VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI, KARNATAKA-590018

Internship Report on

"Data Science"

Submitted in fulfilment for the award of the degree of
Bachelor of Engineering in

Information Science and EngineeringSubmitted By

KUSHAL N [1JT18IS030]

Internship carried out at

Tequed Labs Private Limited

3, 1st Main Rd, Ittamadu, Banashankari 3rd Stage, Banashankari,

Bengaluru, Karnataka 560085

Department of Information Science and Engineering

JYOTHY INSTITUTE OF TECHNOLOGY
Department of Information Science and Engineering
Accredited by NBA, New Delhi
Tataguni,Bengaluru -560082

JYOTHY INSTITUTE OF TECHNOLOGY
Department of Information Science and Engineering
Accredited by NBA, New Delhi
Tataguni,Bengaluru -560082

CERTIFICATE

This is to certify that the Internship titled '**Data science**' carried out by **Mr. KUSHAL N,** a bonafide student of Jyothy Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering**, in **Information Science and Engineering** under Visvesvaraya Technological University, Belagavi, during the year 2021-2022. It is certified that all corrections/suggestions indicated have been incorporated in the report.

The project report has been approved as it satisfies the academic requirements in respect of Internship prescribed for the course Internship / Professional Practice (18CSI85)

Signature of Guide	**Signature of HOD**	**Signature of Principal**
Dr. Kishore GR	**Dr. Harshvardhan Tiwari**	**Dr. K. Gopalkrishna**
Dept. Of ISE Jyothy Institute of Technology	Dept. Of ISE Jyothy Institute of Technology	Jyothy Institute Of Technology

External Viva:

Name of the Examiner Signature with Date

1)_________________ _________________

2)_________________ _________________

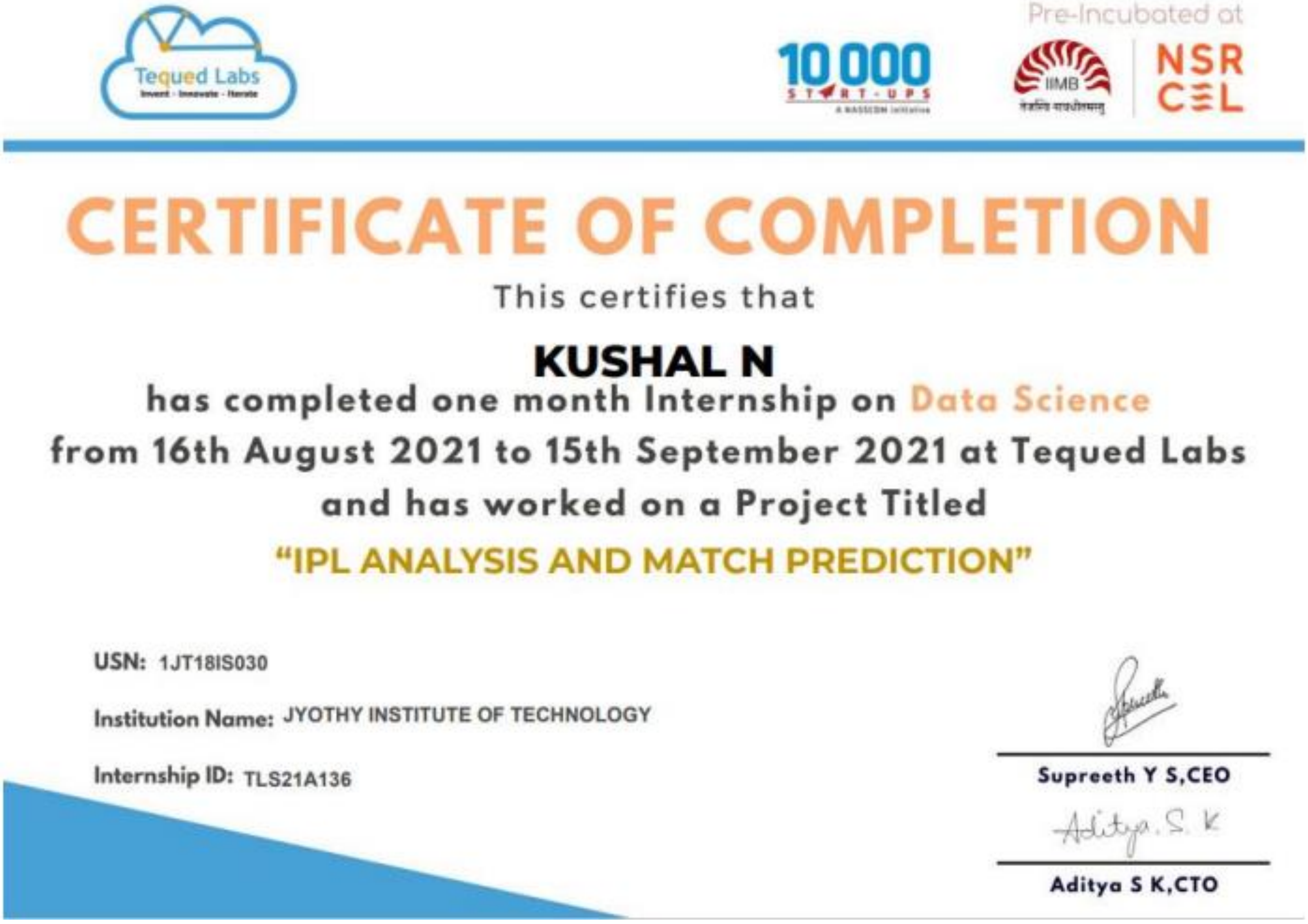
DECLARATION

I, **Kushal N**, final year student of Information Science and Engineering, Jyothy Institute of Technology, Bangalore - 560 082, declare that the Internship has been successfully completed, in **TEQUED LABS**. This report is submitted in partial fulfillment of the requirements for award of Bachelor Degree in Information Science and Engineering, during the academic year 2021-2022.

Date:
Place: Bangalore

USN: 1JT18IS030
NAME: KUSHAL N

CERTIFICATE



Tequed Labs
Invent - Innovate - Iterate

10,000 START-UPS
A NASSCOM initiative

Pre-Incubated at
IIMB NSRCEL

CERTIFICATE OF COMPLETION

This certifies that

KUSHAL N

has completed one month Internship on Data Science from 16th August 2021 to 15th September 2021 at Tequed Labs and has worked on a Project Titled

"IPL ANALYSIS AND MATCH PREDICTION"

USN: 1JT18IS030

Institution Name: JYOTHY INSTITUTE OF TECHNOLOGY

Internship ID: TLS21A136

Supreeth Y S,CEO

Aditya. S. K

Aditya S K,CTO

ACKNOWLEGDEMENT

This Internship is a result of accumulated guidance, direction and support of several important persons. We take this opportunity to express our gratitude to all who have helped us to complete the Internship.

We express our sincere thanks to our Principal **Dr K. Gopalakrishna**, for providing us adequate facilities to undertake this Internship.

We would like to thank **Dr. Harshvardhan Tiwari**, Head of Dept - ISE, for providing us an opportunity to carry out Internship and for his valuable guidance and support.

We would like to thank **Mr. Supreeth Y, Tequed Lab, CE**, for guiding us during the period of internship.

We express our deep and profound gratitude to our guide **Dr. Kishore GR,** Associate Professor- ISE, for his keen interest and encouragement atevery step in completing the Internship.

We would like to thank all the faculty members of ISE department for the support extended during the course of Internship.

We would like to thank the non-teaching members of the Dept of ISE, forhelping us during the Internship.

Last but not the least, we would like to thank our parents and friends without whose constant help, the completion of Internship would have notbeen possible.

KUSHAL N [1JT18IS030]

ABSTRACT

Data mining tools predict the future trends and behaviours which gives an opportunity to predict the outcome of IPL (Indian Premier League) match using data mining algorithms. Data mining algorithms have been applied to IPL dataset and knowledge from each algorithm has been obtained and analysed thoroughly as the results are obtained with good accuracy performance.

A way of predicting the outcome of the matches between various teams can aid in the team selection process. The result has been predicted using the algorithm approaches and have analysed the results of the IPL match using the above approaches. Here we have considered toss winning result and accurate results of previous matches. Thus, we measure the outcome of the IPL upcoming matches using the data-mining algorithms.

Keywords: Data Mining, IPL, Algorithms, Accuracy.

TABLE OF CONTENTS

=

CHAPTER 1

COMPANY PROFILE

1.1 Introduction

Tequed Labs Private Limited is a private company incorporated on 22 January 2018. It is classified as a non-government company and is registered at Registrar of Companies, Bangalore. Tequed Labs is a research and development center and educational institute based in Bangalore.

They are focused on providing quality education on latest technologies and develop products which are of great need to the society. They also involve in distribution and sales of latest electronic innovation products developed all over the globe to their customers.

This work was awarded state's best innovation in IOT domain. This project was the world finalist in the international innovation challenge called MASTERPIECE in Dubai. It has been exhibited in NASSCOM Product Conclave and has received great appreciation from IT giants. This product has been patented bearing a patent number - 201741034208.
Their other research work includes development of a device for blind which can recognize objects and convert it into speech. This innovation has a lot of potential in helping the blind people. Their other products include:

- Automation of production line and remote quality control monitoring system.
- Development of mobile app and website for sales of artistic and antique products.
- Development of an energy conservation system for paper machineries.
- Development of an analytic tool for software-based vehicle condition analysis forresales.

Vision

To be a world-class research and development organization committed to enhancing stakeholder’s value

Mission

To build best products which are socially innovative with high-quality attributes and provide excellent education to all.

Values

- Zeal to excel and zest for change.
- Integrity and fairness in all matters.
- Respect for dignity and potential of individuals
- Strict adherence to commitments.
- Ensure speed of response.
- Faster learning, creativity and team-work.
- Loyalty and pride in the company.

Quality Policy

In the quest to be world-class, TEQUED LABS pursues continual improvement in the quality of its products, services and performance leading to total customer satisfaction and business growth through dedication, commitment and team work of all employees.

Development Sectors of TEQUED LABS

- Software Development
- Embedded System Design
- Application Development
- IOT based home automation System
- Educational Services
- Product Research

CHAPTER 2

ABOUT THE COMPANY

Tequed Labs Private Limited

Tequed Labs Private Limited is a private company incorporated on 22 January 2018. It is classified as a non-government company and is registered at Registrar of Companies, Bangalore. Tequed Labs is a research and development centre and educational institute based in Bangalore.

They are focused on providing quality education on latest technologies and develop products which are of great need to the society. They also involve in distribution and sales of latest electronic innovation products developed all over the globe to their customers.

Development Sectors of TEQUED LABS

- Software Development
- Embedded System Design
- Application Development
- IOT based home automation System
- Educational Services
- Product Research

CHAPTER 3

INTRODUCTION

3.1 Data Science

Data Science is a multidisciplinary approach to extracting actionable insights from the large and ever-increasing volumes of data collected and created by today’s organizations. Data science encompasses preparing data for analysis and processing, performing advanced data analysis, and presenting the results to reveal patterns and enable stakeholders to draw informed conclusions.

Data preparation can involve cleansing, aggregating, and manipulating it to be ready for specific types of processing. Analysis requires the development and use of algorithms, analytics and AI models. It’s driven by software that combs through data to find patterns within to transform these patterns into predictions that support business decision-making. The accuracy of these predictions must be validated through scientifically designed tests and experiments. And the results should be shared through the skilful use of data visualization tools that make it possible for anyone to see the patterns and understand trends.

3.2 Python

Python is open source, interpreted, high level language and provides great approach for object-oriented programming. It is one of the best language used by data scientist for various data science projects/application. Python provide great functionality to deal with mathematics, statistics and scientific function. It provides great libraries to deals with data science application.

In terms of application areas, Data and ML scientists prefer Python as well. When it comes to areas like building fraud detection algorithms and network security, developers leaned towards Java, while for applications like data science, natural language processing (NLP) and sentiment analysis, developers opted for Python, because it provides large collection of libraries that help to solve complex business problem easily, build strong system and data application.

3.3 Most Commonly used python libraries for data science:

Numpy: Numpy is Python library that provides mathematical function to handle large dimension array. It provides various method/function for Array, Metrics, and linear algebra. NumPy stands for Numerical Python. It provides lots of useful features for operations on n-arrays and matrices in Python. The library provides vectorization of mathematical operations on the NumPy array type, which enhance performance and speeds up the execution. It’s very easy to work with large multidimensional arrays and matrices using NumPy.

Pandas: Pandas is one of the most popular Python library for data manipulation and analysis. Pandas provide useful functions to manipulate large amount of structured data. Pandas provide easiest method to perform analysis.

Matplotlib: Matplotlib is another useful Python library for Data Visualization. Descriptive analysis and visualizing data is very important for any organization. Matplotlib provides various method to Visualize data in more effective way. Matplotlib allows to quickly make line graphs, pie charts, histograms, and other professional grade figures.

Scipy: Scipy is another popular Python library for data science and scientific computing. Scipy provides great functionality to scientific mathematics and computing programming. SciPy contains sub-modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers, Statmodel and other tasks common in science and engineering.

Scikit – learn: Sklearn is Python library for machine learning. Sklearn provides various algorithms and functions that are used in machine learning. Sklearn is built on NumPy, SciPy, and matplotlib. Sklearn provides easy and simple tools for data mining and data analysis. It provides a set of common machine learning algorithms to users through a consistent interface. Scikit-Learn helps to quickly implement popular algorithms on datasets and solve real-world problems.

CHAPTER 4

HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements

- Operating System: Windows 7/8/10
- Minimum of 4GB RAM (8GB Recommended)
- 1.2 GHz and above processor speed
- Minimum of 500 GB HDD /256 SSD

4.2 Software Requirements

- Python 3.0 and above
- Jupyter Notebook
- Matplotlib, Seaborn, NumPy, Pandas and other python libraries required

CHAPTER 5

DESIGN AND ANALYSIS

5.1 Product /Project name:

IPL Analysis and match prediction

5.2 Product features:

To predict the winner of the IPL match

5.3 Users:

Cricket analysts, experts, cricket teams and staff members

5.4 Deadline:

To be completed within the internship completion date they mentioned i.e., within the last week of internship

5.5 Analysis:

To Build a predictive model to predict the team with the winning chance between two teams of upcoming matches based on toss decision and winner result of all previous dataset and calculate the accuracy.

The dataset used in this project was downloaded from Kaggle. From this data archive we will be using two data files for matches and batting analysis.

Various tasks needed to be implemented are:

1. Data Acquisition and Cleaning
2. Data Visualization
3. Data Modelling
4. Testing
5. Measurement and Comparison

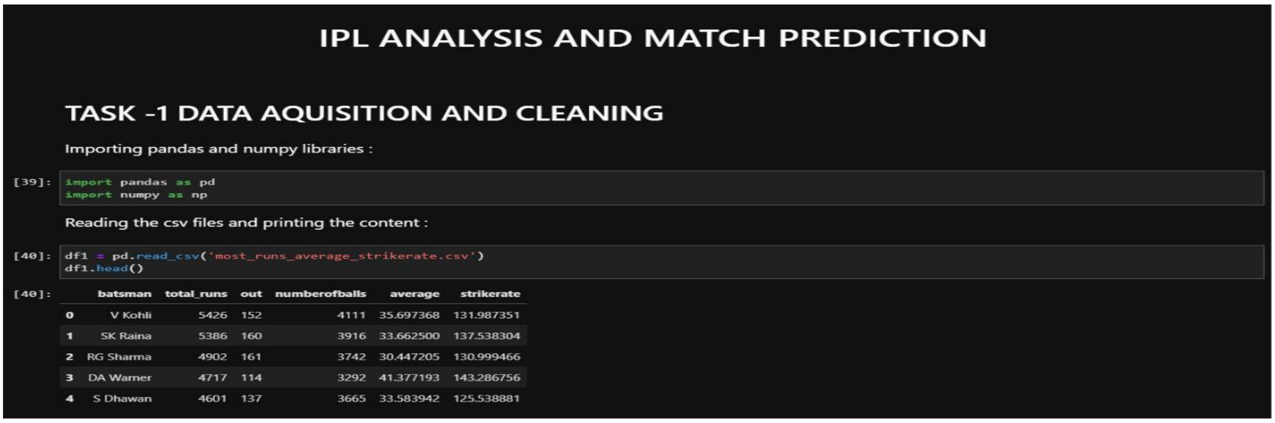
CHAPTER 6

IMPLEMENTATION

6.1 Data Acquisition and Cleaning

Data Acquisition is to collect data from relevant sources before it can be stored, cleaned, preprocessed and used for further analysis.

Data Cleaning is the process of detecting and removing corrupt or inaccurate records from record set, table or database.



IPL ANALYSIS AND MATCH PREDICTION

TASK -1 DATA AQUISITION AND CLEANING

Importing pandas and numpy libraries :

```
[39]: import pandas as pd
      import numpy as np
```

Reading the csv files and printing the content :

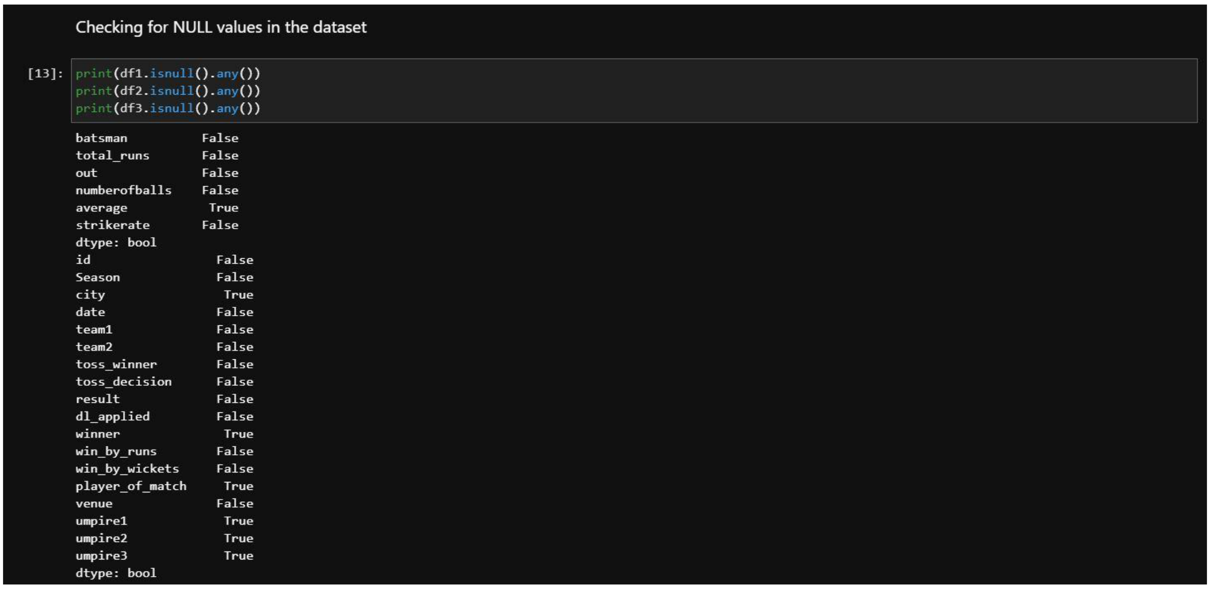
```
[40]: df1 = pd.read_csv('most_runs_average_strikerate.csv')
      df1.head()
```

[40]:

	batsman	total_runs	out	numberofballs	average	strikerate
0	V Kohli	5426	152	4111	35.697368	131.987351
1	SK Raina	5386	160	3916	33.662500	137.538304
2	RG Sharma	4902	161	3742	30.447205	130.999466
3	DA Warner	4717	114	3292	41.377193	143.286756
4	S Dhawan	4601	137	3665	33.583942	125.538881

Fig 6.1: Importing python Libraries and reading the csv files:

In the similar way we read other csv files required for the analysis



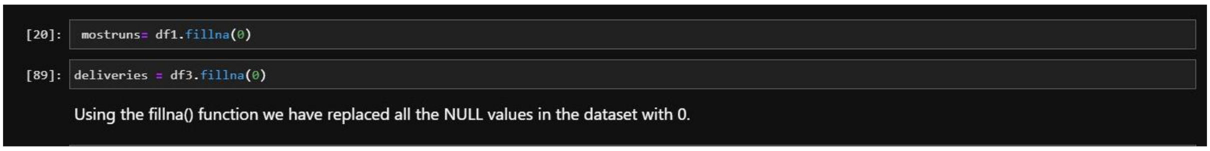
Checking for NULL values in the dataset

```
[13]: print(df1.isnull().any())
      print(df2.isnull().any())
      print(df3.isnull().any())
```

```
batsman           False
total_runs        False
out               False
numberofballs     False
average            True
strikerate        False
dtype: bool
id                  False
Season              False
city                 True
date                False
team1               False
team2               False
toss_winner         False
toss_decision       False
result              False
dl_applied          False
winner               True
win_by_runs         False
win_by_wickets      False
player_of_match      True
venue               False
umpire1              True
umpire2              True
umpire3              True
dtype: bool
```

Fig 6.2: **Checking the dataset for NULL values**

We can see that the dataset consists of NULL values in few of its columns. Values of dataset can not be NULL. So, either we have an option to delete the rows with NULL values or fill the NULL values with any other value of our choice.



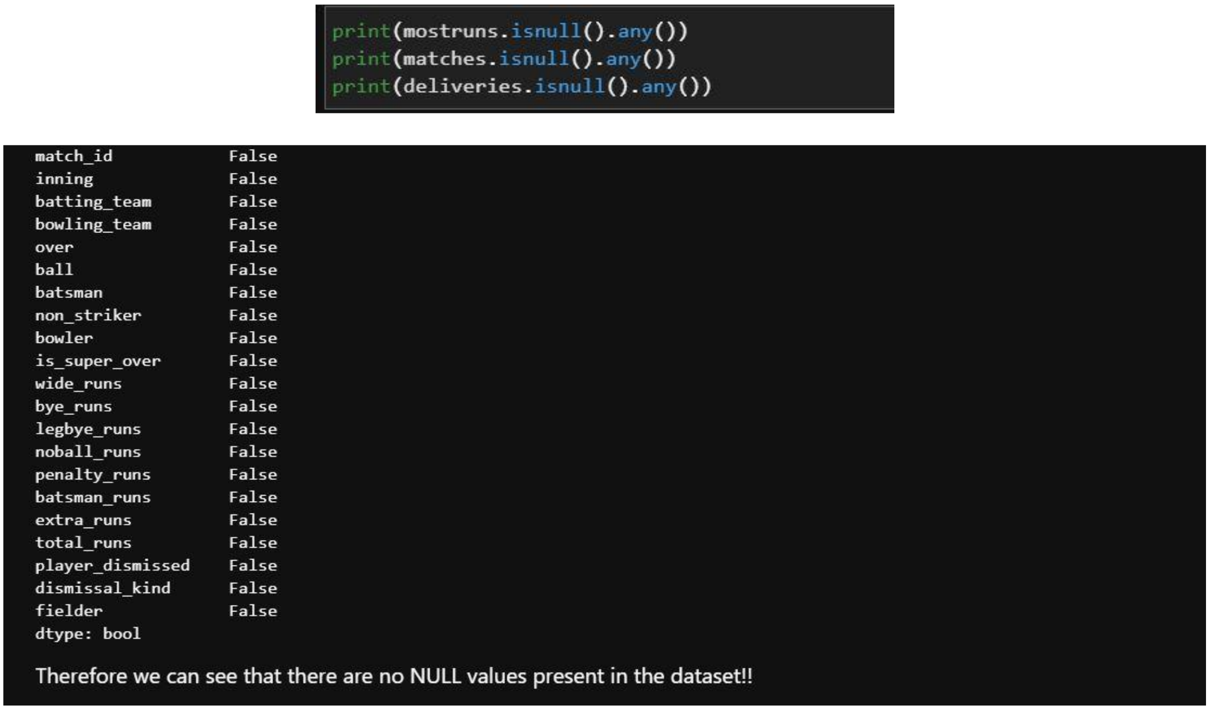
```
[20]:  mostruns= df1.fillna(0)

[89]:  deliveries = df3.fillna(0)
```

Using the fillna() function we have replaced all the NULL values in the dataset with 0.

Fig 6.3. Filling the NULL values

We filled the NULL values to 0 instead of deleting the rows because deletion of rows can lead to deletion of important data for or visualization and the final analysis would deviate a lot. Filling values with 0 is better as in cricket players can have scores, stats equal to 0.



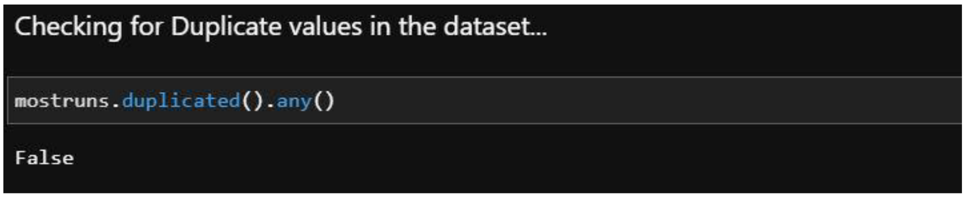
```
print(mostruns.isnull().any())
print(matches.isnull().any())
print(deliveries.isnull().any())
```

```
match_id            False
inning              False
batting_team        False
bowling_team        False
over                False
ball                False
batsman             False
non_striker         False
bowler              False
is_super_over       False
wide_runs           False
bye_runs            False
legbye_runs         False
noball_runs         False
penalty_runs        False
batsman_runs        False
extra_runs          False
total_runs          False
player_dismissed    False
dismissal_kind      False
fielder             False
dtype: bool
```

Therefore we can see that there are no NULL values present in the dataset!!

Fig 6.4: Checking if NULL values are removed successfully

We get the Boolean value ‘False’ for all columns which means all the NULL values have been removed from dataset.



Checking for Duplicate values in the dataset...

```
mostruns.duplicated().any()
```

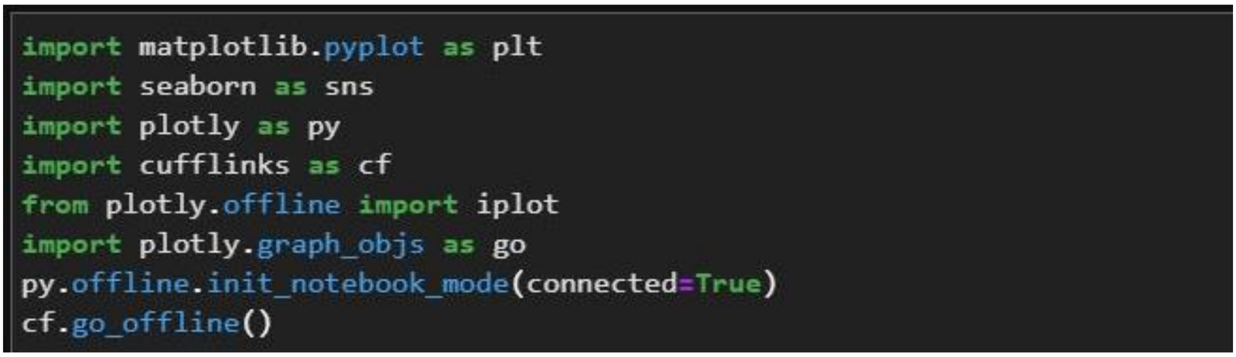
```
False
```

Fig 6.5: Checking for duplicate entries in dataset

We can see that we get Boolean value ‘False’ for checking duplicate values which means there are no duplicate values present in the dataset. Same method is used for every other dataset files. With this, Data Acquisition and Cleaning has been successfully completed.

6.2 Data Visualization

Data Visualization checks the dataset structure, looks for possible problems in data and provides clear understanding of data. The information acquired in this section maybe useful for data modelling.



```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly as py
import cufflinks as cf
from plotly.offline import iplot
import plotly.graph_objs as go
py.offline.init_notebook_mode(connected=True)
cf.go_offline()
```

Fig 6.6: Importing python Libraries

Overall Teams Data Visualization

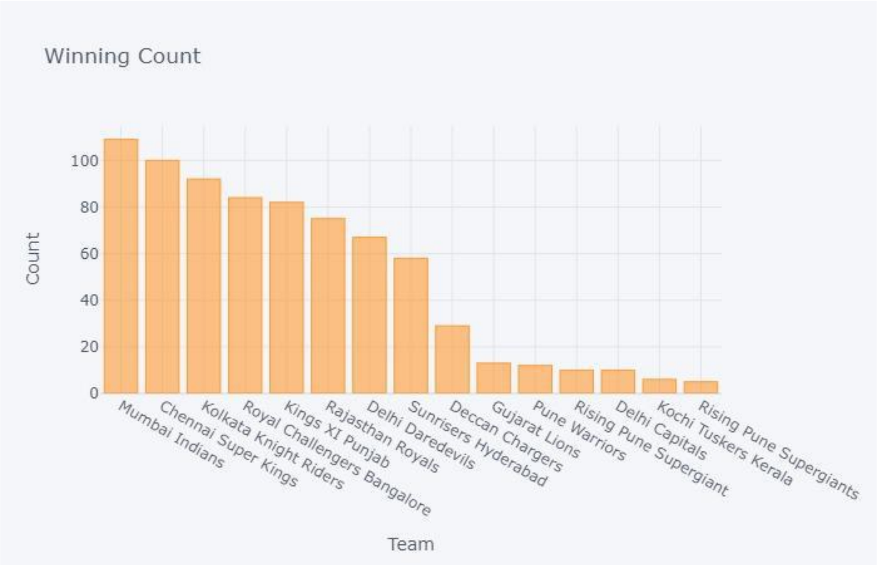
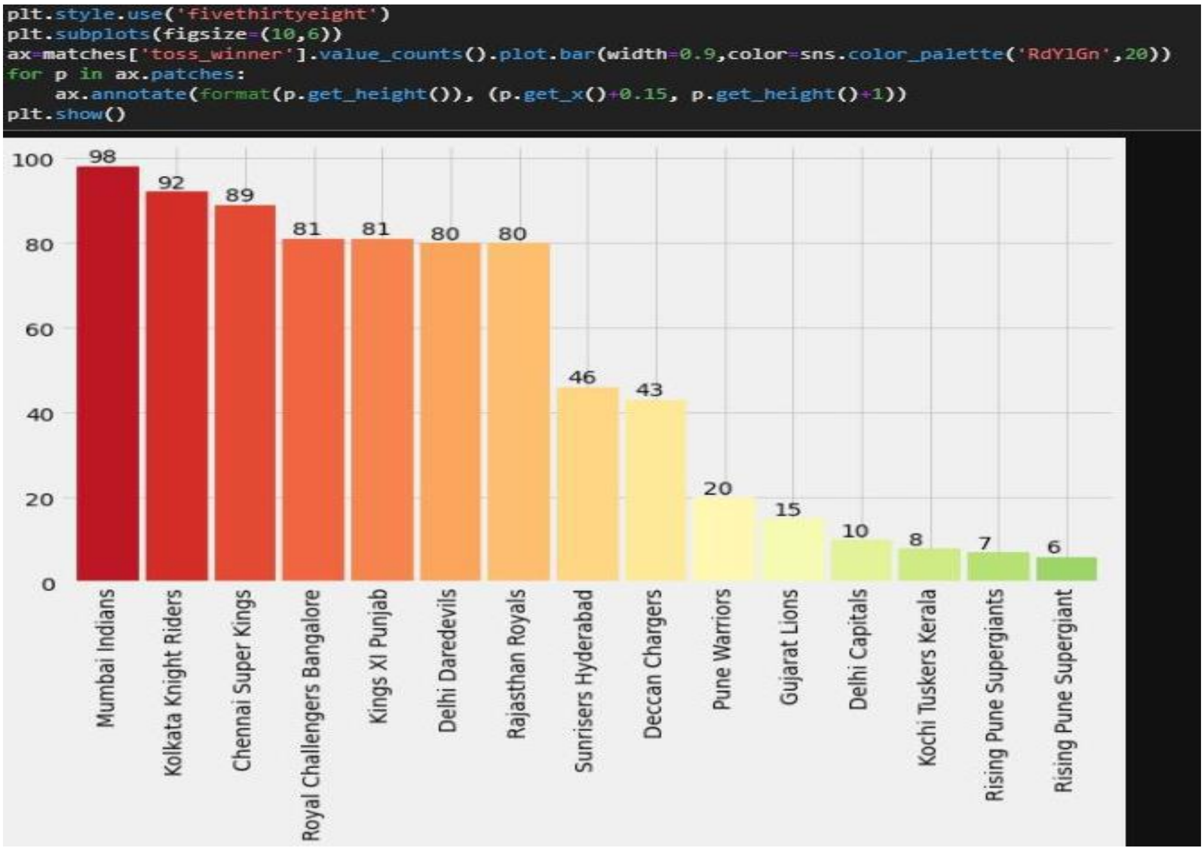


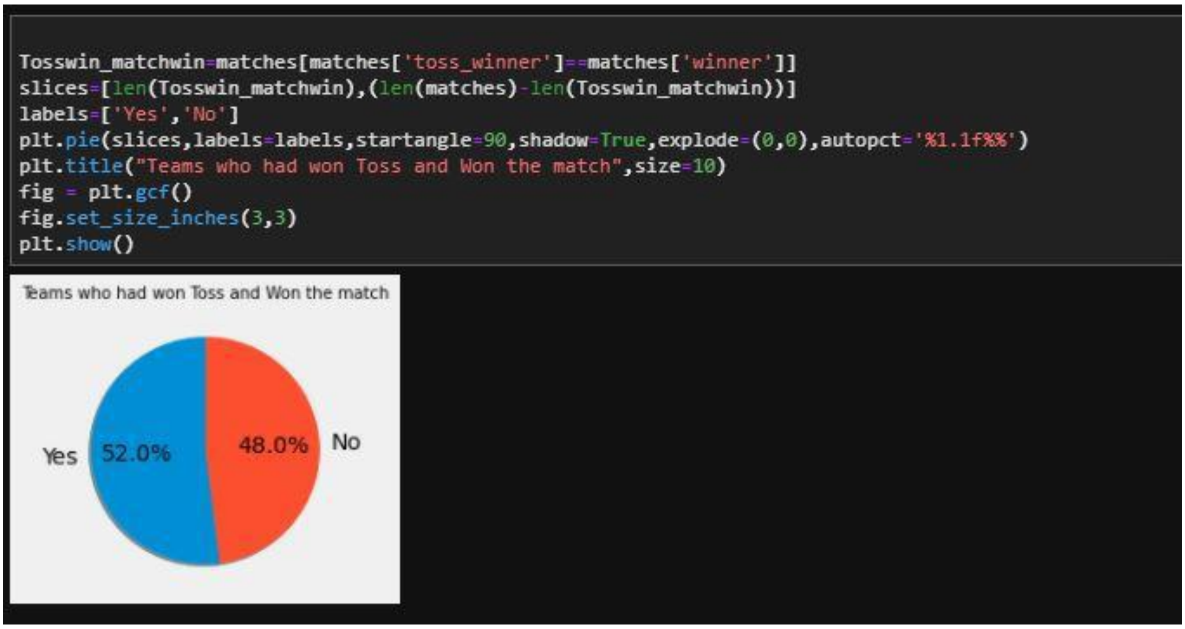
Fig 6.7: Visualizing teams based on winning



```
plt.style.use('fivethirtyeight')
plt.subplots(figsize=(10,6))
ax=matches['toss_winner'].value_counts().plot.bar(width=0.9,color=sns.color_palette('RdYlGn',20))
for p in ax.patches:
    ax.annotate(format(p.get_height()), (p.get_x()+0.15, p.get_height()+1))
plt.show()
```

Fig 6.8: Visualizing Toss Winners

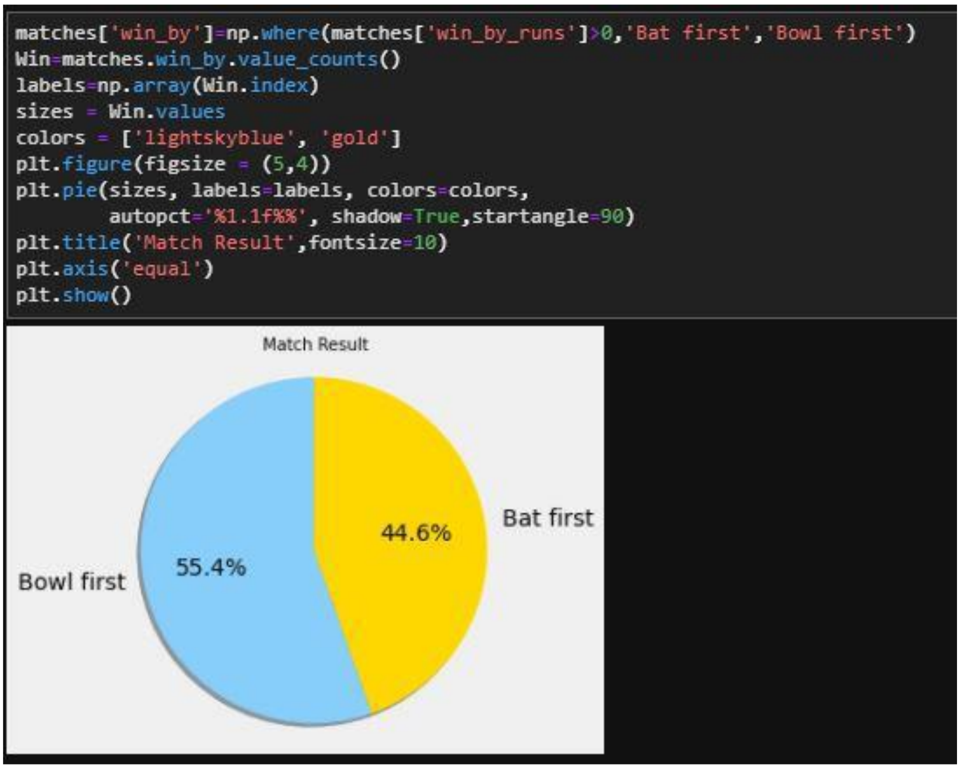
Visualizing how many percent matches have toss winners turned out to be the match winners:



```
Tosswin_matchwin=matches[matches['toss_winner']==matches['winner']]
slices=[len(Tosswin_matchwin),(len(matches)-len(Tosswin_matchwin))]
labels=['Yes','No']
plt.pie(slices,labels=labels,startangle=90,shadow=True,explode=(0,0),autopct='%1.1f%%')
plt.title("Teams who had won Toss and Won the match",size=10)
fig = plt.gcf()
fig.set_size_inches(3,3)
plt.show()
```

Fig 6.9: toss winners turned out to be the match winners

Visualizing the percentage of match winners based on bowling or batting first:



```
matches['win_by']=np.where(matches['win_by_runs']>0,'Bat first','Bowl first')
Win=matches.win_by.value_counts()
labels=np.array(Win.index)
sizes = Win.values
colors = ['lightskyblue', 'gold']
plt.figure(figsize = (5,4))
plt.pie(sizes, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True,startangle=90)
plt.title('Match Result',fontsize=10)
plt.axis('equal')
plt.show()
```

Fig 6.10: match winners based on bowling or batting first

Visualization of overall all teams winning against matches played and win percent:

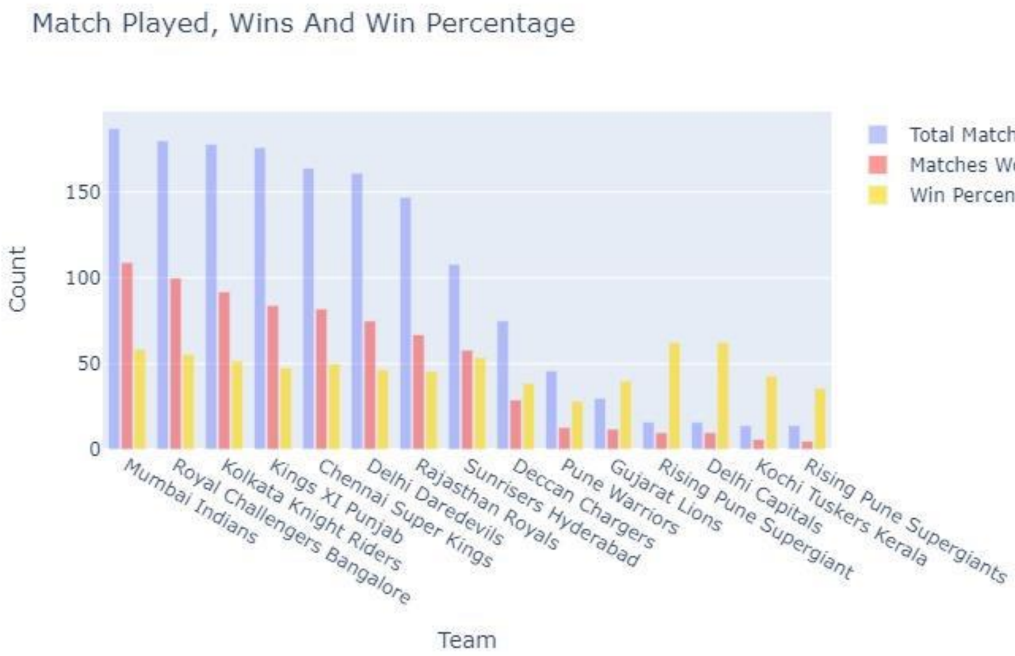


Fig 6.11: overall all teams winning

Visualizing how Royal Challengers Bangalore has done over the years:

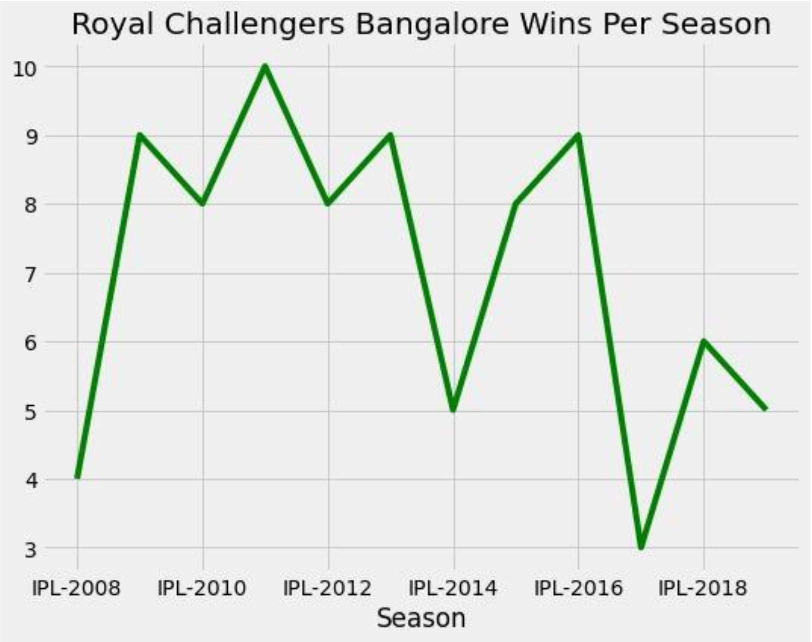
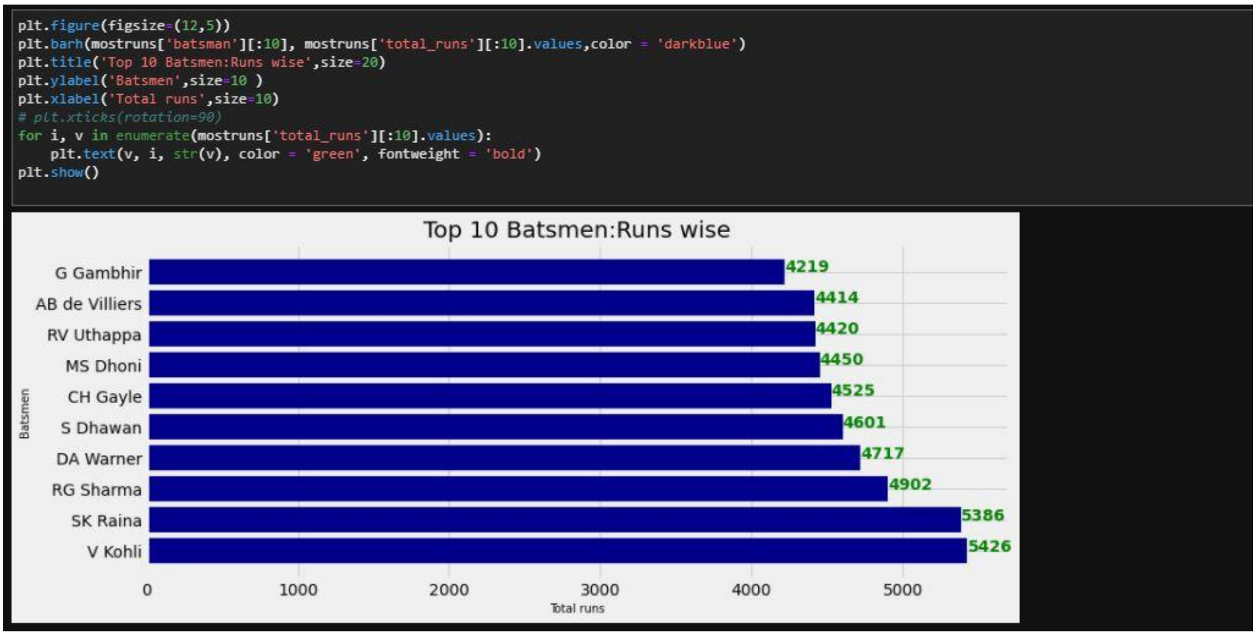


Fig 6.12: Royal Challengers Bangalore Performance

Batsmen Data Visualization:



```
plt.figure(figsize=(12,5))
plt.barh(mostruns['batsman'][:10], mostruns['total_runs'][:10].values,color = 'darkblue')
plt.title('Top 10 Batsmen:Runs wise',size=20)
plt.ylabel('Batsmen',size=10 )
plt.xlabel('Total runs',size=10)
# plt.xticks(rotation=90)
for i, v in enumerate(mostruns['total_runs'][:10].values):
    plt.text(v, i, str(v), color = 'green', fontweight = 'bold')
plt.show()
```

Fig 6.13: Visualizing the top 10 runs scorers in IPL

Average of Batsmen based on number of balls faced:

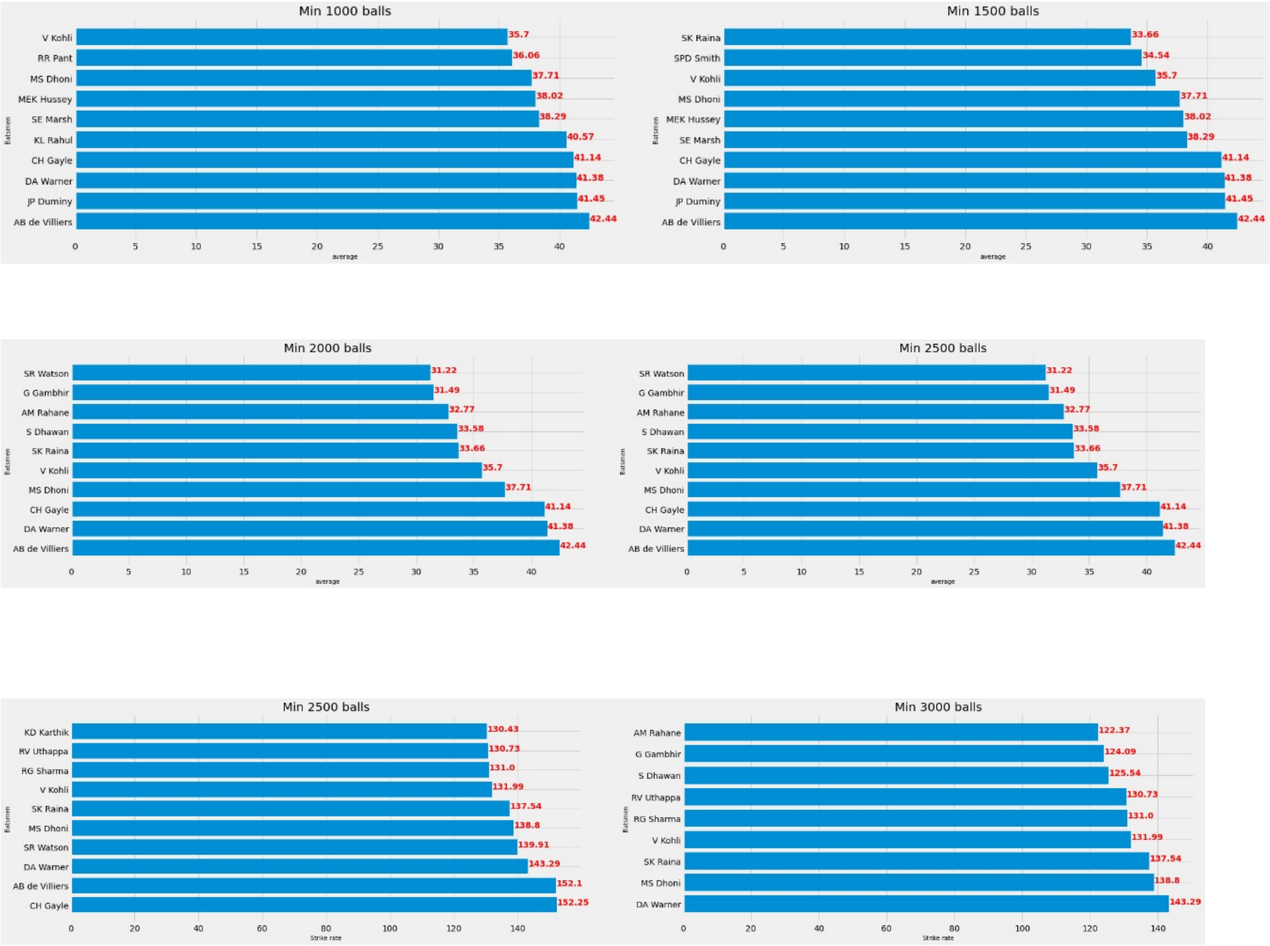


Fig 6.14: Average of Batsmen

Bowlers Data Visualization:

Visualization of highest wicket taking bowlers:

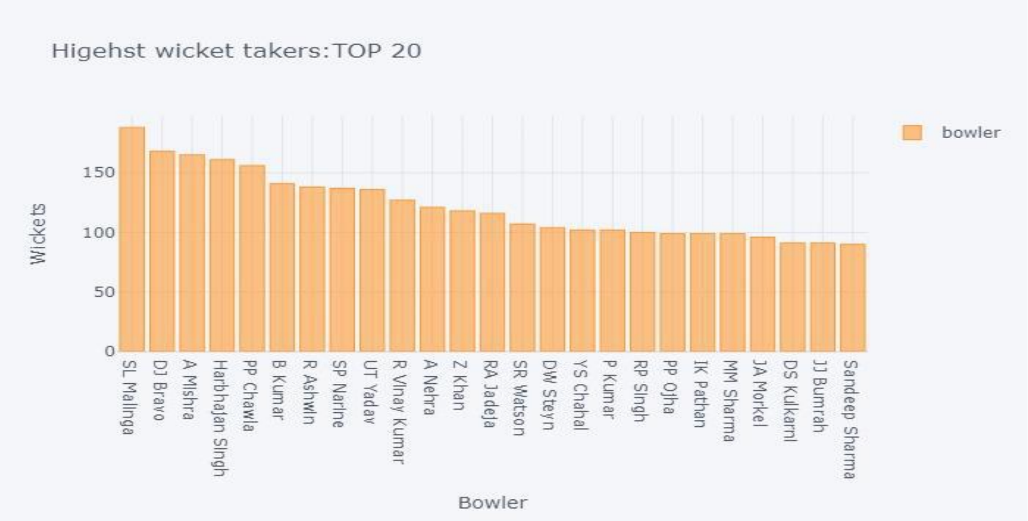


Fig 6.15: highest wicket Bowler

Visualizing the most economical bowlers:

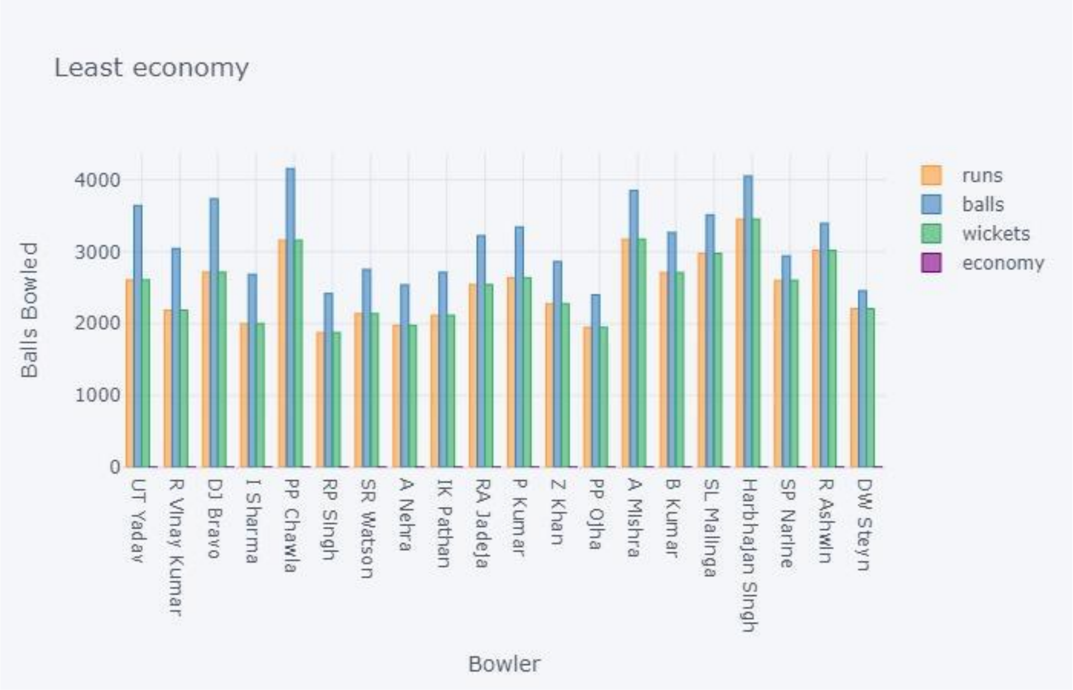


Fig 6.16: Economical bowlers

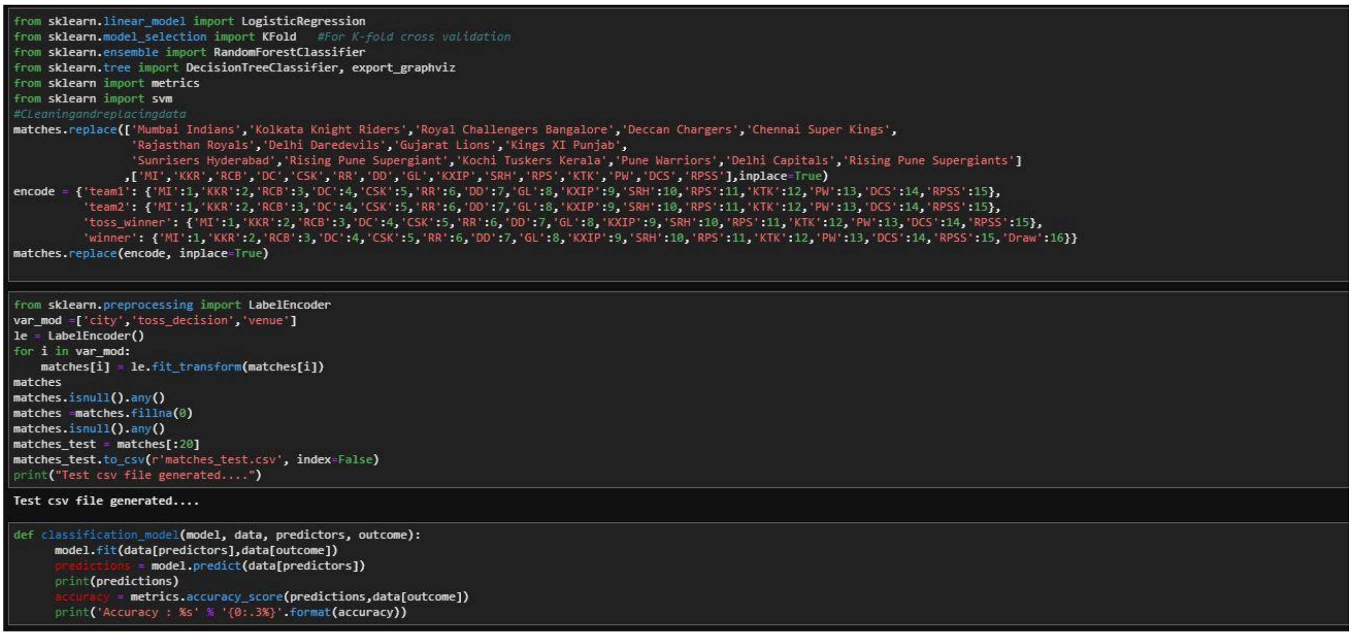
6.3 Data Modelling

Data Modelling is used to ensure efficient use of as a blueprint for construction of new software or for reengineering. Using Data Modelling we can make use of various analytic algorithms for predicting the accuracy of various events based on its previous performance.

There are various algorithms under Data Modelling which can be used to determine the accuracy, but before implementing algorithms we have to replace and label the data which is called as Label Encoding.

Label Encoding:

Label encoding refers to converting the labels into a numeric form so as to convert them into a Machine-readable form, In Label encoding we replace the Categorical value with a numeric value.



```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import KFold   #For K-fold cross validation
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn import metrics
from sklearn import svm
#Cleaningandreplacingdata
matches.replace(['Mumbai Indians','Kolkata Knight Riders','Royal Challengers Bangalore','Deccan Chargers','Chennai Super Kings',
                 'Rajasthan Royals','Delhi Daredevils','Gujarat Lions','Kings XI Punjab',
                 'Sunrisers Hyderabad','Rising Pune Supergiant','Kochi Tuskers Kerala','Pune Warriors','Delhi Capitals','Rising Pune Supergiants']
                ,['MI','KKR','RCB','DC','CSK','RR','DD','GL','KXIP','SRH','RPS','KTK','PW','DCS','RPSS'],inplace=True)
encode = {'team1': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13,'DCS':14,'RPSS':15},
          'team2': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13,'DCS':14,'RPSS':15},
          'toss_winner': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13,'DCS':14,'RPSS':15},
          'winner': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13,'DCS':14,'RPSS':15,'Draw':16}}
matches.replace(encode, inplace=True)
```

```
from sklearn.preprocessing import LabelEncoder
var_mod =['city','toss_decision','venue']
le = LabelEncoder()
for i in var_mod:
    matches[i] = le.fit_transform(matches[i])
matches
matches.isnull().any()
matches =matches.fillna(0)
matches.isnull().any()
matches_test = matches[:20]
matches_test.to_csv(r'matches_test.csv', index=False)
print("Test csv file generated....")
```

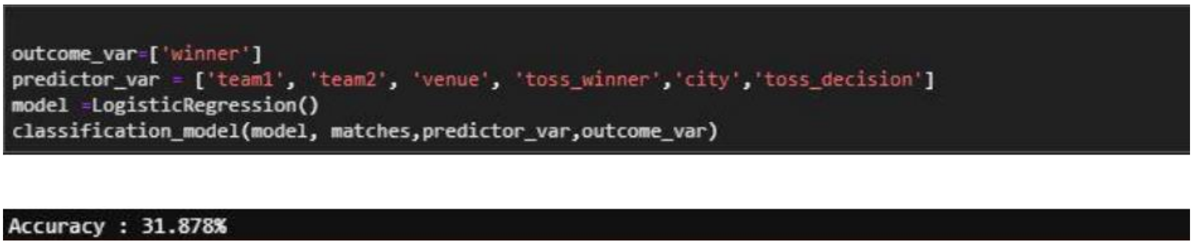
```
Test csv file generated....
```

```
def classification_model(model, data, predictors, outcome):
      model.fit(data[predictors],data[outcome])
      predictions = model.predict(data[predictors])
      print(predictions)
      accuracy = metrics.accuracy_score(predictions,data[outcome])
      print('Accuracy : %s' % '{0:.3%}'.format(accuracy))
```

Fig 6.17: Classification model

6.3.1 Logistic Regression

It is a statistical analysis method used to predict a data value based on prior observations of a data set. The approach allows an algorithm being used in a machine learning application to classify incoming data based on historical data. It is used for binary classification problems i.e, problems with two class values.



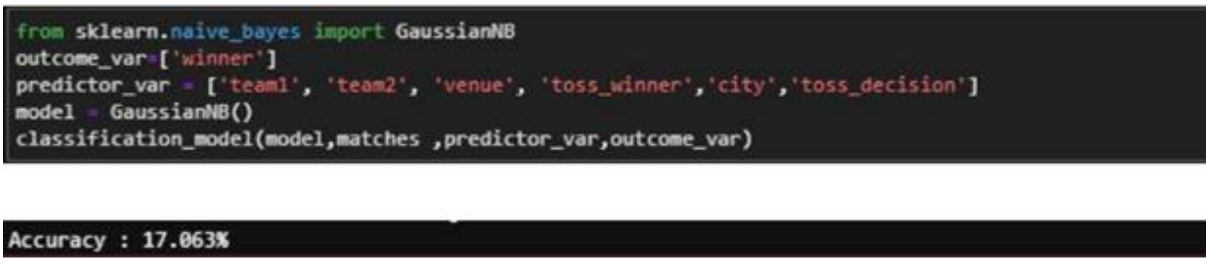
```
outcome_var=['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decision']
model =LogisticRegression()
classification_model(model, matches,predictor_var,outcome_var)
```

```
Accuracy : 31.878%
```

Fig 6.18: Logistic Regression

6.3.2 Gaussian Naive Bayes Algorithm

A Gaussian Naive Bayes algorithm is a special type of NB algorithm. It's specifically used when the features have continuous values. It's also assumed that all the features are following a gaussian distribution i.e, normal distribution.



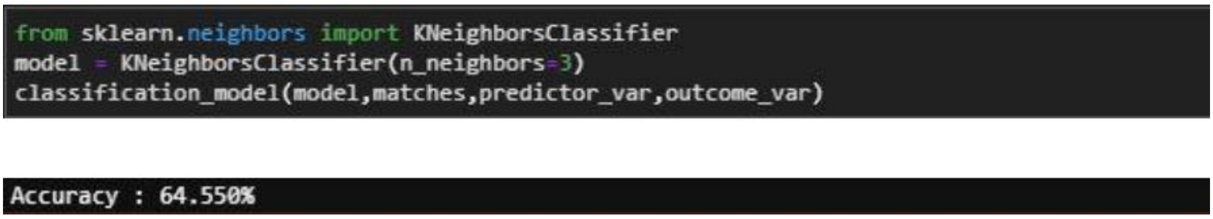
```
from sklearn.naive_bayes import GaussianNB
outcome_var=['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decision']
model = GaussianNB()
classification_model(model,matches ,predictor_var,outcome_var)
```

```
Accuracy : 17.063%
```

Fig 6.19: Gaussian Naive Bayes Algorithm

6.3.3 KNN Algorithm

The k-nearest neighbours (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems.



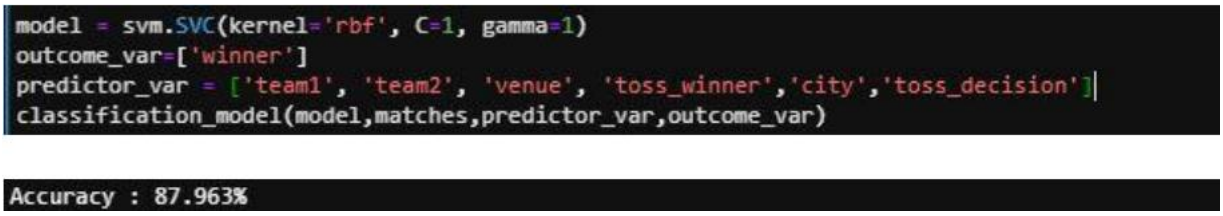
```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=3)
classification_model(model,matches,predictor_var,outcome_var)
```

```
Accuracy : 64.550%
```

Fig 6.20: KNN Algorithm

6.3.4 Support Vector Machine Algorithm

SVM is a linear model for classification or regression problems. the idea of SVM is simple. The algorithm creates a line or a hyperplane which separates the data into classes.



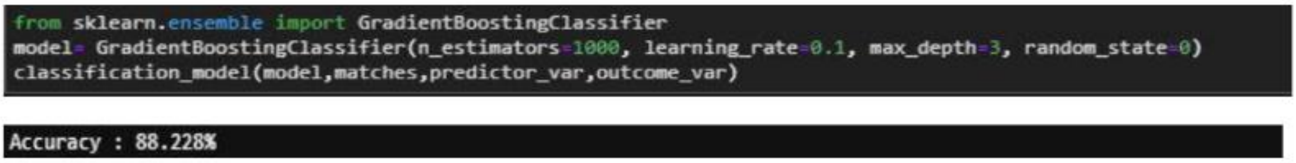
```
model = svm.SVC(kernel='rbf', C=1, gamma=1)
outcome_var=['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decision']
classification_model(model,matches,predictor_var,outcome_var)
```

```
Accuracy : 87.963%
```

Fig 6.21: Support Vector Machine Algorithm

6.3.5 Gradient Boost Algorithm

Gradient boost is a greedy algorithm and can overfit a training dataset quickly. It can benefit from regularization methods that penalize various parts of the algorithm and generall improve the performance of the algorithm by reducing overfitting.



```
from sklearn.ensemble import GradientBoostingClassifier
model= GradientBoostingClassifier(n_estimators=1000, learning_rate=0.1, max_depth=3, random_state=0)
classification_model(model,matches,predictor_var,outcome_var)
```

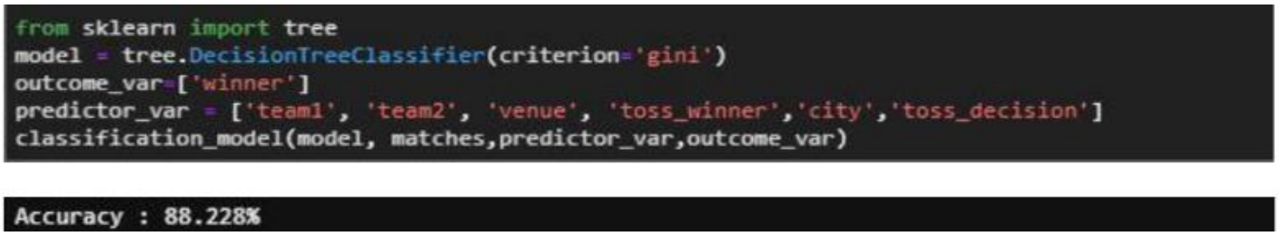
```
Accuracy : 88.228%
```

Fig 6.22: Gradient Boost Algorithm

.

6.3.6 Decision Tree Algorithm

Decision trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of target variable by leaning simple decision rules inferred from data features.



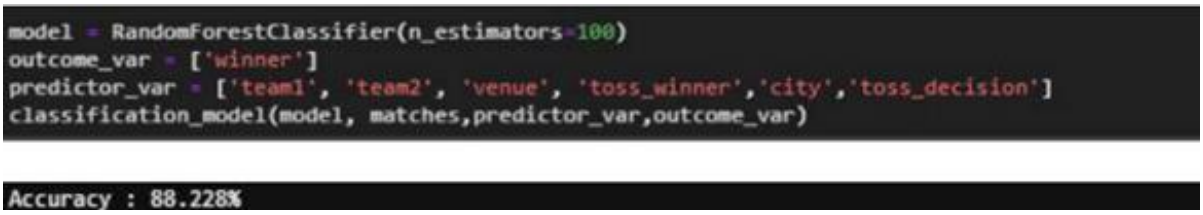
```
from sklearn import tree
model = tree.DecisionTreeClassifier(criterion='gini')
outcome_var=['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decision']
classification_model(model, matches,predictor_var,outcome_var)
```

```
Accuracy : 88.228%
```

Fig 6.23: Decision Tree Algorithm

6.3.7 Random Forest Classifier

Random forest is a machine learning technique that's used to solve regression and classification problems. these operates by constructing the multitude of decision trees at training time.



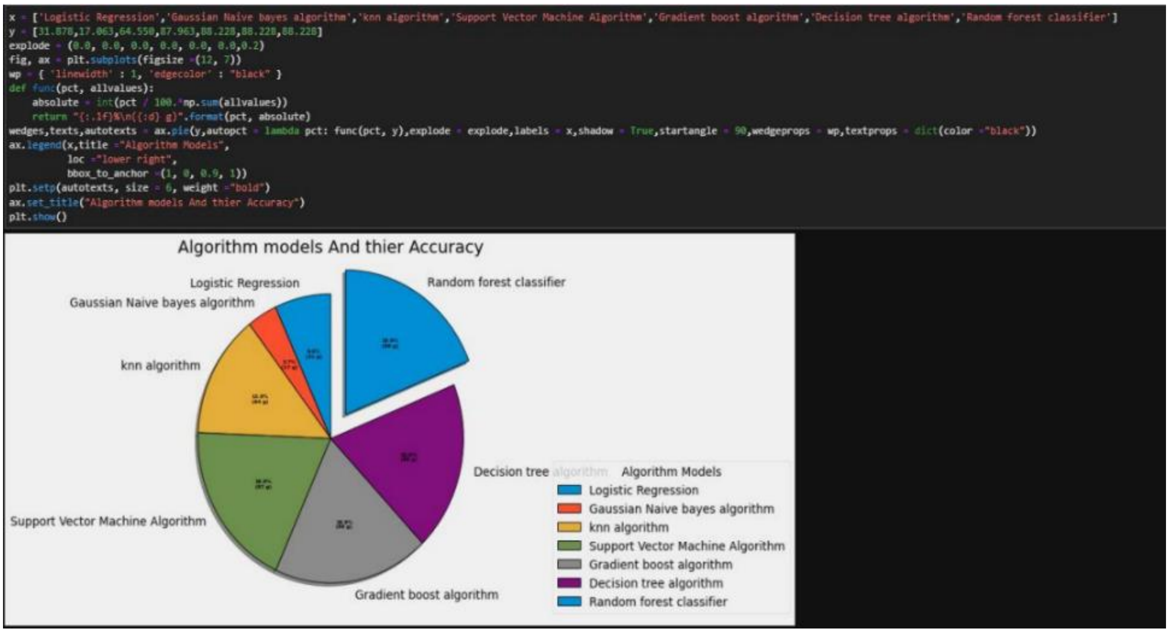
```
model = RandomForestClassifier(n_estimators=100)
outcome_var = ['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decision']
classification_model(model, matches,predictor_var,outcome_var)
```

```
Accuracy : 88.228%
```

Fig 6.24: Random Forest Classifier

Now that we have used many algorithms and found out the accuracy with which each algorithm would provide the prediction, now let's compare all these algorithms to find the best among them.

Comparing all the algorithms models and their accuracy



```
x = ['Logistic Regression','Gaussian Naive bayes algorithm','knn algorithm','Support Vector Machine Algorithm','Gradient boost algorithm','Decision tree algorithm','Random forest classifier']
y = [31.878,17.063,64.550,87.963,88.228,88.228,88.228]
explode = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0,0.2)
fig, ax = plt.subplots(figsize =(12, 7))
wp = { 'linewidth' : 1, 'edgecolor' : "black" }
def func(pct, allvalues):
    absolute = int(pct / 100.*np.sum(allvalues))
    return "{:.1f}%\n({:d} g)".format(pct, absolute)
wedges,texts,autotexts = ax.pie(y,autopct = lambda pct: func(pct, y),explode = explode,labels = x,shadow = True,startangle = 90,wedgeprops = wp,textprops = dict(color ="black"))
ax.legend(x,title ="Algorithm Models",
          loc ="lower right",
          bbox_to_anchor =(1, 0, 0.9, 1))
plt.setp(autotexts, size = 6, weight ="bold")
ax.set_title("Algorithm models And thier Accuracy")
plt.show()
```

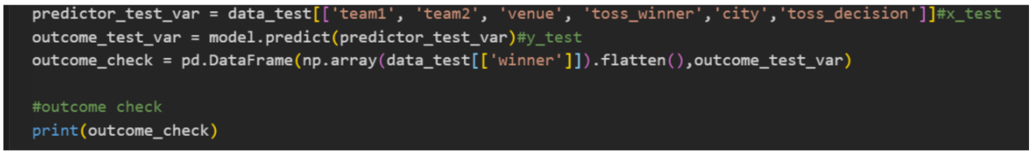
Fig 6.25: algorithms models and their accuracy

The following pie chart has been divided based on the accuracies of various algorithm models. Comparing all the models we can come to the conclusion that among all the models, Random Forest Classifier Algorithm has given the best accuracy.

Since Forest Classifier Algorithm gives the best accuracy, we will use the same algorithm for testing to predict the match winners.

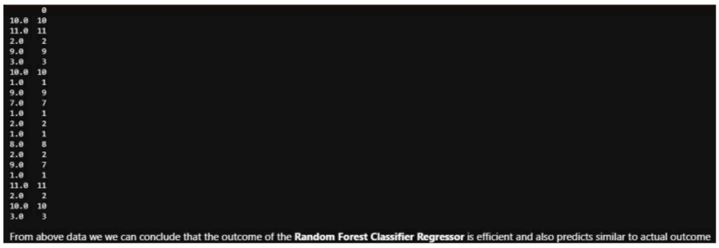
6.4 Testing

Testing is predicting if the given algorithm gives accurate or near accurate output when given an input. Before testing the algorithm, we'd have to train it. We can train the algorithm by providing it with the required dataset so that it can learn the outcome of the certain input given. Training data should always be provided more than the testing data as it'll lead to better predicted values.



```
predictor_test_var = data_test[['team1', 'team2', 'venue', 'toss_winner','city','toss_decision']]#x_test
outcome_test_var = model.predict(predictor_test_var)#y_test
outcome_check = pd.DataFrame(np.array(data_test[['winner']]).flatten(),outcome_test_var)

#outcome check
print(outcome_check)
```



```
       0
10.0  10
11.0  11
2.0    2
9.0    9
3.0    3
10.0  10
1.0    1
9.0    9
7.0    7
1.0    1
2.0    2
1.0    1
8.0    8
2.0    2
9.0    7
1.0    1
11.0  11
2.0    2
10.0  10
3.0    3
```

From above data we we can conclude that the outcome of the **Random Forest Classifier Regressor** is efficient and also predicts similar to actual outcome

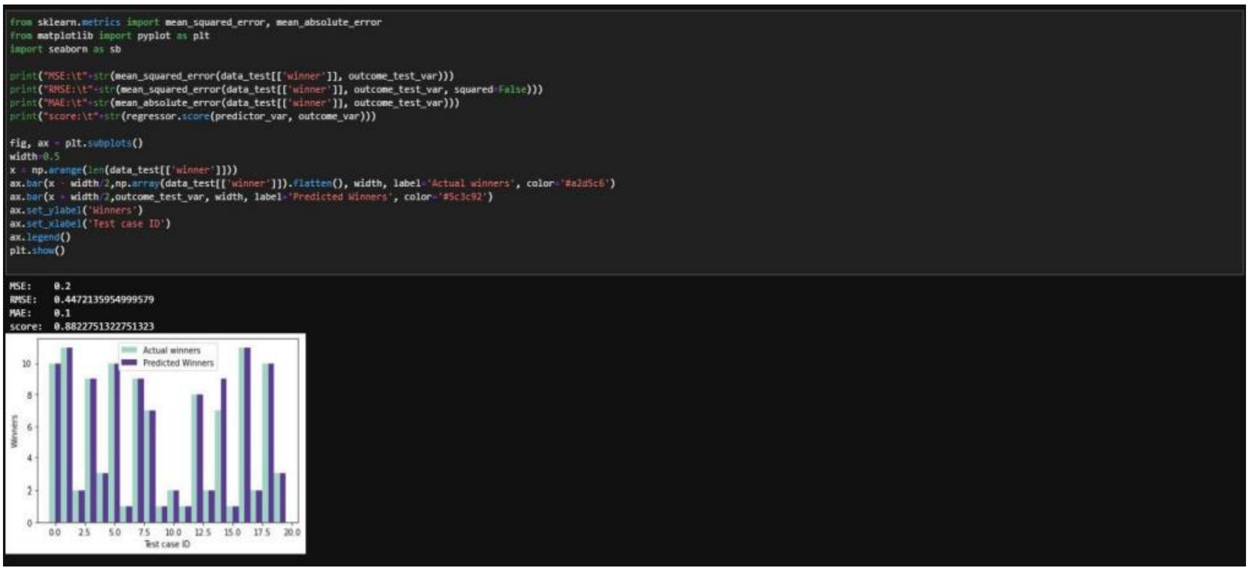
Fig 6.26: Testing based on Random Forest Classifier Algorithm

Here we have encoded each team in IPL with a number. The algorithm has been trained with the previous outcomes of match winners. To test the algorithm, we have provided with 20 test data and received the predicted match winners based on toss winners. From the above data we can conclude that Random Forest Classifier Algorithm is efficient and predicts almost similar to actual data.

6.5 Measurement and Comparison

After the satisfying test of the algorithm, we would have to measure and compare the test results with the actual outcome. This provide us a visual result of how accurate the test results are when compared to actual data.

Computing matrices for performance evaluation MAE, MSE, RMSE and Score



```
from sklearn.metrics import mean_squared_error, mean_absolute_error
from matplotlib import pyplot as plt
import seaborn as sb

print("MSE:\t"+str(mean_squared_error(data_test[['winner']], outcome_test_var)))
print("RMSE:\t"+str(mean_squared_error(data_test[['winner']], outcome_test_var, squared=False)))
print("MAE:\t"+str(mean_absolute_error(data_test[['winner']], outcome_test_var)))
print("score:\t"+str(regressor.score(predictor_var, outcome_var)))

fig, ax = plt.subplots()
width=0.5
x = np.arange(len(data_test[['winner']]))
ax.bar(x - width/2,np.array(data_test[['winner']]).flatten(), width, label='Actual winners', color='#a2d5c6')
ax.bar(x + width/2,outcome_test_var, width, label='Predicted Winners', color='#5c3c92')
ax.set_ylabel('Winners')
ax.set_xlabel('Test case ID')
ax.legend()
plt.show()
```

```
MSE:    0.2
RMSE:   0.4472135954999579
MAE:    0.1
score:  0.8822751322751323
```

Fig 6.23: MAE, MSE, RMSE and Score

The above graph is the comparison of actual winners (Blue) and predicted winners (Purple).

CHAPTER 7

CONCLUSION

Cricket is a bat-and-ball game played between two teams of eleven players each on a cricket field, at the Centre of which is a rectangular 20-metre (22-yard) pitch with a target at each end called the wicket (a set of three wooden stumps upon which two bails sit). Each phase of play is called an innings, during which one team bats, attempting to score as many runs as possible, whilst their opponents bowl and field, attempting to minimize the number of runs scored. When each innings ends, the teams usually swap roles for the next innings (i.e., the team that previously batted will bowl/field, and vice versa). The teams each bat for one or two innings, depending on the type of match. The winning team is the one that scores the most runs, including any extras gained (except when the result is not a win/loss result).

By the help of AI/ML code using python we can analyze any data set given and as per the topic chosen about the IPL data set where we predict the team winning chances between two teams of upcoming matches based upon the toss decision and winner result of all previous dataset taken and calculate accuracy using Data-mining Algorithms.

REFERENCES

[1] The learning from the company portal i.e. Tequed Labs learning portal

[2] Python Data science Handbook-Jake VanderPlas

[3] Practical Statistics for Data Science- Peter Bruce, Andrew Bruce & Peter Gedeck

[4] Stack Overflow

[5] W3Schools Website

[6] Kaggle Website