IPL Match Prediction

ASHIN R

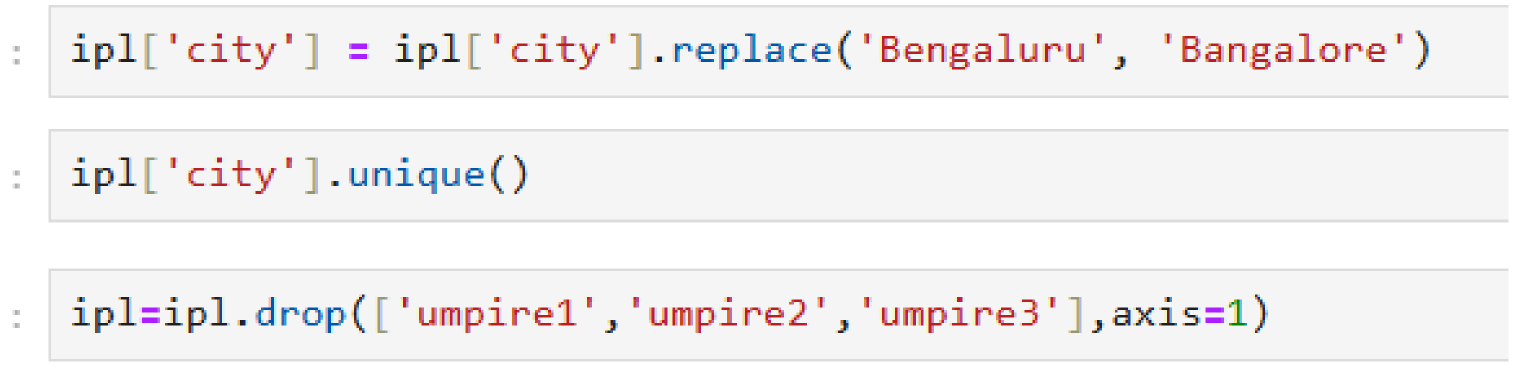
RP-33

OVERVIEW

- Data Cleaning & Preprocessing
- Exploratory Data Analysis
- Model Building
- Model Evaluation
- Hyperparameter Tuning

Data Cleaning & Preprocessing

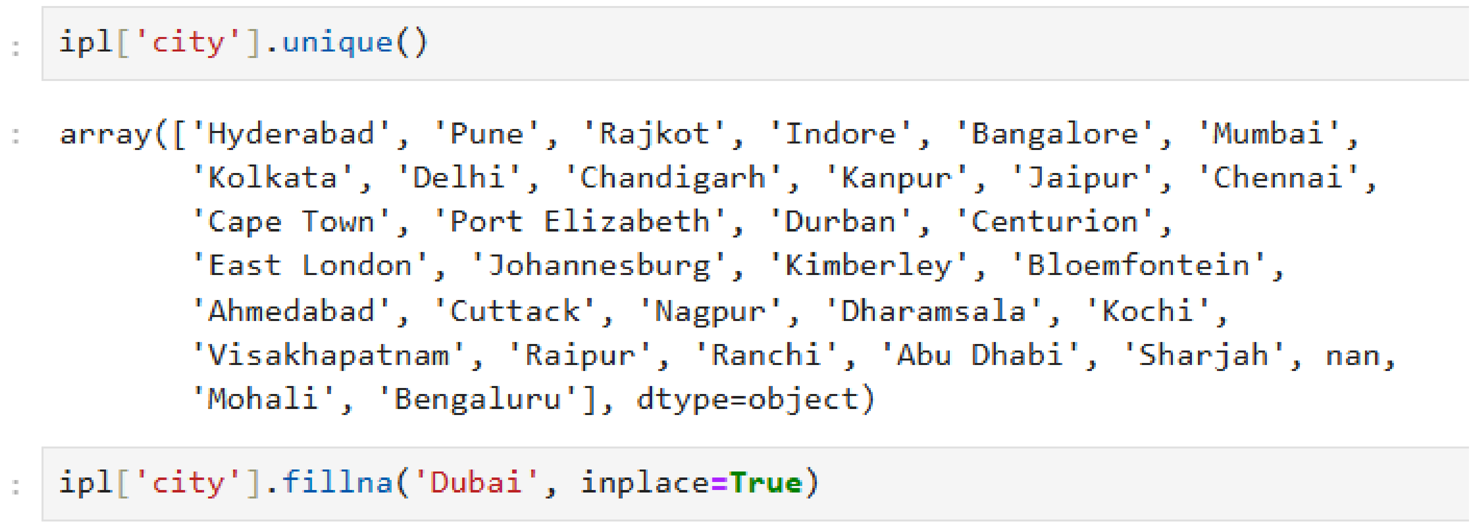
- REMOVE NULL VALUES
- REPLACE
- REMOVE COLUMNS
- SPLIT DATA (TRAIN & TEST)



```
ipl['city'] = ipl['city'].replace('Bengaluru', 'Bangalore')
```

```
ipl['city'].unique()
```

```
ipl=ipl.drop(['umpire1','umpire2','umpire3'],axis=1)
```



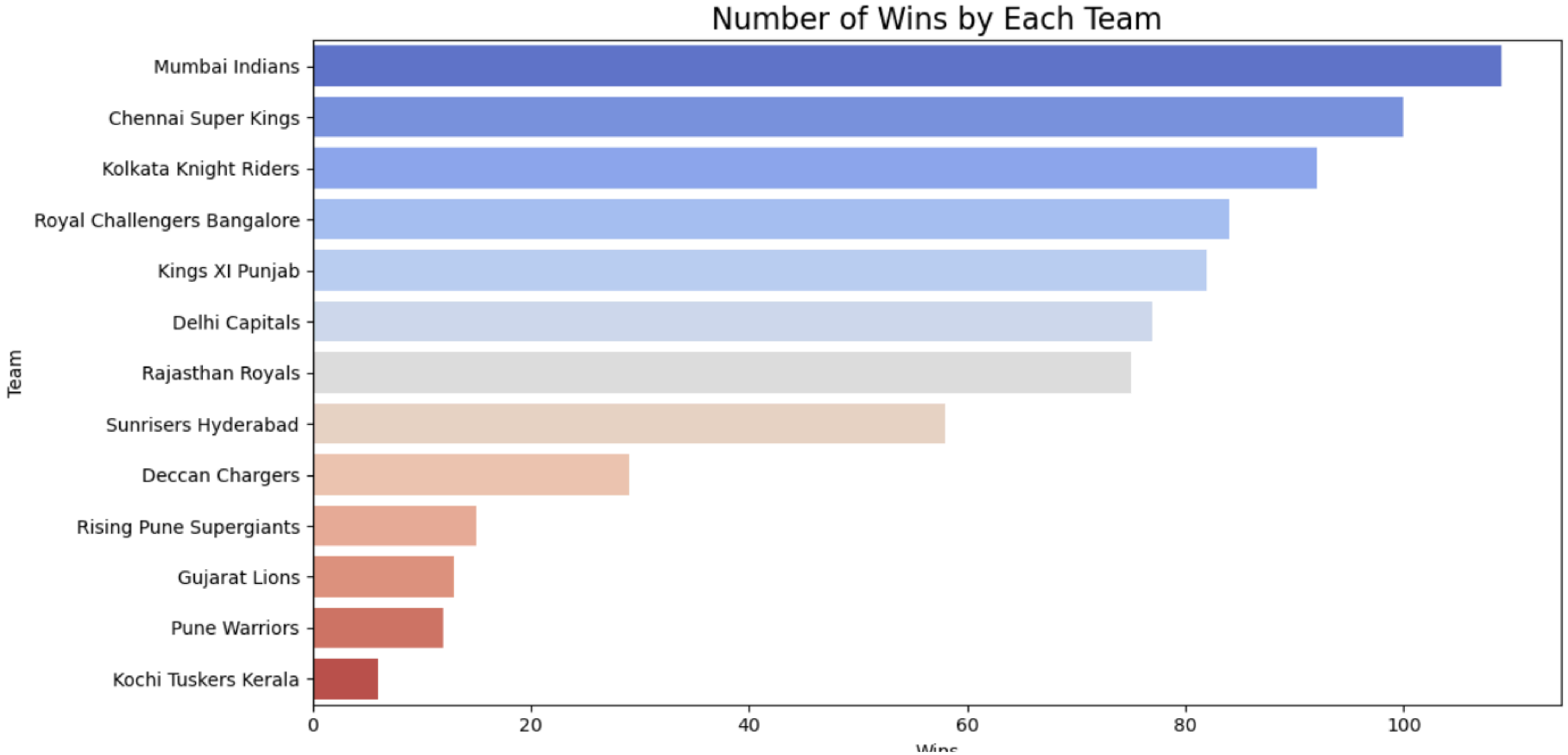
```
ipl['city'].unique()
```

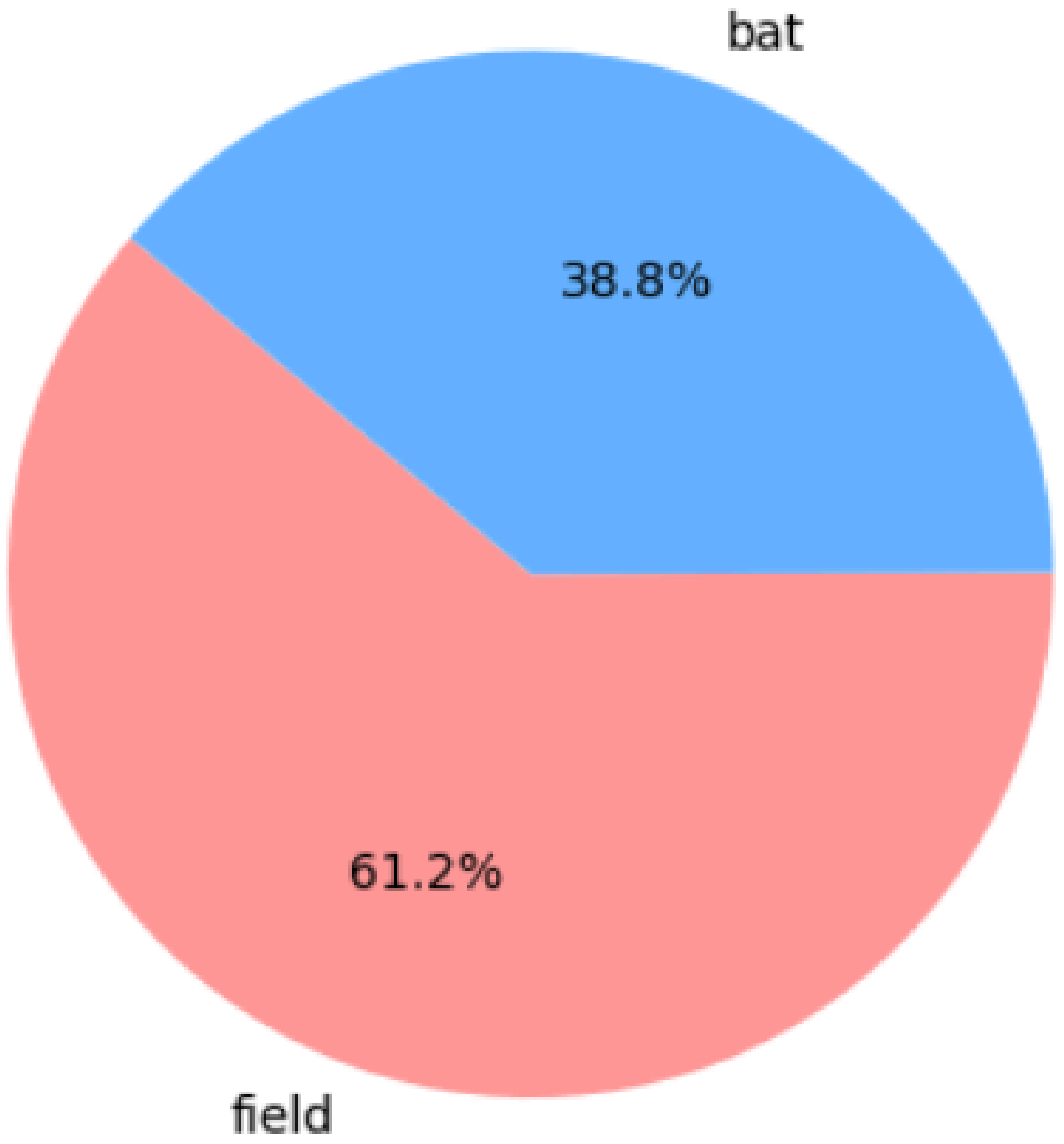
```
array(['Hyderabad', 'Pune', 'Rajkot', 'Indore', 'Bangalore', 'Mumbai',
       'Kolkata', 'Delhi', 'Chandigarh', 'Kanpur', 'Jaipur', 'Chennai',
       'Cape Town', 'Port Elizabeth', 'Durban', 'Centurion',
       'East London', 'Johannesburg', 'Kimberley', 'Bloemfontein',
       'Ahmedabad', 'Cuttack', 'Nagpur', 'Dharamsala', 'Kochi',
       'Visakhapatnam', 'Raipur', 'Ranchi', 'Abu Dhabi', 'Sharjah', nan,
       'Mohali', 'Bengaluru'], dtype=object)
```

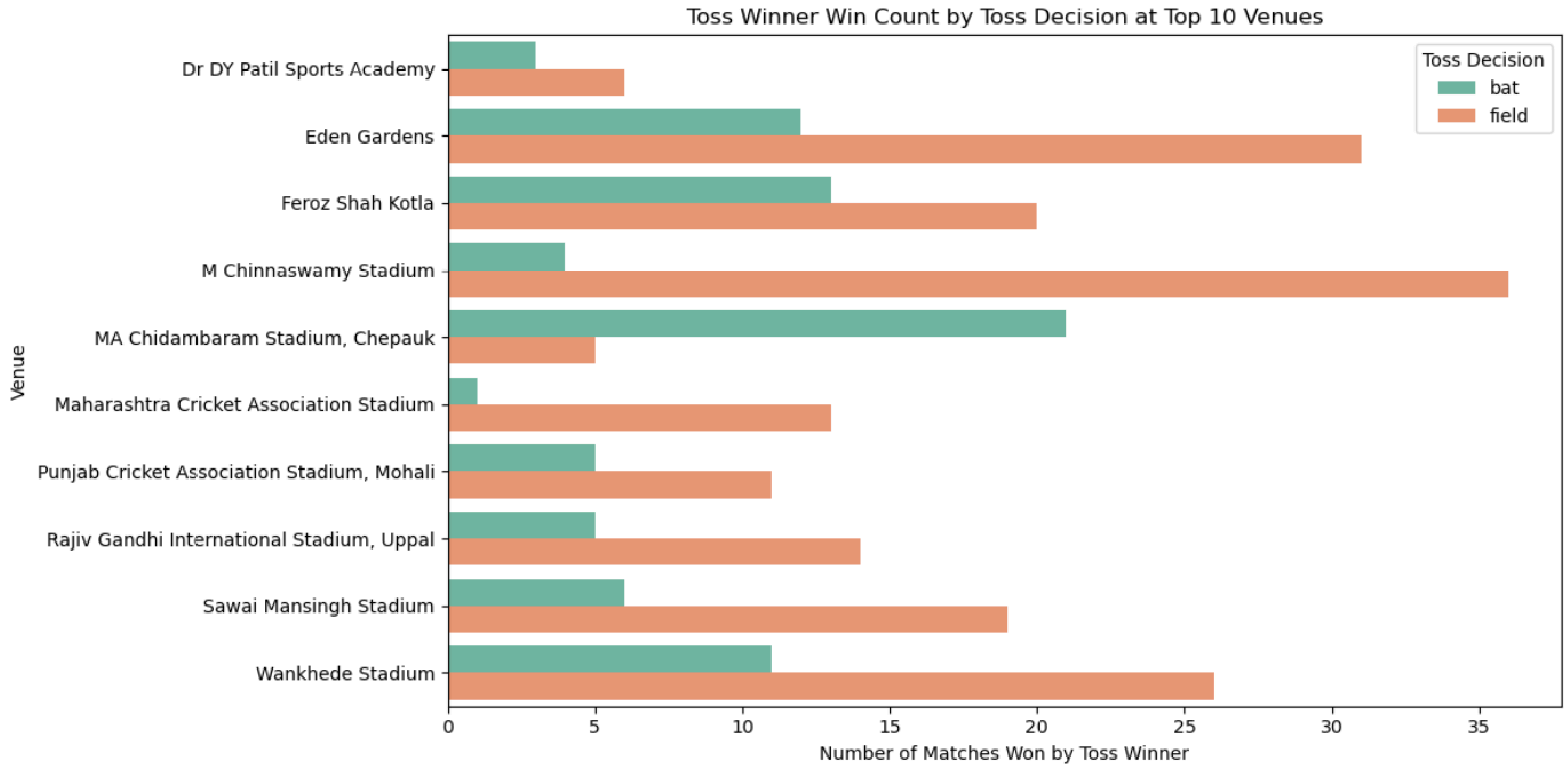
```
ipl['city'].fillna('Dubai', inplace=True)
```

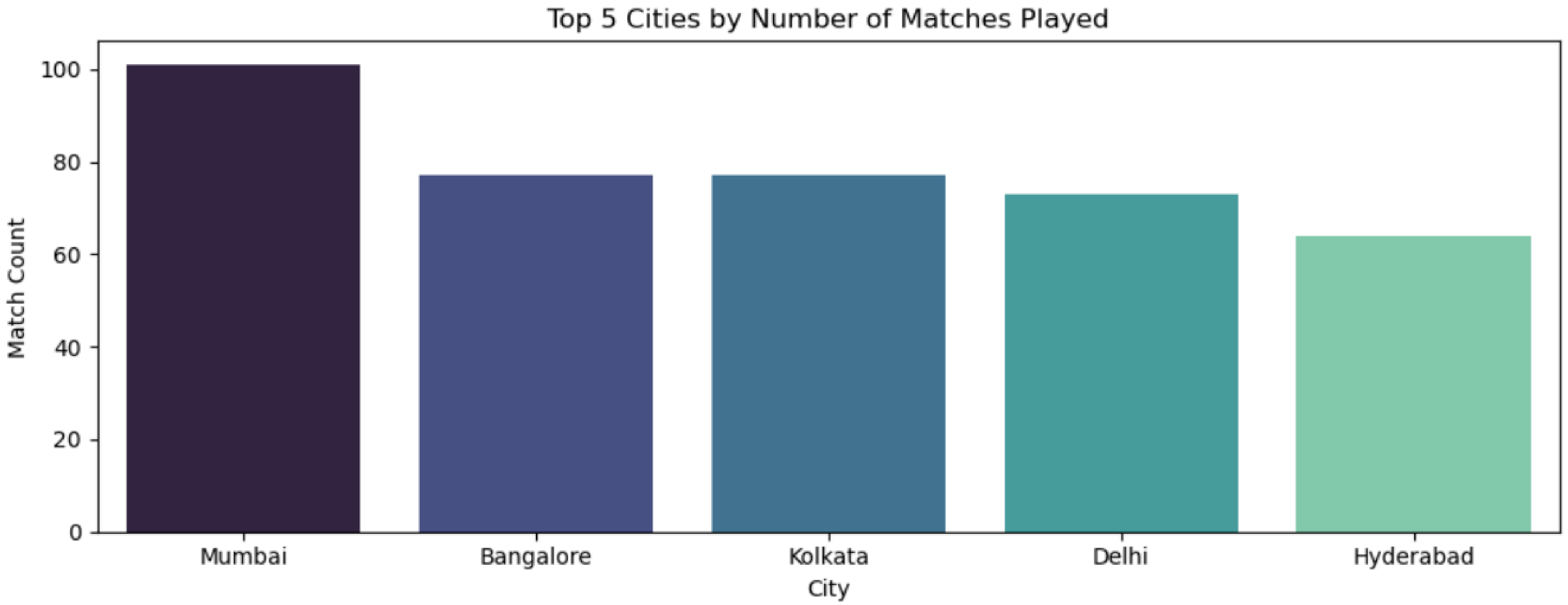
Exploratory Data Analysis

- Most winning teams
- Toss vs match winning
- Venue influence


Number of Wins by Each Team
Team
Mumbai Indians
Chennai Super Kings
Kolkata Knight Riders
Royal Challengers Bangalore
Kings XI Punjab
Delhi Capitals
Rajasthan Royals
Sunrisers Hyderabad
Deccan Chargers
Rising Pune Supergiants
Gujarat Lions
Pune Warriors
Kochi Tuskers Kerala
0
20
40
60
80
100
Wins

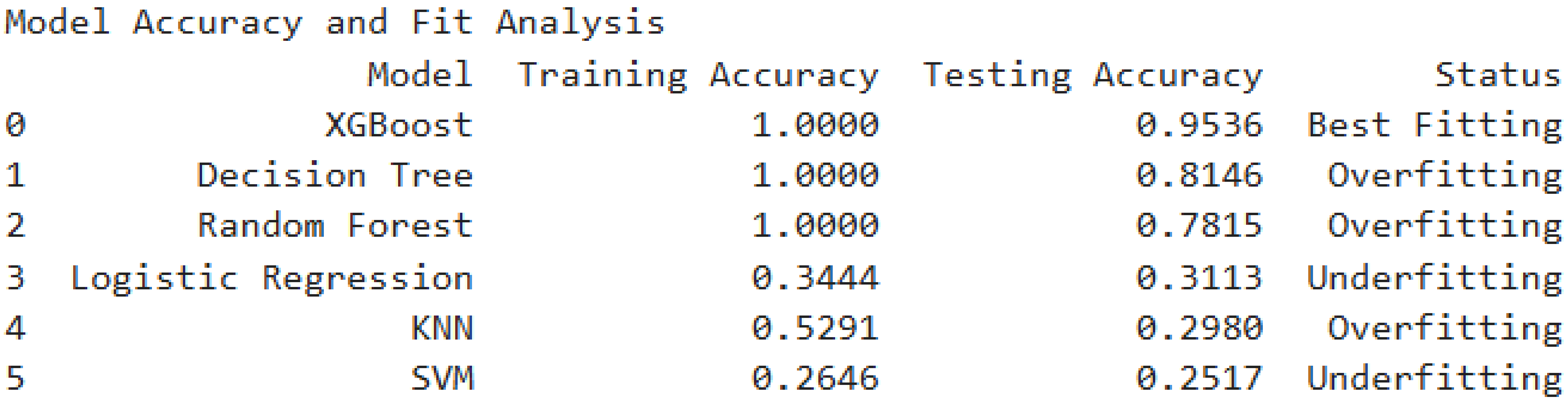

bat
38.8%
61.2%
field


Toss Winner Win Count by Toss Decision at Top 10 Venues
Toss Decision
bat
field
Venue
Dr DY Patil Sports Academy
Eden Gardens
Feroz Shah Kotla
M Chinnaswamy Stadium
MA Chidambaram Stadium, Chepauk
Maharashtra Cricket Association Stadium
Punjab Cricket Association Stadium, Mohali
Rajiv Gandhi International Stadium, Uppal
Sawai Mansingh Stadium
Wankhede Stadium
0
5
10
15
20
25
30
35
Number of Matches Won by Toss Winner


Top 5 Cities by Number of Matches Played
Match Count
100
80
60
40
20
0
Mumbai
Bangalore
Kolkata
Delhi
Hyderabad
City

Model Building

- Check Fit & Accuracy
- Overfitting/Underfitting Check
- Cross-validation
- Confusion Matrix

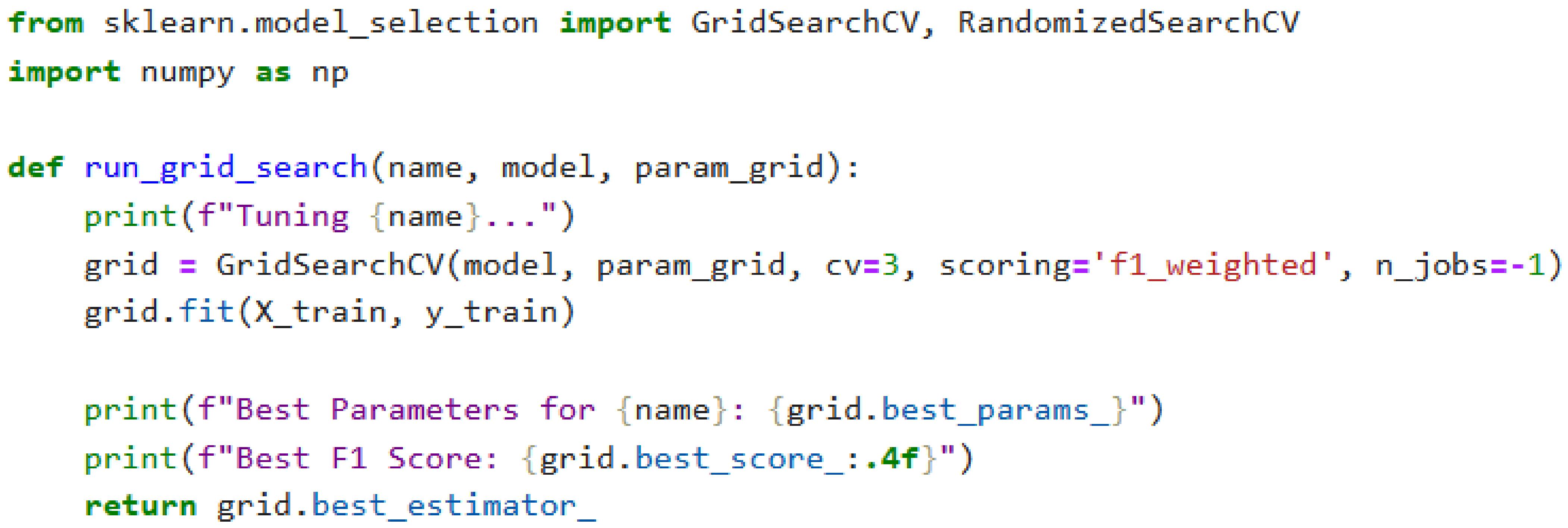


```
Model Accuracy and Fit Analysis
                 Model  Training Accuracy  Testing Accuracy        Status
0              XGBoost             1.0000            0.9536  Best Fitting
1        Decision Tree             1.0000            0.8146   Overfitting
2        Random Forest             1.0000            0.7815   Overfitting
3  Logistic Regression             0.3444            0.3113  Underfitting
4                  KNN             0.5291            0.2980   Overfitting
5                  SVM             0.2646            0.2517  Underfitting
```

- Logistic Regression
- SVM
- KNN
- Decision Tree
- Random Forest
- XGBoost

- Accuracy
- Precision
- Recall
- F1 Score
- Confusion Matrix

Hyperparameter Tuning



```
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
import numpy as np

def run_grid_search(name, model, param_grid):
    print(f"Tuning {name}...")
    grid = GridSearchCV(model, param_grid, cv=3, scoring='f1_weighted', n_jobs=-1)
    grid.fit(X_train, y_train)

    print(f"Best Parameters for {name}: {grid.best_params_}")
    print(f"Best F1 Score: {grid.best_score_:.4f}")
    return grid.best_estimator_
```

CONCLUSION

- Applied thorough EDA to understand winning patterns, toss impact, and team performance.
- Performed data cleaning, encoding, and feature scaling to prepare the dataset for modeling.
- Trained and tested multiple models including:
- Evaluated each model using:
- XGBoost gave the most balanced performance without overfitting.

IPL

Thank you!