IPL SCORE PREDICTION USING MACHINE LEARNING

(COURSE CODE: 23UPCSC4P01)

A mini project submitted in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

IN

DATA SCIENCE

to the

Periyar University, Salem-11

by

SURESH KUMAR S

(Reg. No: U23PG507DTS033)

DEPARTMENT OF COMPUTER SCIENCE

(Supported by UGC-SAP)

PERIYAR UNIVERSITY

(NAAC "A++" Grade-with CGPA 3.61(Cycle-3)

State university-NIRF Rank 59-NIRF innovation Band of 11-50

PERIYAR PALKALAI NAGAR SALEM – 636011

APRIL 2024

MR. E. V. SHARAVANAN, M.C.A, M.Phil.,

TEACHING ASSISTANT,
DEPARTMENT OF COMPUTER SCIENCE,
PERIYAR UNIVERSITY,
SALEM -11.

CERTIFICATE

This is to certify that the report of Mini Project entitled "**IPL SCORE PREDICTION USING MACHINE LEARNING**" submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science to the Periyar University, Salem is a record of bonafide work carried out by **SURESH KUMAR S (Reg.No: U23PG507DTS033)** under my supervision and guidance.

Signature of the Guide　　　　　　　　　　Signature of the HOD

Submitted of the Viva-Voce Examination held on ___________________.

Internal Examiner　　　　　　　　　　**External Examiner**

DECLARATION

I hereby, declare that the project work entitled "**IPL SCORE PREDICTION USING MACHINE LEARNING**" submitted to Periyar University in partial fulfilment of the requirement for the award of the Degree of Master of Science in Data Science is the record work carried out by me, under the supervision **Mr E.V. SHARAVANAN, TEACHING ASSISTANT, DEPARTMENT OF COMPUTER SCIENCE, Periyar University**. To the best of my knowledge, the work reported here is not a part of any other work on the basis of which a degree or award was conferred on an earlier to one or any other candidate.

Place: Salem -11 **Signature of Student**

Date:

ACKNOWLEDGEMENT

First, I would like to thank The Almighty for providing me with everything that I required in completing this project.

I would like to extend my sincere thanks to **Dr. R. JAGANNATHAN, M.Sc Vice Chancellor Periyar University**, Salem who has been an invaluable source for providing the facility to complete this mini project successfully.

I extend my heart-felt gratitude to the **Dr .C. CHANDRASHEKHAR,M.C.A, Ph.D, Head of Department**, Department of Computer Science, Periyar University, Salem -11 for histimely triggers.

I acknowledge with thanks the kind patronage, loving inspiration and timely guidance which I have received from **Mr. E.V. SHARAVANAN, M.C.A, M.phill, Teaching Assitant,** Department of Computer Science**,** Periyar University, Salem.

I acknowledge with thanks the kind patronage, loving inspiration and timely guidance which I have received from **Mr. J. GOKULRAJ, B.Tech, M.E Program Manager**, Data Science at Boston IT Solutions India.

I extend my thanks to my parents and well-wishers for their constant support and encouragement.

Place: Salem-11 **Signature of the Student**

Date:

INDEX

ABSTRACT

The provided code seems to be a web application built using Flask for predicting cricket scores based on various input parameters like venue, batting team, bowling team, overs, runs, wickets, runs scored in the previous 5 overs, and wickets taken in the previous 5 overs. The model used for prediction appears to be a Ridge regression model trained on IPL data. The application allows users to input these parameters and get a predicted score. It incorporates preprocessing steps like data cleaning, feature engineering, scaling, and model serialization. The Ridge regression model achieved a reasonably good performance, with a root mean squared error (RMSE) of around 16.3 and an R-squared score of approximately 0.74.

INTRODUCTION

In the rapidly evolving landscape of cricket analytics, the need for accurate score prediction models is paramount. This project endeavors to fulfill that need by harnessing machine learning techniques, particularly Ridge regression, to forecast cricket scores in Indian Premier League (IPL) matches. Leveraging a dataset encompassing various match parameters such as venue, batting and bowling teams, overs, runs, and wickets, the model is trained and fine-tuned to optimize predictive accuracy. Through meticulous preprocessing, including data cleaning and feature engineering, alongside rigorous evaluation metrics like mean absolute error and R-squared score, the model demonstrates its efficacy. This endeavor not only contributes to enhancing cricket analytics but also serves as a testament to the power of data-driven decision-making in sports.

SYSTEM CONFIGURATION

HARDWARE REQUIREMENT

- RAM: 4.00 GB
- Processor: AMD PRO A4-3350B APU with Radeon R4 Graphics
- Base Frequency: 2.00 GHz
- Operating System: Windows 10 Pro Education, version 22H2.
- This configuration suggests a system with limited RAM and a moderately-powered processor, suitable for basic computing tasks but may encounter performance constraints with resource-intensive applications.

SOFTWARE REQUIRMENT

- Python: Programming language used for development.
- Flask: Web framework for building the application
- Pandas: Data manipulation library for handling datasets
- NumPy: Library for numerical computations
- Pickle: Module for serializing and deserializing Python objects.
- HTML/CSS: For front-end design of the web application.
- Web browser:To access and interact with the web application.

OBJECTIVES

- Python is utilized for its versatility and extensive libraries.
- Flask constructs the web application, handling HTTP requests and HTML rendering
- pandas efficiently manipulates tabular data using DataFrame objects.
- NumPy facilitates numerical computations with array operations.
- Seaborn enables insightful data analysis and visualization through various plots.
- Joblib and Pickle ensure model serialization for saving and loading trained machine learning model
- HTML/CSS create an interactive and visually appealing front-end design
- Users interact with the application via a web browser, inputting data and receiving predictions.

Optional:

- Jupyter Notebook enhances exploratory capabilities with interactive data analysis and code development.
- Key objects such as DataFrame, Scaler, Ridge Regressor, Flask App, Model Object (Regressor), and HTML Form Objects play pivotal roles in data preprocessing, model training, web development, and user interaction.

DATA EXPLORATION

- Summary Statistics:
- Calculate basic statistics like mean, median, mode, standard deviation, etc
- Understand the central tendency and variability of the data.
- Visualization:
- Visualize data distribution, relationships, and trends
- Common plots include histograms, box plots, scatter plots, and heatmaps.
- Feature Analysis:
- Explore individual features to understand their distributions and detect outliers.
- Identify any missing or inconsistent values.
- Correlation Analysis:
- Calculate correlation coefficients between different features.
- Identify relationships and dependencies among features
- Visualize correlations using heat maps.
- Dimensionality Reduction:
- Use techniques like Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE)
- Visualize high-dimensional data in lower-dimensional spaces.
- Cluster Analysis:
- Apply clustering algorithms like K-means or hierarchical clustering
- Identify natural groupings or clusters within the data.
- Time Series Analysis
- Analyze trends, seasonality, and patterns over time
- Use techniques like decomposition and autocorrelation for time series data.

SCOPE

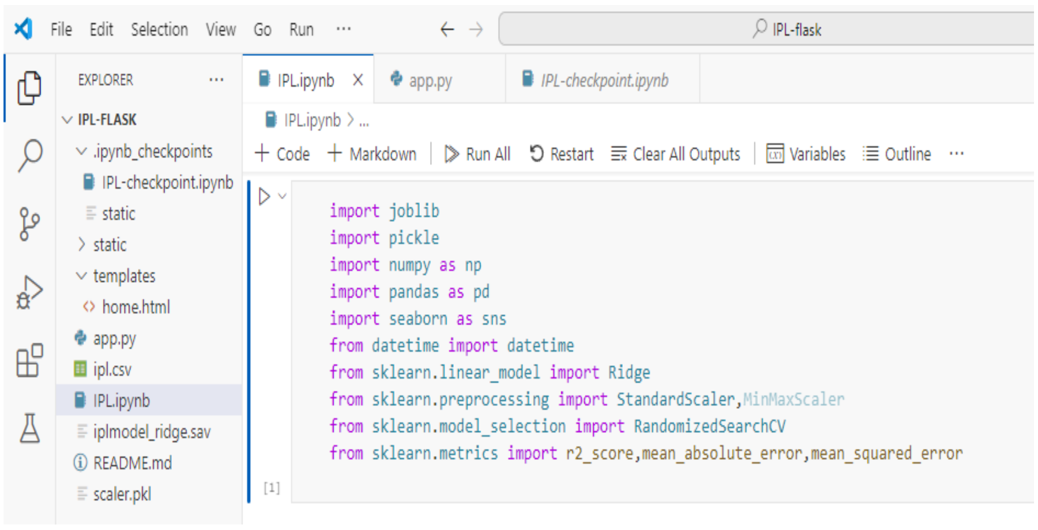
The code you provided seems to be preprocessing IPL cricket data, dropping unnecessary columns, converting date formats, filtering teams, and selecting specific overs.

To explain the scope of this code in simple terms, it's preparing the dataset for further analysis or modeling. Specifically, it's getting the data into a cleaner and more manageable format, focusing on relevant teams and overs for analysis or prediction.

In simpler terms, this code is like preparing ingredients before cooking a dish. It's ensuring that the data is in the right format and contains only the necessary information needed for the next steps, such as building a predictive model or conducting statistical analysis.

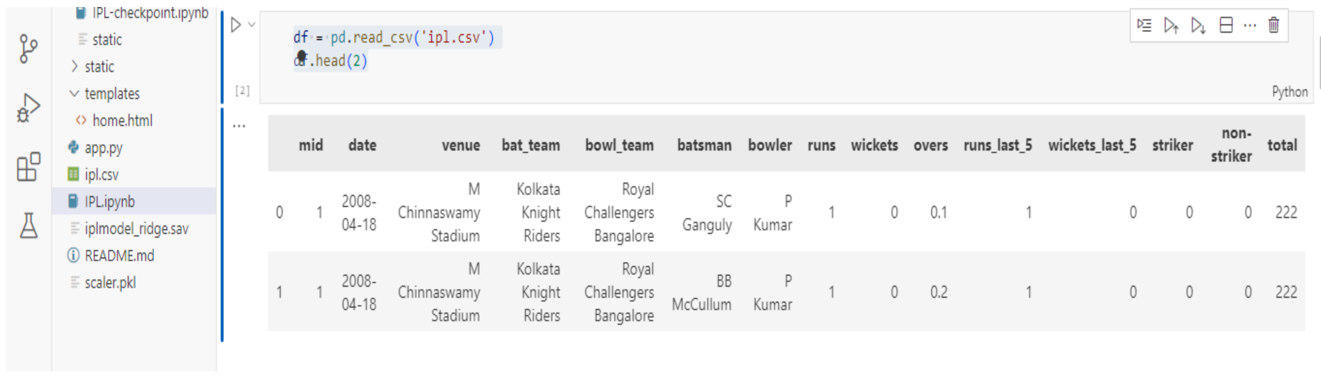
DATA PREPROCESSING

This line imports the pandas library as 'pd' and the matplotlib. pyplot library as 'plt', allowing access to their functions and methods throughout the script and reading the csv file.



```
import joblib
import pickle
import numpy as np
import pandas as pd
import seaborn as sns
from datetime import datetime
from sklearn.linear_model import Ridge
from sklearn.preprocessing import StandardScaler,MinMaxScaler
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
```

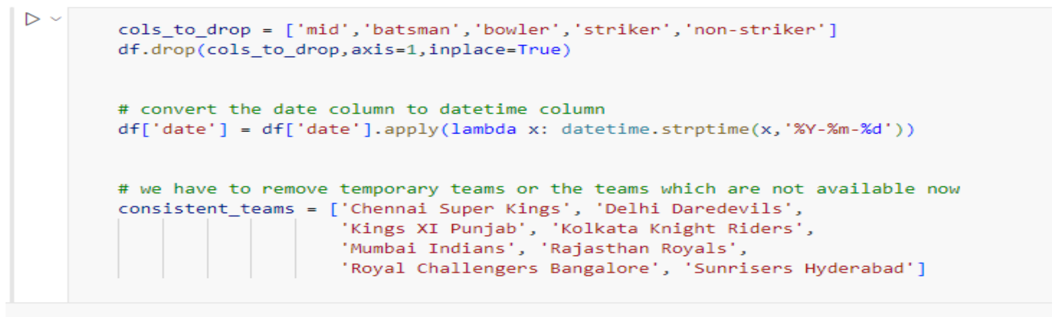
- Display the first five rows of the Data Frame 'df'.



```
df = pd.read_csv('ipl.csv')
df.head(2)
```

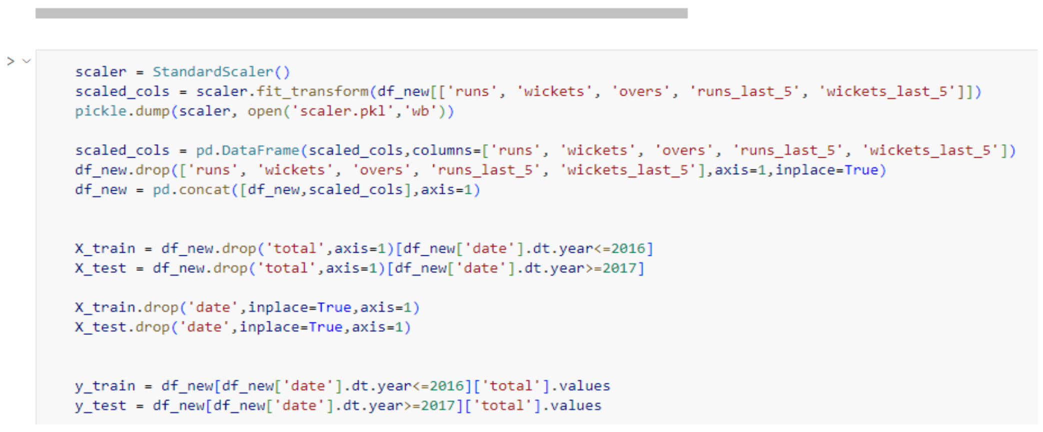
	mid	date	venue	bat_team	bowl_team	batsman	bowler	runs	wickets	overs	runs_last_5	wickets_last_5	striker	non-striker	total
0	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	SC Ganguly	P Kumar	1	0	0.1	1	0	0	0	222
1	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullum	P Kumar	1	0	0.2	1	0	0	0	222

- Select specific columns, rename one, and display the first five rows of the modified DataFrame 'df'.



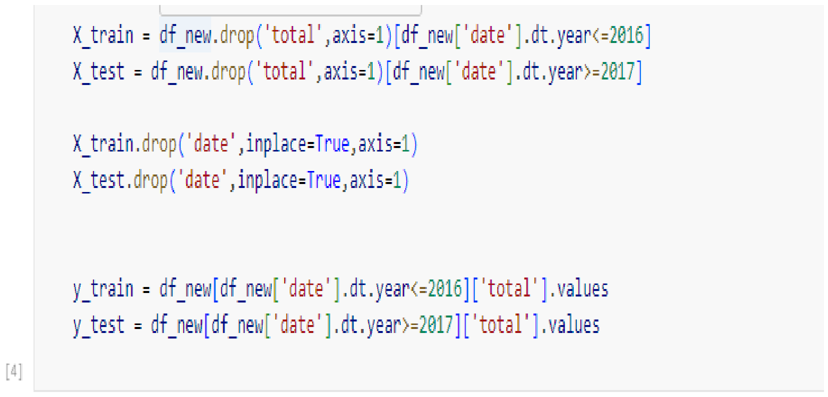
```
cols_to_drop = ['mid','batsman','bowler','striker','non-striker']
df.drop(cols_to_drop,axis=1,inplace=True)


# convert the date column to datetime column
df['date'] = df['date'].apply(lambda x: datetime.strptime(x,'%Y-%m-%d'))


# we have to remove temporary teams or the teams which are not available now
consistent_teams = ['Chennai Super Kings', 'Delhi Daredevils',
                    'Kings XI Punjab', 'Kolkata Knight Riders',
                    'Mumbai Indians', 'Rajasthan Royals',
                    'Royal Challengers Bangalore', 'Sunrisers Hyderabad']
```

Scaler model:



```
scaler = StandardScaler()
scaled_cols = scaler.fit_transform(df_new[['runs', 'wickets', 'overs', 'runs_last_5', 'wickets_last_5']])
pickle.dump(scaler, open('scaler.pkl','wb'))

scaled_cols = pd.DataFrame(scaled_cols,columns=['runs', 'wickets', 'overs', 'runs_last_5', 'wickets_last_5'])
df_new.drop(['runs', 'wickets', 'overs', 'runs_last_5', 'wickets_last_5'],axis=1,inplace=True)
df_new = pd.concat([df_new,scaled_cols],axis=1)


X_train = df_new.drop('total',axis=1)[df_new['date'].dt.year<=2016]
X_test = df_new.drop('total',axis=1)[df_new['date'].dt.year>=2017]

X_train.drop('date',inplace=True,axis=1)
X_test.drop('date',inplace=True,axis=1)


y_train = df_new[df_new['date'].dt.year<=2016]['total'].values
y_test = df_new[df_new['date'].dt.year>=2017]['total'].values
```

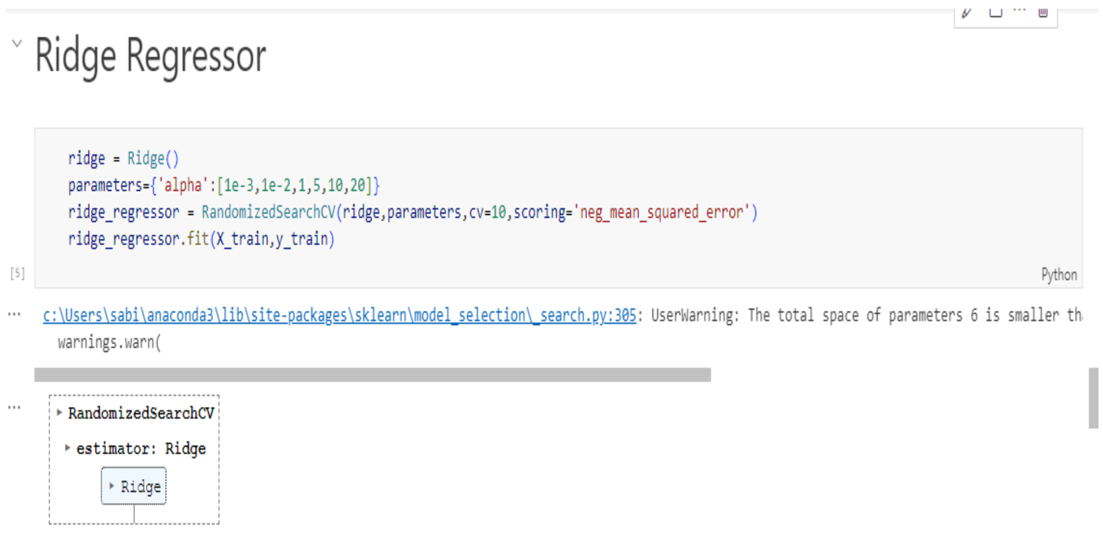
Build the train_test_split model:



```
X_train = df_new.drop('total',axis=1)[df_new['date'].dt.year<=2016]
X_test = df_new.drop('total',axis=1)[df_new['date'].dt.year>=2017]

X_train.drop('date',inplace=True,axis=1)
X_test.drop('date',inplace=True,axis=1)


y_train = df_new[df_new['date'].dt.year<=2016]['total'].values
y_test = df_new[df_new['date'].dt.year>=2017]['total'].values
```
[4]

MODEL BUILDING:

- linear regression model (linear_reg) using features (X) and target variable (y). Then, it predicts y values (y_pred) based on the trained model.



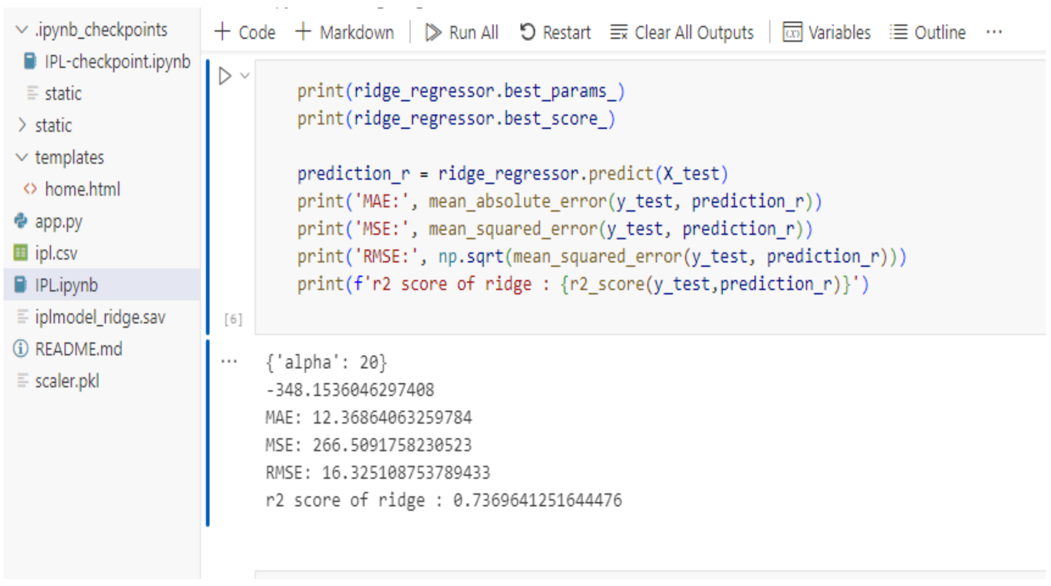
Ridge Regressor

```
ridge = Ridge()
parameters={'alpha':[1e-3,1e-2,1,5,10,20]}
ridge_regressor = RandomizedSearchCV(ridge,parameters,cv=10,scoring='neg_mean_squared_error')
ridge_regressor.fit(X_train,y_train)
```
[5] Python

```
c:\Users\sabi\anaconda3\lib\site-packages\sklearn\model_selection\_search.py:305: UserWarning: The total space of parameters 6 is smaller th
  warnings.warn(
```

```
RandomizedSearchCV
  estimator: Ridge
    Ridge
```

Prediction ridge reggressor model:

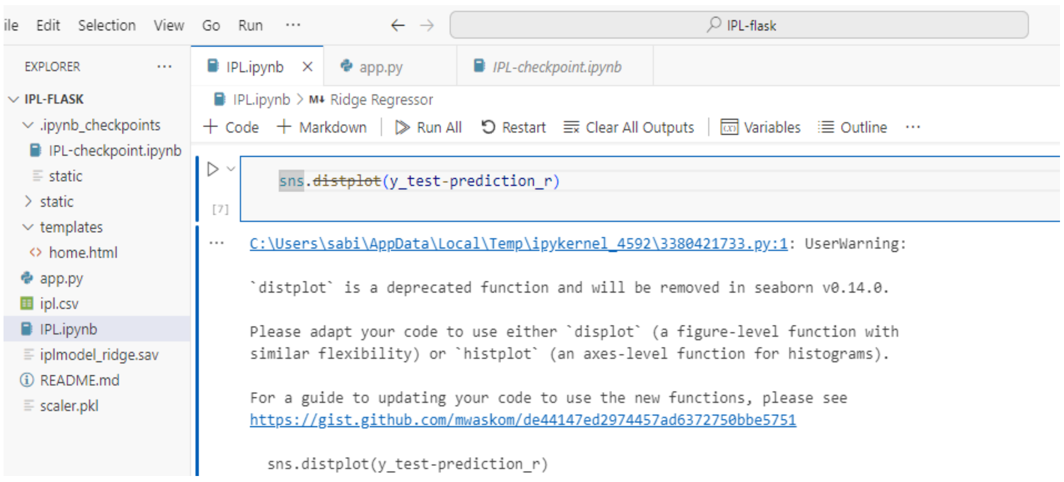


```
print(ridge_regressor.best_params_)
print(ridge_regressor.best_score_)

prediction_r = ridge_regressor.predict(X_test)
print('MAE:', mean_absolute_error(y_test, prediction_r))
print('MSE:', mean_squared_error(y_test, prediction_r))
print('RMSE:', np.sqrt(mean_squared_error(y_test, prediction_r)))
print(f'r2 score of ridge : {r2_score(y_test,prediction_r)}')
```

[6]

```
{'alpha': 20}
-348.1536046297408
MAE: 12.36864063259784
MSE: 266.5091758230523
RMSE: 16.325108753789433
r2 score of ridge : 0.7369641251644476
```

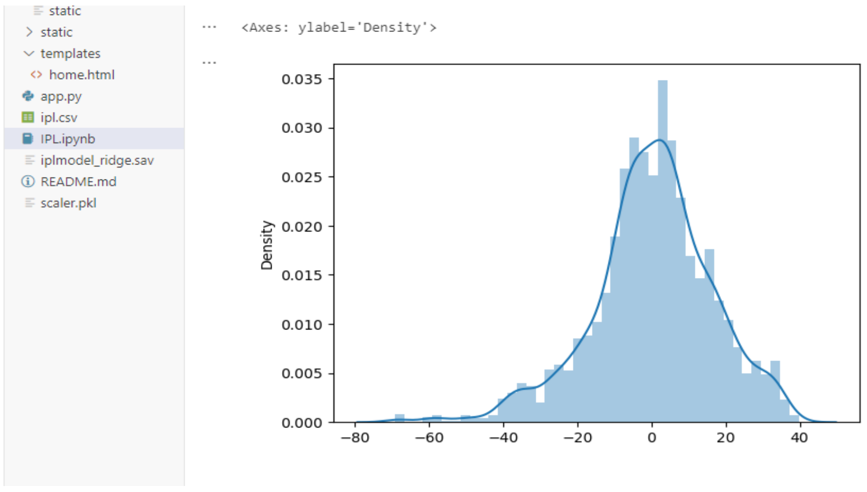
Visualization the prediction model:



```
sns.~~distplot~~(y_test-prediction_r)
```

[7]

```
C:\Users\sabi\AppData\Local\Temp\ipykernel_4592\3380421733.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(y_test-prediction_r)
```

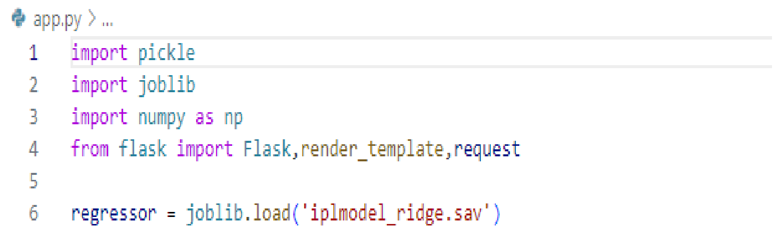


```
static
> static
v templates
  home.html
app.py
ipl.csv
IPL.ipynb
iplmodel_ridge.sav
README.md
scaler.pkl
```

```
<Axes: ylabel='Density'>
```

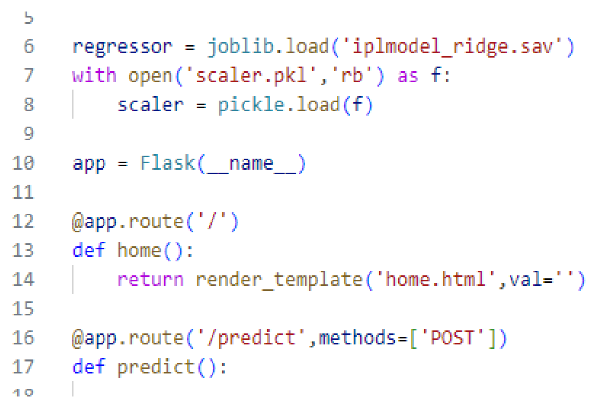
Creating a predict_page.py (app.py):

Import flask to visual app.py:



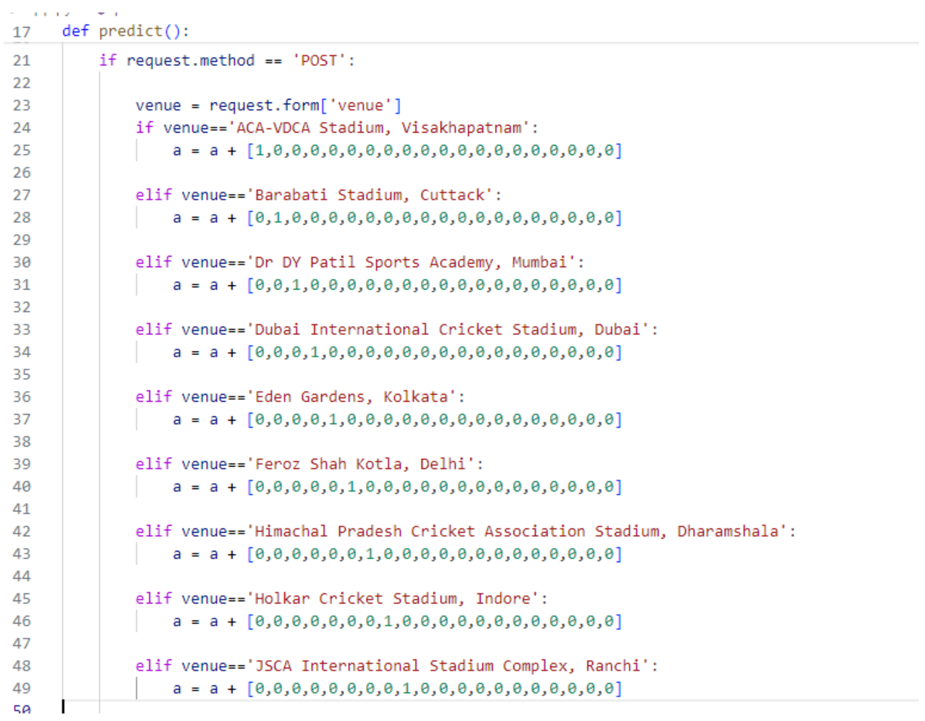
```
app.py > ...
import pickle
import joblib
import numpy as np
from flask import Flask,render_template,request

regressor = joblib.load('iplmodel_ridge.sav')
```

Model from pickle:

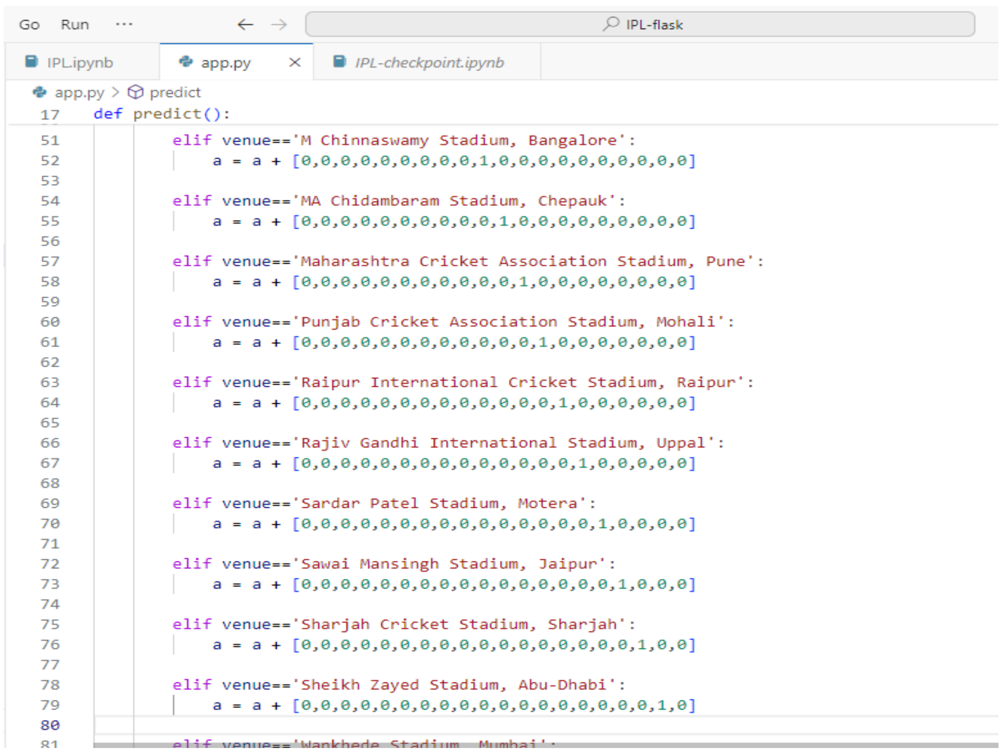


```
regressor = joblib.load('iplmodel_ridge.sav')
with open('scaler.pkl','rb') as f:
    scaler = pickle.load(f)

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('home.html',val='')

@app.route('/predict',methods=['POST'])
def predict():
```

Import the venue app.flask:

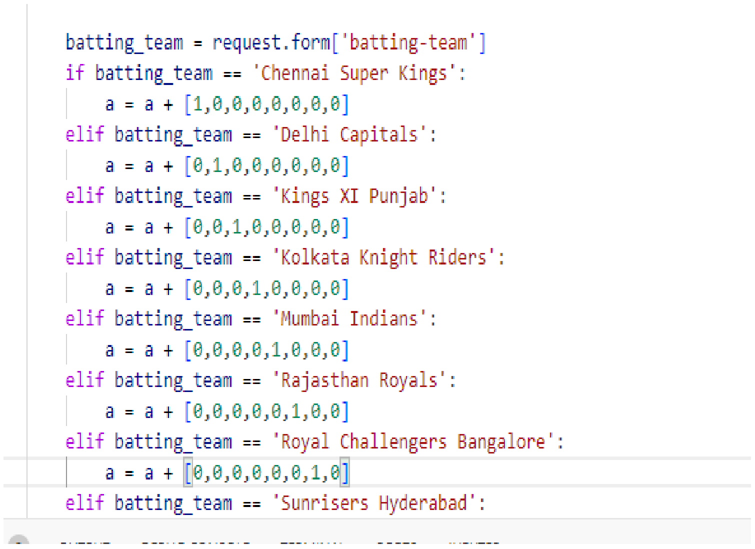


```python
def predict():
    if request.method == 'POST':

        venue = request.form['venue']
        if venue=='ACA-VDCA Stadium, Visakhapatnam':
            a = a + [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

        elif venue=='Barabati Stadium, Cuttack':
            a = a + [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

        elif venue=='Dr DY Patil Sports Academy, Mumbai':
            a = a + [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

        elif venue=='Dubai International Cricket Stadium, Dubai':
            a = a + [0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

        elif venue=='Eden Gardens, Kolkata':
            a = a + [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

        elif venue=='Feroz Shah Kotla, Delhi':
            a = a + [0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

        elif venue=='Himachal Pradesh Cricket Association Stadium, Dharamshala':
            a = a + [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0]

        elif venue=='Holkar Cricket Stadium, Indore':
            a = a + [0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0]

        elif venue=='JSCA International Stadium Complex, Ranchi':
            a = a + [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0]
```



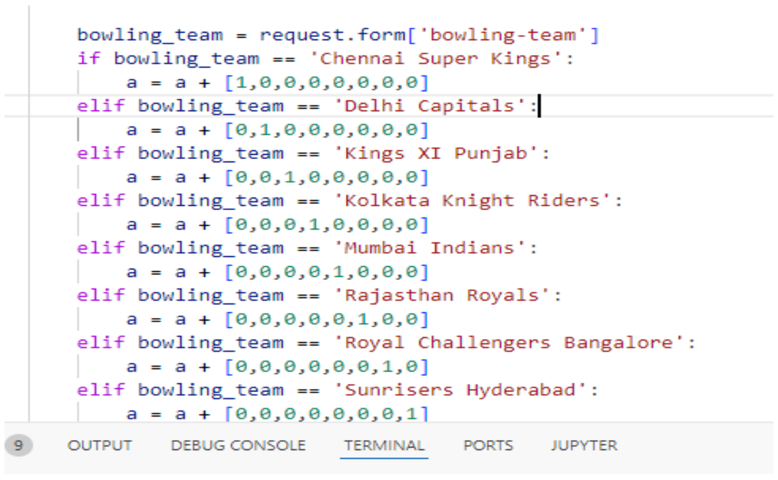
```python
def predict():
        elif venue=='M Chinnaswamy Stadium, Bangalore':
            a = a + [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0]

        elif venue=='MA Chidambaram Stadium, Chepauk':
            a = a + [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0]

        elif venue=='Maharashtra Cricket Association Stadium, Pune':
            a = a + [0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0]

        elif venue=='Punjab Cricket Association Stadium, Mohali':
            a = a + [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]

        elif venue=='Raipur International Cricket Stadium, Raipur':
            a = a + [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0]

        elif venue=='Rajiv Gandhi International Stadium, Uppal':
            a = a + [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0]

        elif venue=='Sardar Patel Stadium, Motera':
            a = a + [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0]

        elif venue=='Sawai Mansingh Stadium, Jaipur':
            a = a + [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0]

        elif venue=='Sharjah Cricket Stadium, Sharjah':
            a = a + [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0]

        elif venue=='Sheikh Zayed Stadium, Abu-Dhabi':
            a = a + [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0]

        elif venue=='Wankhede Stadium, Mumbai':
```

Batting teams :



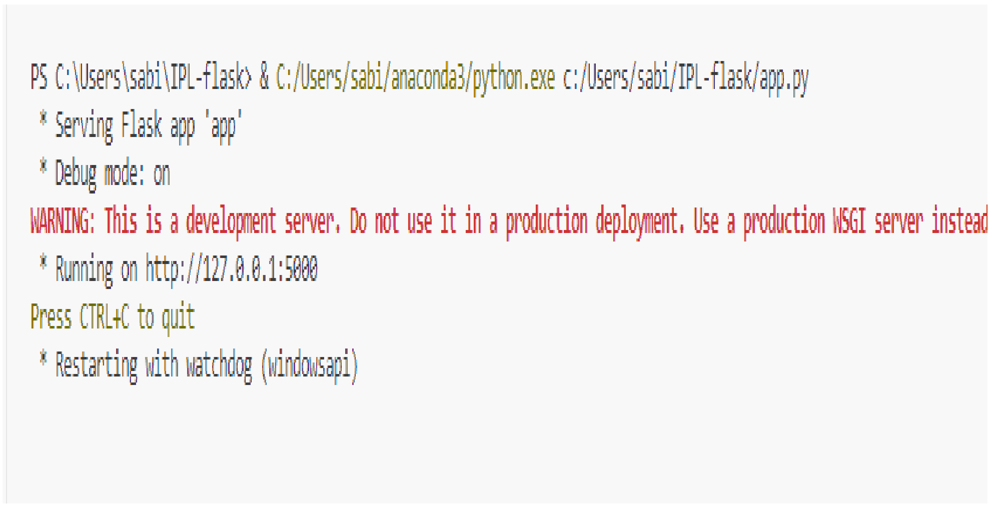
```python
batting_team = request.form['batting-team']
if batting_team == 'Chennai Super Kings':
    a = a + [1,0,0,0,0,0,0,0]
elif batting_team == 'Delhi Capitals':
    a = a + [0,1,0,0,0,0,0,0]
elif batting_team == 'Kings XI Punjab':
    a = a + [0,0,1,0,0,0,0,0]
elif batting_team == 'Kolkata Knight Riders':
    a = a + [0,0,0,1,0,0,0,0]
elif batting_team == 'Mumbai Indians':
    a = a + [0,0,0,0,1,0,0,0]
elif batting_team == 'Rajasthan Royals':
    a = a + [0,0,0,0,0,1,0,0]
elif batting_team == 'Royal Challengers Bangalore':
    a = a + [0,0,0,0,0,0,1,0]
elif batting_team == 'Sunrisers Hyderabad':
```

Bowling teams:



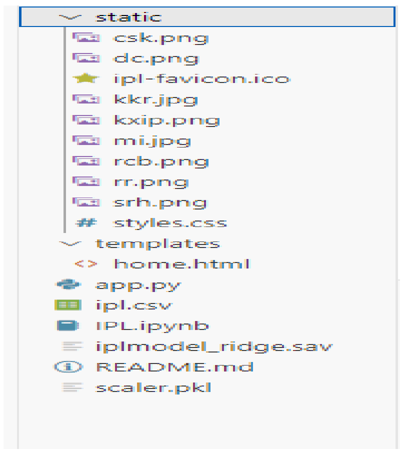
```
bowling_team = request.form['bowling-team']
if bowling_team == 'Chennai Super Kings':
    a = a + [1,0,0,0,0,0,0,0]
elif bowling_team == 'Delhi Capitals':
    a = a + [0,1,0,0,0,0,0,0]
elif bowling_team == 'Kings XI Punjab':
    a = a + [0,0,1,0,0,0,0,0]
elif bowling_team == 'Kolkata Knight Riders':
    a = a + [0,0,0,1,0,0,0,0]
elif bowling_team == 'Mumbai Indians':
    a = a + [0,0,0,0,1,0,0,0]
elif bowling_team == 'Rajasthan Royals':
    a = a + [0,0,0,0,0,1,0,0]
elif bowling_team == 'Royal Challengers Bangalore':
    a = a + [0,0,0,0,0,0,1,0]
elif bowling_team == 'Sunrisers Hyderabad':
    a = a + [0,0,0,0,0,0,0,1]
```

Prediction on webpage:



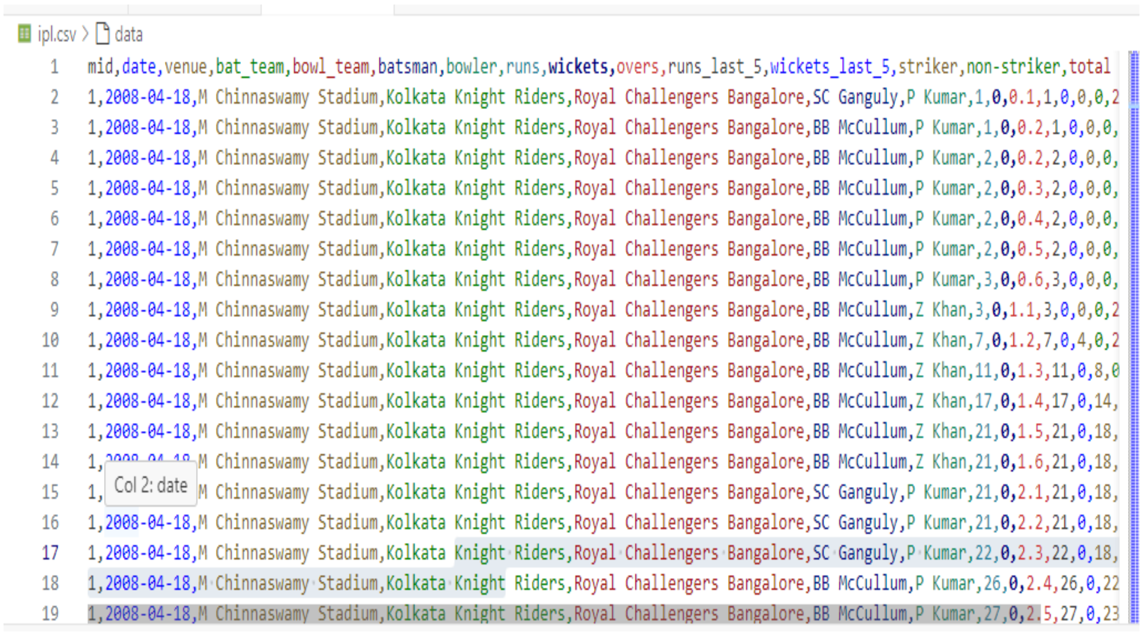
```
PS C:\Users\sabi\IPL-flask> & C:/Users/sabi/anaconda3/python.exe c:/Users/sabi/IPL-flask/app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
```

Static model:



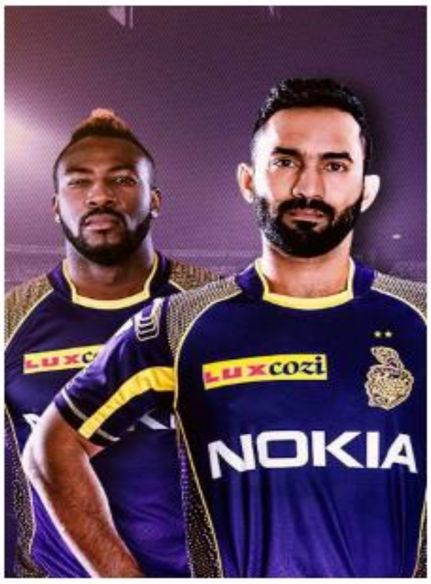
```
static
    csk.png
    dc.png
    ipl-favicon.ico
    kkr.jpg
    kxip.png
    mi.jpg
    rcb.png
    rr.png
    srh.png
    styles.css
templates
    home.html
app.py
ipl.csv
IPL.ipynb
iplmodel_ridge.sav
README.md
scaler.pkl
```

Ipl.csv dataset :

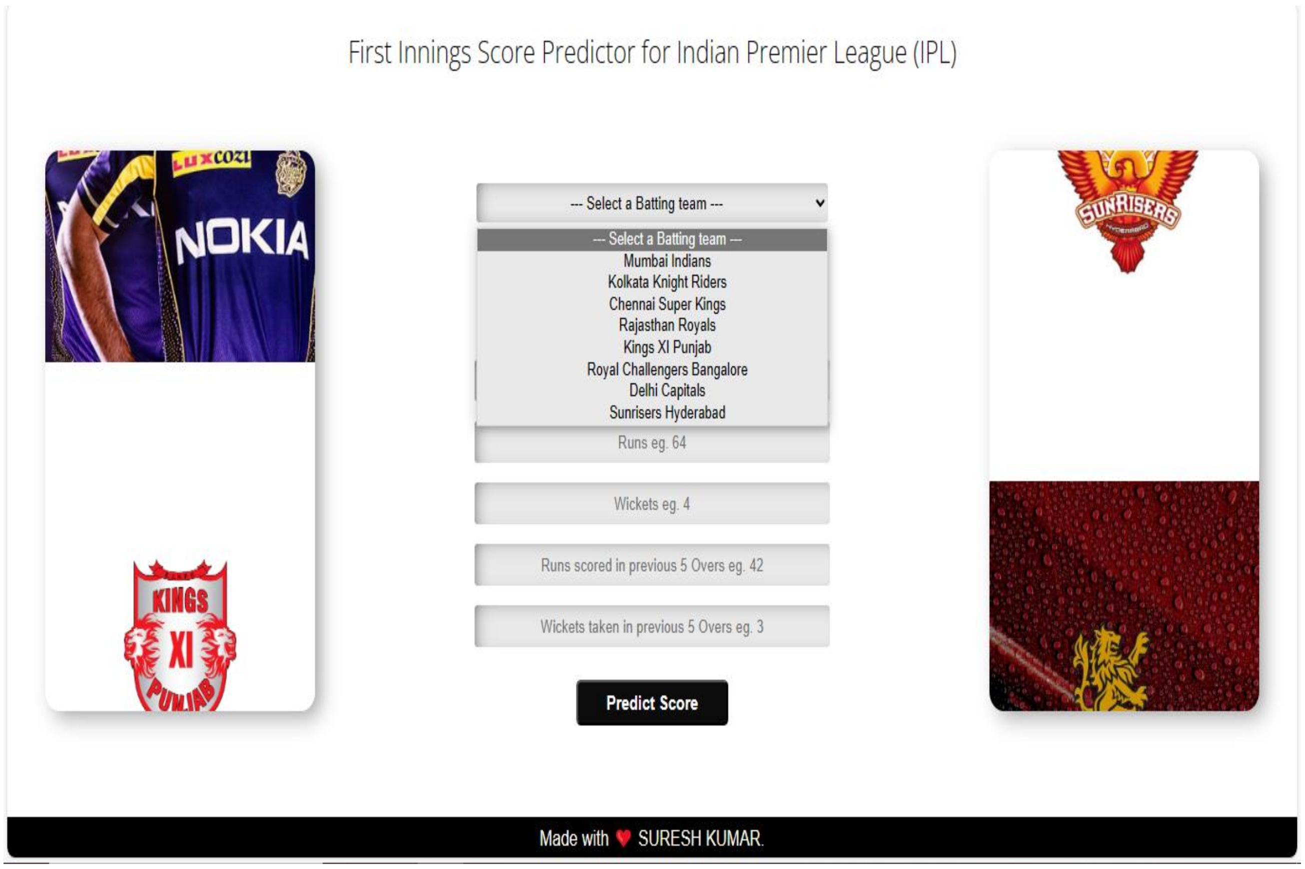


```
mid,date,venue,bat_team,bowl_team,batsman,bowler,runs,wickets,overs,runs_last_5,wickets_last_5,striker,non-striker,total
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,SC Ganguly,P Kumar,1,0,0.1,1,0,0,0,2
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,P Kumar,1,0,0.2,1,0,0,0,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,P Kumar,2,0,0.2,2,0,0,0,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,P Kumar,2,0,0.3,2,0,0,0,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,P Kumar,2,0,0.4,2,0,0,0,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,P Kumar,2,0,0.5,2,0,0,0,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,P Kumar,3,0,0.6,3,0,0,0,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,Z Khan,3,0,1.1,3,0,0,0,2
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,Z Khan,7,0,1.2,7,0,4,0,2
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,Z Khan,11,0,1.3,11,0,8,0
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,Z Khan,17,0,1.4,17,0,14,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,Z Khan,21,0,1.5,21,0,18,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,Z Khan,21,0,1.6,21,0,18,
1,[illegible]M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,SC Ganguly,P Kumar,21,0,2.1,21,0,18,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,SC Ganguly,P Kumar,21,0,2.2,21,0,18,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,SC Ganguly,P Kumar,22,0,2.3,22,0,18,
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,P Kumar,26,0,2.4,26,0,22
1,2008-04-18,M Chinnaswamy Stadium,Kolkata Knight Riders,Royal Challengers Bangalore,BB McCullum,P Kumar,27,0,2.5,27,0,23
```

Static img:


LUXcozi
NOKIA

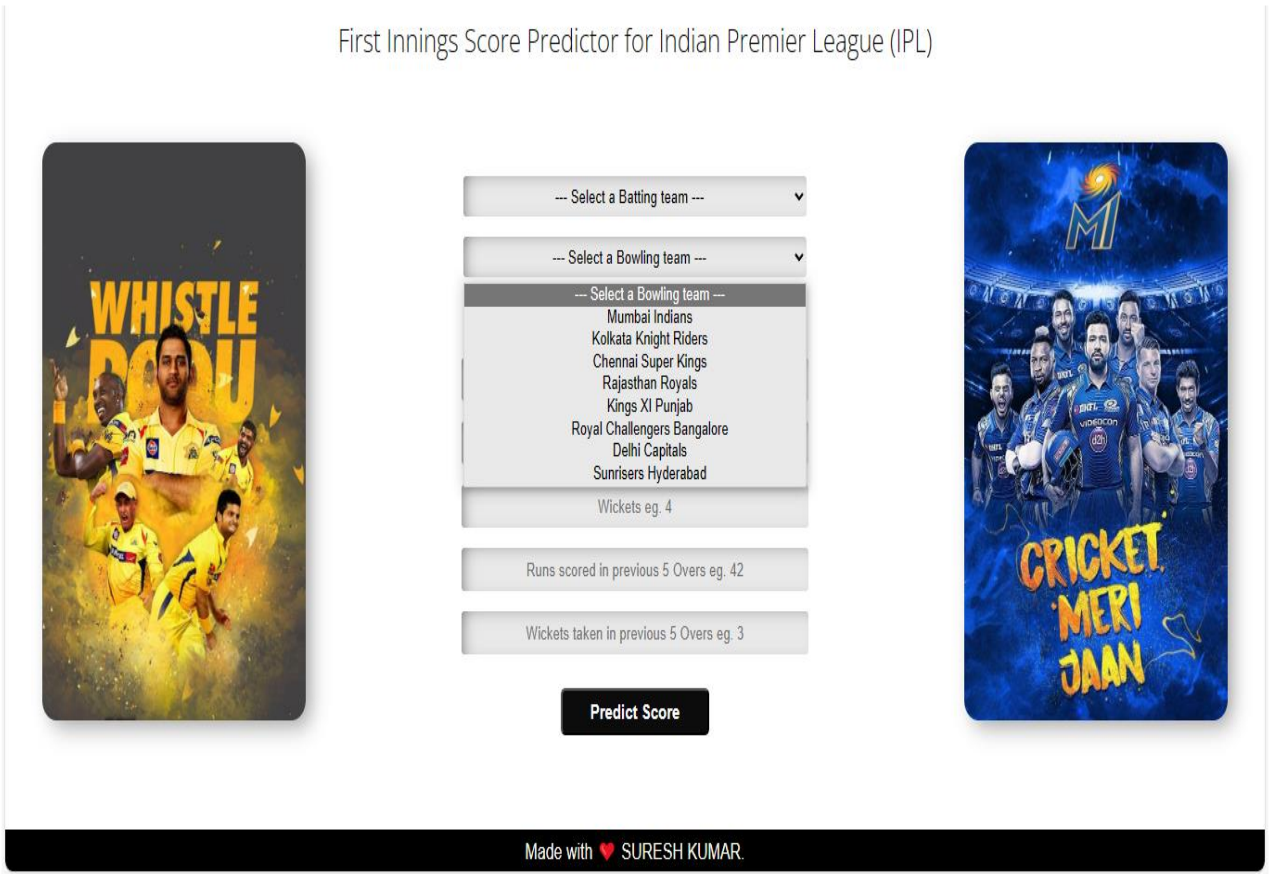
1.Select the batting team :



First Innings Score Predictor for Indian Premier League (IPL)

--- Select a Batting team ---
--- Select a Batting team ---
Mumbai Indians
Kolkata Knight Riders
Chennai Super Kings
Rajasthan Royals
Kings XI Punjab
Royal Challengers Bangalore
Delhi Capitals
Sunrisers Hyderabad
Runs eg. 64
Wickets eg. 4
Runs scored in previous 5 Overs eg. 42
Wickets taken in previous 5 Overs eg. 3
Predict Score
Made with ♥ SURESH KUMAR.

2. select the bowling team:


First Innings Score Predictor for Indian Premier League (IPL)

--- Select a Batting team ---

--- Select a Bowling team ---

--- Select a Bowling team ---
Mumbai Indians
Kolkata Knight Riders
Chennai Super Kings
Rajasthan Royals
Kings XI Punjab
Royal Challengers Bangalore
Delhi Capitals
Sunrisers Hyderabad

Wickets eg. 4

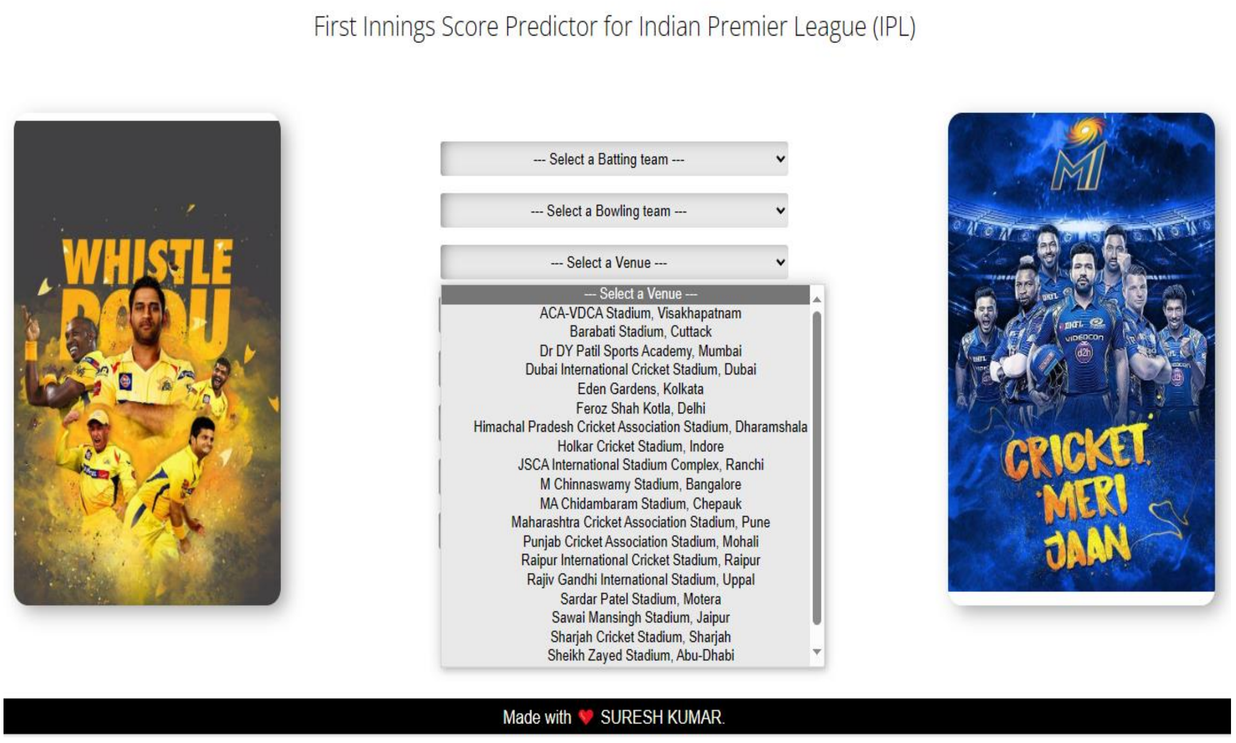
Runs scored in previous 5 Overs eg. 42

Wickets taken in previous 5 Overs eg. 3

Predict Score

Made with ♥ SURESH KUMAR.

3. select a venue:


First Innings Score Predictor for Indian Premier League (IPL)

--- Select a Batting team ---

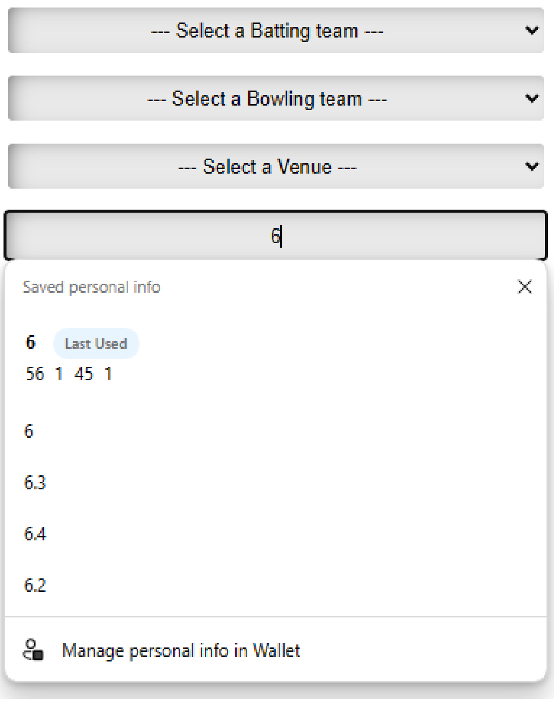
--- Select a Bowling team ---

--- Select a Venue ---

--- Select a Venue ---
ACA-VDCA Stadium, Visakhapatnam
Barabati Stadium, Cuttack
Dr DY Patil Sports Academy, Mumbai
Dubai International Cricket Stadium, Dubai
Eden Gardens, Kolkata
Feroz Shah Kotla, Delhi
Himachal Pradesh Cricket Association Stadium, Dharamshala
Holkar Cricket Stadium, Indore
JSCA International Stadium Complex, Ranchi
M Chinnaswamy Stadium, Bangalore
MA Chidambaram Stadium, Chepauk
Maharashtra Cricket Association Stadium, Pune
Punjab Cricket Association Stadium, Mohali
Raipur International Cricket Stadium, Raipur
Rajiv Gandhi International Stadium, Uppal
Sardar Patel Stadium, Motera
Sawai Mansingh Stadium, Jaipur
Sharjah Cricket Stadium, Sharjah
Sheikh Zayed Stadium, Abu-Dhabi

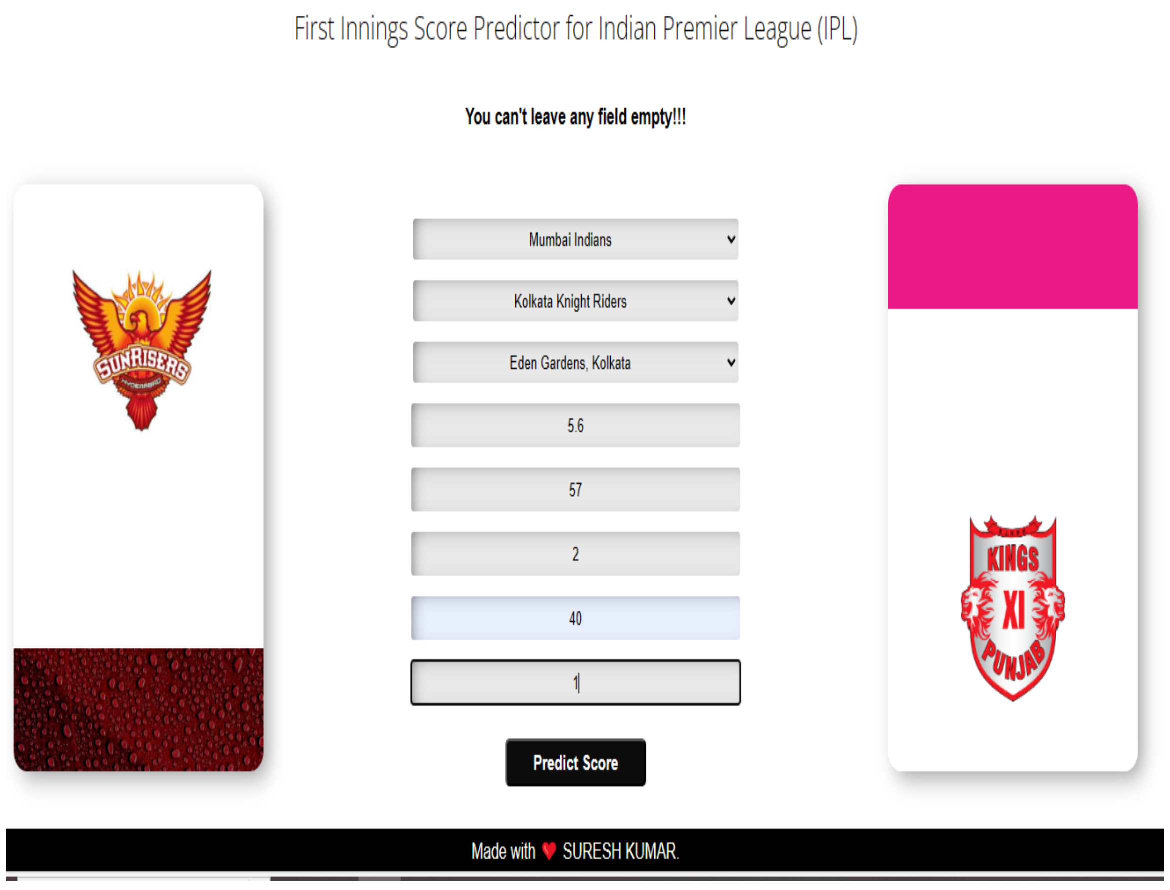
Made with ♥ SURESH KUMAR.

4. Put the no.of Overs:


--- Select a Batting team ---
--- Select a Bowling team ---
--- Select a Venue ---
6
Saved personal info
6 Last Used
56 1 45 1
6
6.3
6.4
6.2
Manage personal info in Wallet

5.Put the no. of runs

And wickets with in overs :


First Innings Score Predictor for Indian Premier League (IPL)

You can't leave any field empty!!!

Mumbai Indians

Kolkata Knight Riders

Eden Gardens, Kolkata

5.6

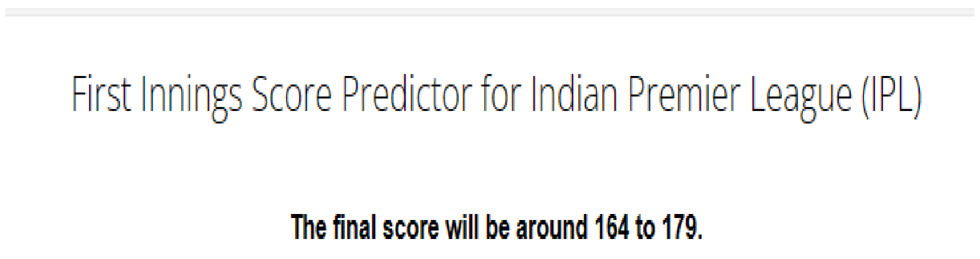
57

2

40

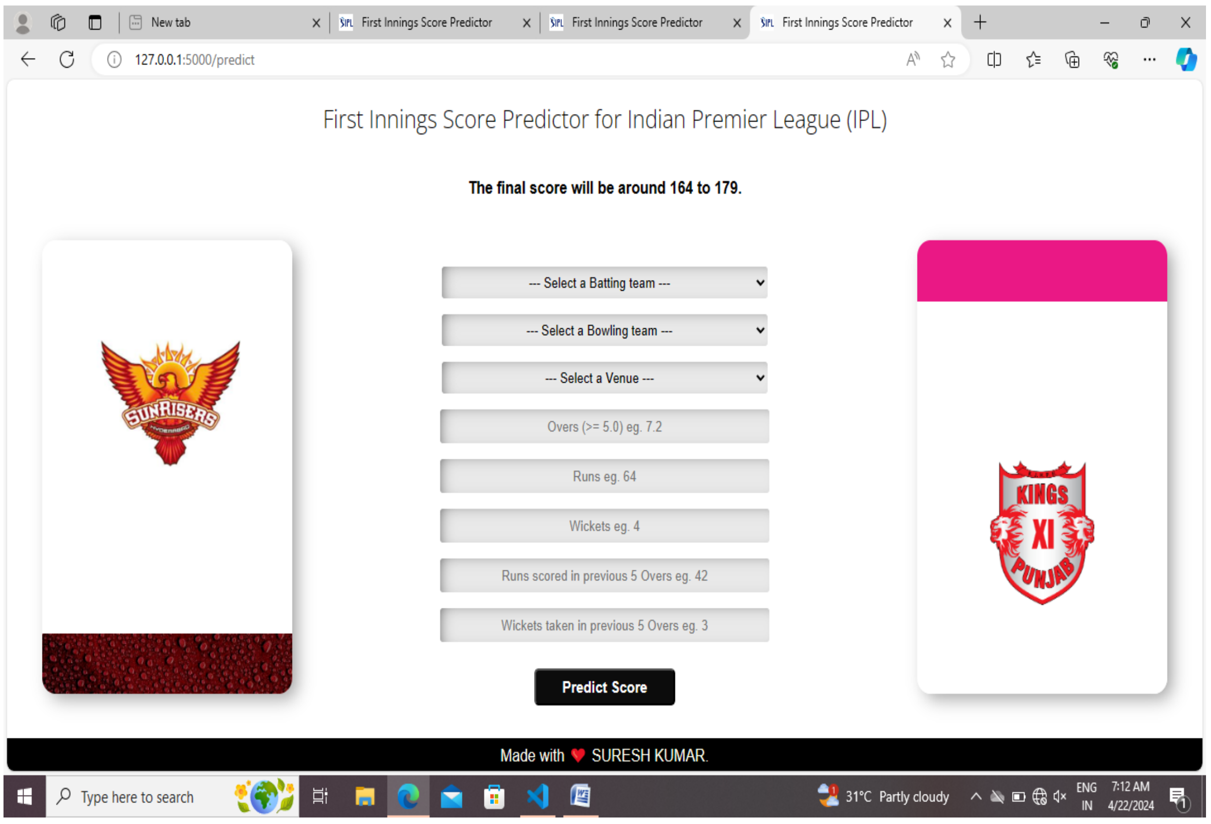
1

6. predict the 1st innings score:



First Innings Score Predictor for Indian Premier League (IPL)

The final score will be around 164 to 179.



New tab | First Innings Score Predictor | First Innings Score Predictor | First Innings Score Predictor

127.0.0.1:5000/predict

First Innings Score Predictor for Indian Premier League (IPL)

The final score will be around 164 to 179.

--- Select a Batting team ---
--- Select a Bowling team ---
--- Select a Venue ---
Overs (>= 5.0) eg. 7.2
Runs eg. 64
Wickets eg. 4
Runs scored in previous 5 Overs eg. 42
Wickets taken in previous 5 Overs eg. 3
Predict Score
Made with ♥ SURESH KUMAR.
Type here to search
31°C Partly cloudy
ENG IN
7:12 AM
4/22/2024

CONCLUSION:

This Python script seems to be part of a Flask web application for predicting cricket scores based on various input parameters such as venue, batting team, bowling team, overs, runs, wickets, runs scored in the previous 5 overs, and wickets lost in the previous 5 overs. Let's summarize the functionality and components of the script:

1. Imports: Necessary libraries like Flask, NumPy, joblib, and pickle are imported.

2. Loading Model and Scaler: The script loads a pre-trained Ridge regression model (`iplmodel_ridge.sav`) and a scaler object (`scaler.pkl`) using joblib and pickle, respectively. These are used to make predictions on the input data.

3. Flask App Setup: An instance of the Flask application is created.

4. Route Definition:

The home route renders an HTML template (`home.html`) and passes an empty string (`val=''`) to it.

predict`: This route handles POST requests for making predictions. It extracts input data from the form submitted by the user, processes it, makes predictions using the loaded model, and then renders the home template with a message displaying the predicted score range.

5. Venue, Batting Team, Bowling Team Encodin: The script encodes the venue, batting team, and bowling team selected by the user into a feature vector based on predefined mappings.

6. Input Validation: The script checks for empty input fields and ensures that the batting team and bowling team selected by the user are not the same.

7. Data Preparation: It prepares the input data by concatenating the encoded venue, batting team, and bowling team vectors with the numerical input features like overs, runs, wickets, runs scored in the previous 5 overs, and wickets lost in the previous 5 overs. The numerical features are also scaled using the loaded scaler object.

8. Prediction: The prepared data is used to make a prediction using the loaded regression model (`regressor`). The predicted score range is then displayed to the user.

9. Running the Flask App: The Flask application is run with debug mode enabled.

Conclusion:

This script provides a simple web interface for users to predict cricket scores based on various match parameters. It uses a pre-trained regression model to make predictions and Flask for handling web requests and rendering HTML templates. With this script, users can input match details and get an estimated score range as output.

REFRENCE:

The provided Python script appears to be a Flask web application for predicting cricket scores based on various match parameters. Unfortunately, I can't browse the internet for specific references. However, you can find similar tutorials or resources on building Flask web applications for machine learning predictions on platforms like Medium, Towards Data Science on Medium, Real Python, or GitHub repositories. Just search for Flask web applications for machine learning predictions, and you're likely to find helpful guides and examples.