GROUP 5 PRESENTATION

Soccer Match Prediction With Decision Tree

INTRODUCTION

The project is developed to analyze previous matches played and predict future matches between two teams.Since the teams are the target, we will build our predictors based on the teams statistics and features.We will design a GUI that will enable a user selected both the home and away team and generate the predictors based on the selected teams to predict the match outcome

DATASET

The Data set is an sqlite file with 7 tables

1. Country: This table has 11 records and 2 columns
2. League : This table has 11 records and 3 columns
3. Match : This table has 25,979 records and 114 columns.
4. Player: This table has 11,060 records and 7 columns.
5. Player_Attributes : This table has 183,978 records and 42 columns.
6. Team : This table has 299 records and 5 columns.
7. Team_Attributes : This table has 1458 records and 25 columns.

DATA PREPROCESSING

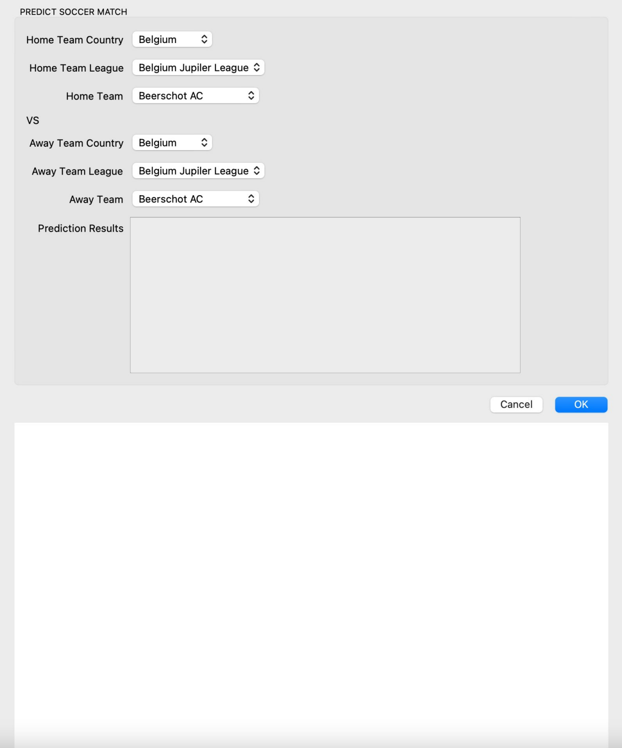
1. The data was extracted using sql queries and converting them into dataframes
2. N/A values were filled with zero in some instances and removed in other situations
3. Using SQL select, we only selected columns we need rather than fetching over 120 columns
4. From the database files, we can get the list of countries, leagues and teams
5. Based on the selected home and away team we can get the predator variables and target variables from the dataset.

GRAPHIC USER INTERFACE

Using Pyqt5, we designed a user interface to get a users input.

1. We created a combo control to select country, league and team for both the home and away team.
2. We created two buttons control for both cancel and submit
3. We created a plaintext edit and disabled it to display the results
4. We created a canvas to display the teams formations
5. We added all the controls created above to a formgroup box and added them as a widget to the layout.
6. We added labels to these controls using the addRow() function calling the Qlabel to make them more descriptive.

GRAPHIC USER INTERFACE RESULT



PREDICT SOCCER MATCH

Home Team Country: Belgium

Home Team League: Belgium Jupiler League

Home Team: Beerschot AC

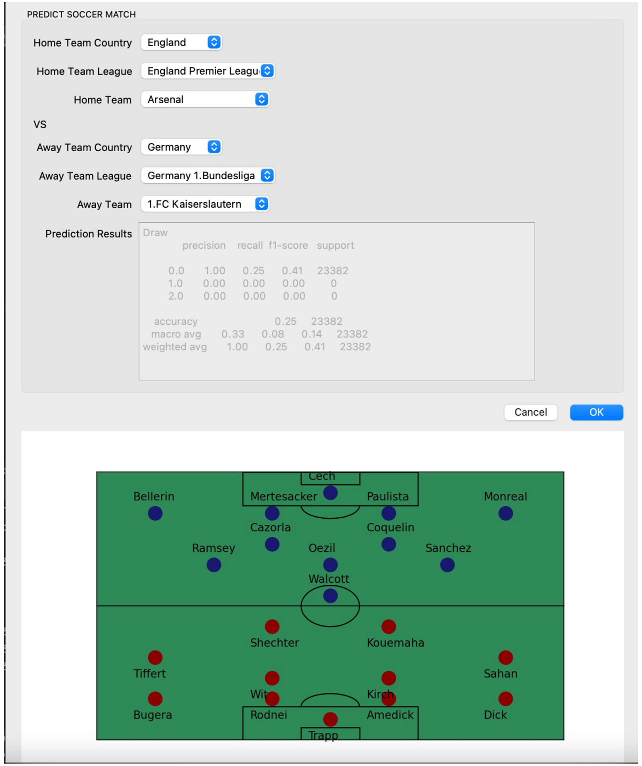
VS

Away Team Country: Belgium

Away Team League: Belgium Jupiler League

Away Team: Beerschot AC

Prediction Results



PREDICT SOCCER MATCH

Home Team Country: England

Home Team League: England Premier Leagu

Home Team: Arsenal

VS

Away Team Country: Germany

Away Team League: Germany 1.Bundesliga

Away Team: 1.FC Kaiserslautern

Prediction Results

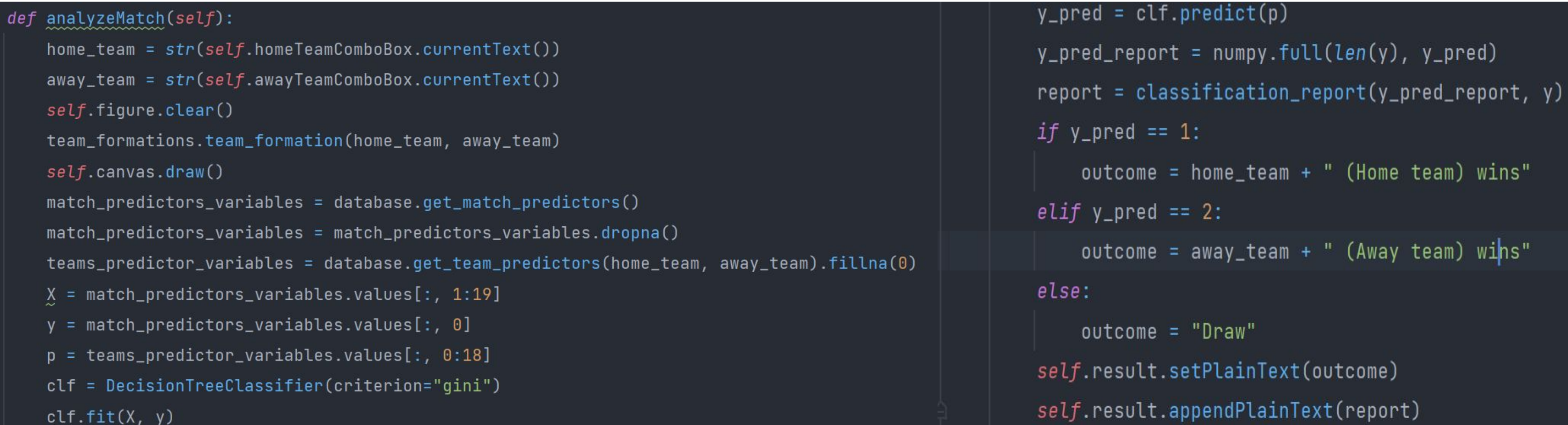
```
Draw
            precision   recall  f1-score   support

       0.0       1.00      0.25      0.41     23382
       1.0       0.00      0.00      0.00         0
       2.0       0.00      0.00      0.00         0

  accuracy                           0.25     23382
 macro avg       0.33      0.08      0.14     23382
weighted avg     1.00      0.25      0.41     23382
```

ALGORITHM: DECISION TREE

- Retrieve match predictors from database based on team performance, like No. of passes and shooting
- The predictors were trained and fitted with the target using the Sklearn Decision tree classifier.
- Gini index was used as metrics.
- We also used matplotlib to plot the teams formations



```
def analyzeMatch(self):
    home_team = str(self.homeTeamComboBox.currentText())
    away_team = str(self.awayTeamComboBox.currentText())
    self.figure.clear()
    team_formations.team_formation(home_team, away_team)
    self.canvas.draw()
    match_predictors_variables = database.get_match_predictors()
    match_predictors_variables = match_predictors_variables.dropna()
    teams_predictor_variables = database.get_team_predictors(home_team, away_team).fillna(0)
    X = match_predictors_variables.values[:, 1:19]
    y = match_predictors_variables.values[:, 0]
    p = teams_predictor_variables.values[:, 0:18]
    clf = DecisionTreeClassifier(criterion="gini")
    clf.fit(X, y)
```

```
    y_pred = clf.predict(p)
    y_pred_report = numpy.full(len(y), y_pred)
    report = classification_report(y_pred_report, y)
    if y_pred == 1:
        outcome = home_team + " (Home team) wins"
    elif y_pred == 2:
        outcome = away_team + " (Away team) wins"
    else:
        outcome = "Draw"
    self.result.setPlainText(outcome)
    self.result.appendPlainText(report)
```

LIMITATION

1. Data is gathered on teams playing with its peers of similar competitiveness
2. Captures teams' "average" performance for 8 years, could not reflect recent changes
    - Could assign weights to more recent matches
3. Suffer from all limitations of linear classifiers: feature class imbalance, overfitting, etc.
4. Unaccounted features: weather, pitch condition, matching strategy and more

SUMMARY

- Implemented learning from class material
- Learned to work with new packages together
- Decision Tree model could yield prediction with certain level of confidence
- Limitation from full dataset, algorithm and unaccounted factors is still holding model performance
- Betting money on this model is not advised!