**Football Game Prediction Project Documentation**

**Project Overview**

This project aims to develop a machine learning model to predict the outcomes of football games. Using historical data from the English Premier League (2010 to 2025), the model will analyze various features to predict match results. The project involves data preprocessing, feature engineering, model training, evaluation, and visualization of predictions.

**Objectives**

1. To collect and preprocess football match data.
2. To explore and analyze historical trends in football results.
3. To build a machine learning model that predicts match outcomes.
4. To evaluate the performance of the model using appropriate metrics.
5. To present insights and results in a clear and concise manner.

**Data Source**

* The dataset is obtained from [Football Data UK](https://www.football-data.co.uk/englandm.php).
* Data includes English Premier League matches from 2010 to 2025.
* It contains key features such as:
  + Date of the match
  + Home and away teams
  + Full-time result (FTR: H, D, A for Home win, Draw, Away win)
  + Goals scored by each team
  + Shots, corners, and fouls
  + Betting odds from various bookmakers

**Features for Prediction**

* Home Team Performance: Past win/loss record, average goals scored.
* Away Team Performance: Past win/loss record, average goals scored.
* Match Context: Home advantage, head-to-head results.
* Team Stats: Shots, corners, fouls, possession percentage.
* Betting Odds: Odds provided by bookmakers.

**Tools and Technologies**

1. Python: Programming language for data analysis and model development.
2. Libraries:
   * pandas for data manipulation.
   * numpy for numerical computations.
   * matplotlib and seaborn for data visualization.
   * scikit-learn for machine learning.
   * xgboost or lightgbm for advanced modeling.
3. Jupyter Notebook: Development environment.
4. Microsoft Excel: For inspecting and cleaning raw data.

**Methodology**

**Step 1: Data Collection**

* Download the English Premier League datasets from [Football Data UK](https://www.football-data.co.uk/englandm.php).
* Load data into a Python environment using pandas.

**Step 2: Data Preprocessing**

* Inspect and clean the data (handle missing values, duplicates).
* Convert categorical features (e.g., team names) into numerical representations.
* Create new features, such as recent form and head-to-head results.
* Split the data into training and testing sets.

**Step 3: Exploratory Data Analysis (EDA)**

* Visualize trends using matplotlib and seaborn.
* Analyze feature importance and correlations.
* Identify patterns in match results.

**Step 4: Model Building**

* Train machine learning models:
  + Logistic Regression
  + Random Forest
  + Gradient Boosting (e.g., XGBoost, LightGBM)
* Use scikit-learn pipelines for preprocessing and modeling.

**Step 5: Model Evaluation**

* Use metrics like accuracy, precision, recall, F1 score, and AUC-ROC.
* Perform cross-validation to ensure model robustness.
* Compare model performance and select the best one.

**Step 6: Predictions and Insights**

* Predict outcomes for future matches.
* Visualize predictions vs. actual results.
* Provide insights based on the model's performance.

**Example Code**

**Data Loading and Preprocessing**

import pandas as pd

# Load data

data = pd.read\_csv('EPL\_2010\_2025.csv')

# Inspect data

print(data.head())

# Handle missing values

data = data.dropna()

# Feature engineering

data['Goal\_Difference'] = data['HomeGoals'] - data['AwayGoals']

# Encode target variable

data['Result'] = data['FTR'].map({'H': 1, 'D': 0, 'A': -1})

# Split data into training and testing sets

from sklearn.model\_selection import train\_test\_split

X = data[['HomeGoals', 'AwayGoals', 'Shots', 'Corners']]

y = data['Result']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**Model Training**

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

# Train model

model = RandomForestClassifier(random\_state=42)

model.fit(X\_train, y\_train)

# Evaluate model

predictions = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, predictions)

print(f'Accuracy: {accuracy \* 100:.2f}%')

**Deliverables**

1. Cleaned dataset of English Premier League matches (2010-2025).
2. Jupyter Notebook with EDA, model training, and evaluation.
3. Visualizations of trends and model performance.
4. Final report summarizing results and insights.

**References**

* [Football Data UK](https://www.football-data.co.uk/englandm.php)
* [scikit-learn Documentation](https://scikit-learn.org/stable/)
* [Matplotlib Documentation](https://matplotlib.org/stable/)
* [Pandas Documentation](https://pandas.pydata.org/)

**APPENDICES**

* **'Div'**: Division (the league in which the match was played, e.g., Premier League)
* **'Date'**: The date the match was played
* **'Time'**: The time the match kicked off
* **'HomeTeam'**: The name of the home team
* **'AwayTeam'**: The name of the away team
* **'FTHG'**: Full-time home goals (number of goals scored by the home team at full-time)
* **'FTAG'**: Full-time away goals (number of goals scored by the away team at full-time)
* **'FTR'**: Full-time result (could be a code indicating Win/Draw/Loss, e.g., "H" for Home win, "A" for Away win, "D" for Draw)
* **'HTHG'**: Half-time home goals (number of goals scored by the home team at half-time)
* **'HTAG'**: Half-time away goals (number of goals scored by the away team at half-time)
* **'HTR'**: Half-time result (similar to 'FTR', indicating Win/Draw/Loss at half-time)
* **'Referee'**: Name of the referee for the match
* **'HS'**: Home team shots (total number of shots taken by the home team)
* **'AS'**: Away team shots (total number of shots taken by the away team)
* **'HST'**: Home team shots on target (number of shots on target by the home team)
* **'AST'**: Away team shots on target (number of shots on target by the away team)
* **'HF'**: Home team fouls (number of fouls committed by the home team)
* **'AF'**: Away team fouls (number of fouls committed by the away team)
* **'HC'**: Home team corners (number of corners taken by the home team)
* **'AC'**: Away team corners (number of corners taken by the away team)
* **'HY'**: Home team yellow cards (number of yellow cards issued to the home team)
* **'AY'**: Away team yellow cards (number of yellow cards issued to the away team)
* **'HR'**: Home team red cards (number of red cards issued to the home team)
* **'AR'**: Away team red cards (number of red cards issued to the away team)
* **'B365H'**, **'B365D'**, **'B365A'**: Bet365 odds for Home win, Draw, and Away win, respectively
* **'BWH'**, **'BWD'**, **'BWA'**: Odds from a different betting agency (e.g., William Hill) for Home win, Draw, and Away win
* **'IWH'**, **'IWD'**, **'IWA'**: Odds from another betting agency (e.g., Interwetten) for Home win, Draw, and Away win
* **'PSH'**, **'PSD'**, **'PSA'**: Odds from a different betting service (e.g., Pinnacle) for Home win, Draw, and Away win
* **'WHH'**, **'WHD'**, **'WHA'**: Odds from William Hill, potentially for different types of betting (e.g., over/under goals)
* **'MaxH'**, **'MaxD'**, **'MaxA'**: Maximum odds for Home win, Draw, and Away win
* **'AvgH'**, **'AvgD'**, **'AvgA'**: Average odds for Home win, Draw, and Away win
* **'B365>2.5'**, **'B365<2.5'**: Bet365 odds for over/under 2.5 goals in the match
* **'P>2.5'**, **'P<2.5'**: Other odds for over/under 2.5 goals (possibly from Pinnacle)
* **'Max>2.5'**, **'Max<2.5'**: Maximum odds for over/under 2.5 goals
* **'Avg>2.5'**, **'Avg<2.5'**: Average odds for over/under 2.5 goals
* **'AHh'**: Asian Handicap Home team (odds for Asian handicap on home team)
* **'B365AHH'**, **'B365AHA'**: Bet365 Asian Handicap odds for Home and Away teams
* **'PAHH'**, **'PAHA'**: Pinnacle Asian Handicap odds for Home and Away teams
* **'MaxAHH'**, **'MaxAHA'**: Maximum Asian Handicap odds for Home and Away teams
* **'AvgAHH'**, **'AvgAHA'**: Average Asian Handicap odds for Home and Away teams
* **'B365CH'**, **'B365CD'**, **'B365CA'**: Bet365 odds for different types of bets related to corner statistics
* **'BWCH'**, **'BWCD'**, **'BWCA'**: Odds for corner-related bets from William Hill
* **'IWCH'**, **'IWCD'**, **'IWCA'**: Odds for corner-related bets from Interwetten
* **'PSCH'**, **'PSCD'**, **'PSCA'**: Odds for corner-related bets from Pinnacle
* **'WHCH'**, **'WHCD'**, **'WHCA'**: Odds for corner-related bets from William Hill
* **'VCCH'**, **'VCCD'**, **'VCCA'**: Odds for corner-related bets from a different betting service
* **'MaxCH'**, **'MaxCD'**, **'MaxCA'**: Maximum odds for corner-related bets
* **'AvgCH'**, **'AvgCD'**, **'AvgCA'**: Average odds for corner-related bets
* **'B365C>2.5'**, **'B365C<2.5'**: Bet365 odds for over/under 2.5 goals in corners
* **'PC>2.5'**, **'PC<2.5'**: Other odds for over/under 2.5 corners
* **'MaxC>2.5'**, **'MaxC<2.5'**: Maximum odds for over/under 2.5 corners
* **'AvgC>2.5'**, **'AvgC<2.5'**: Average odds for over/under 2.5 corners
* **'AHCh'**: Asian Handicap odds for corners