Introduction

Every year there are over 2,000+ matches played in Europe. Many factors determine the outcome of these matches. Some of these factors include players, teams, infrastructure, strategy etc. Making a prediction on a match outcome is quite difficult due to many factors to be put into consideration.

This main outcome of this project is to develop an application to predict the outcome of a football game between any two teams in Europe. The shared tasks for developing the application are data preprocessing, algorithm implementation for match outcome prediction, developing appropriate visualizations and lastly bringing it all together onto a GUI for ease of use.

Personal Contributions

1. Team formation visualization

The team_formation(home_team_name, away_team_name) method is used for plotting the home team vs away team formation in the application on the PyQt5 Canvas. This method is invoked when the ok button is clicked. It takes in 2 inputs: home_team_name and away_team_name which the user selects on the GUI and plots their respective formations.

The player name and position data is queried from the "Team" table for the most recent match played. It is preprocessed and then plotted onto 2 subplots. We use the scatter() function for displaying the player position markers and the annotate() function for displaying the player names. We also add the field markings using the Arc(), Rectangle() methods from Matplotlib onto the subplots to make the visualization better.

Steps:
1. Query Team and Match tables for latest data of home and away players (player id, player X coordinate, player Y coordinate)
2. Split the player names and coordinates in different lists
3. Get the player names for home and away player ids
4. Set Goalkeeper X coordinate to 5 (center of field)
5. Plot player positions using X and Y coordinates, player names and necessary field markings for both subplot
6. Plot the Matplotlib graph which is further transferred to PyQt5 Canvas on the application GUI

2. Decision Tree Classifier for Match Outcome prediction

The analyze_match() method is used for predicting the match outcome for home and away team selected on the GUI by the user. This method is invoked when the ok button is clicked.

We use the decision tree classifier with 'gini' criterion to train the model.
We have used 18 features (9 home team + 9 away team) from the "Team Attributes" table to train the model with the match outcome (0-Draw, 1-Home Team Win, 2- Away Team Win) as the target variable. These features can be generalized into 3 groups:

i. buildUpPlay – Speed, Dribbling, Passing
ii. chanceCreation – Crossing, Shooting, Passing
iii. defence – Pressure, Aggression, TeamWidth

For predicting the outcome of a game selected by the User in the GUI, we feed the features for the home and away teams based on their average performance in past matches to the predict() function.

Results

The model is able to predict the outcomes with ~44% accuracy. (calculated by using train, test splits and not a part of the application)

Summary and conclusions

Predicting soccer games is difficult. Although form is crucial for winning, we cannot say on how the team will play on a particular day based on their past performances. Furthermore, there are external factors involved like pitch conditions, weather, player injuries, red cards etc.

Future improvements that can be made to improve the model accuracy are addition of other features that decide the match outcome, selection of an alternative model, hyperparameter tuning.

Code Percentage

(100 – 30)/ (100 + 60) = 43.75%

References

1. https://www.kaggle.com/hugomathien/soccer
2. https://www.geeksforgeeks.org/pyqt5-set-fix-window-size-for-height-or-width/
3. https://pythonprogramminglanguage.com/pyqt5-center-window/#:~:text=To%20center%20a%20Python%20PyQt,the%20center%20of%20the%20screen.
4. https://www.tutorialspoint.com/pyqt5/pyqt5_tutorial.pdf
5. https://datacarpentry.org/python-ecology-lesson/09-working-with-sql/index.html