NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by
NAAC 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

Machine Learning

MINI PROJECT

On

IPL Score and Winner Predictor

Course Code: 20CS601

Academic Year – 2022-2023

Semester: 5 Section: C

Submitted To,

Course Instructor:

Mrs Soumya Ashwath
Assistant professor GD-1
Dept. of Computer Science and Engineering
NMAMIT, Nitte

Submitted By:

Name: Nischitha Shetty USN: 4NM20CS122

Name: Prathiksha Kini USN: 4NM20CS139

Date of submission: 03-05-2023

NITTE EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)

Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by

NAAC 08258 - 281039 – 281263, Fax: 08258 – 281265

Department of Computer Science and Engineering

CERTIFICATE

"IPL Score and Winner Prediction using Machine Learning" is a bonafide work carried out by Nischitha Shetty (4NM20CS122) and Prathiksha Kini (4NM20CS139) in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the year 2022-2023.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide Signature of HOD

Abstract

The Indian Premier League (IPL) is a professional Twenty20 cricket league in India, which is highly popular among cricket fans worldwide. Predicting the scores and the winner of IPL matches is a challenging task due to the complexity of the game and various factors that can affect the outcome of a match. In recent years, machine learning has shown promising results in predicting the scores and winners of cricket matches.

In this project, we aim to develop a machine learning model that can predict the scores and the winner of IPL matches. We will use a dataset containing historical data of IPL matches, including batting and bowling statistics, venue, and weather conditions. The dataset will be preprocessed and cleaned to remove any missing or irrelevant data. We will then use various machine learning algorithms such as regression and classification to build models that can predict the scores and the winner of IPL matches.

To evaluate the performance of our models, we will use various performance metrics such as accuracy, RMSE, MAE etc. Finally, we will deploy the best-performing model as a web application that can be used to predict the scores and the winner of upcoming IPL matches.

The outcome of this project can be highly beneficial for cricket fans, sports analysts, and betting enthusiasts. It can also be used by team owners and coaches to plan their strategies for upcoming matches based on predicted scores and outcomes.

TABLE OF CONTENTS

Introduction

Machine learning has shown great potential in predicting the outcome of sports events, including cricket matches. In this project, we aim to develop a machine learning model that can accurately predict the scores and winners of IPL matches using historical data. We will preprocess and clean the dataset, train and test various machine learning models, and evaluate their performance using various metrics. Finally, we will deploy the best-performing model as a web application, which can be used to predict the scores and winners of upcoming IPL matches. This project can be highly beneficial for cricket fans, team owners, coaches, sports analysts, and betting enthusiasts alike.

By developing an IPL score and winner predictor machine learning model, we hope to provide a valuable tool for cricket fans to enhance their viewing experience and for team owners and coaches to improve their strategies. The project also has potential implications for the sports industry, where accurate predictions can lead to better decision-making, improved fan engagement, and increased revenue.

Problem Statement

The problem we aim to solve with the implementation of a web portal using machine learning for IPL score and winner prediction is the challenge of accurately predicting the outcome of IPL matches. Predicting the scores and winners of cricket matches is a complex task that requires the analysis of various factors such as player form, playing conditions, and team strategies. The web portal will use machine learning models trained on historical IPL data to provide accurate predictions of scores and winners for upcoming matches. This will enhance the viewing experience of cricket fans by providing them with insightful predictions that they can use to follow the matches with a greater understanding of the game. In summary, the implementation of the web portal using machine learning for IPL score and winner prediction aims to improve the accuracy of predictions for IPL matches and enhance the experience of cricket fans and professionals.

Objectives

The main objective of implementing a web portal using machine learning for IPL score and winner prediction is to provide accurate predictions of the scores and winners of IPL matches. The specific objectives of the project are:

- Preprocess and clean the IPL historical data to create a dataset suitable for machine learning.
- Develop and train machine learning models using various algorithms such as regression and classification to predict the scores and winners of IPL matches.
- Evaluate the performance of the machine learning models using various metrics such as accuracy, RMSE, MAE.
- Develop a web portal that can display predictions of IPL match scores and winners based on the trained machine learning models.
- Deploy the web portal on a cloud-based platform to make it accessible to cricket fans, team owners, coaches, and sports analysts.
- Continuously update the machine learning models with the latest IPL data to improve the accuracy of the predictions.
- By achieving these objectives, we aim to provide an accurate and reliable IPL score and winner prediction web portal that can be used by cricket fans and professionals to enhance their viewing experience and make informed decisions.

Literature review

SL NO.	Topic	JOURNAL PUBLICA-TION YEAR	OBJECTIVES	METHODO-LOGY	SCOPE FOR IMPROV-EMENT	CONCLUSION
1	IPL Cricket Score and Winning Prediction using Machine Learning Techniques	May 2021	To predict the IPL first Inning Match Score and IPL Match Winning Prediction.	The algorithms used to predict the IPL first Inning Match Score are linear, lasso and ridge regression and for the IPL Match Winning Prediction, the classifier used here are SVC classifier, decision tree classifier and most important Random forest classifier	-	Using all the information they have developed a website. In Score Prediction analysis accuracy of Linear Regression is more than Ridge and Lasso Regression and in winning prediction analysis among SVC, Decision tree classifier and Random forest classifier, they have got Random forest classifier accuracy more than other 2, with all 90%, 80%, 75%, 70% training data
2	Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning	September 2018	With millions of people following the Indian Premier League (IPL), developing a model for predicting the outcome of its matches is a real-world problem. A cricket match depends upon various factors, and in this work, the factors	Six commonly used classification-based machine learning algorithms, viz. Naive Bayes, Extreme Gradient Boosting, Support Vector Machine, Logistic Regression, Random	The accuracy of the MLP classifier would have improved further if the team weight was calculated immediately after the end of each match. Because this is	In this study, the various factors that influence the outcome of an Indian Premier League matches were identified. The seven factors which significantly influence the result of an IPL match include the home team, the away team, the toss winner, toss

			which significantly influence the outcome of a Twenty20 cricket match are identified. Each player's performance in the field is considered to find out the overall weight (relative strength) of the teams.	Forests, and Multilayer Perceptron (MLP) are trained on the IPL dataset.	the only way, the classifier gets fed with real-time performance of the participating teams.	decision, the stadium, and the respective teams' weight. A multi-variate regression based model was formulated to calculate the points earned by each player based on their past performances which include (i) number of wickets taken, (ii) number of dot balls given, (iii) number of fours hit, (iv) number of sixes hit, (v) number of catches, and (vi) number of stumpings.

Hardware and Software Requirements

Hardware Requirements:

There are no specific hardware requirements.

Software Requirements:

- Jupyter Notebook
- Visual Studio Code
- Flask 2.2.3
- Python 3.10.6
- Scikit-learn 1.2.2
- Pandas 1.5.3
- Numpy 1.24.2
- Seaborn 0.12.2

For more information refer to requirements.txt file

Dataset

For our project we have used 2 datasets: matches.csv and ipl.csv, which are used for analyzing and predicting the winner of the match and target score respectively.

matches.csv:

It is a structured dataset, organized into rows and columns, with each row representing a single match and each column representing a specific variable associated with that match.

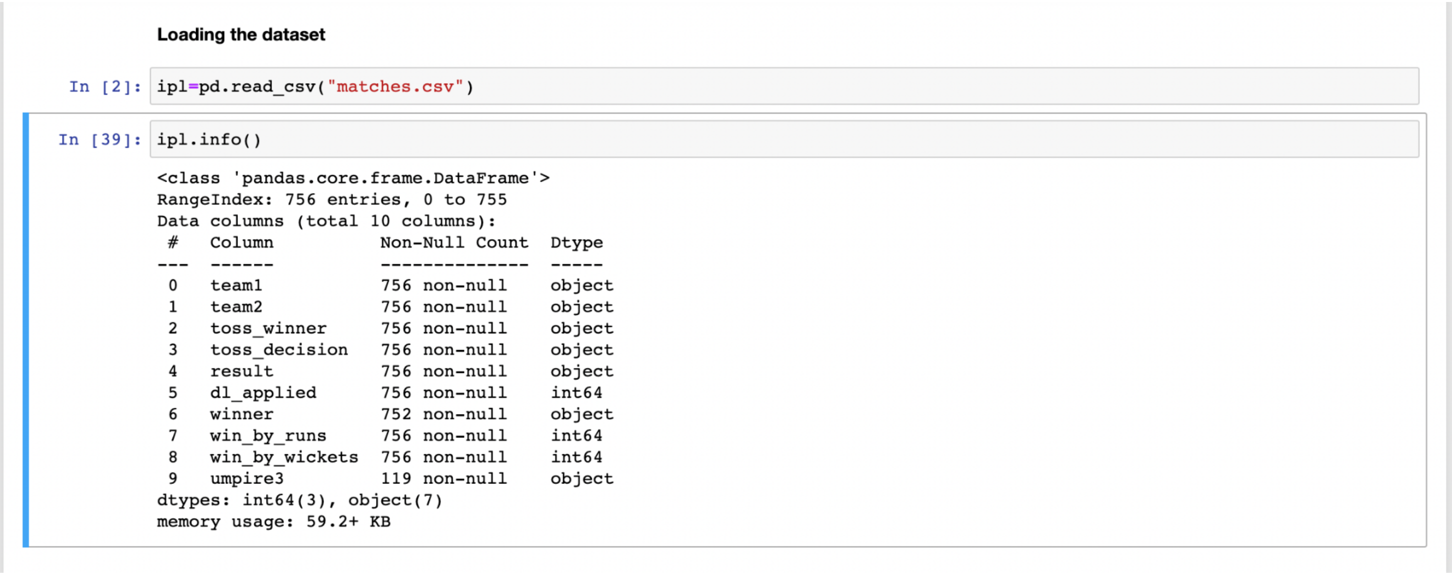
The columns in the dataset include:

- team1: The name of the first team in the match.
- team2: The name of the second team in the match.
- toss_winner: The name of the team that won the coin toss at the start of the match.
- toss_decision: The decision made by the team that won the toss - to bat or field first.
- result: The outcome of the match - normal, tie, or no result.
- dl_applied: Whether the Duckworth-Lewis method was applied to adjust targets due to interruptions in play, with a value of 1 indicating that it was applied, and 0 indicating that it was not.
- winner: The name of the winning team in the match.
- win_by_runs: The number of runs by which the winning team won, if they won

by runs.

- win_by_wickets: The number of wickets by which the winning team won, if they won by wickets.
- umpire3: The name of the third umpire for the match.

The dataset contains information on 816 IPL matches that were played between 2008 and 2021. The data has been sourced from various official IPL websites and is reliable for analysis and research purposes. It can be used for exploring various aspects of IPL matches, such as team performance, player performance, the impact of various factors on the outcome of the matches, and so on.



Loading the dataset

```
In [2]: ipl=pd.read_csv("matches.csv")
```

```
In [39]: ipl.info()
```

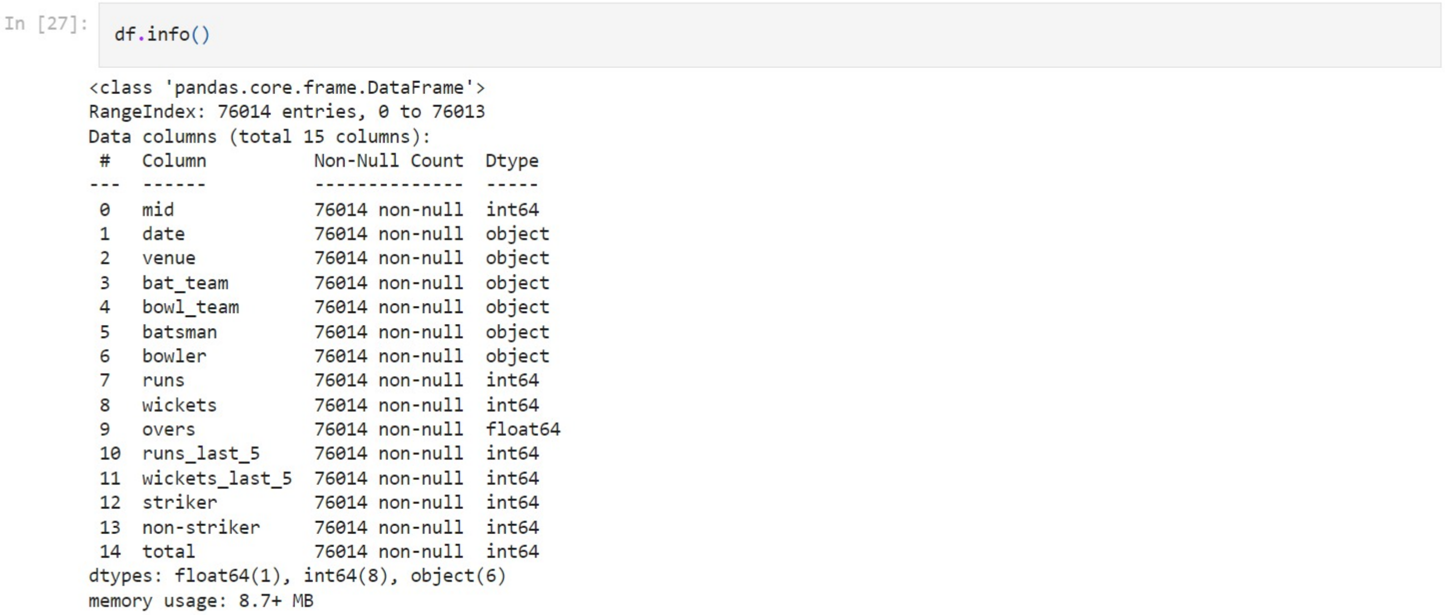
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   team1            756 non-null    object
 1   team2            756 non-null    object
 2   toss_winner      756 non-null    object
 3   toss_decision    756 non-null    object
 4   result           756 non-null    object
 5   dl_applied       756 non-null    int64 
 6   winner           752 non-null    object
 7   win_by_runs      756 non-null    int64 
 8   win_by_wickets   756 non-null    int64 
 9   umpire3          119 non-null    object
dtypes: int64(3), object(7)
memory usage: 59.2+ KB
```

ipl.csv:

The dataset contains information about the match, such as the date, venue, batting team, bowling team, and the players who batted and bowled, along with their individual statistics.

Each row in the dataset represents a ball bowled during the match, and the columns provide information about various aspects of that ball, such as the runs scored, wickets taken, overs bowled, and the current score of the batting team. The dataset contains information about every ball bowled during the match, and can be used to perform various analyses, such as predicting the outcome of the match, identifying the key players, and understanding the tactics used by the teams during the match.

- date: the date of the match
- venue: the location of the match
- bat_team: the team batting
- bowl_team: the team bowling
- batsman: the batsman facing the ball
- bowler: the bowler bowling the ball
- runs: the runs scored off the ball
- wickets: the number of wickets taken off the ball
- overs: the number of overs bowled in the innings up to that point
- runs_last_5: the number of runs scored in the last five balls
- wickets_last_5: the number of wickets taken in the last five balls
- striker: the batsman on strike
- non-striker: the batsman at the non-striker's end
- total: the total runs scored by the batting team up to that point in the innings.



```
In [27]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 76014 entries, 0 to 76013
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   mid             76014 non-null  int64  
 1   date            76014 non-null  object 
 2   venue           76014 non-null  object 
 3   bat_team        76014 non-null  object 
 4   bowl_team       76014 non-null  object 
 5   batsman         76014 non-null  object 
 6   bowler          76014 non-null  object 
 7   runs            76014 non-null  int64  
 8   wickets         76014 non-null  int64  
 9   overs           76014 non-null  float64
 10  runs_last_5     76014 non-null  int64  
 11  wickets_last_5  76014 non-null  int64  
 12  striker         76014 non-null  int64  
 13  non-striker     76014 non-null  int64  
 14  total           76014 non-null  int64  
dtypes: float64(1), int64(8), object(6)
memory usage: 8.7+ MB
```

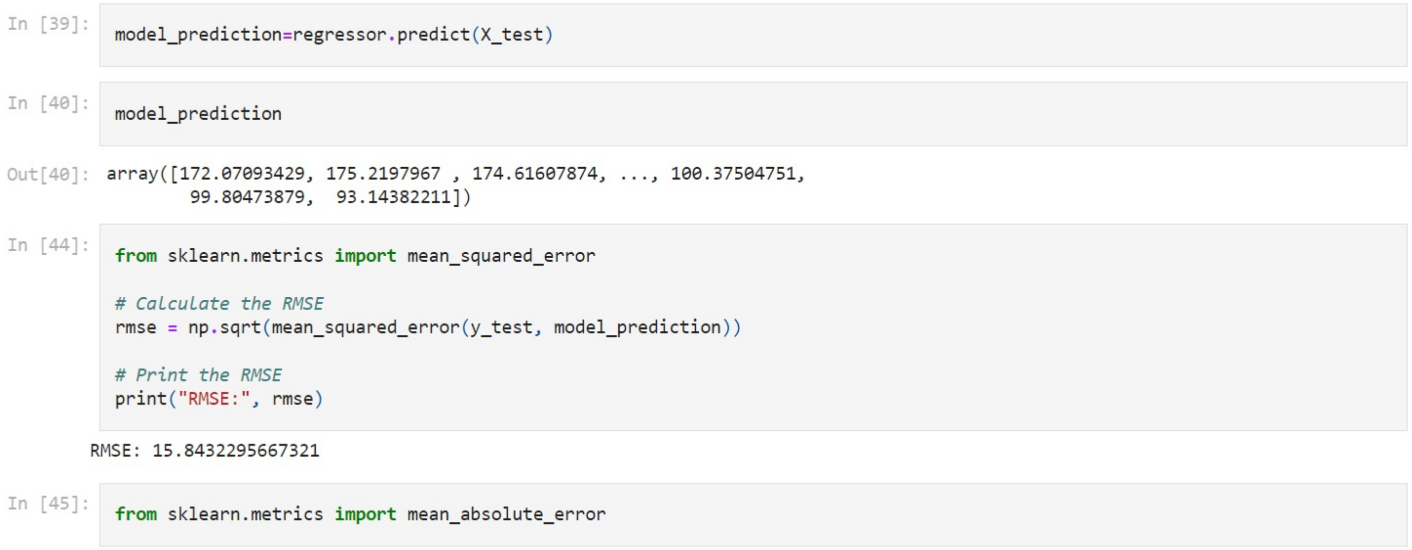
Implementation

The two algorithms used for the project are Linear regression and Random Forest Classifier for ipl.csv and matches.csv datasets respectively.

Linear Regression:

Linear Regression is a statistical method used to model the relationship between a dependent variable (also called response variable or target variable) and one or more independent variables (also called predictor variables or explanatory variables) that may affect the dependent variable.

The goal of linear regression is to use the model to make predictions of the dependent variable based on the values of the independent variable(s). The accuracy of the model can be evaluated using various measures such as the R-squared value, which represents the proportion of the variation in the dependent variable that can be explained by the independent variable(s) in the model.



```
In [39]: model_prediction=regressor.predict(X_test)
```

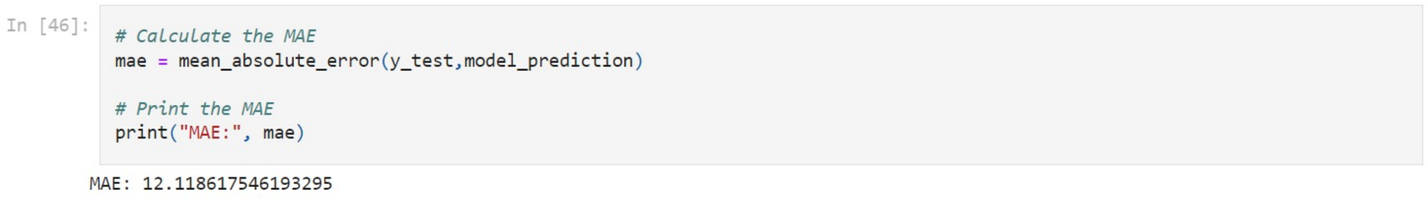
```
In [40]: model_prediction
```

```
Out[40]: array([172.07093429, 175.2197967 , 174.61607874, ..., 100.37504751,
                 99.80473879,  93.14382211])
```

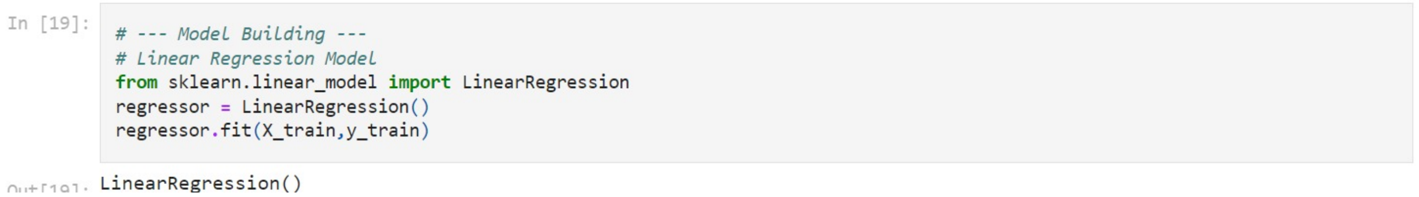
```
In [44]: from sklearn.metrics import mean_squared_error

         # Calculate the RMSE
         rmse = np.sqrt(mean_squared_error(y_test, model_prediction))

         # Print the RMSE
         print("RMSE:", rmse)
```

```
RMSE: 15.8432295667321
```

```
In [45]: from sklearn.metrics import mean_absolute_error
```



```
In [46]:  # Calculate the MAE
          mae = mean_absolute_error(y_test,model_prediction)

          # Print the MAE
          print("MAE:", mae)

         MAE: 12.118617546193295
```



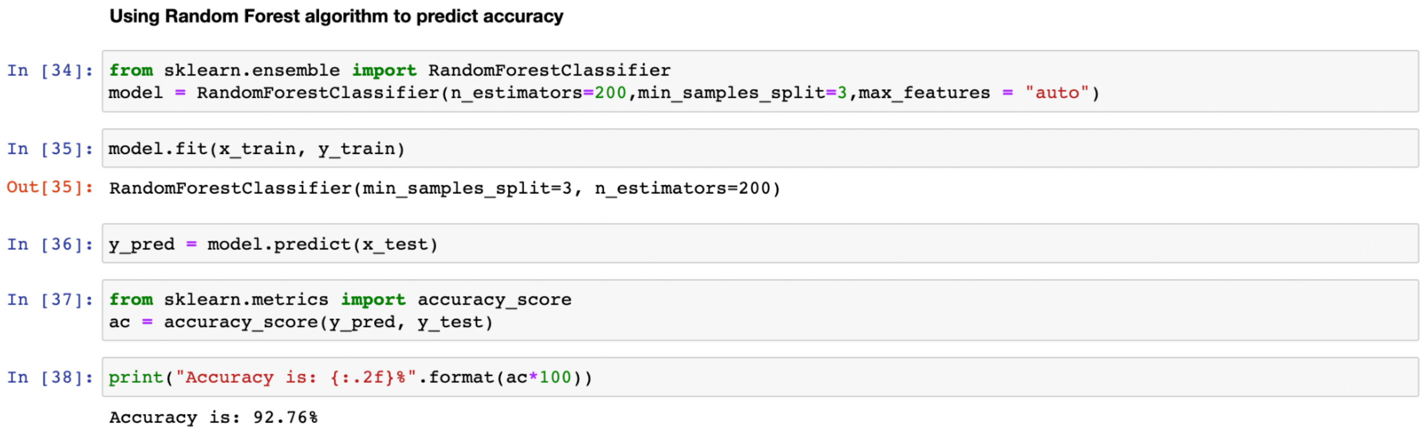
```
In [19]:  # --- Model Building ---
          # Linear Regression Model
          from sklearn.linear_model import LinearRegression
          regressor = LinearRegression()
          regressor.fit(X_train,y_train)

Out[19]:  LinearRegression()
```

Random Forest Classifier:

Random forest classifier is a type of ensemble learning algorithm that combines multiple decision trees to make predictions.

The main idea behind a random forest classifier is to create a "forest" of decision trees by randomly selecting a subset of features from the input data for each tree and then training each tree on a random sample of the training data. This helps to reduce overfitting, improve the accuracy of the predictions, and increase the generalizability of the model.

During prediction, the random forest classifier combines the predictions of all the individual decision trees in the forest to arrive at a final prediction. The final prediction is based on a majority vote (in classification) or average (in regression) of the predictions of all the individual trees in the forest.



Using Random Forest algorithm to predict accuracy

```
In [34]: from sklearn.ensemble import RandomForestClassifier
         model = RandomForestClassifier(n_estimators=200,min_samples_split=3,max_features = "auto")
```

```
In [35]: model.fit(x_train, y_train)

Out[35]: RandomForestClassifier(min_samples_split=3, n_estimators=200)
```

```
In [36]: y_pred = model.predict(x_test)
```

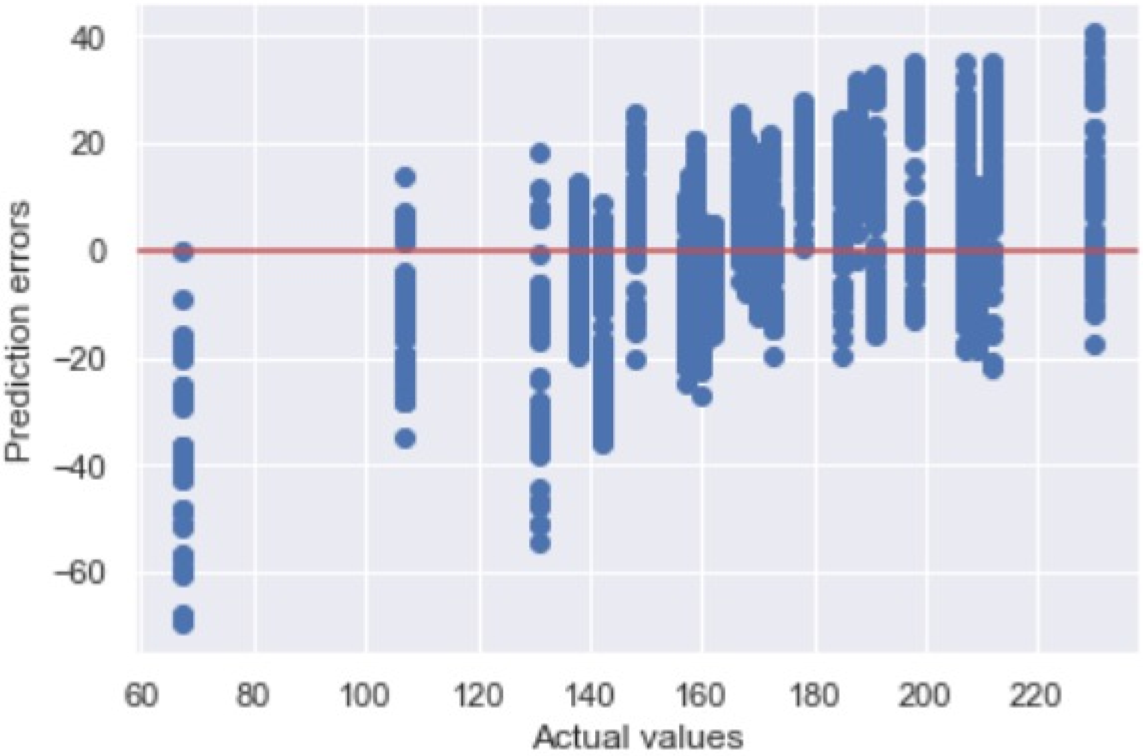
```
In [37]: from sklearn.metrics import accuracy_score
         ac = accuracy_score(y_pred, y_test)
```

```
In [38]: print("Accuracy is: {:.2f}%".format(ac*100))

         Accuracy is: 92.76%
```

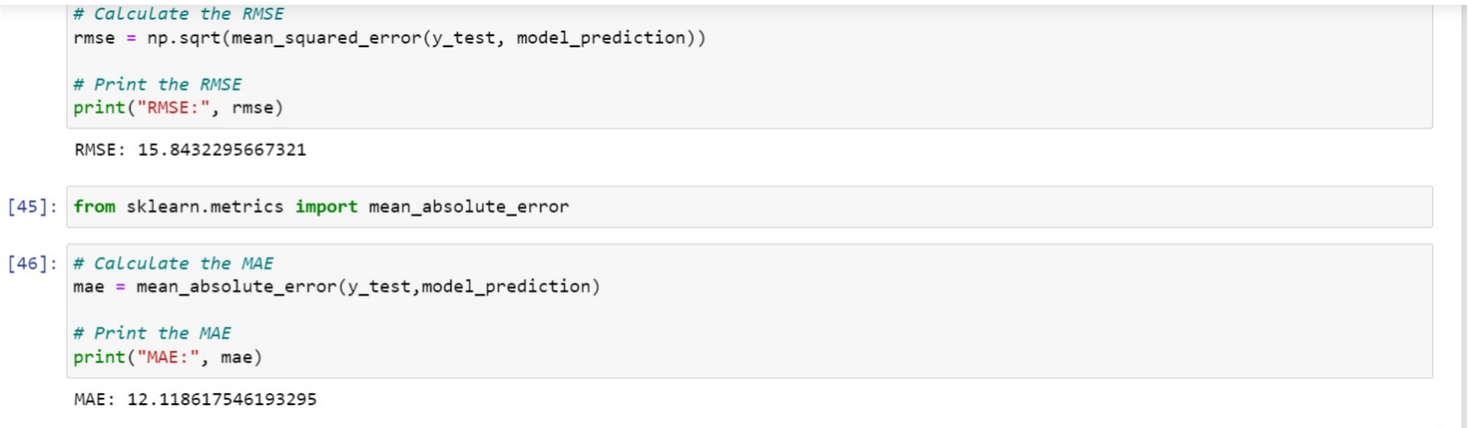
Results

Our model achieved a mean absolute error of 12.118617546193295 and a root mean squared error of 15.8432295667321 when predicting the target variable for ipl.csv dataset using linear regression algorithm. The accuracy achieved for matches.csv dataset was 89.47% using random forest classifier. These results indicate that our model performed well in predicting the target variable.

ipl.csv: Prediction of target score



Prediction Error Plot



```
# Calculate the RMSE
rmse = np.sqrt(mean_squared_error(y_test, model_prediction))

# Print the RMSE
print("RMSE:", rmse)
```

```
RMSE: 15.8432295667321
```

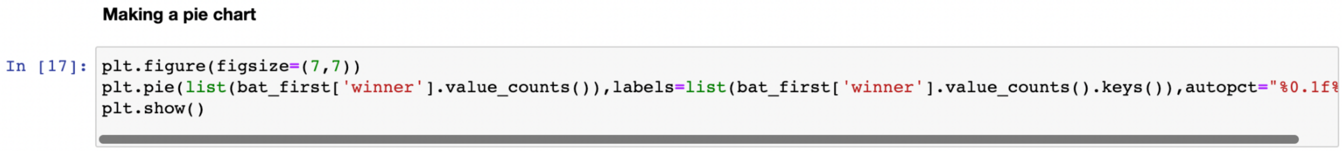
```
[45]: from sklearn.metrics import mean_absolute_error
```

```
[46]: # Calculate the MAE
mae = mean_absolute_error(y_test,model_prediction)

# Print the MAE
print("MAE:", mae)
```

```
MAE: 12.118617546193295
```

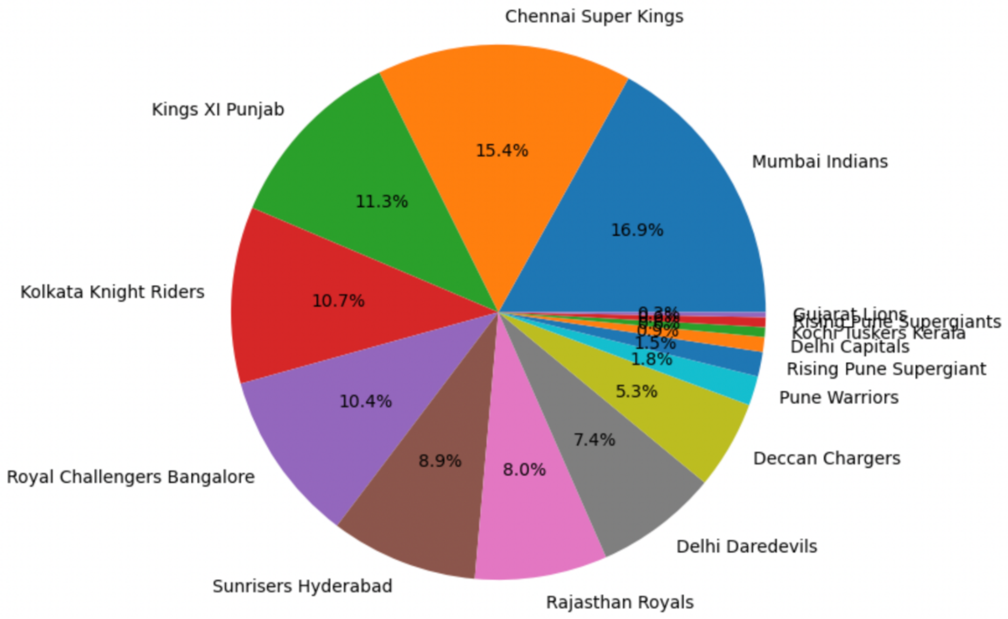
RMSE and MAE

matches.csv: Winner of the match

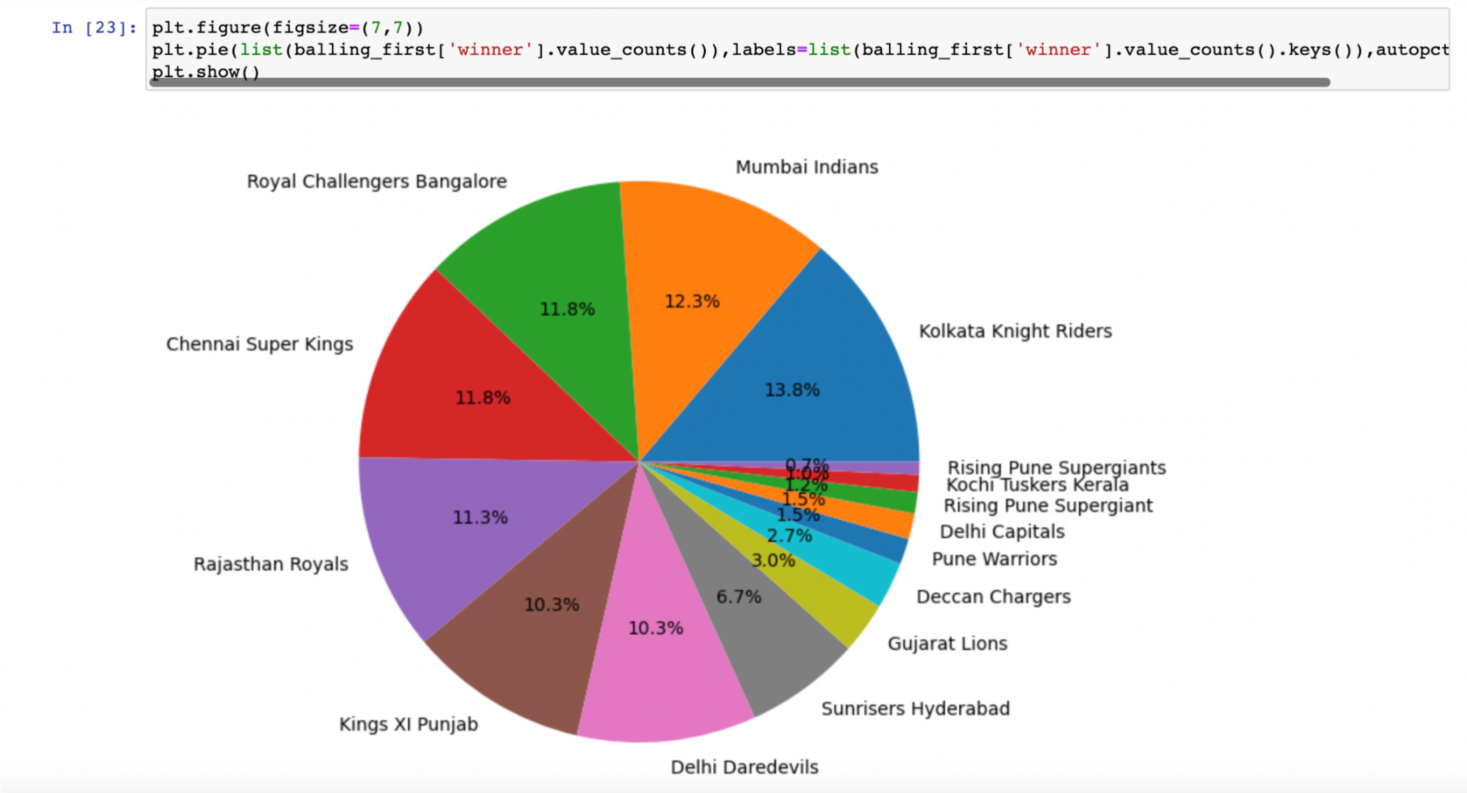


Making a pie chart

```
In [17]: plt.figure(figsize=(7,7))
plt.pie(list(bat_first['winner'].value_counts()),labels=list(bat_first['winner'].value_counts().keys()),autopct="%0.1f%
plt.show()
```

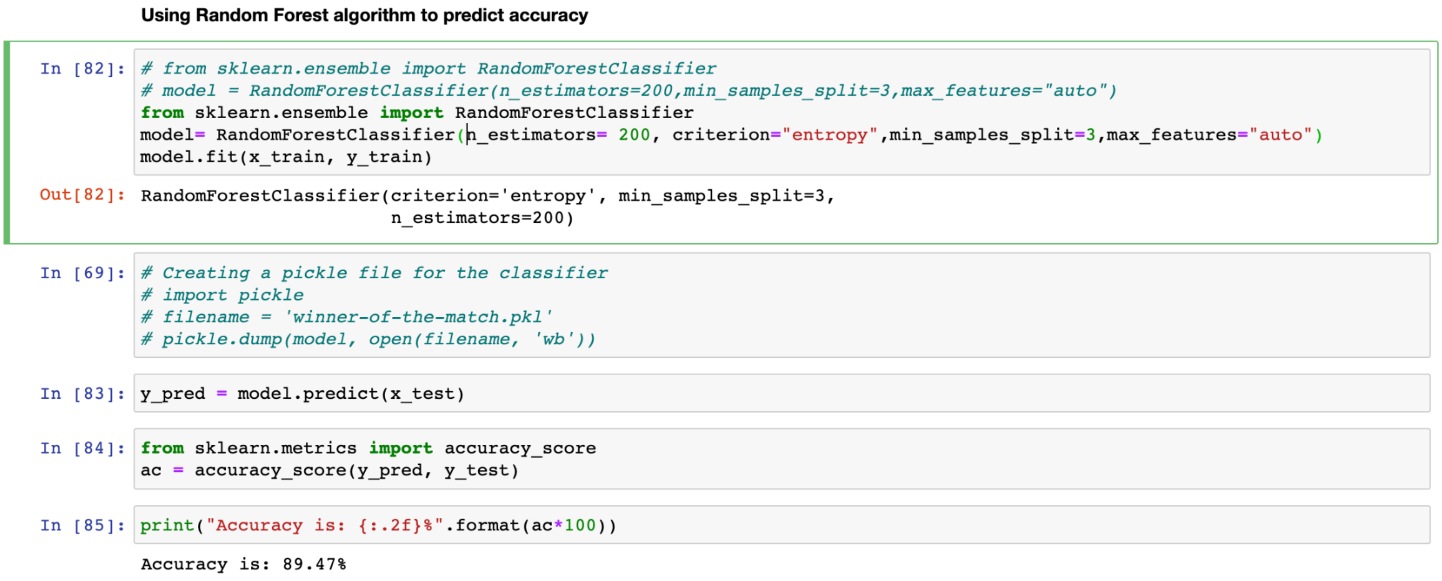


Winner of the match after winning the toss and choosing batting first



```
In [23]: plt.figure(figsize=(7,7))
         plt.pie(list(balling_first['winner'].value_counts()),labels=list(balling_first['winner'].value_counts().keys()),autopct
         plt.show()
```

Winner of the match after winning the toss and choosing balling first



Using Random Forest algorithm to predict accuracy

```
In [82]: # from sklearn.ensemble import RandomForestClassifier
         # model = RandomForestClassifier(n_estimators=200,min_samples_split=3,max_features="auto")
         from sklearn.ensemble import RandomForestClassifier
         model= RandomForestClassifier(n_estimators= 200, criterion="entropy",min_samples_split=3,max_features="auto")
         model.fit(x_train, y_train)

Out[82]: RandomForestClassifier(criterion='entropy', min_samples_split=3,
                                n_estimators=200)

In [69]: # Creating a pickle file for the classifier
         # import pickle
         # filename = 'winner-of-the-match.pkl'
         # pickle.dump(model, open(filename, 'wb'))

In [83]: y_pred = model.predict(x_test)

In [84]: from sklearn.metrics import accuracy_score
         ac = accuracy_score(y_pred, y_test)

In [85]: print("Accuracy is: {:.2f}%".format(ac*100))

         Accuracy is: 89.47%
```

Accuracy

Conclusion

The implementation of a web portal that can predict the target scores and winner of an IPL match using ML algorithms is a promising idea that can provide valuable insights and entertainment to cricket fans. Based on the performance of the model, it can be concluded that the predictions can be accurate to a certain extent.

The future scope for this implementation could involve further fine-tuning of the ML algorithms used to make more accurate predictions. Additionally, including more variables such as weather, pitch conditions, player performance history, and team dynamics could further improve the accuracy of the predictions.

Furthermore, this web portal can be expanded to cover other cricket leagues and tournaments globally, providing users with more opportunities to engage with the platform. Finally, the platform could be monetized by offering premium services to users, such as customized predictions and analytics based on user preferences and data.

References

- Scikit-learn documentation: https://scikit-learn.org/stable/index.html
- Flask documentation: https://flask.palletsprojects.com/en/1.1.x/quickstart/
- Dataset links:
    - ipl.csv: https://drive.google.com/file/d/1nGYmbEB3GjJIuSF9lt5jrxlTxvf799eN/view?usp=sharing
    - matches.csv: https://drive.google.com/file/d/1KFkurdApiH1OmyFx_2G0ABRHTtHRVHBw/view?usp=sharing
- Python Documentation: https://www.python.org/doc/