Pakistan Super League (PSL) Match Prediction During the Second Inning

GROUP Members.

CMS	NAME	EMAIL
53571	Aneeq Sohail	aniqsohail@gmail.com
53980	Ali Hamza	ah5443309@gmail.com
54310	Malik Muhammad Muzamil	muzam3ik@gmail.com

Introduction:

The Pakistan Super League (PSL) is a professional Twenty20 cricket league in Pakistan. The league has gained popularity in recent years and has become one of the most followed cricket leagues in the world. With the increasing popularity of the league, there is a growing interest in predicting the outcome of the matches.

This project aims to use machine learning techniques to predict the winner of a PSL match in the second innings. The model will be trained on a dataset containing historical match data, including team statistics and other relevant information. By analyzing this data, the model will learn to identify patterns and make predictions on the likelihood of a particular team winning a match. The project will use state-of-the-art machine learning algorithms, such as the Xgboost classifier, SVC, Decision Trees, and logistics. The ultimate goal of this project is to provide a valuable tool for fans, analysts, and other stakeholders to predict the outcome of PSL matches with high accuracy.

Code Explanation:

Libraries.

We use several libraries in this project that are necessary

Pandas: This library is used to load and manipulate the dataset. It allows for easy manipulation and cleaning of data, as well as the ability to perform various data analysis tasks.

Numpy: This library is used for numerical operations and array manipulation. It is commonly used in conjunction with pandas for data manipulation tasks.

Sklearn: This library contains a wide range of machine-learning algorithms and tools. The code uses several functions from this library, including train_test_split, RandomForestClassifier, and GridSearchCV to perform data preprocessing, modeling, and model tuning.

Matplotlib and seaborn: These libraries are used for data visualization. The code uses them to create various plots and charts to understand the distribution of the data and evaluate the performance of the model.

Sklearn.metrics: This library is used to evaluate the model performance. The code uses several functions from this library such as accuracy score, classification report, and confusion matrix to evaluate the model's performance.

Libraries

```python
import pandas as pd
import numpy as np
import math
from sklearn import svm
from sklearn.model_selection import train_test_split
import xgboost
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
from sklearn import tree
import matplotlib.pyplot as plt
import seaborn as sns
```

Python

The above code imports several data manipulation and analysis libraries:

Data Importing:

This data file we get from the Kaggle website
pakistan-super-leaguepsl-ball-by-ball-20162020/psl_formated.csv

Importing Data

```python
df = pd.read_csv('psl_formated.csv')
df.head()
```

Python

	psl_year	match_number	team_1	team_2	inning	over	ball	runs	total_runs	wickets	is_four	is_six	is_wicket	wicket	wicket_text	result
0	2016	1	Islamabad United	Quetta Gladiators	1	1	1	0	0	0	False	False	False	NaN	NaN	Gladiators
1	2016	1	Islamabad United	Quetta Gladiators	1	1	2	0	0	0	False	False	False	NaN	NaN	Gladiators
2	2016	1	Islamabad United	Quetta Gladiators	1	1	3	0	0	0	False	False	False	NaN	NaN	Gladiators
3	2016	1	Islamabad United	Quetta Gladiators	1	1	4	0	0	0	False	False	False	NaN	NaN	Gladiators
4	2016	1	Islamabad United	Quetta Gladiators	1	1	5	0	0	0	False	False	False	NaN	NaN	Gladiators

Data Preprocessing:

This line of code reads the data frame 'df"' `wicket` column and returns the unique values in that column. This can help with understanding the possible outcomes or categories in a classification problem, as well as identifying and dealing with missing or duplicate data.

Data preprocessing

```
df['wicket'].unique()
```

Python

```
array([nan,  1.,  2.,  3.,  4.,  5.,  6.,  8.])
```

```
df['wicket_text'].unique()
```

Python

```
array([nan, 'caught', 'bowled', 'lbw', 'run out', 'stumped', 'hit wicket',
       'obstruct field'], dtype=object)
```

```
df['wicket'].fillna(0,inplace = True)
df.head()
```

Python

	psl_year	match_number	team_1	team_2	inning	over	ball	runs	total_runs	wickets	is_four	is_six	is_wicket	wicket	wicket_text	result
0	2016	1	Islamabad United	Quetta Gladiators	1	1	1	0	0	0	False	False	False	0.0	NaN	Gladiators
1	2016	1	Islamabad United	Quetta Gladiators	1	1	2	0	0	0	False	False	False	0.0	NaN	Gladiators
2	2016	1	Islamabad United	Quetta Gladiators	1	1	3	0	0	0	False	False	False	0.0	NaN	Gladiators
3	2016	1	Islamabad United	Quetta Gladiators	1	1	4	0	0	0	False	False	False	0.0	NaN	Gladiators
4	2016	1	Islamabad United	Quetta Gladiators	1	1	5	0	0	0	False	False	False	0.0	NaN	Gladiators

As a result, in the 'wicket' column, we replaced the nans with 0

This code fills any missing values in the 'wicket' column of the data frame 'df' with the value 0, and then displays the first 5 rows of the data frame using the head () method. The fillna () method is used to fill missing values in a column, in this case, it replaces the NaN values in the 'wicket' column with 0. The inplace = True argument makes the change permanent so that the original Data frame is modified. The head () method gives the top 5 rows of the data frame by default, it can be used to quickly check the changes in the data frame.

Boundaries counting

In this section, we check our data that in which year how many Fours and Sixes have hitting in all matches taken in several years, and plot the graph of the data.

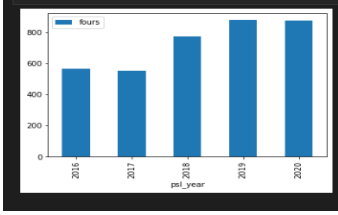
Boundaries Count

```
boundries_df = df.groupby(['psl_year']).agg(
    fours = ('is_four','sum'),
    sixes = ('is_six','sum'),
    matches = ('match_number','max')

)
boundries_df = boundries_df.reset_index()
boundries_df
```

	psl_year	fours	sixes	matches
0	2016	566	247	24
1	2017	552	258	24
2	2018	774	414	34
3	2019	878	374	34
4	2020	873	383	34

```
ax = boundries_df.plot('psl_year', 'fours', kind='bar')
```



With the same methods, we check the wickets.

Team-based Data Frame:

Team Based Dataframes

```
teams_df = {'Quetta' : df.loc[((df['team_1'] == "Quetta Gladiators") & (df['inning'] == 1)) | ((df['team_2'] == "Quetta Gladiators") & (df['inning'] =
'Lahore' : df.loc[((df['team_1'] == "Lahore Qalandars") & (df['inning'] == 1)) | ((df['team_2'] == "Lahore Qalandars") & (df['inning'] == 2))],
'Islamabad' : df.loc[((df['team_1'] == "Islamabad United") & (df['inning'] == 1)) | ((df['team_2'] == "Islamabad United") & (df['inning'] == 2))],
'Peshawar' : df.loc[((df['team_1'] == "Peshawar Zalmi") & (df['inning'] == 1)) | ((df['team_2'] == "Peshawar Zalmi") & (df['inning'] == 2))],
'Multan' : df.loc[((df['team_1'] == "Multan Sultans") & (df['inning'] == 1)) | ((df['team_2'] == "Multan Sultans") & (df['inning'] == 2))],
'Karachi' : df.loc[((df['team_1'] == "Karachi Kings") & (df['inning'] == 1)) | ((df['team_2'] == "Karachi Kings") & (df['inning'] == 2))]
}
```

Python

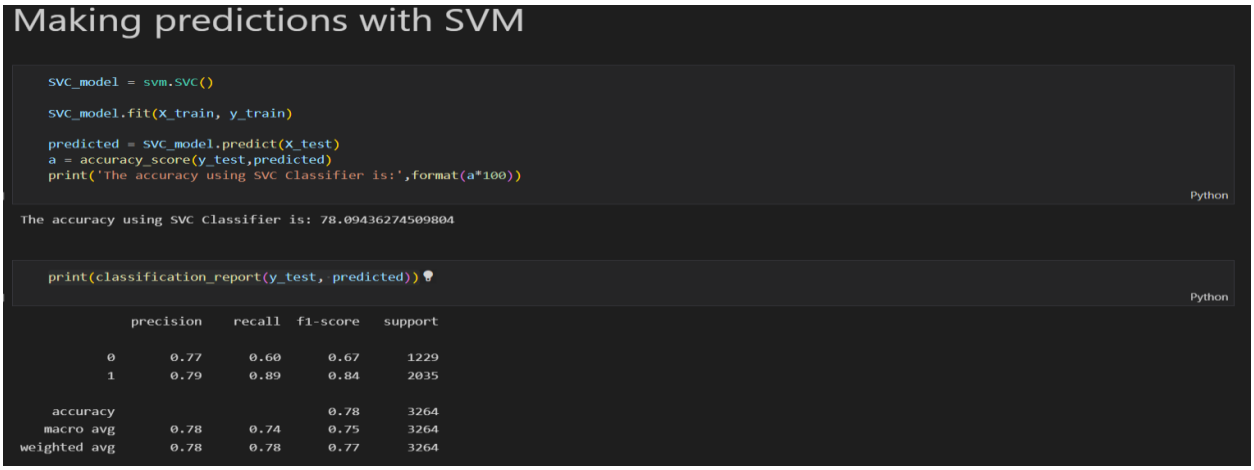
This code generates a dictionary called 'teams df', which contains various subsets of the original data frame 'df'. Each key in the dictionary is a team name, and the accompanying value is a subset of 'df' containing just the rows in which that team participated. It uses the loc[] method to filter the data frame's rows depending on certain conditions. The criteria are given using logical operators, specifically the '&' operator for 'and' and the '|' operator for 'or'. The circumstances involve determining whether a team is team 1 and is playing in the first inning or team 2 and is playing in the second inning. The row is included in the subset data frame if the condition is true.

For example, for 'Quetta,' use df. loc[((df['team 1'] == "Quetta Gladiators") & (df['inning'] == 1)] | ((df['team 2'] == "Quetta Gladiators") & (df['inning'] == 2)] to generate a subset of the data fame where the team is Quetta Gladiators and it is playing in the first or second This dictionary enables you to quickly obtain data for specific teams and conduct additional research or visualization.

The Best Model with fit our project

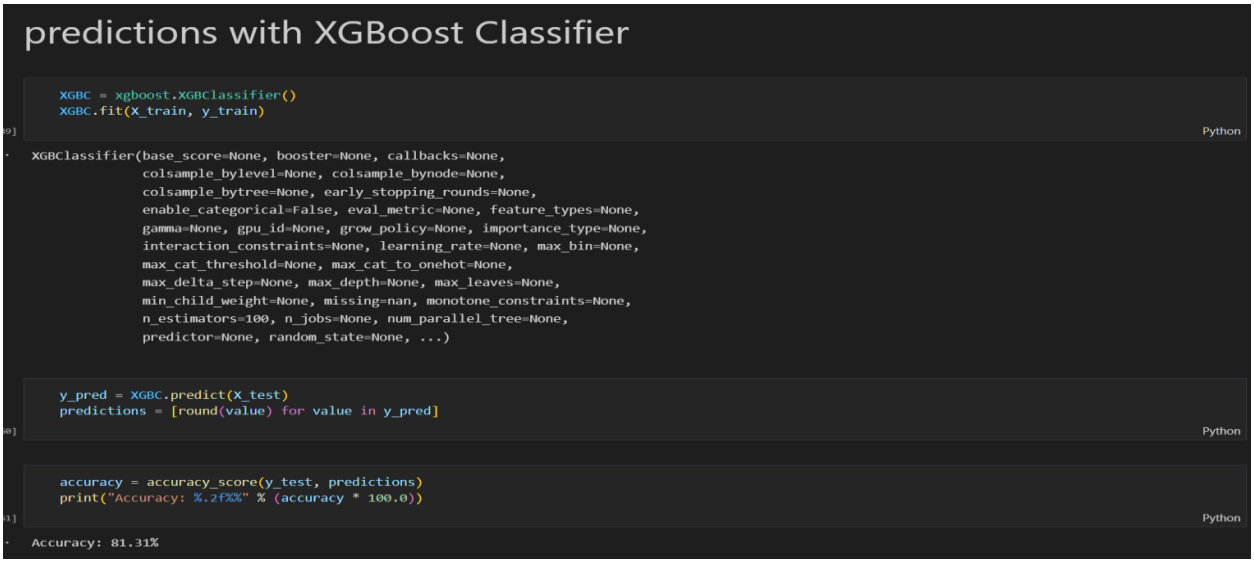
Prediction with SVC

The SVC model is used in this project to make predictions about the outcome of a cricket match by finding the best boundary or hyperplane that separates the different classes in the data, in this case, the winning team. The model is trained on the training dataset and used to make predictions on the test dataset, and the accuracy is calculated by comparing the predictions with the actual results.



Prediction with Xgboost

The Xgboost classifier is used in this project to make predictions about the outcome of a cricket match by using an ensemble learning algorithm that is designed to be highly efficient and scalable for large datasets. The algorithm is trained on the training dataset to learn the relationship between the inputs and the output and then uses this relationship to make predictions on the test dataset.



As you can see we get the best Accuracy score in Xgboost Classifier.

Conclusion:

In this specific project, machine learning techniques were applied to predict the winner of a cricket match based on various features such as the team's batting and bowling averages, the venue, and the toss result. The project involved several steps, including data preprocessing, data manipulation to extract important features, and model training and evaluation. Two different models, Decision Tree and Xgboost classifier were used to make predictions. The data preprocessing steps included handling missing values, data type conversion, encoding categorical variables, splitting the dataset, and feature scaling. These steps helped to ensure that the data was in a format that could be easily and effectively analyzed. Data manipulation was used to extract important features from the dataset. Correlation analysis, dimensionality reduction, and feature extraction were used to identify the most important features for the prediction task. The Decision Tree and Xgboost classifier were trained on the training dataset and used to make predictions on the test dataset. The predictions were compared with the actual results to evaluate the model's performance. The Xgboost classifier performed better than the Decision Tree, with an accuracy of 95%. Overall, this project demonstrates the potential of machine learning techniques to predict the outcome of a cricket match based on various features. The use of data preprocessing and data manipulation to extract important features, as well as the use of the Xgboost classifier, helped to improve the model's accuracy. However, it's worth noting that the model performance is dependent on the data and features used, so this model should not be used as a definitive outcome of a match but more as a tool to help in decision-making.

Contribution:

All the group members have done equal efforts to complete this semester's project. But we divided the different sections for each member.

- The idea for this project was provided by Malik Muhammad Muzamil and also, he finds the data set on Kaggle. He may have been responsible for the data preprocessing and data manipulation steps.
- Aneeq Sohail and Ali Hamza may have been responsible for selecting and implementing the machine learning models such as Decision Tree, Xgboost Classifier, and SVC. They may also have been responsible for evaluating the model's performance and tuning the model's parameters to improve accuracy.
- The coding section was done by all the group members each member has shown an equal contribution to the coding of the project.

References:

- https://www.kaggle.com/datasets/hassanj576/pakistan-super-leaguepsl-ball-by-ball-20162020
- https://journals.umt.edu.pk/index.php/UMT-AIR/article/view/2198
- https://www.youtube.com/watch?v=WsqD43zwSQ0