EAST WEST UNIVERSITY

Project Title: T-20 cricket match score prediction model.

CSE366 - Artificial Intelligence

Project Report

Submitted by

Sumiaya Ahmed (2019-3-60-117)

Submitted to

Amit Mandal

Lecturer

Department of Computer Science & Engineering

Introduction: Cricket's Twenty20 (T-20) format has taken the sporting world by storm, offering high-octane entertainment. Accurate match score prediction in T-20 cricket is of paramount importance, given its fast-paced nature. This report presents our efforts to build a predictive model using decision trees and random forests to enhance score forecasting accuracy, mean absolute error, mean square error, R^2value catering to cricket enthusiasts, teams, and analysts alike.

Methodology:

Decision Tree: The decision tree algorithm is a fundamental machine-learning technique used in this project for predicting T-20 cricket match scores. They are a supervised learning method that works well for regression tasks.

For the decision tree model, several hyperparameters were considered and tuned to optimize performance. Key hyperparameters are: maximum depth, minimum samples per leaf, and mean squared error criterion. The dataset was divided into a training set and a testing set, to facilitate model training and evaluation.

The decision tree model was trained on the training data, where it learned to make predictions based on the features provided. Special attention was given to handling categorical variables if present in the dataset.

Random Forest: Random Forest, a type of ensemble learning, were chosen for their ability to improve prediction accuracy and mitigate overfitting by combining multiple decision trees. They are well-suited for regression tasks.

The random forest model required tuning of various hyperparameters to optimize performance. Key hyperparameters included: Number of Trees, Maximum Depth of Trees, Number of Features to Consider.

Similar to the decision tree model, the dataset was split into training and testing sets for random forest training and evaluation. This model was trained on the training data. We assessed feature importance within the random forest ensemble to understand which features contributed most to the predictions. Feature importance scores provided insight into the relevance of different variables in score prediction.

Data Selection: The dataset used in this project was sourced from Kaggle. The dataset consisted of essential attributes, including total_balls, match_id, batting_team, ball, run, and venue. The choice of data source, the specified time frame, and the relevant attributes were carefully considered to ensure the dataset's suitability for predicting T-20 cricket match scores.

This dataset, comprising a significant number of matches and pertinent match-related features, served as the foundation for the development and evaluation of our T-20 Cricket Match Score Prediction Model.

Data Processing:

In the data processing phase:

We addressed missing values, scaled numeric features, and encoded categorical variables. Prior to model training, the dataset underwent data preprocessing. This included:

Handling Missing Values:

Any missing data points were addressed using appropriate techniques, such as imputation or removal. We drop the attributes that are not important for prediction.

Data transformation:

We transform and prepare the dataset for analysis. Here, we calculated the sum of the last five overs, current run rate, wickets, current score, final score etc. which are new attributes essential for predicting the score.

Feature engineering techniques were applied to capture critical cricket aspects.

Data Scaling:

Numeric features were scaled to ensure uniformity in magnitude.

Categorical Encoding: Categorical variables, such as team names and venues, were encoded for Model compatibility.

The dataset was split into training (70%) and testing (30%) sets to facilitate model evaluation.

Model Training and Testing:

In the model training and testing phase, our T-20 Cricket Match Score Prediction Model underwent a systematic process to evaluate its predictive capabilities.

We trained the models, including Decision Trees and Random Forests, on the provided training dataset. These models learned to make predictions based on the relevant features of T-20 cricket matches.

The evaluation process took place on a separate testing dataset, distinct from the training data. This allowed us to assess how well the models could generalize and make predictions on unseen data.

Performance Evaluation:

Our model is trained by algorithms such as Random Forest and Decision Tree. We conducted a comprehensive performance evaluation of our T-20 Cricket Match Score Prediction Model, in terms of accuracy, R2 score, Mean Square error (MSE), and mean absolute error (MAE).

Decision Tree Model:

MAE: ~4.40

MSE: ~170.63

R²: ~0.84

Training Accuracy: 0.99874

Testing Accuracy: 0.845

Random Forest Model:

MAE: ~4.82

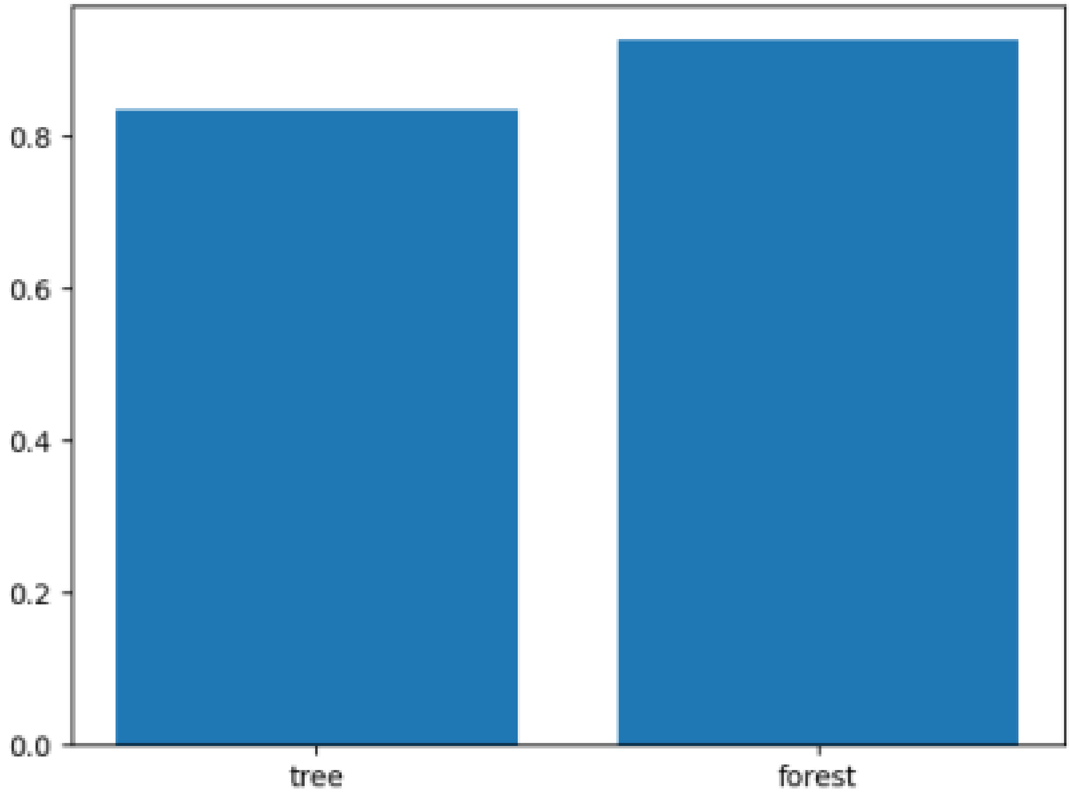
MSE: ~78.62

R²: ~0.92

Training Accuracy: 0.987

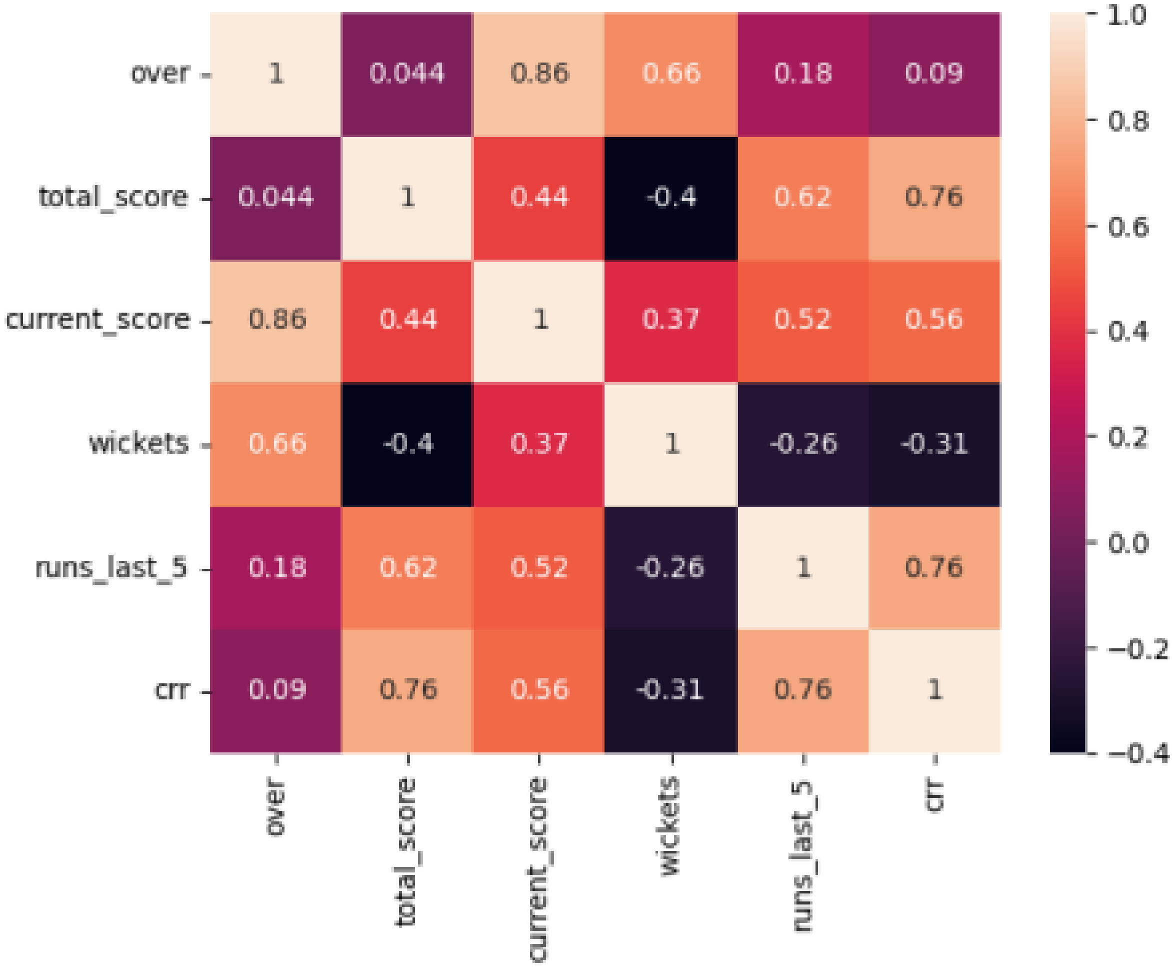
Testing Accuracy: 0.920

These metrics confirm that both models offer better predictions but Random Forest model exhibits slightly better accuracy and exceptional explanatory power, capturing around 92% of testing accuracy.



Data Visualization:

Correlation:



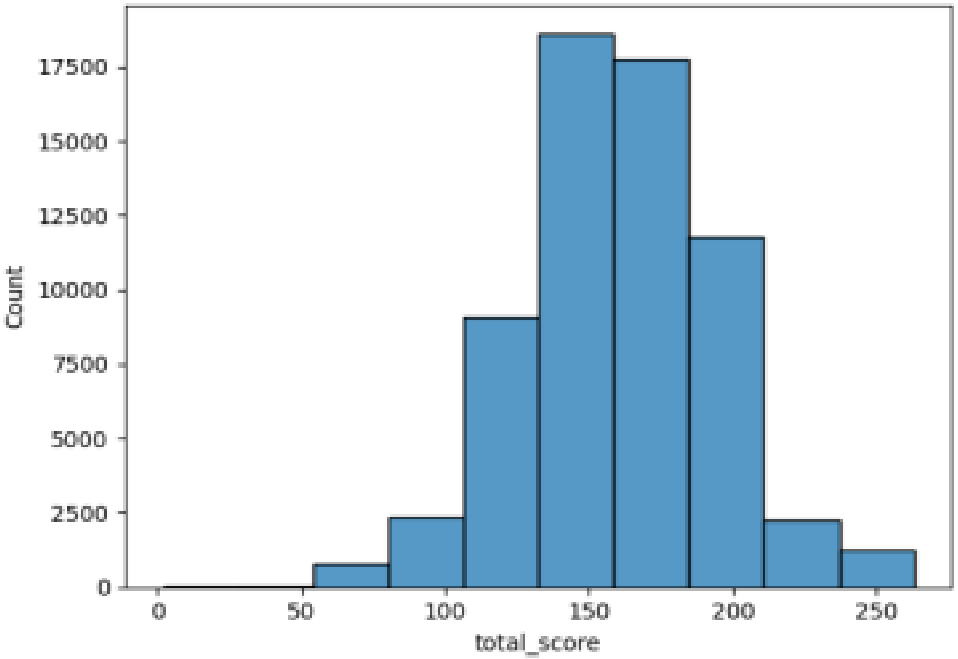
Here, in the correlation matrix, firstly, the current run rate has a very high positive correlation with the total score or final score which is 0.76. Secondly, final score has second-highest positive correlation with runs last five over. Current score and final score have moderate positive correlation and wickets have moderate negative correlation with final score.

Additionally, over and current score have very high positive correlation and so on.

Histograms:

Total score vs Number of matches

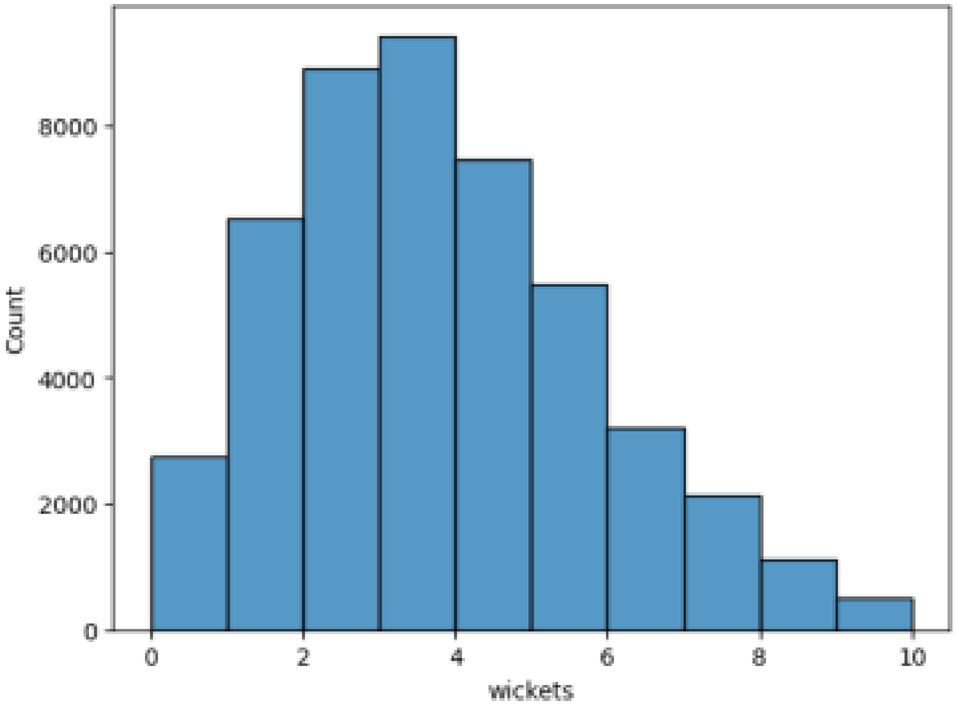
In the following histogram, most of the matches have scores between 110 and



190.

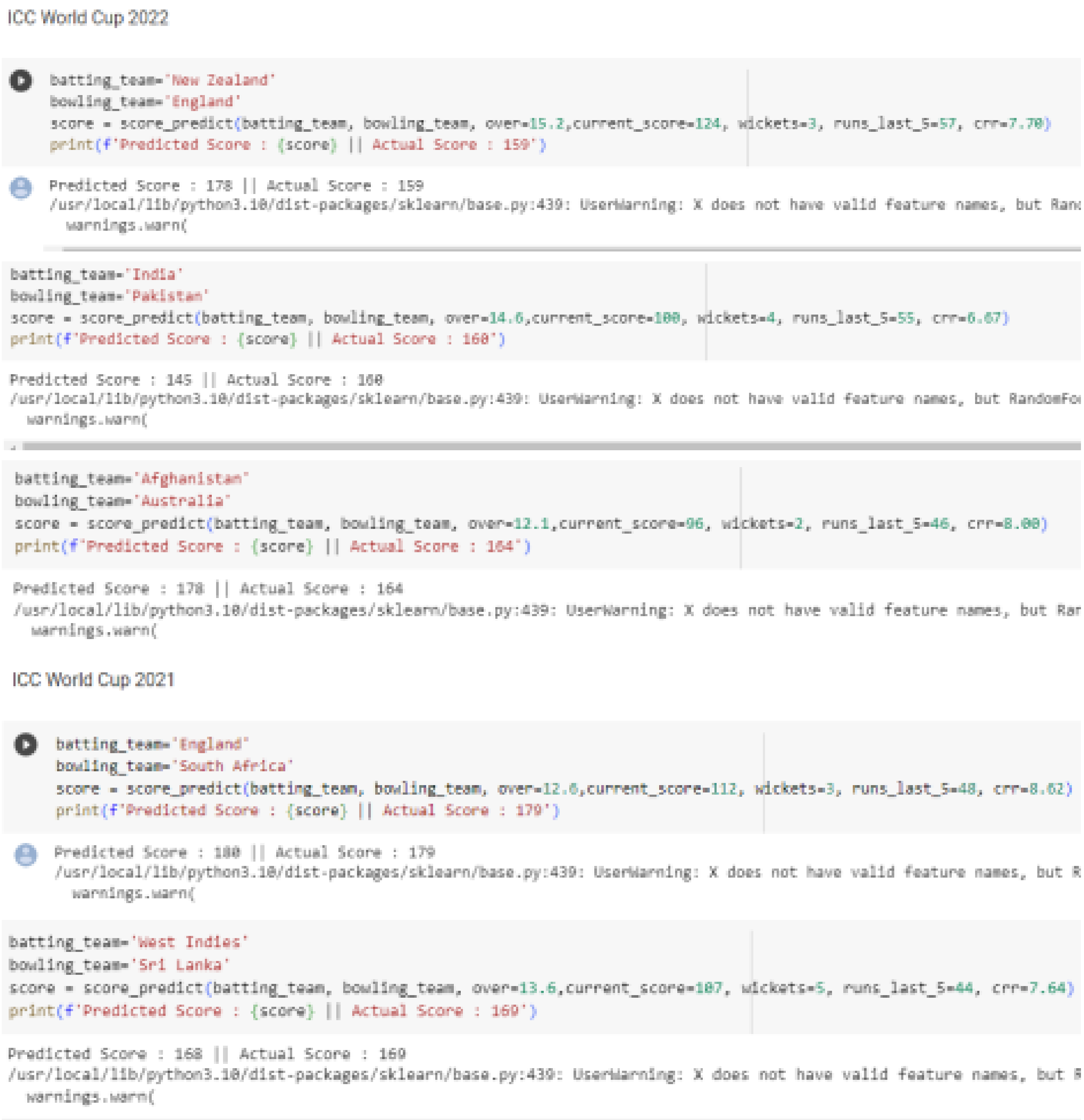
Wickets count vs Number of matches

In the following histogram, Most of the matches have wickets between 2 to 6.



Result:

We have tested our model for various match predictions. Some examples are: for ICC World Cup 2022, in the match between India vs Pakistan, for 14.6 over it predicted score of 145 whereas the actual score was 160. Again, for the match between England vs South Africa it predicted score is 180 whereas the actual score was 179.



ICC World Cup 2022

```
batting_team='New Zealand'
bowling_team='England'
score = score_predict(batting_team, bowling_team, over=15.2,current_score=124, wickets=3, runs_last_5=57, crr=7.70)
print(f'Predicted Score : {score} || Actual Score : 159')
```

```
Predicted Score : 178 || Actual Score : 159
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but Ranc
  warnings.warn(
```

```
batting_team='India'
bowling_team='Pakistan'
score = score_predict(batting_team, bowling_team, over=14.6,current_score=100, wickets=4, runs_last_5=55, crr=6.67)
print(f'Predicted Score : {score} || Actual Score : 160')
```

```
Predicted Score : 145 || Actual Score : 160
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but RandomFo
  warnings.warn(
```

```
batting_team='Afghanistan'
bowling_team='Australia'
score = score_predict(batting_team, bowling_team, over=12.1,current_score=96, wickets=2, runs_last_5=46, crr=8.00)
print(f'Predicted Score : {score} || Actual Score : 164')
```

```
Predicted Score : 178 || Actual Score : 164
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but Rar
  warnings.warn(
```

ICC World Cup 2021

```
batting_team='England'
bowling_team='South Africa'
score = score_predict(batting_team, bowling_team, over=12.6,current_score=112, wickets=3, runs_last_5=48, crr=8.62)
print(f'Predicted Score : {score} || Actual Score : 179')
```

```
Predicted Score : 180 || Actual Score : 179
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but R
  warnings.warn(
```

```
batting_team='West Indies'
bowling_team='Sri Lanka'
score = score_predict(batting_team, bowling_team, over=13.6,current_score=107, wickets=5, runs_last_5=44, crr=7.64)
print(f'Predicted Score : {score} || Actual Score : 169')
```

```
Predicted Score : 168 || Actual Score : 169
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but R
  warnings.warn(
```

Future Scope and conclusion: The objective of the model is to predict the match score of T 20 cricket. As we have not used venue, player selection in this model for T20 match score prediction, therefore, for the future scope , there can develop winning percentage, venue, player selection etc in the future model . Adding that, we have used decision tree and random forest. For future we can use any other regression models.

<u>**Code:**</u>

```
import numpy as np
import pandas as pd

df = pd.read_csv('t20i_info.csv')
df
df.shape
df.dtypes
df.columns
df.isnull().sum()
df.drop(columns=['city','venue','total_balls'],inplace=True)
df
df.isnull().sum()
df['batting_team'].unique()
df['bowling_team'].unique()
total_df = df.groupby('match_id').sum()['runs'].reset_index()
df = df.merge(total_df,on='match_id')
df
df['current_score'] = df.groupby('match_id').cumsum()['runs_x']
df
df.rename(columns = {'runs_y':'total_score'}, inplace = True)
df
df['player_dismissed'] = df['player_dismissed'].apply(lambda x:0 if x=='0' else
1) df['player_dismissed'] = df['player_dismissed'].astype('int')
df['wickets'] = df.groupby('match_id').cumsum()['player_dismissed']
df
groups = df.groupby('match_id')
match_ids = df['match_id'].unique()
last_five = []
for id in match_ids:
 last_five.extend(groups.get_group(id).rolling(window=30).sum()['runs_x'].values.tolist())
df['runs_last_5'] = last_five
df
df['over'] = df['ball'].apply(lambda x:str(x).split(".")[0])
df['balls'] = df['ball'].apply(lambda x:str(x).split(".")[1])
df
```

```
df['balls_bowled'] = (df['over'].astype('int')*6) +
df['balls'].astype('int') df['crr'] =
round((df['current_score']*6)/df['balls_bowled'],2)

import matplotlib.pyplot as plt
import seaborn as sns
sns.histplot(data = df, x='total_score', bins=10)
sns.histplot(data = df, x='wickets', bins=10)

df.columns
df.drop(columns=['runs_x','player_dismissed','balls_bowled','over','balls'],inplace=True
) df.drop(columns=['match_id'],inplace=True)
df.rename(columns = {'ball':'over'}, inplace = True)
df
df
df = df[df['over'] >= 5.0]
df
df.columns
df.shape
df.dtypes
from seaborn import heatmap
heatmap(data=df.corr(), annot=True)
df.columns
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
columnTransformer = ColumnTransformer([('encoder',
 OneHotEncoder(),
 [0, 1])],
 remainder='passthrough')
df= np.array(columnTransformer.fit_transform(df))
df.shape
cols = ['batting_team_Australia', 'batting_team_India', 'batting_team_Bangladesh',
'batting_team_New Zealand', 'batting_team_South Africa', 'batting_team_England',
 'batting_team_West Indies',
'batting_team_Afghanistan','batting_team_Pakistan','batting_team_Sri Lanka',

 'bowling_team_Australia', 'bowling_team_India', 'bowling_team_Bangladesh',
'bowling_team_New Zealand', 'bowling_team_South Africa', 'bowling_team_England',
'bowling_team_West Indies', 'bowling_team_Afghanistan', 'bowling_team_Pakistan',
'bowling_team_Sri Lanka','over',
 'total_score','current_score', 'wickets', 'runs_last_5', 'crr']
df = pd.DataFrame(df, columns=cols)
df.head()
X = df.drop(['total_score'], axis=1)
y = df['total_score']
from sklearn.model_selection import train_test_split
# Split the data for train and test
X_train,X_test,y_train,y_test =
train_test_split(X,y,train_size=0.7,random_state=100) print(X_train.shape)
print(X_test.shape)
models = dict()

from sklearn.tree import DecisionTreeRegressor
tree = DecisionTreeRegressor()
# Train Model
tree.fit(X_train,y_train)
# Predict using test data
y_predict = tree.predict(X_test)
y_predict
# Predicted Score of train data
train_accuracy = tree.score(X_train, y_train)
train_accuracy
d_test_accuracy = tree.score(X_test, y_test)
d_test_accuracy
models["tree"] = d_test_accuracy
print('Mean absolute error = ', mean_absolute_error(y_test,y_predict))
print('Mean square error = ', mean_squared_error(y_test,y_predict))
print('R2 Score = ', r2_score(y_test,y_predict))

from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor()
# Train Model
forest.fit(X_train,y_train)
# Predict using test data
```

```
y_predict = forest.predict(X_test)
y_predict
# Predicted Score of train data
train_accuracy = forest.score(X_train, y_train)
train_accuracy
f_test_accuracy = forest.score(X_test, y_test)
f_test_accuracy
models["forest"] = f_test_accuracy
print('Mean absolute error = ',
mean_absolute_error(y_test,y_predict)) print('Mean square error = ',
mean_squared_error(y_test,y_predict))
print('R2 Score = ', r2_score(y_test,y_predict))
models.values()
import matplotlib.pyplot as plt
model_names = list(models.keys())
accuracy = list(map(float, models.values()))
# creating the bar plot
plt.bar(model_names, accuracy)
df.columns
df.head()
def score_predict(batting_team, bowling_team,over, current_score, wickets, runs_last_5, crr,
model=forest):
 prediction_array = []
 # Batting Team
 if batting_team == 'Australia':
 prediction_array = prediction_array + [1,0,0,0,0,0,0,0,0,0]
 elif batting_team == 'India':
 prediction_array = prediction_array + [0,1,0,0,0,0,0,0,0,0]
 elif batting_team == 'Bangladesh':
 prediction_array = prediction_array + [0,0,1,0,0,0,0,0,0,0]
 elif batting_team == 'New Zealand':
 prediction_array = prediction_array + [0,0,0,1,0,0,0,0,0,0]
 elif batting_team == 'South Africa':
 prediction_array = prediction_array + [0,0,0,0,1,0,0,0,0,0]
 elif batting_team == 'England':
 prediction_array = prediction_array + [0,0,0,0,0,1,0,0,0,0]
 elif batting_team == 'West Indies':
```

```
 prediction_array = prediction_array + [0,0,0,0,0,0,1,0,0,0]
 elif batting_team == 'Afghanistan':
 prediction_array = prediction_array + [0,0,0,0,0,0,0,1,0,0]
 elif batting_team == 'Pakistan':
 prediction_array = prediction_array + [0,0,0,0,0,0,0,0,1,0]
 elif batting_team == 'Sri Lanka':
 prediction_array = prediction_array + [0,0,0,0,0,0,0,0,0,1]
 # Bowling Team
 if bowling_team == 'Australia':
 prediction_array = prediction_array + [1,0,0,0,0,0,0,0,0,0]
 elif bowling_team == 'India':
 prediction_array = prediction_array + [0,1,0,0,0,0,0,0,0,0]
 elif bowling_team == 'Bangladesh':
 prediction_array = prediction_array + [0,0,1,0,0,0,0,0,0,0]
 elif bowling_team == 'New Zealand':
 prediction_array = prediction_array + [0,0,0,1,0,0,0,0,0,0]
 elif bowling_team == 'South Africa':
 prediction_array = prediction_array + [0,0,0,0,1,0,0,0,0,0]
 elif bowling_team == 'England':
 prediction_array = prediction_array + [0,0,0,0,0,1,0,0,0,0]
 elif bowling_team == 'West Indies':
 prediction_array = prediction_array + [0,0,0,0,0,0,1,0,0,0]
 elif bowling_team == 'Afghanistan':
 prediction_array = prediction_array + [0,0,0,0,0,0,0,1,0,0]
 elif bowling_team == 'Pakistan':
 prediction_array = prediction_array + [0,0,0,0,0,0,0,0,1,0]
 elif bowling_team == 'Sri Lanka':
 prediction_array = prediction_array + [0,0,0,0,0,0,0,0,0,1]
 prediction_array = prediction_array + [over, current_score, wickets, runs_last_5, crr]
prediction_array = np.array([prediction_array])
 pred = model.predict(prediction_array)
 return int(round(pred[0]))

batting_team='New Zealand'
bowling_team='England'
score = score_predict(batting_team, bowling_team, over=15.2,current_score=124, wickets=3,
runs_last_5=57, crr=7.70)
```

```
print(f'Predicted Score : {score} || Actual Score : 159')
batting_team='India'
bowling_team='Pakistan'
score = score_predict(batting_team, bowling_team, over=14.6,current_score=100, wickets=4,
runs_last_5=55, crr=6.67)
print(f'Predicted Score : {score} || Actual Score : 160')
batting_team='Afghanistan'
bowling_team='Australia'
score = score_predict(batting_team, bowling_team, over=12.1,current_score=96, wickets=2,
runs_last_5=46, crr=8.00)
print(f'Predicted Score : {score} || Actual Score : 164')
batting_team='England'
bowling_team='South Africa'
score = score_predict(batting_team, bowling_team, over=12.6,current_score=112, wickets=3,
runs_last_5=48, crr=8.62)
print(f'Predicted Score : {score} || Actual Score : 179')
batting_team='West Indies'
bowling_team='Sri Lanka'
score = score_predict(batting_team, bowling_team, over=13.6,current_score=107, wickets=5,
runs_last_5=44, crr=7.64)
print(f'Predicted Score : {score} || Actual Score : 169')
```