not a vital decision maker for pitch type prediction. Because we did not observe a significant difference between two approaches in overall perspective for ensemble, decision tree and support vector machine classifier, however naive bayes classifier showed down trends in accuracy metrics. As the classifier algorithms have weak and strong points in various conditions, this problem can be handled without zone information. We can also decrease the cost of zone dimension by eliminating zone information from feature set with appropriate classifier algorithm as we mentioned at the beginning of this research.

REFERENCES

Ait-Mlouk, A., F. Gharnati, and T. Agouti (2017). An improved approach for association rule mining using a multi-criteria decision support system: a case study in road safety. *European Transport Research Review*, 40.

Ao, S.-I., B. B. Rieger, and S.-S. Chen (2008). *Advances in Computational Algorithms and Data Analysis*.

Bharati, M. and M. Ramageri (2010). Data mining techniques and applications. *Indian Journal of Computer Science and Engineering*, 1.

Bible, U. (2020). Pitch type. urlhttp://www.umpirebible.com/ubBlog/archives/228. Online; Accessed: 2020-01-04.

Buhlmann, P. (2012). Bagging, boosting and ensemble methods. *Handbook of Computational Statistics*.

Dhakar, L. (2020). Pitch type. urlhttp://lokeshdhakar.com. Online; Accessed: 2020-01-04.

Evgeniou, T. and M. Pontil (2001). Support vector machines: Theory and applications. pp. 249–257.

Ganeshapillai, G. and J. Guttag (2012). Predicting the next pitch.

Google (2020). Classification. urlhttps://developers.google.com/machine-learning/guides/text-classification. Online; Accessed: 2020-01-04.

Goyal, M. and S. Kumar (2014). Improving the initial centroids of k-means clustering algorithm to generalize its applicability.

Hamilton, M., P. Hoang, L. Layne, J. Murray, D. Padget, C. Stafford, and H. Tran (2014). Applying machine learning techniques to baseball pitch prediction.

Han, J., J. Pei, and M. Kamber (2011). *Data mining: concepts and techniques*.

Hoang, P., M. Hamilton, H. Tran, J. Murray, and C. Stafford (2014). A dynamic feature selection based lda approach to baseball pitch prediction.

Jawad, F., T. U. R. Choudhury, and A. Najeeb (2015). Data mining techniques to analyze the reason for home birth in bangladesh.

Kaviani, P. and S. Dhotre (2017). Short survey on naive bayes algorithm. *International Journal of Advance Research in Computer Science and Management 04*.

Kidokoro, S., Y. Matsuzaki, and R. Akagi (2020). Does the combination of different pitches and the absence of pitch type information influence timing control during batting in baseball. *Human Movement Science*, 554–563.

Kim, H. and W.-S. JuUNG (2018). Does pitch type - zone uncertainty matter to a pitcher's performance. *New Physics: Sae Mulli*, 624–629.

Li, C.-C., C.-W. Lin, and J.-Y. Yu (2010). Statistical pitch type recognition in broadcast baseball videos.

Liu, Y., Y. Zhou, S. Wen, and C. Tang (2014). A strategy on selecting performance metrics for classifier evaluation. *International Journal of Mobile Computing and Multimedia Communications*, 20–35.

Omran, M., A. Engelbrecht, and A. Salman (2007). An overview of clustering methods. *Intell. Data Anal.*, 583–605.

Prajapati, D. J., S. Garg, and N. Chauhan (2017). Interesting association rule mining with consistent and inconsistent rule detection from big sales data in distributed environment. *Future Computing and Informatics Journal*.

Prasad (2011). Using association rule mining for extracting product sales patterns in retail store transactions. *International Journal of Computer Science and Engineering,*

Volume 3, Issue 5.

Schale, P. (2019). Mlb pitch data 2015-2018. url:https://www.kaggle.com/pschale/mlb-pitch-data-20152018. Online; Accessed: 2019-08-08.

Tran, H. T. (2017). Using multi-class classification methods to predict baseball pitch types. Technical report.

Van Stralen, K. J., V. S. Stel, and J. B. Reitsma (2009). Diagnostic methods i: sensitivity, specificity, and other measures of accuracy. *Kidney international*, 1257–63.

Vorontsov, I., I. Kulakovskiy, and V. Makeev (2013). Jaccard index based similarity measure to compare transcription factor binding site models. *Algorithms for molecular biology : AMB*, 23.

Williams Jr, G. R. and M. Kelley (2000). Management of rotator cuff and impingement injuries in the athlete. *Journal of athletic training*, 300–15.

Xiaohu, W., W. Lele, and L. Nianfeng (2012). An application of decision tree based on id3. *Physics Procedia*, 1017–1021.