The Game Of Numbers

MSc Research Project
Data Analytics

Sahil Chordia
Student ID: x20203993

School of Computing
National College of Ireland

Supervisor: Mr. Aaloka Anant

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sahil Chordia
Student ID:	x20203993
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Mr. Aaloka Anant
Submission Due Date:	15/12/2022
Project Title:	The Game Of Numbers
Word Count:	6100
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

The Game Of Numbers

Sahil Chordia
x20203993

Abstract

As the analysis of massive data-sets connected to sports performance has grown in importance within the sporting environment, the relevance of specialists and analysts with prior training in data analytics increases. Sports analysis has been a hot topic for development and research in a number of areas, including field analytics, scouting analysis, and sports betting. Field analytics has been used for making game strategies, player selections for a game, etc. As field analysis helps teams to improve their success percentage in a league and get ahead of their opponents before the game starts, many teams have invested a hefty amount of resources in analytics. This paper studies Gaussian Naive Bayes and its implementations to predict a potential value of a player in football, which could help coaches and management staff as a decision support system to team for a game. This paper also focuses on how a classification algorithm solves a regression problem and whether it needs any method to convert continuous data into categorical data.

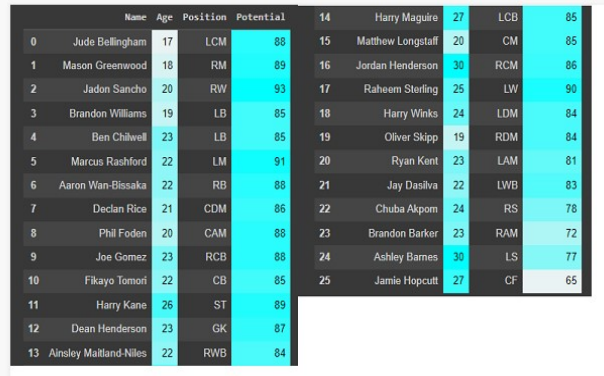
Keywords— Data analytics, Field analytics, Potential, Decision Support System, Gaussian Naive Bayes, Classification Algorithm, Regression Problem, Continuous data

1 Introduction

Football is a team sport and in terms of players and viewers, football is the most popular sport in the world. Football has become more popular recently and has a significant economic impact on the world. Whereas the total revenue for football teams in Europe in 2017 was estimated at $27 billion Al-Asadi (2018). Since there is no method or scientific equation used to determine the preferred position for a player in a team, team management is considered one of the major challenges in football, particularly those related to selecting the right player for the right position in a particular formation. Since coaches typically complete assignments using their views and experiences with players, this makes the selection of players prone to numerous biases.

The success of each football game depends on the players chosen, which is a difficult choice for football managers to make. Football managers might require a decision support system to help them make decisions. These decision support systems can make decisions based on characteristics like among other things, the player's specific abilities and performance data, player combinations, physical fitness, psychological variables, and injuries. To assess the value of each quality, some coaches may additionally employ importance weights. Football managers can benefit from important weights since they show how a specific attribute affects the likelihood of a good outcome.

This paper proposes a new approach for building a decision support system with the help of machine learning algorithm and appropriate data collected from all the games and practice data. Various attributes were chosen from a set of 92 features to increase the accuracy by dropping features which were not useful. The decision support system will predict the potential score of each player from which team will be selected for a game based on the positions required according to opposition and game strategy.



	Name	Age	Position	Potential
0	Jude Bellingham	17	LCM	88
1	Mason Greenwood	18	RM	89
2	Jadon Sancho	20	RW	93
3	Brandon Williams	19	LB	85
4	Ben Chilwell	23	LB	85
5	Marcus Rashford	22	LM	91
6	Aaron Wan-Bissaka	22	RB	88
7	Declan Rice	21	CDM	86
8	Phil Foden	20	CAM	88
9	Joe Gomez	23	RCB	88
10	Fikayo Tomori	22	CB	85
11	Harry Kane	26	ST	89
12	Dean Henderson	23	GK	87
13	Ainsley Maitland-Niles	22	RWB	84
14	Harry Maguire	27	LCB	85
15	Matthew Longstaff	20	CM	85
16	Jordan Henderson	30	RCM	86
17	Raheem Sterling	25	LW	90
18	Harry Winks	24	LDM	84
19	Oliver Skipp	19	RDM	84
20	Ryan Kent	23	LAM	81
21	Jay Dasilva	22	LWB	83
22	Chuba Akpom	24	RS	78
23	Brandon Barker	23	RAM	72
24	Ashley Barnes	30	LS	77
25	Jamie Hopcutt	27	CF	65

Figure 1: England Football Players

The list of football players playing for England alongside their Age, Position and their overall potential score is shown in image 1. Having a potential score for every player with its position based on various factors, makes it easy for team manager and coach to select their final 11 players for a game.

1.1 Research Question and Objective

How does Gaussian Naive Bayes support off-field analytics to select players for a team in football on the basis of potential scores?

1.1.1 Research Objective

- The main objective for this research is to built an decision making system which helps a club or a football team management to choose their players on the basis of their potential score. Selecting players through these decision support system would avoid favour-ism and discrimination between player.

- To find out the compatibility and efficiency of a classification algorithm with continuous data and regression problems.

- Find out methods that help a classification algorithm solve a regression problem efficiently.

1.2 Research Motivation

- Similar kinds of models have been in use in basketball in various leagues in USA for scouting and selecting team which uses Gaussian Naïve Bayes.

- In football mostly used algorithms are Linear Regression, Random Forest and Ada-Boost.

- So the motivation behind using Gaussian Naïve Bayes in this research was to implement similar model for selecting team in football, which is used in basketball. And to check how does the model perform with continuous data.

2 Related Work

In this section an detail analysis on previous research taken place in Sports predictions (section 2.1) and methods used to improve Gaussian Naive Bayes as a classifier (section 2.2) are explained.

2.1 Sports Prediction

A neural network model was built to predict the best-suited team for a football match(Enyindah et al. (2015)). Football is a team sport, and selecting a team for a particular game is important as team players are responsible for the team's winning and losing. The author used various attributes related to players, such as attack awareness, defense, passing, body balance, stamina, dribble speed, kicking power, injury tolerance, and many more. These attributes where then divided into four groups such as:-

- Players' Technique: Features like "attack awareness, defense, form, tenacity, teamwork, and acceleration."
- Players Speed: Features like "response, explosive power, dribble speed, and top speed."
- Player's Statistical Status : Features like "body balance, stamina, jumping, kicking power, injury tolerance."
- Players Resistance: Features like " attack, defense, header accuracy, dribble accuracy, short pass accuracy/speed, long pass accuracy and speed, shot accuracy, ball control, etc."

A neural network model was then applied to each group to find out a score for each player in all four groups. Matlab was used to implement this model. This model had one hidden layer with total of 20 neurons. This values were then used to find out the optimal weighted values for each player and players were selected on the basis of this Optimal weights. Results were then evaluated with the values of RMSE (Root Mean Error Square) and regressions. The values were increasing as the number of tests increased.

A method was proposed for selecting teams in cricket based on previous match data and using machine learning algorithms like random forest and support vector algorithms by (Shetty et al. (2020)). All of the attributes were divided into batsmen, bowlers, and all-rounders by the author. It also had common attributes such as opponent, runs scored, team averages, etc. But as the data didn't have too many features, various other features like pitch, ground, and weather were taken into account. These models took bowlers and batters into account while constructing an 11-player squad. But all-rounders will always receive lower ratings when compared to bowlers and batsmen. All-rounders are therefore taken into account when developing the model to produce an official team. The author created the interface with Flask API and the model with Jupyter notebook. Data was divided into ration of 20:80 for training and testing respectively. The accuracy rate for this model is 76% for batsmen, 67% for bowlers, and 95% for all-rounder. The outcomes were confirmed for 20% of the data-set, and the aforementioned outcomes were achieved. The team for a game was predicted by this algorithm, but the drawback of this method was that it had to divide the data based on player categories, which are batsmen, bowlers, and all-rounders, which could be a disadvantage as it would predict the whole team in bits and not as a whole group of 11 players. It cannot use all the features of the model at once. Similar to cricket, football also has a team of 11 players that can be selected, taking various features into consideration. But unlike cricket, football has more features to be taken into consideration. As a result, a more sophisticated algorithm should be used to treat all features as one and not in bits.

In this research paper, the author discusses how player transfers and team construction can be planned using machine learning (ML). In this study, the author proposed three definitions of a successful transfer as well as a number of parameters for player evaluation. The success of a player transfer is predicted using the Random Forest, Naive Bayes, and Ada-Boost algorithms in this article. To train and evaluate the classifiers, (Ćwiklinski et al. (2021)) used accuracy. The author conducted a number of experiments; they vary in terms of the parameters' weights, what constitutes a successful transfer, and other elements. The results were encouraging (accuracy = 0.82, precision = 0.84, recall = 0.82, and F1-score = 0.83), according to the author. The evidence that has been presented thus far supports the idea that professional football team development can benefit from machine learning. Ćwiklinski et al. (2021)

The goal of the study is to utilize machine learning algorithms to identify the on-field positions of a team of football players based on their tactical-technical behavior ((García-Aliaga et al. (2021)). It does this by using game information from various seasons and national leagues. There were 4 types of outliers found in the data, which were then removed by the author to increase accuracy. The author used the RIPPER algorithm to predict on-field positions for a player on a team. The author ran the model through three scenarios based on different positions to find the best player on a team for a particular position. The first scenario was based on positions for strikers and wingers. The output of this group was that the accuracy was 79% which distinguished them from offensive play, and 80% with more defensive variables. In terms of offensive factors, the WGs shot less accurately and made more crosses. In terms of the defensive factors, the WGs had more recoveries. The second scenario was based on positions for the defensive midfielder and central midfielder. With just 5 rules and 7 variables, it was possible to distinguish between the players with approximately 94.14% accuracy. Because they carry out fewer clearances and result in fewer offside situations, the DMCs are identified by the regulations. They also clear in a specific direction, and they use long passes more frequently than other players. DMCs are players who are more frequently dribbled past by their opponents but provide more offensive play. Compared to DMCs, these players make more passes to the hole, make more shots, and create more individual plays. DMCs, on the other hand, frequently fell short in one-on-one battles.

2.2 Gaussian Naive Bayes and Data Transformation

The Naive Bayes (NB) approach of machine learning is particularly efficient. A NB model views prediction as binary classification; it builds predictors by studying historical data from multiple modules and bases decisions on them Wang and Li (2010). This study examined some crucial Naive Bayes software fault prediction model elements. The author contrasted the decision tree J48, which was also employed in many other studies, with the Naive Bayes software fault prediction model. Results of the experiment proved the usefulness and significance of Nave Bayes for prediction and shown that its performance is plainly superior. Wang and Li (2010) has also described various types of Naive Bayes and their working. Various types of Naive Bayes mentioned in this papers are Multinomial Naive Bayes, Multi-variants Gauss Naive Bayes, and Flexible Bayes.

The goal of this study performed by Chu et al. (2020) is to assess how three different data types—Text, Numeric, and Text + Numeric—affect the effectiveness of the Random Forest, k-Nearest Neighbor (kNN), and Naive Bayes (NB) algorithms as classifiers. In this paper, the classification issues are investigated in terms of mean accuracy and the implications of modifying algorithm parameters over various data-set types. Eight distinct data-sets from UCI were used to investigate this content analysis and train-test the models for all three techniques. The findings of this investigation clearly demonstrate that Random forest algorithm and k - nearest neighbors perform better than NB. Furthermore, the mean accuracy performance of kNN and RF is comparable, but kNN requires less time to train a model. Random Forest is unaffected by the amount of characteristics in a data-set changing, however Naive Bayes' mean accuracy fluctuates and ends up with a lower mean accuracy whereas kNN's mean accuracy rises and ends up with a higher accuracy. The average accuracy of a Random forest classifier is unaffected by changes in the number of trees, but training the model now takes much longer.

According to the research of Chu et al. (2020), the most effective classifiers algorithms for any type of data-set are found to be Random Forest Classifier and k-Nearest Neighbor. Therefore, if the feature variables are independent and located in the issue space, Nave Bayes can beat the other two algorithms. Random Forest Classifier requires the most processing time, while Naive Bayes requires the least.

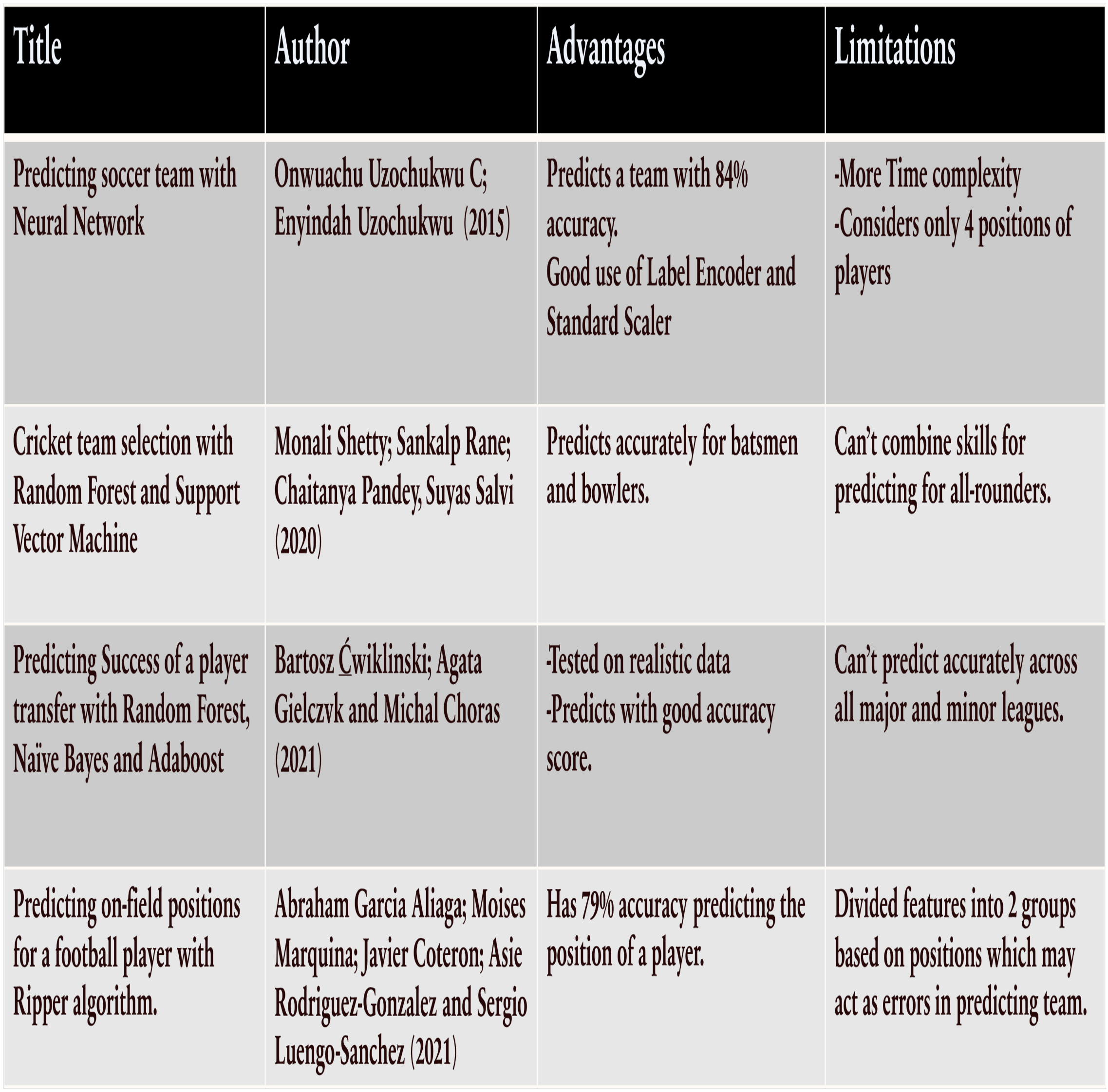
Jishan et al. (2015) has proposed a method which makes Gaussian Naive Bayes an efficient classifier for continuous data by converting it into categorical data with optimal equal width binning for predicting the final exam score of every student.

It's possible that choosing the bin width value at random won't improve our accuracy. As a result, Jishan et al. (2015) has created a discretization technique that is based on error minimization and equal width binning. According to author, dynamically search for the bin width value for a continuous property will help find the best one. The dynamic searching suggests that iteration is necessary in order to locate the ideal bin width value. Additionally, data sets may contain several continuous qualities. If these attributes are distinct from one another, determining the optimal bin width value for all of the continuous attributes in the data set would improve performance in general. According to this research Binning enhances the performance of Gaussian Naive Bayes classifier for predicting values from continuous data.

2.3 Gap Analysis

Paper which are closely related to the research topic have been summarized in table 2 with its advantages and limitations.

After observing the table 2, for all the models, data is divided into groups and then passed through the model, which increases the time complexity and space complexity of the model. To overcome this problem, I have proposed a system in which all the features in the data will be used as one group to train and test the model. This will help build a more integrated model to predict the potential scores of all players and select the final team for a game based on those potential groups. As stated by Chu et al. (2020), Naive Bayes outperforms other classifier such as Random Forest and kNN classifier when it comes to independent feature variable and takes less computational time as compared to other classifier, Gaussian Naive Bayes better suits the research.



Title	Author	Advantages	Limitations
Predicting soccer team with Neural Network	Onwuachu Uzochukwu C; Enyindah Uzochukwu (2015)	Predicts a team with 84% accuracy. Good use of Label Encoder and Standard Scaler	-More Time complexity -Considers only 4 positions of players
Cricket team selection with Random Forest and Support Vector Machine	Monali Shetty; Sankalp Rane; Chaitanya Pandey, Suyas Salvi (2020)	Predicts accurately for batsmen and bowlers.	Can't combine skills for predicting for all-rounders.
Predicting Success of a player transfer with Random Forest, Naïve Bayes and Adaboost	Bartosz Ćwiklinski; Agata Gielczvk and Michal Choras (2021)	-Tested on realistic data -Predicts with good accuracy score.	Can't predict accurately across all major and minor leagues.
Predicting on-field positions for a football player with Ripper algorithm.	Abraham Garcia Aliaga; Moises Marquina; Javier Coteron; Asie Rodriguez-Gonzalez and Sergio Luengo-Sanchez (2021)	Has 79% accuracy predicting the position of a player.	Divided features into 2 groups based on positions which may act as errors in predicting team.

Figure 2: Related Work

3 Methodology

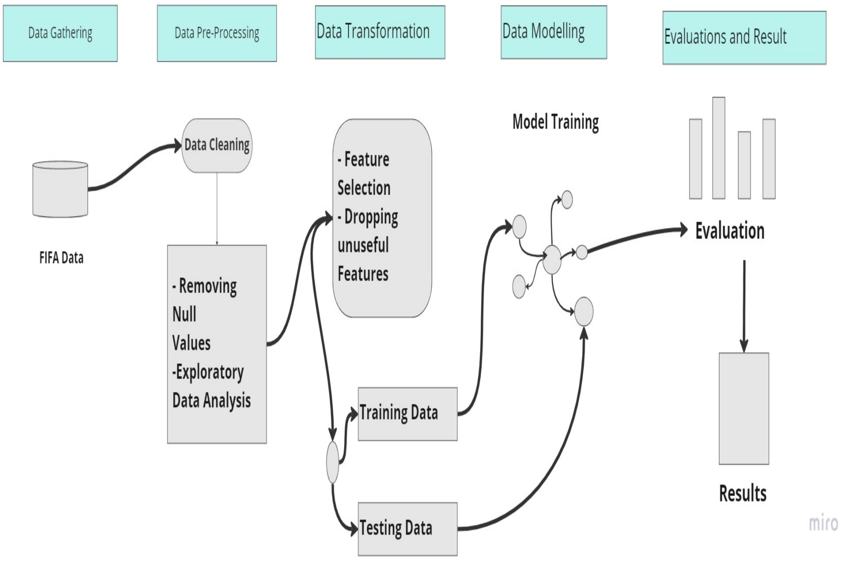


Figure 3: CRISP-DM Methodology

In this research, CRISP-DM methodology has been used. Figure 3 provides an illustration of the research methodology, from feature extraction, data collection, and processing, to modeling. This research is divided into total 5 parts each has its own significance and importance in this project.

3.1 Data Gathering

Data used for this research has been downloaded from kaggle. And data is open source so there are no ethical or copyright issues.

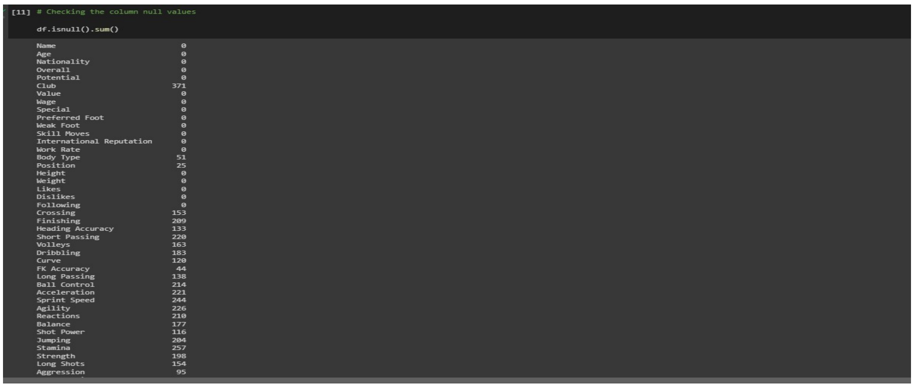
3.2 Data Pre-Processing

The initial stage in data mining is data preprocessing. The datasets used in research must be preprocessed before being used for any kind of classification. These datasets can include information that produces inaccurate results. Therefore, before doing an analysis, the quality of the data should be improvedKamel et al. (2019).

Fifa Data has 18,541 rows and 92 columns. This data has various columns such as Name, Age, Nationality and many more. All this features cannot be used for modeling as they are of no use and will slow down the processing time. Out of 92 features only features are used for modeling the algorithm.

3.2.1 Null Values

There are many null values in many features which may reduce the accuracy of the model. Figure 4 shows the count of null values in every column.



```
[11] # Checking the column null values

     df.isnull().sum()

     Name                          0
     Age                           0
     Nationality                   0
     Overall                       0
     Potential                     0
     Club                        371
     Value                         0
     Wage                          0
     Special                       0
     Preferred Foot                0
     Weak Foot                     0
     Skill Moves                   0
     International Reputation      0
     Work Rate                     0
     Body Type                    51
     Position                     25
     Height                        0
     Weight                        0
     Likes                         0
     Dislikes                      0
     Following                     0
     Crossing                    153
     Finishing                   209
     Heading Accuracy            133
     Short Passing               220
     Volleys                     163
     Dribbling                   183
     Curve                       120
     FK Accuracy                  44
     Long Passing                138
     Ball Control                214
     Acceleration                221
     Sprint Speed                244
     Agility                     226
     Reactions                   210
     Balance                     177
     Shot Power                  116
     Jumping                     204
     Stamina                     257
     Strength                    198
     Long Shots                  154
     Aggression                   95
```

Figure 4: Count of Null Values per Column

To overcome this problem we have replaced all the null values by mean of that column that is by the average value of that column. For example: There are 143 null values in the column "Volleys" of Fifa Dataset and the mean of that column is 44.68, hence all the 143 null values are replaced by 44.68.

3.2.2 Normality of Data

Gaussian Naive Bayes is based on Gaussians Distribution so it is better to know if data is normally distributed or not. Figure 3 shows a histogram of data which describes the distribution of data.

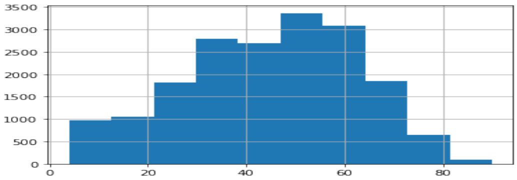
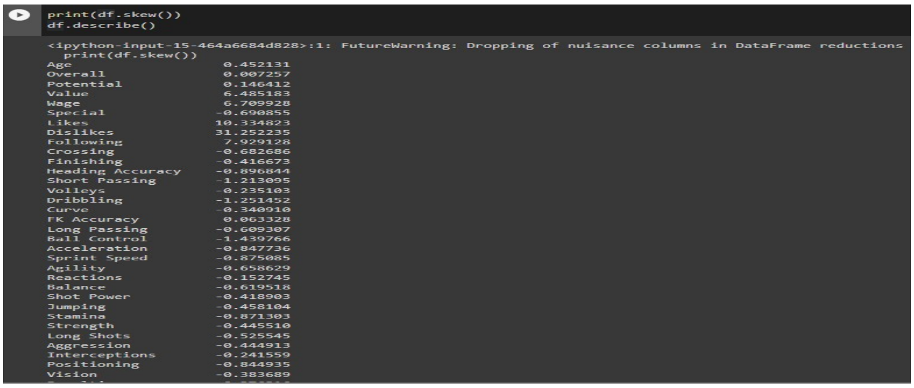


Figure 5: Histogram of Data

After observing Figure 5 we can identify that the data is slightly skewed towards left side.



```
print(df.skew())
df.describe()

<ipython-input-15-464a6684d828>:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions
  print(df.skew())
Age                   0.452131
Overall               0.007257
Potential             0.146412
Value                 6.485183
Wage                  6.709928
Special              -0.690855
Likes                10.334823
Dislikes             31.252235
Following             7.929128
Crossing             -0.682686
Finishing            -0.416673
Heading Accuracy     -0.896844
Short Passing        -1.213095
Volleys              -0.235103
Dribbling            -1.251452
Curve                -0.340910
FK Accuracy           0.063328
Long Passing         -0.609307
Ball Control         -1.439766
Acceleration         -0.847736
Sprint Speed         -0.875085
Agility              -0.658629
Reactions            -0.152745
Balance              -0.619518
Shot Power           -0.418903
Jumping              -0.458104
Stamina              -0.871303
Strength             -0.445510
Long Shots           -0.525545
Aggression           -0.444913
Interceptions        -0.241559
Positioning          -0.844935
Vision               -0.383689
```

Figure 6: Skewness of Data

To confirm weather the data is symmetrically distributed or asymmetrically we used skew() function. Figure 6 shows the result of skew() function as maximum columns have a negative value which means the data is left skewed and distributed asymmetrically.

3.2.3 Exploratory Data Analysis

It is important to know what is there in the data. To know the data inside out we have performed Exploratory Data Analysis on our data. This section shows the initial analysis done on FIFA data.

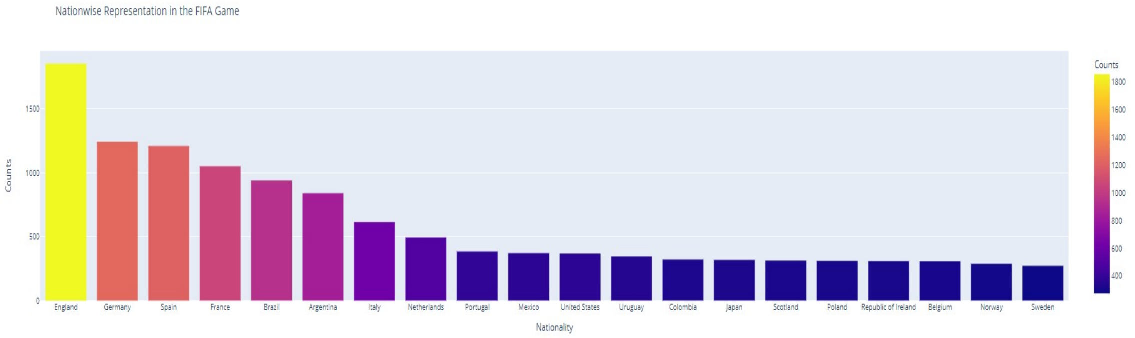


Figure 7: Nation-wise player representation in FIFA

Figure 7 represents the count of players who is playing in FIFA representing their countries. As per the observation England has the most number of player that is 1856 player in FIFA, followed by 1245 by German and 1212 by Spain.

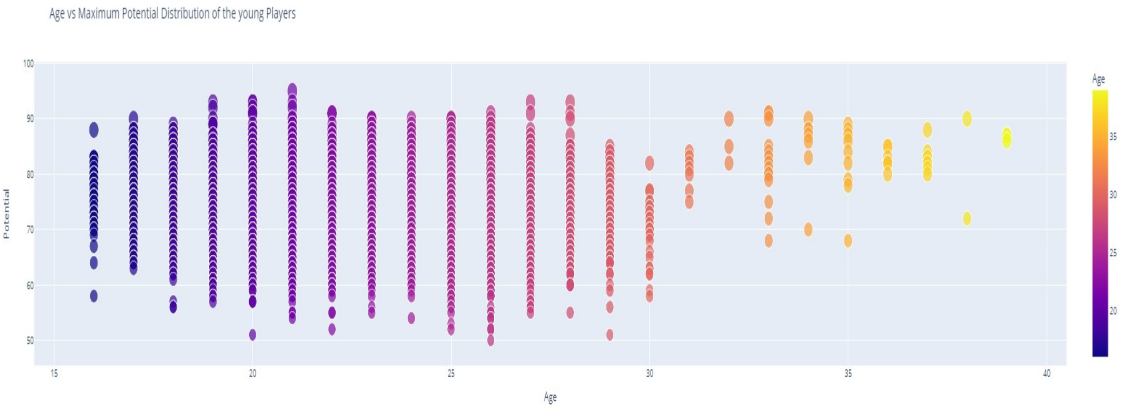


Figure 8: Age vs Potential Distribution of players

Figure 8 depicts the potential distribution based on a player's age ranging from 15 to 40 years old. After observing the graph, it shows that most of the players in the age group of 20 to 30 have a potential score above 85, and very few players in the age group of 30 to 40 have a potential score above 85.

3.3 Data Transformation

3.3.1 Feature Selection

It is well recognized that the features (or representations) used in machine learning methods have a significant impact on how well they work. This feature engineering requires a significant amount of work in order to enable efficient machine learning Luo et al. (2018). In order to train model, 53 features have been selected based on Correlation matrix and relevance of a feature in predicting potential score of a player. Out of 92 columns, 54 columns are taken into account for correlation matrix to find out if these columns are interconnected to each other. Hall (1999) proposed a method for feature selection in training an algorithm based on correlation. When two columns have 90% or more correlation between them either one of two column is dropped to avoid confusion or processing of same data in two columns.

Following is a list of features chosen for x varaible in the modeling: Name, Age, Nationality, Overall, Club, Value, Wage, Special, Preferred Foot, Weak Foot, Skill Moves, International Reputation, Work Rate, Body Type, Position, Height, Weight, Likes, Dislikes, Following, Crossing, Finishing, Heading Accuracy, Short Passing, Volleys, Dribbling, Curve, FK Accuracy, Long Passing, Ball Control, Acceleration, Sprint Speed, Agility, Reactions, Balance, Shot Power, Jumping, Stamina, Strength, Long Shots, Aggression, Interceptions, Positioning, Vision, Penalties, Composure, Standing Tackle, Sliding Tackle, GK Diving, GK Handling, GK Kicking, GK Positioning, GK Reflexes.

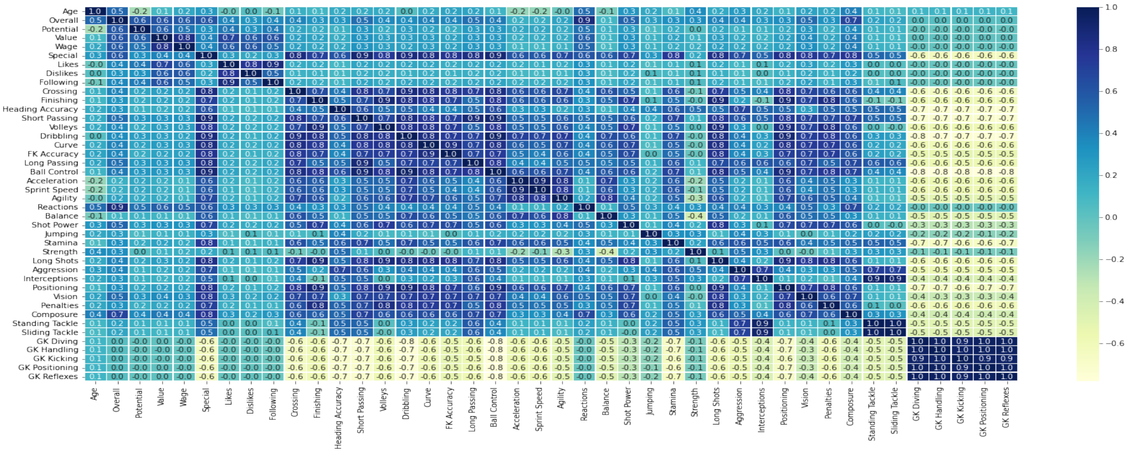


Figure 9: Correlation Matrix

Figure 9 shows a heat-map of correlation matrix between 54 columns. Value represented by light blue are the least co-related columns, as the color gets darker correlation increases. As per observation Standing Tackle and Sliding Tackle are completely co-related with other so Standing Tackle column has been dropped. Similary GK Diving, GK Handling, GK Kicking, GK Positioning and GK Reflexes are all highly co-related to each other so GK Handling, GK positioning, GK Kicking and GK Diving have been dropped.

3.4 Data Modelling

As the data is labeled, supervised learning techniques is used for modeling, where the model learns to predict the target variable based on input data. The model aids in result prediction for unobserved data by learning from labeled training data. The two types of supervised learning are further divided as follows:

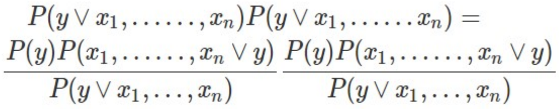
- Classification is a method of predictive modeling with a categorical target variable.
- Regression is a predictive modeling technique using a numerical target variable.

The Problem solved in this research is a regression problem and the algorithm used for modeling is classification algorithm such as Different techniques like Binning, Label Encoder, Train-Test-Split and k-Fold cross validation.

3.4.1 Machine Learning Algorithm

The main application of the Naive Bayes algorithm, a supervised learning algorithm, is to address classification issues. It is considered naive since the occurrence of one feature is unrelated to the occurrence of other traits. The likelihood of an event is used to make predictions. The parameters in the data-set will be used along with the occurrences of the actions, and predictions on the likelihood of the events will then be made.

- Gaussian Naive Bayes: - Gaussian Naive Bayes offers strong performance metrics, as the method relies on conditional independence, the accuracy for classifications is also highly attractive. Because a node's dependencies are spread both equally and unequally, the correctly classified class supports the wrongly classified class, and vice versa Choube et al. (2022). As a result, dependencies that are still strong can balance out dependencies that are weak. When applying the Bayes theorem to feature pairs and given a class value, Naive Bayes is a group of supervised learning algorithms that takes into account the naive conditional probability as indicated in figure 10.



$$P(y \vee x_1, \ldots\ldots, x_n)P(y \vee x_1, \ldots\ldots x_n) = \frac{P(y)P(x_1, \ldots\ldots, x_n \vee y)}{P(y \vee x_1, \ldots, x_n)} \frac{P(y)P(x_1, \ldots\ldots, x_n \vee y)}{P(y \vee x_1, \ldots, x_n)}$$

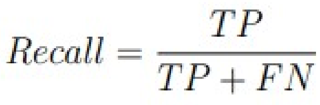
Figure 10: Gaussian Naive Bayes Formula

In the above formula of Gaussian Naive Bayes P(y), the probability of y being a value of x is predicted by comparing all x values from x1 to xn.

3.5 Evaluation and Result

3.5.1 Recall

How accurately the predictions are made out of all the positive classifications is what is meant by "recall." The higher the recall level, the better the model.

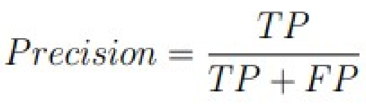


$$Recall = \frac{TP}{TP + FN}$$

Figure 11: Recall
Krstinić et al. (2020)

3.5.2 Precision

The quantity of accurately predicted positive classes that are positive is known as precision.



$$Precision = \frac{TP}{TP + FP}$$

Figure 12: Precision
Krstinić et al. (2020)

3.5.3 Accuracy

Based on the true values and expected values, accuracy is calculated. The model performs better when accuracy is high.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Figure 13: Accuracy
Krstinić et al. (2020)

3.5.4 F1-score

The F1-score is used to simultaneously determine precision and recall. Comparing models with high recall and low precision can be challenging. F1-score comes to the rescue in order to solve such issues.

$$F\text{-}measure = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

Figure 14: F1 Score
Krstinić et al. (2020)

All the metrics mentioned above in the formulas are explained below.

- TP (True positive): The expected values are true and number of predicted values are positive.
- TN (True Negative): Since there are fewer anticipated values than actual values, the actual values are also true.
- FP (False Positive): Also known as Type 1 Error. There are more positive anticipated values than negative actual values.
- FN (False Negative): As well known as Type 2 Error. Although the actual values are fake, the anticipated values are negative.

4 Design Specification

In this section, a detailed overview of the design specifications used for this research is explained. Figure 15, shows the design specification of this research. Once the data was gathered, it was prepped for the model. In data preparation steps such as cleaning null values and replacing them with mean values, exploratory data analysis was conducted. After data preparation methods such as binning, label encoder, and standard scalar were applied for transforming the data, it best suited the model. After data transformation, data was divided into training and testing data by using the train-test-split method from SKLEARN. Binning, Label Encoder, and Train-Test-Split are explained in detail in sections 5.1, 5.2, and 5.3 respectively. After separating the data into training and testing, the data was trained and tested on the model using three different experiments, the results of which are explained in the section 6.

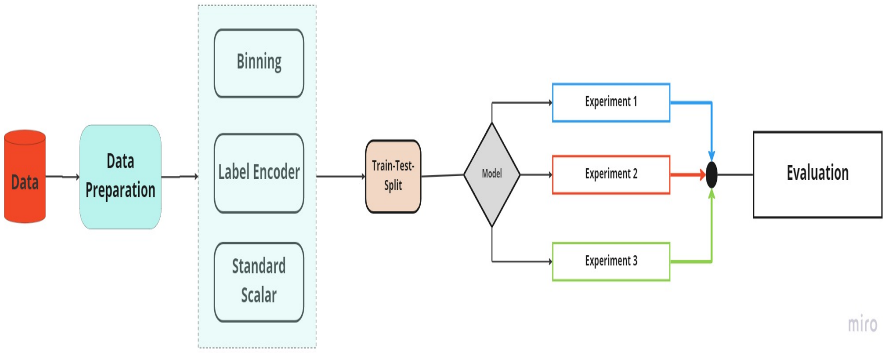


Figure 15: Design Specification

5 Implementation

In this section, all the methods used for modeling are explained in detail.

5.1 Bucketing and Binning

Gaussian Naive Bayes best suits for solving classification problem and normally distributed data, but the problem in the research is regression problem and data is distributed asymmetrically which is left skewed. To overcome this problem for Gaussian Naive Bayes, a method called bucketing and binning is used to convert the problem into classification problem. Binning is a technique which is used to convert the continuous data into categorical data. There are two types of binning as binning by frequency and binning by distance Kola and Muriki (2021). The interval's range of values will be reduced to a single value or each category will have a label name assigned to it. This method helps to increase the accuracy of model.

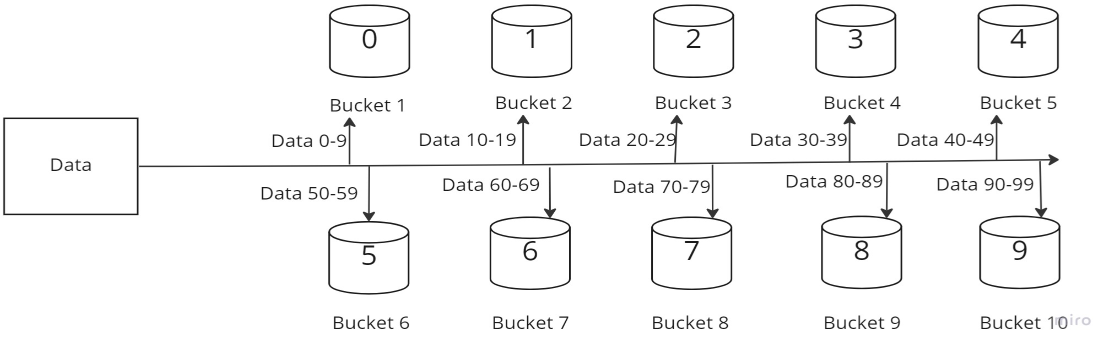
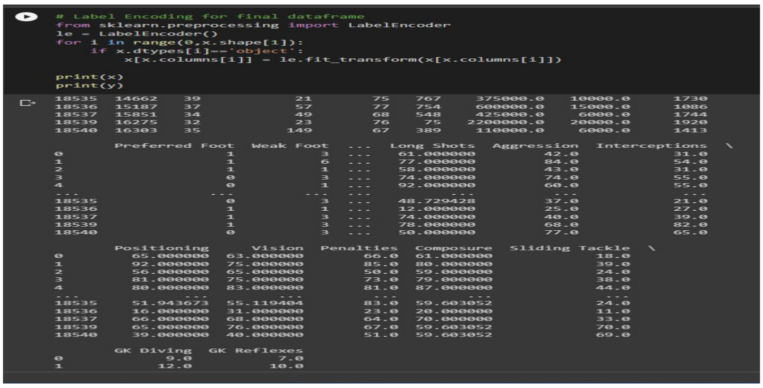


Figure 16: Binning Architecture

Figure 16 shows the binning and bucketing architecture used in this research. Buckets 1 through 10 were named after the ten bins that were created. Once the bins were created, the potential values were then categorized into this bins based on their value at ten's place. For example, a potential value of 18 was placed in Bucket 2, which holds all the values from 10 to 19, as shown in Figure 9. To do so we have made a copy of the potential column in a variable name as ”Potential_new” so we don’t lose the actual data. In this way we keep the original data as well as the bucketed potential values.

5.2 Label Encoder

With label encoding, each value in a textual column is simply turned into a number. It is a simply process but very helpful. All the features selected are converted into numerical values Jackson and Agrawal (2019). This method encodes target variable with values between 0 and n-1, where n is number of classes. Results of Label Encoding are shown in image 17.



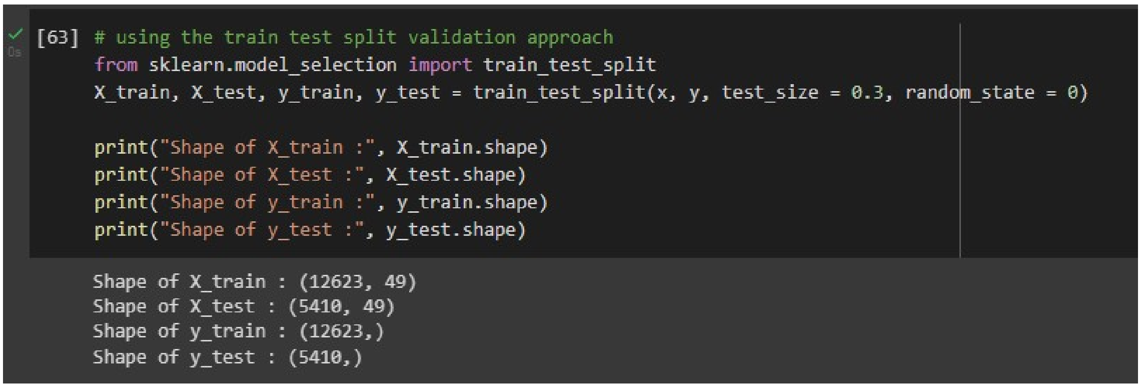
```
# Label Encoding for final dataframe
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for i in range(0,x.shape[1]):
    if x.dtypes[i]=='object':
        x[x.columns[i]] = le.fit_transform(x[x.columns[i]])

print(x)
print(y)
```

```
18535  14662   39            21        75   767    375000.0    10000.0       1730
18536  15187   37            57        77   754    600000.0    15000.0       1086
18537  15851   34            49        68   548    425000.0     6000.0       1744
18539  16275   32            23        76    75   2200000.0    20000.0       1920
18540  16303   35           149        67   389    110000.0     6000.0       1413

       Preferred Foot  Weak Foot  ...  Long Shots  Aggression  Interceptions  \
0                   1          3  ...   61.000000        42.0           31.0
1                   1          6  ...   77.000000        84.0           54.0
2                   1          1  ...   58.000000        43.0           31.0
3                   0          3  ...   74.000000        74.0           55.0
4                   0          1  ...   92.000000        60.0           55.0
...               ...        ...  ...         ...         ...            ...
18535               0          3  ...   48.729428        37.0           21.0
18536               1          1  ...   12.000000        25.0           27.0
18537               1          3  ...   74.000000        40.0           39.0
18539               1          3  ...   78.000000        68.0           82.0
18540               0          3  ...   50.000000        77.0           65.0

       Positioning     Vision  Penalties  Composure  Sliding Tackle  \
0        65.000000  63.000000       66.0  61.000000            18.0
1        92.000000  75.000000       85.0  80.000000            39.0
2        56.000000  65.000000       50.0  59.000000            24.0
3        81.000000  75.000000       73.0  79.000000            38.0
4        80.000000  83.000000       81.0  87.000000            44.0
...            ...        ...        ...        ...             ...
18535    51.943673  55.119404       83.0  59.603052            24.0
18536    16.000000  31.000000       23.0  20.000000            11.0
18537    66.000000  68.000000       64.0  70.000000            33.0
18539    65.000000  76.000000       67.0  59.603052            70.0
18540    39.000000  40.000000       51.0  59.603052            69.0

       GK Diving  GK Reflexes
0            9.0          7.0
1           12.0         10.0
```

Figure 17: Label Encoding x variable

5.3 Training and Testing Data

The dependent variable (represented by y) which in the data-set is potential column, whereas the other variables except the co-related columns are independent (denoted by x). The dependent and independent variables, a test data size of 0.3, and random state was assigned to an integer value were supplied as parameters when using train-test-split from the scikit-learn library.



```
[63] # using the train test split validation approach
     from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 0)

     print("Shape of X_train :", X_train.shape)
     print("Shape of X_test :", X_test.shape)
     print("Shape of y_train :", y_train.shape)
     print("Shape of y_test :", y_test.shape)
```

```
Shape of X_train : (12623, 49)
Shape of X_test : (5410, 49)
Shape of y_train : (12623,)
Shape of y_test : (5410,)
```

Figure 18: Train-Test-Split Function

As shown in image 18 data used for testing makes up 30%, while training data makes up 70%. Using the pandas.shape() method, the outcome variable’s count was verified for both the train and test sets of data. Training data has 12,623 rows and Testing data has 5,410 rows.

5.4 k-Fold Cross-Validation

One of the most popular methods for classifier model selection and error estimation is the K-fold Cross Validation (KCV) methodology. The KCV method divides a data into k subsets, some of which are iteratively used to learn the model and the others to evaluate its performance. However, despite the KCV's efficiency, the number and cardinality of the subsets can only be determined by practical rule-of-thumb techniquesAnguita et al. (2012). There is only one parameter, k, which determines how many splits will be made on a data-set in this straightforward and extensively used procedure. If the wrong value of k is used, the model will perform poorly or it may be very biased toward a particular class label. K can have a value as high as 12. In this research, Elbow chart has been used to find the accurate k value for cross validation based on the behaviour of the data. Figure 19 shows the exact value of k represented by Elbow plot.

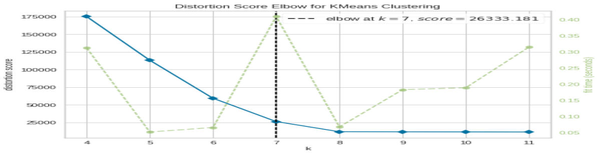


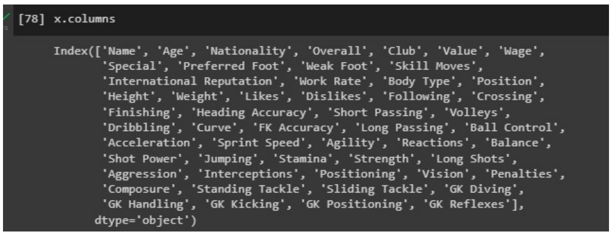
Figure 19: Elbow plot

6 Evaluation

To evaluate the model, Gaussian Naive Bayes was used for first two experiments to find the best accuracy out of all two experiments and for the 3rd experiment, Linear regression was used to find out how does an regression algorithm performs on continuous data. In this section, all three experiments are described and explained, along with their results.

6.1 Experiment 1:-

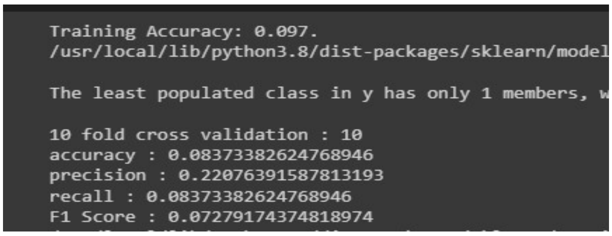
In this experiment, all the features were selected and assigned to the "X" variable, while potential was assigned to "Y." Further, X and Y were divided into training and testing data in a 7:3 ratio, respectively. Label encoder and standard scalar were applied for both variables, and after this, the data was passed to the model to predict the potential of a player.



```
[78] x.columns

    Index(['Name', 'Age', 'Nationality', 'Overall', 'Club', 'Value', 'Wage',
           'Special', 'Preferred Foot', 'Weak Foot', 'Skill Moves',
           'International Reputation', 'Work Rate', 'Body Type', 'Position',
           'Height', 'Weight', 'Likes', 'Dislikes', 'Following', 'Crossing',
           'Finishing', 'Heading Accuracy', 'Short Passing', 'Volleys',
           'Dribbling', 'Curve', 'FK Accuracy', 'Long Passing', 'Ball Control',
           'Acceleration', 'Sprint Speed', 'Agility', 'Reactions', 'Balance',
           'Shot Power', 'Jumping', 'Stamina', 'Strength', 'Long Shots',
           'Aggression', 'Interceptions', 'Positioning', 'Vision', 'Penalties',
           'Composure', 'Standing Tackle', 'Sliding Tackle', 'GK Diving',
           'GK Handling', 'GK Kicking', 'GK Positioning', 'GK Reflexes'],
          dtype='object')
```

Figure 20: Features for Experiment 1

As shown in 21, the model's training accuracy was 09% and its testing accuracy was 08%, indicating that the model is performing very poorly with continuous data. The reason behind this is that Gaussian Naive Bayes works best on classification problems and on categorical data. But the data used for this model is continuous, and the problem is regression. To overcome this problem, a method named "binning" (bucketing) was used in the next experiments.

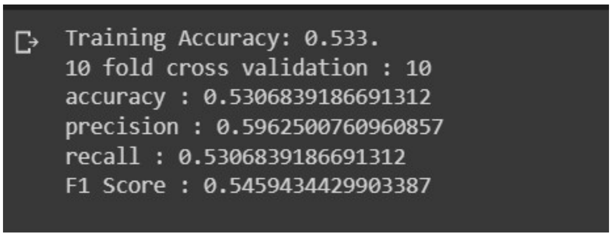


```
Training Accuracy: 0.097.
/usr/local/lib/python3.8/dist-packages/sklearn/model

The least populated class in y has only 1 members, w

10 fold cross validation : 10
accuracy : 0.08373382624768946
precision : 0.22076391587813193
recall : 0.08373382624768946
F1 Score : 0.07279174374818974
```

Figure 21: Result of Experiment 1

6.2 Experiment / Case Study 2

To find the solution to the problem faced in Experiment 1, binning was used to change the problem into a classification problem and convert the data from continuous to categorical. For this experiment, features are selected on the basis of a correlation matrix. If a feature is related to another feature by 90% or more, then either one of the columns is dropped from the "X" variable. Features such as standing tackle, GK diving, GK handling, GK kicking, and GK positioning were dropped. Potential was then distributed into 10 bins according to their numbers, for example, a player with potential 68 was placed in bucket 7. After this, a label encoder and standard scalar were used, and then data was passed onto the model. Image 20 shows the list of feature used for experiment



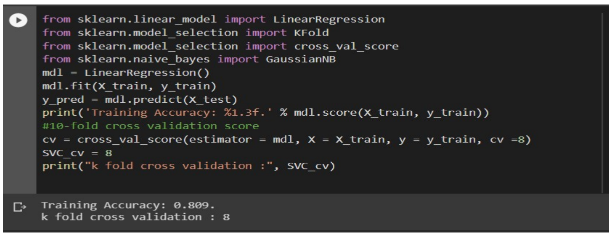
```
Training Accuracy: 0.533.
10 fold cross validation : 10
accuracy : 0.5306839186691312
precision : 0.5962500760960857
recall : 0.5306839186691312
F1 Score : 0.5459434429903387
```

Figure 22: Results for Experiment 2

As shown in 22, after modeling, Gaussian Naive Bayes predicts the potential of a player with 54% accuracy in training while with 53% in testing, which shows that the model is performing well after using the Binning Method.

6.3 Experiment / Case Study 3

In this experiment, instead of using a classification algorithm, we have used a regression algorithm to find out how the model performs. The algorithm used in this experiment is linear regression, and features are selected based on the correlation of columns.



```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import GaussianNB
mdl = LinearRegression()
mdl.fit(X_train, y_train)
y_pred = mdl.predict(X_test)
print('Training Accuracy: %1.3f.' % mdl.score(X_train, y_train))
#10-fold cross validation score
cv = cross_val_score(estimator = mdl, X = X_train, y = y_train, cv =8)
SVC_cv = 8
print("k fold cross validation :", SVC_cv)
```

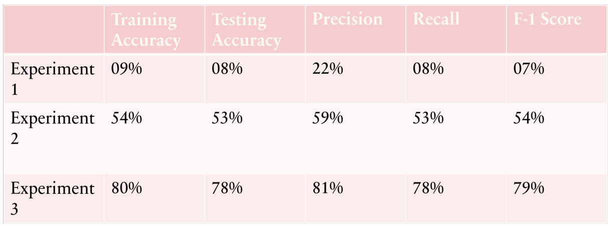
```
Training Accuracy: 0.809.
k fold cross validation : 8
```

Figure 23: Results for Experiment 3

In this experiment, as shown in 23, after modeling, the training accuracy is 80%, which indicates that the model is predicting more accurately when an regression algorithm that is Linear Regression is used.

6.4 Discussion

After completing all the experiments, all the results are shown in 24, with training accuracy, testing accuracy, precision, recall, and an F-1 score for all three experiments.



	Training Accuracy	Testing Accuracy	Precision	Recall	F-1 Score
Experiment 1	09%	08%	22%	08%	07%
Experiment 2	54%	53%	59%	53%	54%
Experiment 3	80%	78%	81%	78%	79%

Figure 24: Evaluation Summary

After observing 24, the following are a few findings from experiments conducted in this research.

- Gaussian Naive Bayes performs poorly on continuous data and provides predictions less accurately when solving a regression problem.
- To gain more accuracy from Gaussian Naive Bayes,the binning method can be used to convert the continuous data into categorical data along with selecting features on the basis of correlation. Binning improves the accuracy of Gaussian Naive Bayes prediction to 53%, while the model predicts with 08% accuracy without binning.
- Regression algorithm like Linear Regression better suits this type of data that is continuous and are efficient in solving regression problems.

Chart 25 refers to all the results displayed through a bar graph.

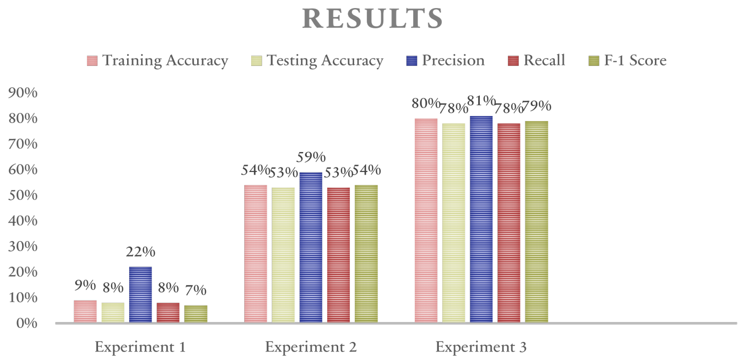


Figure 25: Evaluation Barchart

7 Conclusion

Every algorithm needs a bit of adjustment and adaptation when solving a different problem in order to increase its performance. As the results of Gaussian Naive Bayes had an average accuracy of 53%, while while linear regression, which is a regression algorithm, predicts with an accuracy of 80%, with further adaptations on Gaussian Naive Bayes algorithm to increase the accuracy, it can be used in real time. As this method is already used in Basketball, similar can be done in football and with further research, can be used for various other aspects in football itself.

7.1 Future Work

As this model predicts with average accuracy, it can be tested in real-time test cases to find out how it performs in real time, with some adaptation and tuning of the model to increase the accuracy. Once this is successful in predicting teams for football clubs in real time, this model can be further used in different aspects of football, such as for scouting purposes or predicting the right price for a player transfer. This model can also be used in different sports such as cricket, hockey, baseball, etc. for predicting teams or scouting purposes. This study can be expanded to include visual stimulation for player performance in various aspects of football.

7.2 Acknowledgement

I would like to thank Mr. Aaloka Anant for guiding us throughout the research and for all the valuable inputs. I would also like to thank all the authors whose references were used in this research.

References

Al-Asadi, M. A. M. (2018). Decision support system for a football team management by using machine learning techniques, *Xinyang Teachers College* **10**(2): 1–15.

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L. and Ridella, S. (2012). The 'k'in k-fold cross validation, *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, i6doc. com publ, pp. 441–446.

Choube, G., Dudhmande, G. R., Pushparaj, J., Anand, C. and Suresh, S. (2022). Predicting modalities of dyslexic students using neuro-linguistic programming to enhance learning method, *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, pp. 1–6.

Chu, S.-C., Dao, T.-K., Pan, J.-S. et al. (2020). Identifying correctness data scheme for aggregating data in cluster heads of wireless sensor network based on naive bayes classification, *EURASIP Journal on Wireless Communications and Networking* **2020**(1): 1–15.

Ćwiklinski, B., Giełczyk, A. and Choraś, M. (2021). Who will score? a machine learning approach to supporting football team building and transfers, *Entropy* **23**(1): 90.

Enyindah, P. et al. (2015). A machine learning application for football players' selection, *International Journal of Engineering Research & Technology* .

García-Aliaga, A., Marquina, M., Coterón, J., Rodríguez-González, A. and Luengo-Sánchez, S. (2021). In-game behaviour analysis of football players using machine learning techniques based on player statistics, *International Journal of Sports Science & Coaching* **16**(1): 148–157.

Hall, M. A. (1999). *Correlation-based feature selection for machine learning*, PhD thesis, The University of Waikato.

Jackson, E. and Agrawal, R. (2019). Performance evaluation of different feature encoding schemes on cybersecurity logs, *2019 SoutheastCon*, pp. 1–9.

Jishan, S. T., Rashu, R. I., Haque, N. and Rahman, R. M. (2015). Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique, *Decision Analytics* **2**(1): 1–25.

Kamel, H., Abdulah, D. and Al-Tuwaijari, J. M. (2019). Cancer classification using gaussian naive bayes algorithm, *2019 International Engineering Conference (IEC)*, pp. 165–170.

Kola, L. and Muriki, V. (2021). A comparison on supervised and semi-supervised machine learning classifiers for diabetes prediction.

Krstinić, D., Braović, M., Šerić, L. and Božić-Štulić, D. (2020). Multi-label classifier performance evaluation with confusion matrix, *Comput Sci Inf Technol* **10**: 1–14.

Luo, Y., Qin, X., Tang, N. and Li, G. (2018). Deepeye: Towards automatic data visualization, *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 101–112.

Shetty, M., Rane, S., Pandita, C. and Salvi, S. (2020). Machine learning-based selection of optimal sports team based on the players performance, *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, pp. 1267–1272.

Wang, T. and Li, W.-h. (2010). Naive bayes software defect prediction model, *2010 International Conference on Computational Intelligence and Software Engineering*, pp. 1–4.