A

Group Project Report On

**“Predicting Football Results Using Machine Learning techniques”**

Submitted in Partial Fulfillment of the Requirements For the award of the Degree

**of**

**BACHELOR OF TECHNOLOGY IN**

**Electronics & Computer Engineering (ECM)**

**By**

**Kontham Akhilesh(18311A1925)**

**Katta Rakesh (18311A1921)**

**Rapolu Anil Kumar(18311A1945)**

Under the Guidance of

**Dr. C. SUNIL KUMAR**

**PROFESSOR**



DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

Sreenidhi Institute of Science & Technology

**(Affiliated to JNT University, Hyderabad)**

Yamnampet, Ghatkesar, Hyderabad – 501 301.

## 2020-2021

**DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING**

## SREENIDHI INSTITUTE OF SCIENCE & TECHNOLOGY (AUTONOMOUS)



**CERTIFICATE**

This is to certify that the Group Project work entitled **“Predicting Football Results Using Machine Learning Techniques”**, submitted by **Kontham Akhilesh (18311A1925), Katta Rakesh (18311A1921), Rapolu Anil Kumar (18311A1945)** towards partial fulfillment for the award of Bachelor’s Degree in Electronics & Computer Engineering from Sreenidhi Institute of Science & Technology, Ghatkesar, Hyderabad, is a record of bonafide work done by them. The results embodied in the work are not submitted to any other University or Institute for award of any degree or diploma.

**Internal Guide Project Coordinator External Guide H.O.D**

**Dr. C. SUNIL KUMAR KASI BANDLA Dr. D. MOHAN**

Professor Asst. Professor Department of ECM

# DECLARATION

This is to certify that the work reported in the present project titled **“Predicting Football Results Using Machine Learning Techniques**” is a record work done by us in the **Department of Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology,** Yamnampet, Ghatkesar**.**

The report is based on the project work done entirely by us and not copied from any other source.

KONTHAM AKHILESH (18311A1925)

KATTA RAKESH (18311A1921)

RAPOLU ANIL KUMAR (18311A1945)

# ACKNOWLEDGEMENT

We convey our sincere thanks to **Dr**. **P. Narsimha Reddy,** Director and **Dr. Shiva Reddy**, Principal, Sreenidhi Institute of Science and Technology, Ghatkesar for providing resources to complete this seminar. We are very thankful to **Dr. D. Mohan**, Head of ECM Department, **Kasi Bandla** Associate Professor and **Dr. C. Sunil Kumar** Professor of Sreenidhi Institute of Science and Technology, Ghatkesar for providing an initiative to this group project and giving valuable timely suggestions over our group project work and for their kind co-operation in the completion of the group project. We convey our sincere thanks to all the faculties of ECM department, Sreenidhi Institute of Science and Technology, for their continuous help, co-operation, and support to complete this seminar.

**ABSTRACT**

Predicting the results of football matches poses an interesting challenge due to the fact that the sport is so popular, played nearly around 200+ countries so it is a widespread sport and much popular. However, predicting the outcomes is also a difficult problem because of the number of factors which must be taken into account that cannot be quantitatively valued or modeled. Several methods or approaches are developed for the prediction of the end result of this popular sport.

Modern methods for processing data, in combination with strong computational power of computers, allow us to predict the results of the next matches by applying machine learning algorithms such as Logistic Regression, SVM, XG-Boost on the pre-processed datasets.

**CONTENTS**

**Chapter Page No.**

1. INTRODUCTION 1
2. LITERATURE SURVEY 2
3. PROPOSED MODELS 3
   1. LOGISTIC REGRESSION 3
   2. SUPPORT VECTOR MACHINE 4
   3. XG-BOOST 6
4. ARCHITECTURE 8
   1. DATA SET COLLECTION 8
   2. DATA PRE-PROCESSING 9
   3. DATA SPLITTING 12
   4. MODEL BUILDING 12
   5. EVALUATION 12
   6. PARAMETER TUNING 12
5. DATA VISUALIZATIONS 13
   1. CORRELATION 13
   2. SCATTER MATRIX 14
   3. PAIR PLOT 15
   4. OTHER VISUALIZATIONS 16
6. IMPLEMENTATION 17
   1. LOGISTIC REGRESSION ALGORITHM 17
   2. SUPPORT VECTOR MACHINE ALGORITHM 18
   3. XG-BOOST ALGORITHM 19
7. RESULTS 20
8. CONCLUSION AND FUTURE SCOPE 21
9. REFERENCES 22

**LIST OF FIGURES AND TABLES**

# Figures Page No.

# Logistic Regression 4

# Support Vector Machine 4

# SVM Hyper Plane-1 5

# SVM Hyper Plane-2 5

# SVM Hyper Plane-3 6

# XG-BOOST Decision Tree 6

# Process Flow 8

# Dataset 9

# Dataset features 9

# Dataset Null values 10

# Label Encoding 11

# One Hot Encoding 11

# Splitting Dataset 12

# Correlation Matrix 13

# Scatter Matrix 14

# Pair Plot 15

# Hist Plot 16

# Pie Chart 16

# Logistic Regression Output 17

# SVM Output 18

# XG-Boost Output 19

# Heat map 20

1. **INTRODUCTION**

Sporting is one of the most followed event in the world and among all other sports football (also called as soccer) tops among the most followed or most watched across the world. Different teams come with different strategies and the football world cup happens every four years with lot of surprises and the prediction would be more difficult when they change their strategy to the defensive strategy because of the no of goals scored by individual teams would be less and the model will not be able to predict it well.

The fans of each team will try to predict the winner of the match with more enthusiasm, that may be due to make some money or for fun or entertainment, this model helps them in making predictions before match and in-between the match progress and with good accuracy and ideally the prediction with 100% is not possible due to different factors like luck, weather.

Football is regarded as a complex and dynamic sport when compared to all other sports. Different models were developed to predict the full time results of the English Premier League by considering the datasets of the previous 10 seasons matches of the English Premier League which is the most popular football league in the world. Which also helps the analysts of their respective football match to judge the outcome based on previous statistics and to implement the necessary changes in the team whenever required in different conditions of the season.

1. **LITERATURE SURVEY**

The prediction of the full time result of the football match in any season is done using different emerging technologies which involves Machine Learning, Deep Learning, Neural Networking, IOT, Image Processing and many more technologies. In each technology they use different types of approaches or different tools or different mechanisms to get the output prediction based on only the previous matches of the seasons and which includes the features in the dataset.

The interface is created and the entries are made for the required match prediction and the analysis is done by the users or clients, but there is large no of sources which helps in predicting the results and by convoluting all the results in different models or approaches they will not be able to predict the actual winner of the full time match.

There are different models developed by using Machine Learning like Logistic Regression, Naive Bayes, Support Vector Machine, K-Nearest Neighbours, Bayesian Network. There are pros and cons for each model developed till now. The availability and consideration of the features in the dataset will play a major role in the prediction of the full time match results. The accuracy of the models developed till now are not up to the mark, the accuracy score is lying in the range of 55% - 62% by the different models.

The availability of the datasets is not accurate in the past and not large in number, when the dataset contains less no of records then the dataset may lead to biasing of the model which will turn the model into the wrong directed path. There is no collection of data across the different seasons of the league, there would be more null values and the dependency of the one feature is less compared with the other feature which will not be considered as a good dataset and will lead to the less accurate predicted model by Machine Learning approach.

1. **PROPOSED MODELS**
   1. **LOGISTIC REGRESSION**

Regression models are one of the most popular statistical models (or techniques) used for predictive modelling and data mining. Through regression models, we can be able to predict the target variables (either continuous or categorical). Many of the data scientists use regression models for solving their problem statements. Briefly, the goal of regression model is to build a mathematical equation that defines ‘y’ as a function of ‘x’ variables. The two types of regression are:

**Linear regression** is mainly used when there is a relationship between dependent and independent variables. Here, the increase/decrease in one variable leads to an increase/decrease of the other.

**Logistic regression** is a supervised machine learning algorithm (a machine learning algorithm where the target variable is present) mainly used to predict the probability of a target variable. The nature of the target variable is ‘dichotomous’ which means there would be only two possible cases. Here the outcome is categorical (i.e. the target is yes/no). In simple words, the dependent variable is binary in nature having data coded as either ‘0’ (stands for failure or no) or ‘1’ (stands for success or yes). It is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick.

**Types of Logistic Regression:** Based on the no of categories, Logistic Regression can be divided into 2 types:

1. **Binary/Binomial:** Here the dependent variable will have only 2 possible types either ‘0’ or ‘1’. For ex; these variables may represent success/failure, yes/no, win/loss, etc.
2. **Multinomial:** Here the dependent variable can have 3 or more possible unordered types. For ex; these variables may represent Type A, Type B, Type C.

**Types of Variables:**

1. **Categorical or Discrete:** For Ex; Yes, or No, true or false, 0 or 1 but instead of giving the exact values as 0 or 1, it gives the probabilistic values which lie between 0 and 1.
2. **Ordered:** Here, there can be 3 or more possible ordered types of dependent variable such as Low, Medium, High.

Logistic Regression is similar to Linear Regression except that how they are used. Linear Regression is used for solving regression problems whereas logistic regression is used for solving classification problems. In Logistic Regression, instead of fitting a regression line, we fit an ‘S’ shaped logistic function which predicts two maximum values (0 or 1). For example, Linear Regression could help us predict the students test score on the scale of 0 – 100, i.e. Linear Regression predictions are continuous. But Logistic Regression could help us predict whether the student passed or failed.

For our dataset logistic regression is very much suitable, where the target variable to be predicted is clearly defined with the required data elements that are to be trained.

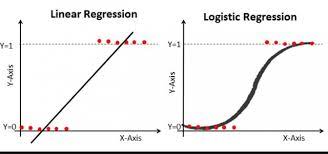


Fig1: Logistic Regression

* 1. **SUPPORT VECTOR MACHINE**

Support Vector Machines (SVMs) are Machine Learning models used for both classification and regression problems. An SVM model represents the training data as points in space so that examples falling in different categories are divided by a hyperplane that is as far as possible from the nearest data point.

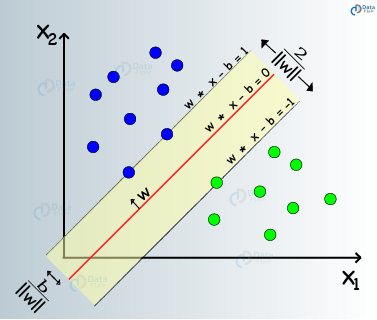


Fig2: Support Vector Machine

New inputs are mapped in the same way as the training data and classified as the category they fall into (which side of the hyper-plane). When the data is not linearly separable, the kernel trick can be used, by using different possible kernel functions such as Radial Basis Functions (RBF) or polynomial functions, in order to map the data into high-dimensional feature spaces and find a suitable high-dimensional hyper-plane. The above classification problem can be extended to solving regression problems in a similar way, by depending only on a subset of the training data to generate a regression prediction. Advantages for using Support Vector Machines include that they are effective in high dimensional spaces, that they are memory efficient thanks to the use of a subset of training points in the decision function, and finally that they are versatile through the use of different possible kernel functions. On the other hand, using SVMs can have some disadvantages: they do not directly provide probability estimates for classification problems, and correctly optimizing the kernel function and regularization term is essential to avoid overfitting.

**How does the algorithm select the right hyper-plane?**

**Scenario 1:**



Fig3: SVM Hyper Plane-1

Here, we have 3 hyper-planes (A, B and C). The SVM algorithm selects the hyper-plane which can differentiate the 2 classes better tham the other hyper-planes. Here, in this scenario, hyper-plane B has performed the job excellently.

**Scenario 2:**



Fig4: SVM Hyper Plane-2

Here, we have 3 hyper-planes (A, B and C) and all the hyper-planes are differentiating the 2 classes very well. The SVM algorithm selects the hyper-plane which maximizes the distance between the nearest data points of both the classes and the hyper plane. This perpendicular distance between the data point and the hyper-plane is called as **Margin.** So in this scenario, hyper-plane C has performed the job excellently than A and B. If the hyper-plane having low margin is selected, then there is a high chance for miss-classification.

**Scenario 3:**



Fig5: SVM Hyper Plane-3

Here, we have 2 hyper-planes (A and B). Hyper-plane A is differentiating the classes very well and hyper-plane B is having higher margin compared to A. In this scenario, the SVM algorithm selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and hyper-plane A has classified the classes correctly. Therefore, the right hyper-plane is A.

* 1. **XG-BOOST (**eXtreme Gradient Boosting**)**

**XG-Boost** is a decision-tree based Machine Learning algorithm, uses a gradient boosting framework. This is used in a wide range it can be used in different types of prediction problems like image detection, machine learning predictions and many more. It has more predictive power than other gradient boosting algorithms.

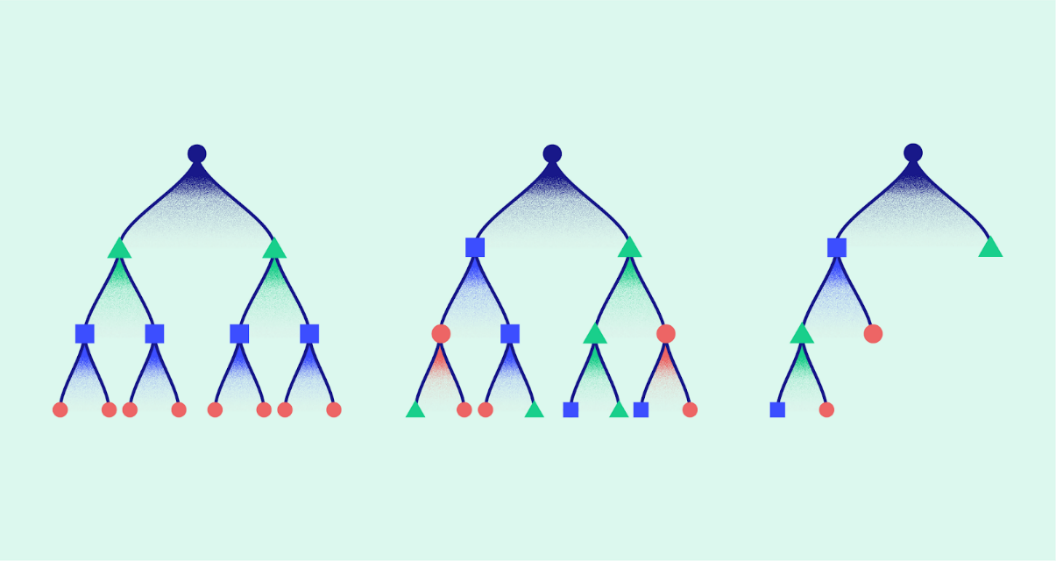


Fig6: XG-Boost Decision Tree

It uses the concept of decision tree, the attributes or features in the data set are selected and treated as the base node and the residual values for each tuple or each row inside the data set with each attribute or feature is calculated and based on the result of the value of the residual, using the concept of binary tree the position of the sub node is made i.e. if the residual value is less than the root node value then it is placed in the left side of the root node, if the value is greater than the root node then it is placed on the right side of the root node.

Considering the second feature also similarly by considering the first feature the residuals are calculated and the decision tree is formed based on that the prediction is made by calculating the residual value of the new record.

1. **ARCHITECTURE**

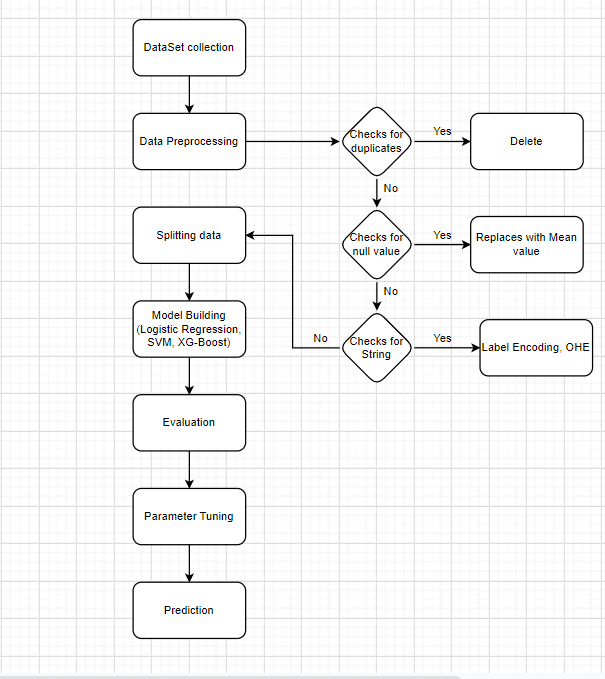
****

Fig7: Process flow

**4.1 DATASET COLLECTION**

Dataset collection is a process of collecting or gathering data from various sources, here the data set contains a single database table or statistical data matrix in which each column represents a particular variable and row represents here the member of the data set. In our project, we are considering the data set collected from the past ten seasons of the English Premier League. We collected this dataset from the kaggle dataset distributors. The Data Set Consist a total of 65 attributes we consider those attributes which have more weight in predicting the output. They are home team points, away team points, full-time result, home team streak hm1, home team goal difference, away team goal difference, different form points (streak goal or penalty goals).

Dataset Collection plays a major role on the prediction model, if the availability of the dataset is less than the predicted model will not work fine because of the less data will lead to bias of the model. If the dataset is more in no with high records or tuples it will work good and the model will predict well, the dataset should contain less no of missing values and less no of strings and less no of dependent features.

In our dataset it consists of 6900 records which indicates that the model developed is well and the prediction will be more accurate and the loss margin/error margin will be less.

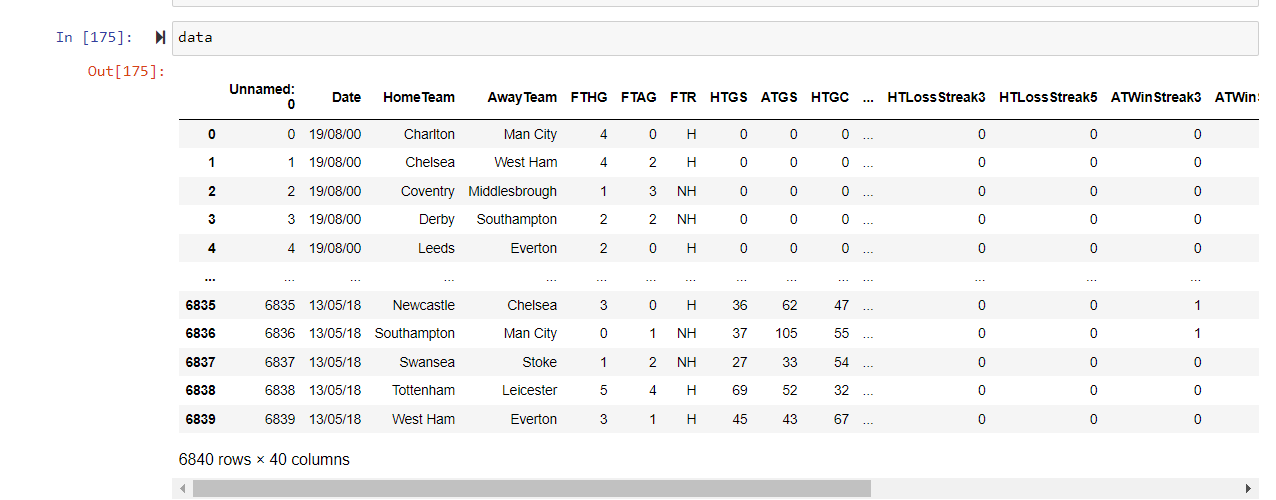


Fig8: Dataset

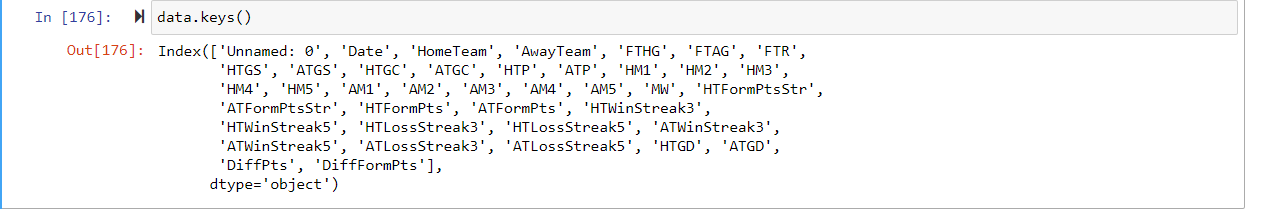


Fig9: Dataset Features

* 1. **DATA PRE-PROCESSING**

Data pre-processing is a process of creating suitable data by cleaning, integrating, transforming, and reducing raw data and extracting required data from it which is suitable for a machine learning model. Data pre-processing plays a major role while creating a machine learning model. In general, the raw data that's been collected from various sources consist of a lot of potentially incorrect data and instrument faulty errors by Data pre-processing those errors will be removed and the model can be developed well.

The steps involved in the Data Pre-Processing are:

**Step 1:** Checks for duplicate values:

If there are any duplicate or repeated rows present in the raw dataset they will be deleted or removed, if there is more no of duplicate data in the dataset it is treated as a poor dataset, if the dataset consists of less no of duplicate records then it is treated as good dataset.

**Step 2**: Checks for Null Value:

In our dataset if there is an empty field with no record in any feature it is treated as a Null Value or it is also called as Empty field. If there are less no of null values in the dataset, then the null value records are deleted. If the no of null values is more in no then they are replaced with the mean, median, mode values of the features in the dataset.

To handle missing values, we will use **Scikit-learn** library which contains various modules for building machine learning models. Here we will use **Imputer** class of **sklearn.preprocessing** library.

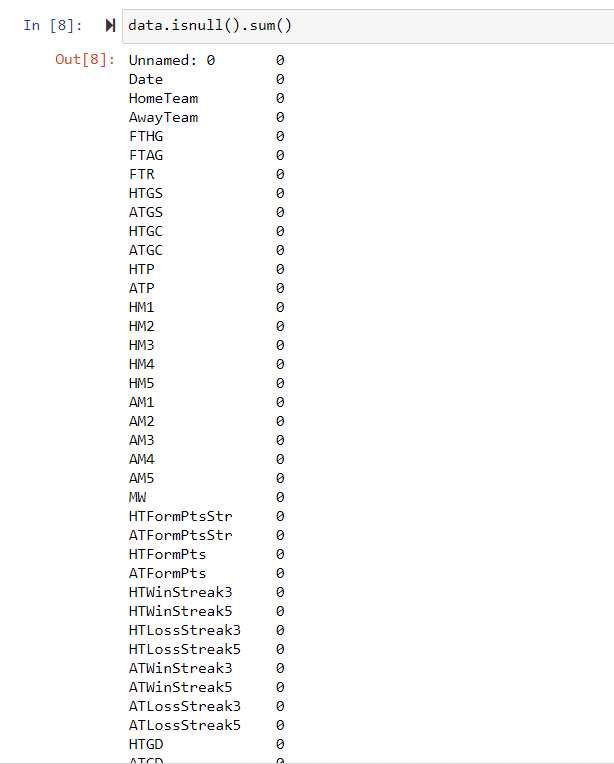


Fig10: Dataset Null values

**Step 3:** Checks for strings:

In general machine learning always work by binary values, if our dataset contains any string values that feature is called as the Categorical variable and the values are to be changed into the numerical for the working of the model. Using the Library called Label Encoder the strings are converted into the numerical by assigning each value as 0, 1, 2. After the process of label encoding One Hot Encoding is done which helps in removing the dummy variable traps. Which removes the dependent variables and make all the dependent variables as independent variables.

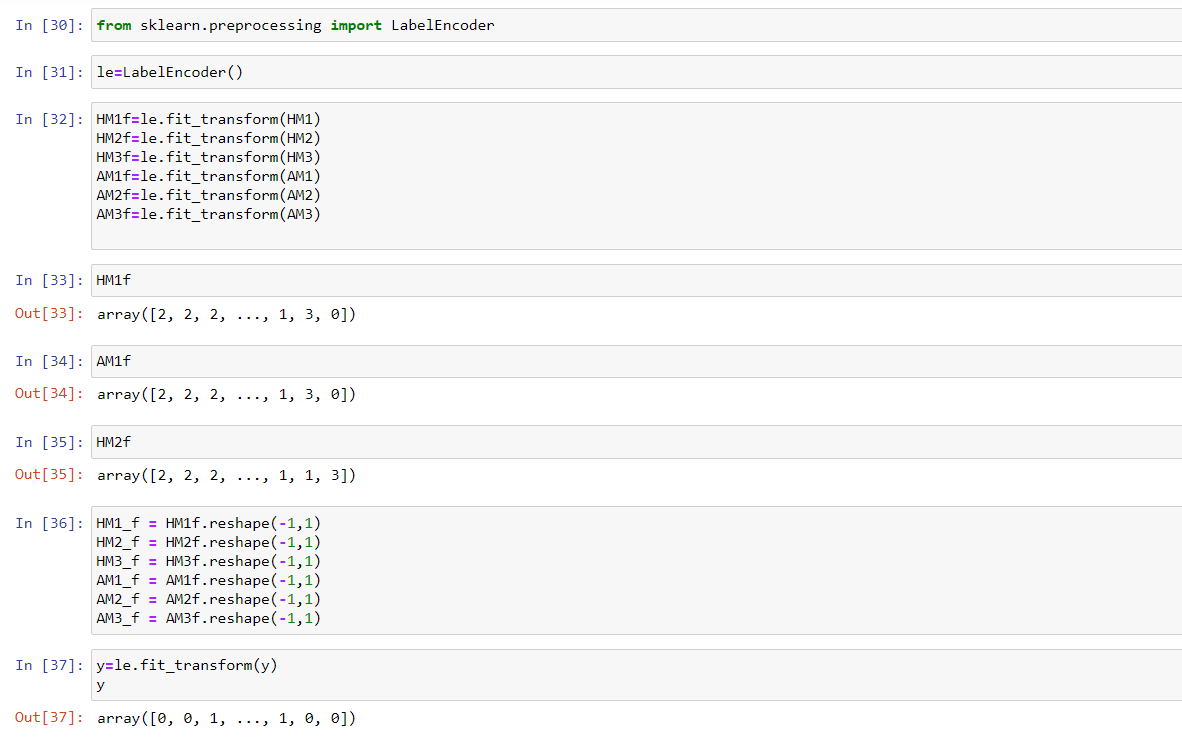


Fig11: Label Encoding

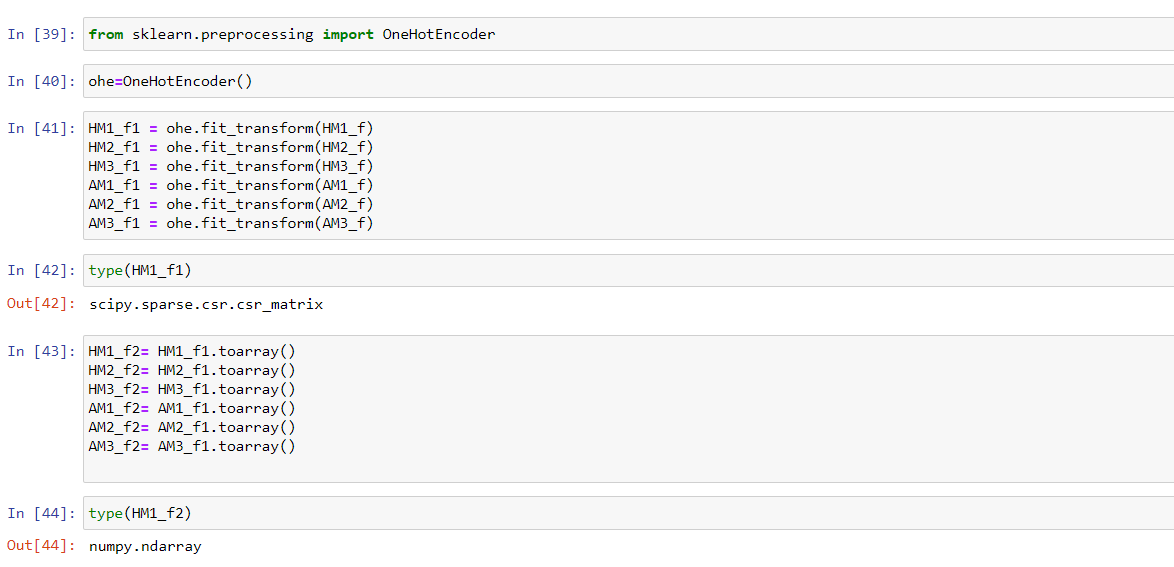


Fig12: One Hot Encoding

* 1. **DATA SPLITTING**

Data is divided into two parts Training and Testing dataset because if we test our training set with a whole different testing set we can’t predict the model and correlations properly. So we divide the same data set into training and testing data.

Training data: a subset of data set to train a model

Testing data: a subset of data set to test the model

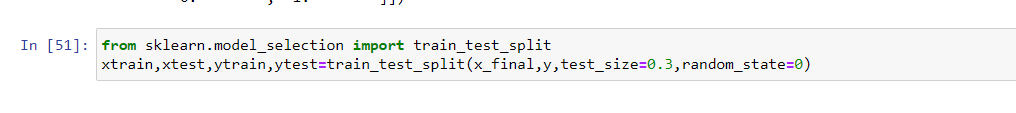
In our project, the new data set with high weighted attributes created will be divided into two parts training data and testing data where the Training set consists of 70% of data and the testing set consists of 30% of data. 

Fig13: Splitting Dataset

* 1. **MODEL BUILDING**

In this model Building process a model is built using the different approaches or different methodology or different techniques which helps in the predicting the value of the required record by considering the feature dataset.

In our project the model building is built using the three different model building mechanisms they are Logistic Regression, Support Vector Machine, XG-Boost.

* 1. **EVALUATION**

We compare the output of the three machine learning algorithms and consider the algorithm which has the highest prediction rate and use that to predict the result.

* 1. **PARAMETER TUNING**

Tuning is the process of maximizing a model's performance without overfitting or creating too high of a variance. In machine learning, this is accomplished by selecting appropriate 'hyper parameters'.

This process helps in increasing the accuracy by identifying the most weighted parameter and it is treated as the hyper parameter.

1. **DATA VISUALIZATIONS**

The data visualization is a process for the better understanding of the work done by the program which helps in identifying the features with high weights and features with high importance. These visualizations are helpful for human to understand much better than by seeing code there will be no use by data visualization to the system or the model. It helps us to predict the picking the correct features available in the dataset.

* 1. **CORRELATION**

It states the relationship between the two or more no of variables which helps in identifying the required features for the model building using the dataset. There are types of correlation they are highly positive correlation, less positive correlation, highly negative correlation, less negative correlation all this indicate the relationship between the two variables which plays a major role in identifying the features in the dataset.

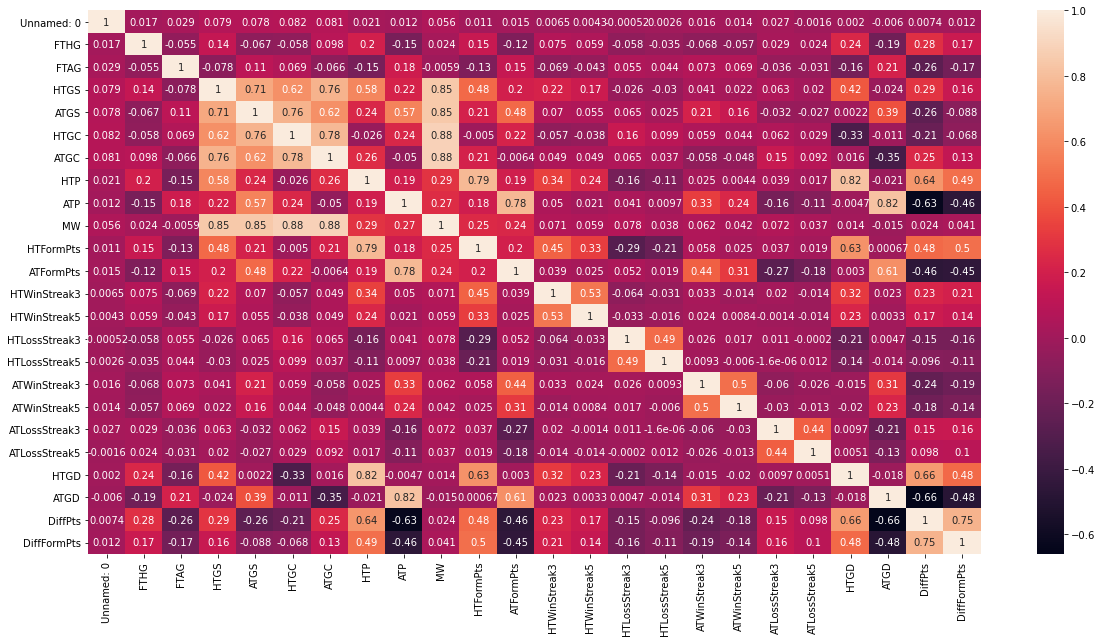
****

Fig14: Correlation Matrix

* 1. **SCATTER MATRIX**

A **scatter matrix** consists of several pair-wise **scatter** plots of variables presented in a **matrix** format. It can be used to determine whether the variables are correlated and whether the correlation is positive or negative.

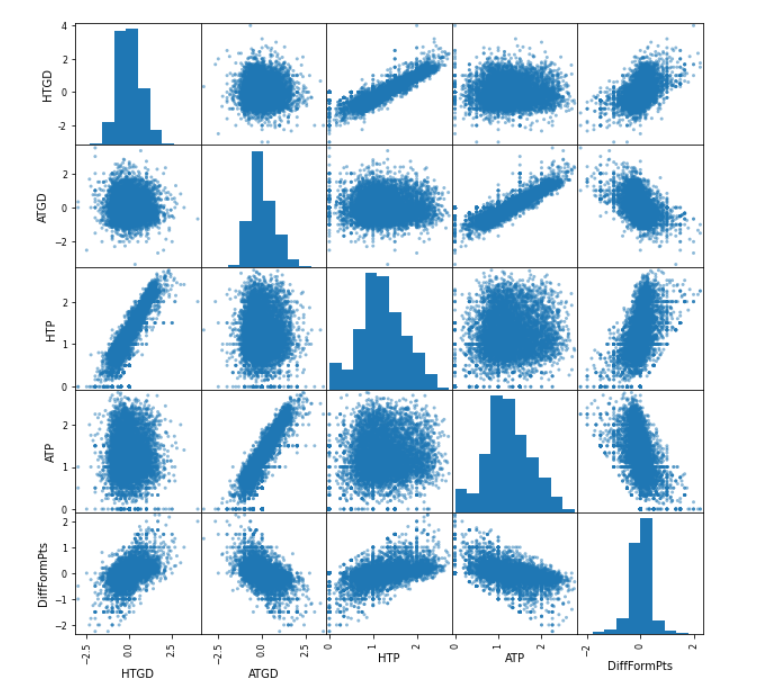
****

Fig15: Scatter Matrix

* 1. **PAIR PLOT**

**Pair Plots** are really simple way to visualize relationships between each variable. It produces a matrix of relationships between each variable in your data for an instant examination of our data. It can also be a great jumping off point for determining types of regression analysis to use.

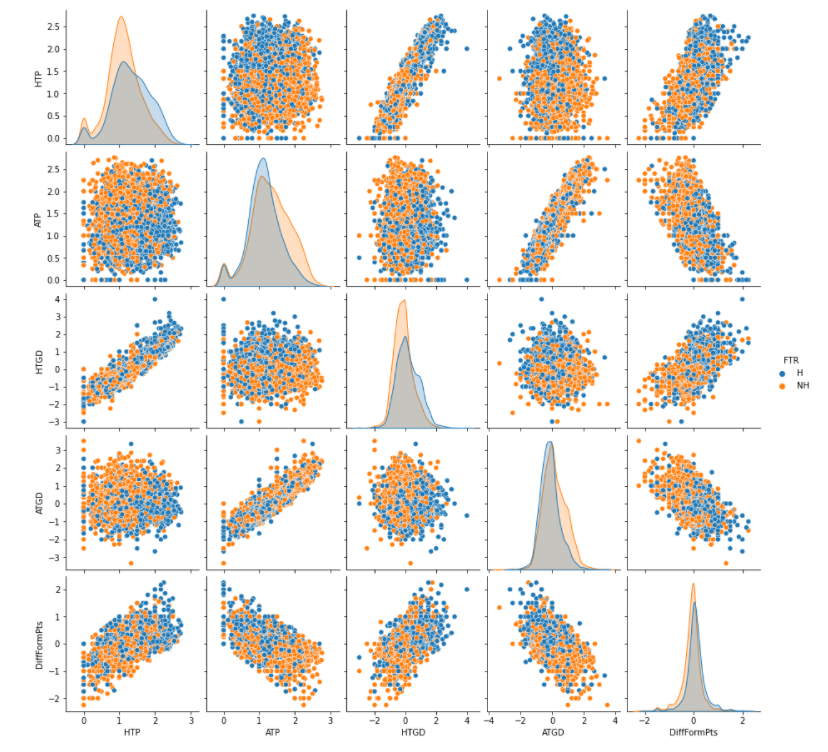
****

Fig16: Pair Plot

* 1. **OTHER VISUALIZATIONS**

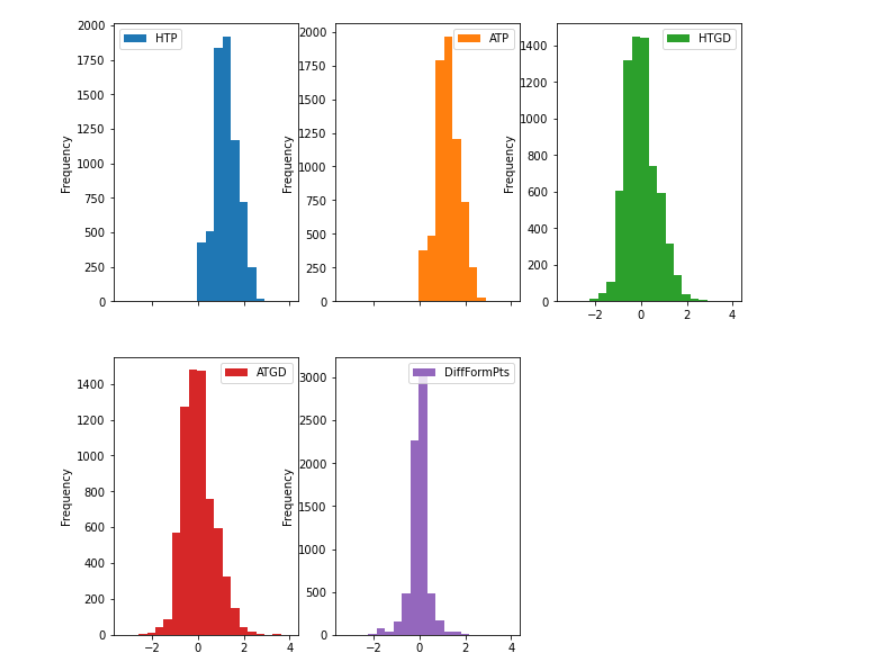
****

Fig17: Hist Plot

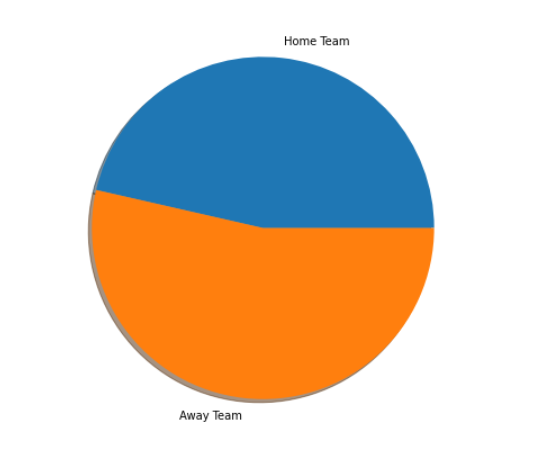
****

Fig18: Pie chart

1. **IMPLEMENTATION**
   1. **LOGISTIC REGRESSION ALGORITHM**

The steps involved in the building of the Logistic Regression model are:

Step 1: Datasets are collected from previous football matches of English Premier League.

Step 2: Data pre-processing is done on the datasets to use the data for building the model.

Step 2.1: finding the null values, if any replace them with mean, median or mode.

Step 2.2: Removing the duplicate tuples.

Step 2.3: Label Encoding, converting the string datatypes to Numbers.

Step 2.4: One Hot Encoding, Creating the Dummy Variables from the categorical variable

Step 3: Splitting the data into training and testing sets. (70%, 30%)

Step 4: Model Building

Step 4.1: from sklearn.linear\_model import LogisticRegression

Step 4.2: model=LogisticRegression(random\_state=42)

Step 4.3: y\_pred=model.predict(xtest)

Step 4.4: y\_pred == ytest

Step 4.5: model\_accuracy = numpy.mean(y\_pred == ytest)

Step 5: Evaluating the accuracy of the model. (model\_accuracy)

Step 6: Optimizing the model using the accuracy scores of different models.



Fig19: Logistic Regression Output

* 1. **SUPPORT VECTOR MACHINE ALGORITHM**

The steps involved in the building of the Support Vector Machine model are:

Step 1: Datasets are collected from previous football matches of English Premier League.

Step 2: Data pre-processing is done on the datasets to use the data for building the model.

Step 2.1: Finding the null values, replace them with mean, median or mode (if any)

Step 2.2: Removing the duplicate tuples.

Step 2.3: Label Encoding, converting the string datatypes to Numbers.

Step 2.4: One Hot Encoding, Creating the Dummy Variables from the categorical variable

Step 3: Splitting the data into training and testing sets. (70%, 30%)

Step 4: Model Building

Step 4.1: from sklearn.svm import SVC

classifier = SVC(kernel='rbf',random\_state = 1) //Radial basis function kernel

classifier.fit(xtrain,ytrain)

Step 4.2: y\_pre = classifier.predict(xtest)

Step 4.3: SVM\_accuracy=numpy.mean(y\_pre == ytest)

Step 5: Evaluating the accuracy of the model. (SVM\_accuracy)

Step 6: Optimizing the model using the accuracy scores of different models.

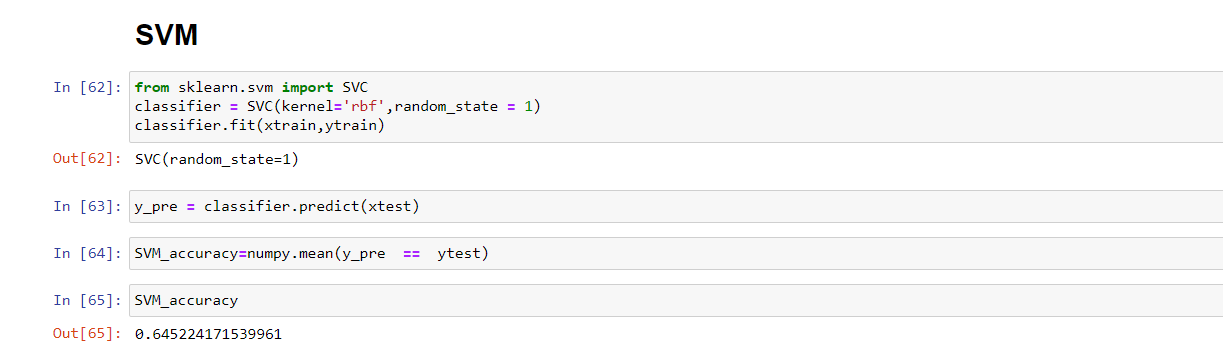


Fig20: SVM Output

* 1. **XG-BOOST ALGORITHM**

The steps involved in the building of the Support Vector Machine model are:

Step 1: Datasets are collected from previous football matches of English Premier League.

Step 2: Data pre-processing is done on the datasets to use the data for building the model.

Step 2.1: Finding the null values, replace them with mean, median or mode (if any)

Step 2.2: Removing the duplicate tuples.

Step 2.3: Label Encoding, converting the string datatypes to Numbers.

Step 2.4: One Hot Encoding, Creating the Dummy Variables from the categorical variable

Step 3: Splitting the data into training and testing sets. (70%, 30%)

Step 4: Model Building

Step 4.1**:** from xgboost import XGBClassifier

classifier1=XGBClassifier(seed=82)

classifier1.fit(xtrain,ytrain)

Step 4.2: ypred=classifier1.predict(xtest)

Step 4.3: XGboost\_accuracy=numpy.mean(y\_pre == ytest)

Step 5: Evaluating the accuracy of the model. (XGboost\_accuracy)

Step 6: Optimizing the model using the accuracy scores of different models.

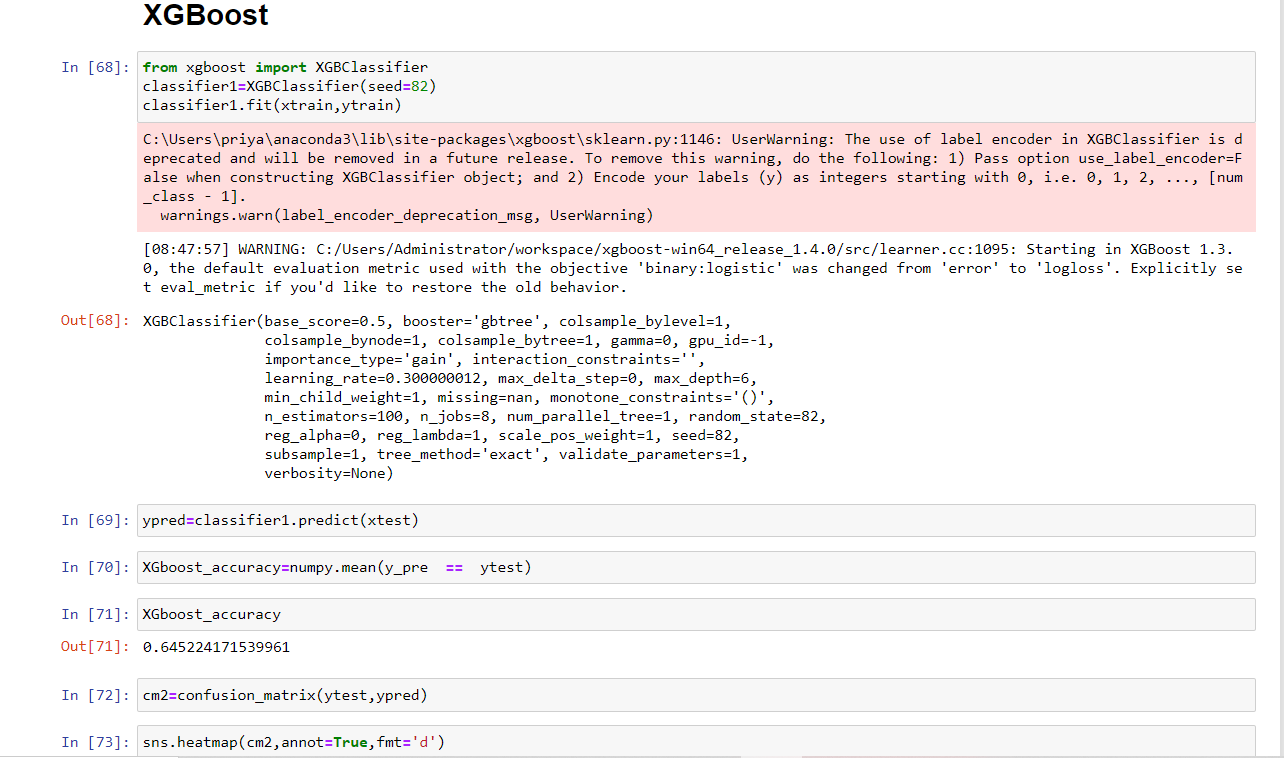
****

Fig21: XG-Boost Output

1. **RESULTS**

The model is built successfully and by using Logical Regression algorithm the accuracy is about 65.8%, by using SVM algorithm the accuracy is about 64.5% and by using XG-Boost algorithm the accuracy is about 69.75%.

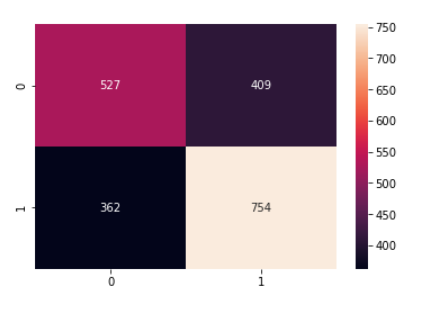
****

Fig 22: Heat map

**8. CONCLUSION AND FUTURE SCOPE**

The project is ready to predict the football match winner using the machine learning models by consideration of past datasets and the necessary changes are made to the team to move towards the winning. The model is able to make more fairly accurate predictions, although the model accuracy is good it’s not guaranteed that every time it will predict correctly. It can be further made more accurate by building the model using different parameters like Trending hashtags in twitter by players and fans, sentiment analysis, individual player performance in recent series, individual coach performance in recent series, by including these features in the model it will be more accurate and the model will able to predict well.

**9. REFERENCES**

[1] Hucaljuk, J & Rakipović, A. (2011). Predicting football scores using machine learning techniques. 2011 Proceedings of the 34th International Convention MIPRO,, 1623-1627.

[2] Abel Hijmans. Dutch football prediction using machine learning classifiers (unpublished)

[3] T. Cheng, D. Cui, Z. Fan, J. Zhou, and S. Lu, “A new model to forecast the results of matches based on hybrid neural networks in the soccer rating system,” Proceedings Fifth International Conference on

Computational Intelligence and Multimedia Applications. ICCIMA 2003.

[4] R.K. Balla, "Soccer Match Result Prediction using Neural Networks" in

[5] Farzin, O., Parinaz, E., &Faezeh, S. M. Football result prediction with Bayesian network in Spanish league-Barcelona team. International Journal of Computer Theory and Engineering, 5(5), 2013, 812-815

[6] Darwin, P & Dra, H (2016). Predicting Football Match Results with Logistic Regression. 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA).

[7] G. Baio and M. Blangiardo. "Bayesian hierarchical model for the prediction of football results." University College London Department of Statistical Sciences, Gower Street, London WC1 6BT

[8] N. Tax and Y. Jousts. "Predicting The Dutch Football Competition Using Public Data: A Machine Learning Approwh."

[9] Baboota, Rahul & Kaur, Harleen. (2018). Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting. 35. 10.1016/j.ijforecast.2018.01.003.

[10] D. Prasetio and D. Harlili, "Predicting football match results with logistic regression," 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), George Town, 2016, pp. 1-5.