Jason Chapman
December 6th, 2022
CS – 249

Final Project Report

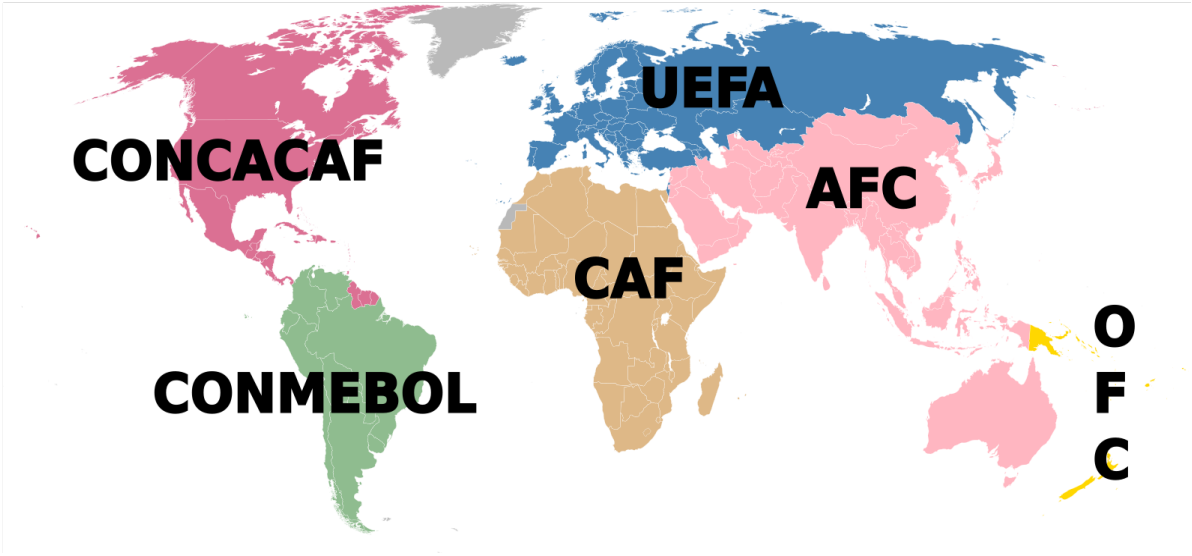
This final project was centered around a prediction model for the 2022 FIFA World Cup. The World Cup is a 32-team international soccer tournament that takes place every four years and is the ultimate achievement for international play. Throughout the 29-day tournament, 64 matches are played. The goal of this project is to predict the outcome of each of the 64 matches and determine the ultimate champion of the 2022 FIFA World Cup. To predict the outcomes of each match, a logistic regression model was trained on previous match data, team rankings, and other informative features. After predicting the results of each match, the model predicted that Brazil would be crowned the champions of the 2022 FIFA World Cup over France in the final match.

The goal of this project is to create a match outcome predictor for the 2022 FIFA World Cup in Qatar. This predictor will use international fixture results and FIFA international rankings since 1993 to create a model that will predict the outcomes of the upcoming matches. The motivation behind studying this problem is that the 2022 FIFA World Cup in Qatar began November 20th and will go until December 18th. This is an event that comes once every four years, and many individuals across the world become invested in the outcomes of every single match. An additional motivation is that I am constantly checking websites like FiveThirtyEight.com for their predictions across all sports including the World Cup. And after taking this course, I was confident I had the ability to create my own predictive modeler.

In the history of the World Cup, which started in 1930, only eight countries have ever won the tournament. Those countries are Brazil, Germany, Italy, Argentina, Uruguay, England, Spain, and current champions France. According to FiveThirtyEight's 2022 World Cup Predictions, the teams with the highest likelihood of winning before the tournament began were Brazil at 22%, Spain at 11%, France at 9%, Argentina at 8%, and Portugal at 8% [1].

There were three primary datasets used for this project. These included international soccer matches since 1872 [2], FIFA International Team Rankings since August 1993 [3], as well as Ballon d'Or voting data since 1956 [4]. FIFA International Team Rankings are significant because it sheds light on the quality of the international teams. The Ballon d'Or is an annual award presented to the best Men's soccer player for that year considering performance in both international and club soccer play. The Ballon d'Or dataset includes about 20 nominees for the Ballon d'Or award per year and is significant because it puts a numerical value for the number of "elite" players on an international team for a given match. These datasets were cleaned, and the appropriate features were combined to create a training dataset for the model.

The predictive model uses many informative features in order to be an accurate predictor. Those features include the FIFA rank of the home team, the FIFA rank of the away team, the home team's FIFA rank change from the previous year, the away team's FIFA rank change from the previous year, whether the match was played at the home team's stadium or on neutral ground, the home team's Confederation, the away team's Confederation, and the number of Ballon d'Or nominees on each of the home or away team. The Confederation that a team belongs to is one of either UEFA, CONCACAF, CONMEBOL, AFC, OFC, or CAF, and is reflective of where the country is located.

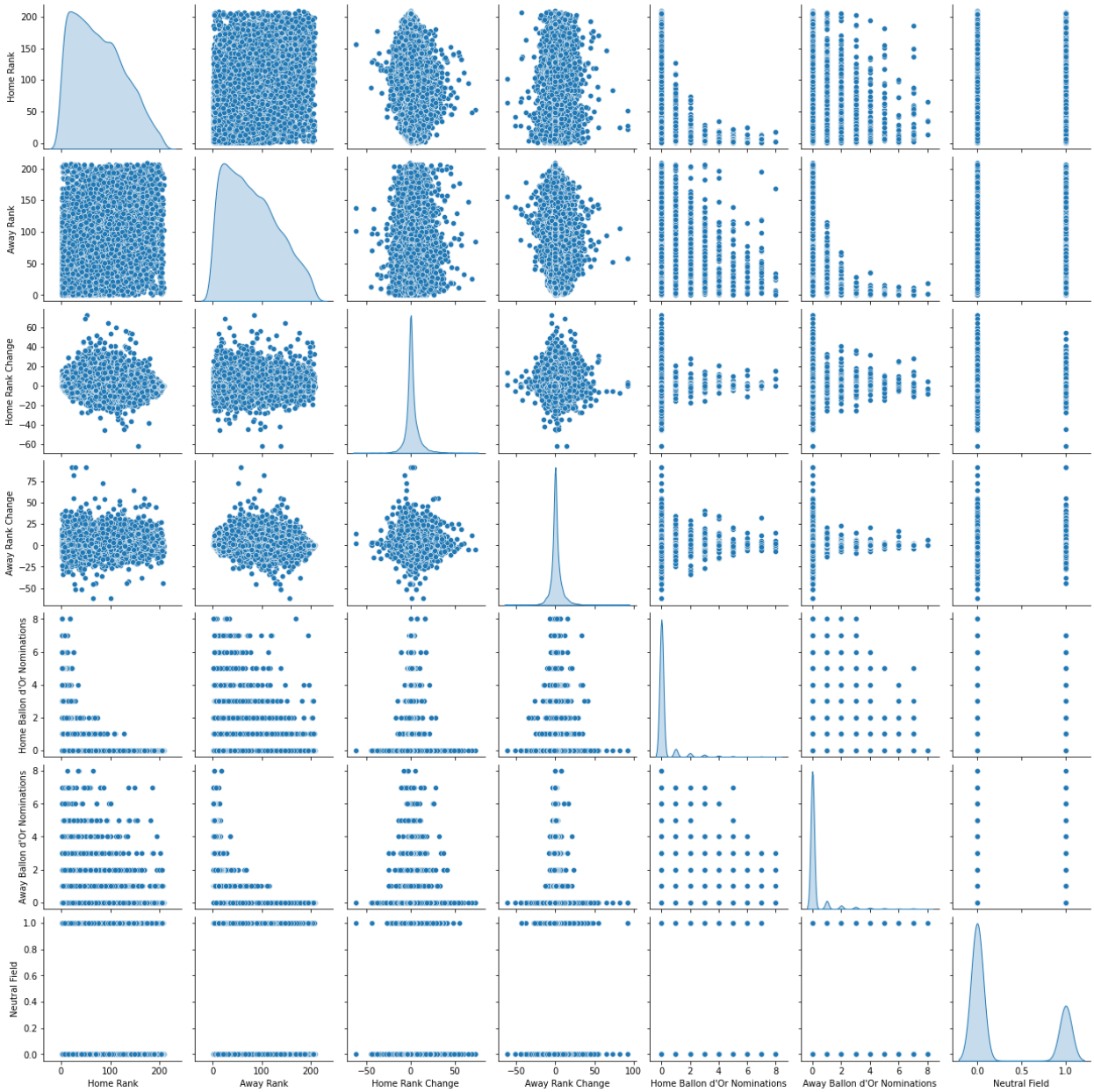


To clarify, the "home team" and "away team" are simply classifiers of the two teams in the match. Even in the World Cup (when played on a neutral field) there is a "home" and "away" team. So, the informative features discussed such as "home team rank", "away team confederation", "home team rank change", etc. are independent of where the match is being played. The team's current FIFA ranking is reflective of the quality of the team as a whole on the date of the match. The team's FIFA rank change feature is reflective of the improvements of the team from the previous ranking and shows how the team's form is improving or regressing. The Confederation that the team belongs to is informative because the teams from UEFA and CONMEBOL tend to dominate international competition. And finally, the neutral field feature is meant to be an indicator of "home field advantage". This will likely play only a small role in this predictor because only one team, Qatar, will have home field advantage, but should be included, nevertheless.

Creating a training dataset that includes these informative features required cleaning of the individual datasets that were found online. The primary step of cleaning the data was to trim the datasets so that they spanned the same timeframe (from August 1993 until December 2018). Then, the FIFA ranking features and Ballon d'Or nominee feature were appended to the results dataset for the correct match date and team. There were some results with unidentified team confederations or were unranked by FIFA at the date of the match. So, it was necessary to determine a method for handling missing values. It was determined that the best method

would be to simply drop the missing values because the samples with unidentified team confederations or undefined FIFA rankings were usually very small countries/territories that could not even qualify for a World Cup according to FIFA's rules and should therefore not be included in the dataset. Then, features not included in the predictive model were dropped from the dataset and the resultant dataset was the complete training dataset. Finally, an outcome vector was created with three classes (home win, away win, and tie).

Once the data had been cleaned, it was then appropriate to explore the data and try to draw some insights from the data before using it to predict the outcome of World Cup matches. In order to analyze the data visually, a simple seaborn "pairplot" was generated from the numerical informative features (excluding the Confederation features).



A few interesting insights can be drawn from the above image. Firstly, it is shown that home rank change and away rank change are uniformly distributed around 0 meaning that there aren't dramatic shifts in the FIFA rank changes among this data. Both the home and away Ballon d'Or nominations are highly left skewed which indicates that having even a single Ballon d'Or nominated player on a team is rare. Home and away FIFA rank are also left-skewed indicating that this dataset is comprised of matches played by relatively high ranked teams, which is good because the distribution of team ranks for the FIFA World Cup matches to be predicted should be similar. The neutral field feature is multimodal, which is expected because it is an indicator variable, and it shows that most of the matches in this dataset are not played on a neutral field. However, there is still a large portion of the dataset (29%) is matches played on a neutral field. As expected, there doesn't appear to be any significant correlation between the variables.

After gathering, cleaning, and exploring the data, the predictive model for the outcome of a World Cup match needed to be chosen. There are three different models that may be appropriate for this problem. The first would be to represent the win or loss of a game as a binary variable for logistic regression where a 1 indicates the home team winning and a 0 indicates the home team losing. However, this model does not account for tie games. Given the fact that many of the samples in the training set have results of a tie and that several of the games in the upcoming FIFA World Cup will result in ties, dropping samples that resulted in a tie and using this model would not be the most appropriate. Another model that was considered was to use a Poisson regression to count the number of goals per game. It is likely that the number of goals in a soccer match follows a Poisson distribution, so this is what FiveThirtyEight uses for their World Cup predictor. However, this model is slightly more complex and indirect to build a model that predicts scores when the goal is to predict a win, loss, or tie. So, it was determined that the best model to use for this prediction problem is to use a multinomial logistic regression with three different categories for the result: home win, home loss, or tie.

After determining that multinomial logistic regression was the appropriate model, it was necessary to determine the accuracy of the predictive model using cross validation. First, the categorical data in the dataset (Confederations) was converted to numerical data by converting them to multiple indicator variables. It was necessary to scale the data in order get the regression model to converge. For each feature, the data is centered and scaled to unit variance. To perform cross validation, 10-fold cross validation was used. When cross validation was first performed, it was seen that the accuracy of the predictor was not very high. So, a few different methods to improve the performance of the predictor were considered. The first method that was considered was the use an appropriate regularization parameter. However, it was determined that the best regularization parameter is no regularization (C=1). The second method that was considered was to use a different classifier. A decision tree classifier is appropriate. However, this resulted in less accuracy than the multinomial logistic regression. Lastly, feature selection was considered. After using forward feature selection, it was suggested to get rid of a fraction of the team Confederation indicator features. However, only including some would be erroneous. So, the accuracy of the model without any of the Confederation features was found, and it was determined to be less than when it is included. The results from

cross validation are shown below. It is important to note that the scoring method for multinomial logistic regression is the mean accuracy of the given test data and label.

Model	Accuracy
Multinomial Logistic Regression	*0.6277*
Multinomial Logistic Regression w/ Feature Selection (No Confederation Features)	*0.6246*
Decision Tree Classifier	*0.5128*

The best way to increase the accuracy of this model would be to add more informative features. Unfortunately, there was not enough time and access to appropriate datasets to add these features for this project. Some improvement methods that will be implemented are discussed in the conclusion. Thus, it was determined to move forward using this prediction model. After using a 10% train-validation split, the accuracies of a train set, and validation set were compared to determine if there was any overfitting. The train set accuracy was 0.6299, and the validation set accuracy was 0.6167. There is no considerable difference in accuracy between the training and validation sets, which suggests that there is no overfitting in this model.

The accuracy of this model improves considerably when simplified to simple logistic regression with two classes (1: Home Win, 0: Away Win). A simple logistic regression is used for the final 16 games of the tournament because there can be no ties after the Group Stage round. So, for the consequent rounds a new model was created and cross validation was done again. After using a 10% train-validation split, the accuracies of a train set, and validation set were compared to determine if there was any overfitting. The train set accuracy was 0.8247, and the validation set accuracy was 0.8071. There is no considerable difference in accuracy between the training and validation sets, which suggests that there is no overfitting in this model.

Because the test set for this project is the actual matches of the 2022 FIFA World Cup, the training set for the model is the entire dataset. So, after fitting the entire dataset to the model, it was informative to look at the coefficients and interpret them.

	Home Rank	Away Rank	Home Rank Change	Away Rank Change	Home Ballon d'Or Nominations	Away Ballon d'Or Nominations	Neutral Field
Away Win	-0.818334	0.96125	-0.215187	0.244631	-0.194636	0.137314	0.21929
Home Win	0.849988	-0.833907	0.270299	-0.237502	0.17674	-0.12841	-0.185304
Tie	-0.0316538	-0.127343	-0.0551111	-0.007129	0.0178962	-0.0089041	-0.0339858

These coefficients match intuition. For instance, a one unit increase in Home Rank, given the other variables remain constant, corresponds to an increase in the odds of Home Win by 0.849988 units. All the "home" features correspond to positive coefficients for home win and vice versa for "away" features. This suggests that having a higher rank, a greater positive rank change, and more Ballon d'Or Nominations correspond to greater likelihood of winning. This is as expected. Additionally, playing on a neutral field benefits the away team when compared to playing on a non-neutral field. Again, as expected. It was also informative to create a confusion matrix. The confusion matrix for the training set is shown below. The relatively low prediction accuracy explains the large numbers on the non-diagonals.

		Predicted			
		Away Win	Home Win	Tie	Total
Actual	Away Win	2300	743	261	*3304*
	Home Win	717	4547	330	*5594*
	Tie	1008	1310	286	*2604*
	Total	*4025*	*6600*	*877*	*11502*

Finally, it was time to start predicting matches of the 2022 FIFA World Cup. The first step in predicting the outcomes of the 2022 FIFA World Cup matches was to generate the appropriate test set. Generating a test set for the first 48 games was simple because the team matchups are known, as well as all the informative features included in the training set. So, the first test set will include 48 samples, one for each game of the Group Stage. Then, the results from predicting the outcomes of the matches using the multinomial logistic regression model were evaluated. The teams that advance out of the Group Stage and into the Knockout Stage of the FIFA World Cup are the top 2 teams in each of 8 groups of 4. The ranking of the teams in each group is determined by a point system from the 3 games that each team plays in the group stage (3 points for a win, 1 point for a tie, 0 points for a loss). After evaluating the results of the prediction model for these games, it was determined which two teams in each group were moving onto the next round [A]. Then, a new test set was created from the remaining 16 teams that made up the knockout stages and the predicted results from using the simple logistic regression model were evaluated. And a similar process was repeated three more times in order to predict the 2022 World Cup Champion [B]. The results from each phase resulted in the following completed bracket, showing that Brazil will be the 2022 FIFA World Cup Champion after beating France in the finals.

These results are as expected, with two interesting results. It was interesting that the model predicted that Costa Rica and Qatar will make it out of the group stages. It is of note that Qatar was the only team to have a “home field advantage” and Costa Rice had the most dramatic FIFA rank change with +15 from the last ranking. So, these two features may have exaggerated impact in the training dataset.

Overall, the prediction model predicted that Brazil will beat France in the 2022 World Cup Finals. This is a reasonable result and is consistent with FiveThirtyEight’s 2022 World Cup Predictor. Thus, the results from this model not only match intuition, but also match results from similar prediction models. Additionally, at the time of submission of this project, the quarterfinal matches of the actual 2022 World Cup will be set. The model has correctly predicted six out of eight of the quarterfinalists. It will be interesting to see if the remaining actual results of the World Cup match the predicted results.

One of the main reasons that I decided to study this problem as my final project was because I knew that I would be motivated to not only work hard on it before submitting, but to continue to work on it after the course is over. So, I will continue to work on this project outside of this assignment, and my first goal is to increase the prediction accuracy. As mentioned above, the best way to improve the prediction accuracy would be to add new features. There are several additional features that could be added to improve the accuracy of this model. Two simple examples would be the number of players on the team that participates in the

Champion's League (the premier club team tournament), and the record of the manager with the team. After watching World Cup games for the past few weeks, both seem like they would be informative features that would be beneficial to add. However, I did not have the time to add them here because I could not find sufficient datasets. Another interesting direction for this project would be tackle the more complex method of trying to predict the match of an outcome by predicting the number of goals scored by each team assuming the number of goals follows a Poisson distribution (as FiveThirtyEight does). This method requires you to define a rating system to evaluate the rating of each team. FiveThirtyEight uses the Soccer Power Index (SPI), which is their own rating system and is used to determine the appropriate parameters to define the Poisson distribution. This would be an interesting, yet complex, direction for this problem.

References:

[1] - https://projects.fivethirtyeight.com/2022-world-cup-predictions/?ex_cid=rrpromo

[2] - https://www.kaggle.com/datasets/martj42/international-football-results-from-1872-to-2017

[3] - https://www.kaggle.com/datasets/cashncarry/fifaworldranking

[4] - https://data.world/datasets/ballon-dor

Appendix:

[A]

Country	Points	Group	A
Netherlands	9	Winner	Netherlands
Senegal	1	Runner Up	Qatar
Ecuador	3	**Group**	**B**
Qatar	4	Winner	England
England	9	Runner Up	USA
USA	4	**Group**	**C**
Iran	3	Winner	Argentina
Wales	1	Runner Up	Poland
Argentina	9	**Group**	**D**
Poland	4	Winner	France
Mexico	1	Runner Up	Denmark
Saudi Arabia	3	**Group**	**E**
France	9	Winner	Germany
Australia	1	Runner Up	Costa Rica
Tunisia	1	**Group**	**F**
Denmark	6	Winner	Belgium
Japan	0	Runner Up	Croatia
Spain	3	**Group**	**G**
Germany	9	Winner	Brazil
Costa Rica	6	Runner Up	Switzerland
Morocco	1	**Group**	**H**
Croatia	6	Winner	Portugal
Belgium	9	Runner Up	Uruguay
Canada	1		
Brazil	9		
Switzerland	6		
Cameroon	0		
Serbia	3		
Portugal	9		
Korea Republic	3		
Uruguay	6		
Ghana	0		

[B]

Home Team	Away Team	Home Rank	Away Rank	Home Confederation	Away Confederation	Home Rank Change	Away Rank Change	Home Ballon d'Or Nominations	Away Ballon d'Or Nominations	Neutral Field			
Netherlands	USA	8	14	UEFA	CONCACAF	2	-3	1	0	1		Home	Netherlands
Argentina	Denmark	3	10	CONMEBOL	UEFA	2	-1	0	0	1		Home	Argentina
France	Poland	4	26	UEFA	UEFA	-1	1	4	1	1		Home	France
England	Qatar	5	49	UEFA	AFC	-1	-1	3	0	0		Home	England
Germany	Croatia	11	15	UEFA	UEFA	1	0	2	1	1		Home	Germany
Brazil	Uruguay	1	13	CONMEBOL	CONMEBOL	1	4	3	1	1		Home	Brazil
Belgium	Costa Rica	2	34	UEFA	CONCACAF	-1	15	2	0	1		Home	Belgium
Portugal	Switzerland	9	16	UEFA	UEFA	-1	-3	4	0	1		Home	Portugal
Home Team	**Away Team**	**Home Rank**	**Away Rank**	**Home Confederation**	**Away Confederation**	**Home Rank Change**	**Away Rank Change**	**Home Ballon d'Or Nominations**	**Away Ballon d'Or Nominations**	**Neutral Field**			
Netherlands	Argentina	8	3	UEFA	CONMEBOL	2	2	1	0	1		Home	Netherlands
Germany	Brazil	11	1	UEFA	CONMEBOL	1	1	2	3	1		Away	Brazil
England	France	5	4	UEFA	UEFA	-1	-1	3	4	1		Away	France
Portugal	Belgium	9	2	UEFA	UEFA	-1	-1	4	2	1		Home	Portugal
Home Team	**Away Team**	**Home Rank**	**Away Rank**	**Home Confederation**	**Away Confederation**	**Home Rank Change**	**Away Rank Change**	**Home Ballon d'Or Nominations**	**Away Ballon d'Or Nominations**	**Neutral Field**			
Netherlands	Brazil	8	1	UEFA	CONMEBOL	2	1	1	3	1		Away	Brazil
France	Portugal	4	9	UEFA	UEFA	-1	-1	4	4	1		Home	France
Home Team	**Away Team**	**Home Rank**	**Away Rank**	**Home Confederation**	**Away Confederation**	**Home Rank Change**	**Away Rank Change**	**Home Ballon d'Or Nominations**	**Away Ballon d'Or Nominations**	**Neutral Field**			
Brazil	France	1	4	CONMEBOL	UEFA	1	-1	3	4	1		Home	Brazil

```
# CS-249 Final Project
# Jason Chapman
# World Cup Predictor

import pandas as pd
import numpy as np
import seaborn as sns

# Load in Datasets
df_ballondor = pd.read_csv('BallonDOr.csv')
df_fifarank = pd.read_csv('fifa_ranking.csv')
df_results = pd.read_csv('results.csv')

# Filter out results not in date range 1993-2018 and Friendly games
df_fifarank['rank_date'] = pd.to_datetime(df_fifarank['rank_date'])
df_fifarank['year'], df_fifarank['month'] = df_fifarank['rank_date'].dt.year, df_fifarank['rank_date'].dt.month
df_results['date'] = pd.to_datetime(df_results['date'])
df_results['year'], df_results['month'] = df_results['date'].dt.year, df_results['date'].dt.month
first_date = df_fifarank.rank_date[1]
res = df_results[~(df_results['date'] < first_date)]
res = res[~(res['tournament'] == 'Friendly')]

# Creat new dataframe with ballon d'or nominee numbers for each year
df_bdor = pd.DataFrame(columns = ['Country', 'Nominations', 'Year'])
years = df_ballondor['Year'].unique()
for yr in years:
    df_temp = df_ballondor.loc[df_ballondor['Year']==yr]
    counts = df_temp['Nationality'].value_counts()
    counts = counts.to_frame()
    counts = counts.reset_index()
    year = np.full(len(counts),yr)
    counts['Year'] = year
    counts.rename({'index':'Country', 'Nationality':'Nominations'}, axis=1, inplace=True)
    df_bdor = pd.concat([df_bdor,counts],ignore_index=True)

# Create dataframe with informative features
df_train = pd.DataFrame()
home_team = []; away_team = []; home_rank = []; away_rank = []; home_confed = []; away_confed = [];
home_rank_move = []; away_rank_move = []; match_result = []; home_bdor_noms = []; away_bdor_noms
= [];
neutral_field = []
for i in res.index:

    # Outcome of match
    home_team_score = res.home_score[i]
    away_team_score = res.away_score[i]
    if home_team_score > away_team_score:
        match_result.append('Home')
    elif home_team_score < away_team_score:
        match_result.append('Away')
    elif home_team_score == away_team_score:
        match_result.append('Tie')
    else:
        match_result.append(np.nan)
    home_team.append(res.home_team[i])
```

```
away_team.append(res.away_team[i])

# Played on neutral field
if res.neutral[i] == True:
    neutral_field.append(1)
elif res.neutral[i] == False:
    neutral_field.append(0)
else:
    neutral_field.append(np.nan)

# Find FIFA Ranking for each team on date of match
# Find FIFA Ranking movement for each team on date of match
# Find team Confederation
month = res.month[i]
year = res.year[i]
temp = df_fifarank[~(df_fifarank['year'] != year)]
temp = temp[~(temp['month'] != month)]
if len(temp) == 0: # Check if no data for that month and go to the previous ranking
    if month == 1:
        month = 12
        year = year-1
    else:
        month = month-1
    temp = df_fifarank[~(df_fifarank['year'] != year)]
    temp = temp[~(temp['month'] != month)]
i_hometeam = temp[temp['country_full'] == res.home_team[i]].index.values
i_awayteam = temp[temp['country_full'] == res.away_team[i]].index.values
if len(i_hometeam) == 0:
    home_rank.append(np.nan)
    home_confed.append(np.nan)
    home_rank_move.append(np.nan)
elif len(i_hometeam) == 2: # fifarank dataset has some countries listed twice (error in dataset)
    i_hometeam = i_hometeam[1]
    home_rank.append(temp['rank'][int(i_hometeam)])
    home_confed.append(temp['confederation'][int(i_hometeam)])
    if home_confed[-1] == 'OFC': home_confed[-1] = np.nan
    home_rank_move.append(temp['rank_change'][int(i_hometeam)])
else:
    home_rank.append(temp['rank'][int(i_hometeam)])
    home_confed.append(temp['confederation'][int(i_hometeam)])
    if home_confed[-1] == 'OFC': home_confed[-1] = np.nan
    home_rank_move.append(temp['rank_change'][int(i_hometeam)])
if len(i_awayteam) == 0:
    away_rank.append(np.nan)
    away_confed.append(np.nan)
    away_rank_move.append(np.nan)
elif len(i_awayteam) == 2:
    i_awayteam = i_awayteam[1]
    away_rank.append(temp['rank'][int(i_awayteam)])
    away_confed.append(temp['confederation'][int(i_awayteam)])
    if away_confed[-1] == 'OFC': away_confed[-1] = np.nan
    away_rank_move.append(temp['rank_change'][int(i_awayteam)])
else:
    away_rank.append(temp['rank'][int(i_awayteam)])
    away_confed.append(temp['confederation'][int(i_awayteam)])
```

```
        if away_confed[-1] == 'OFC': away_confed[-1] = np.nan
        away_rank_move.append(temp['rank_change'][int(i_awayteam)])

    # Find # of Ballon d'Or Nominations for that year
    if ((df_bdor['Year']==res.year[i]) & (df_bdor['Country']==res.home_team[i])).any():
        index = df_bdor.index[((df_bdor['Year']==res.year[i]) & (df_bdor['Country']==res.home_team[i]))]
        home_bdor_noms.append(df_bdor['Nominations'][index[0]])
    else:
        home_bdor_noms.append(0)
    if ((df_bdor['Year']==res.year[i]) & (df_bdor['Country']==res.away_team[i])).any():
        index = df_bdor.index[((df_bdor['Year']==res.year[i]) & (df_bdor['Country']==res.away_team[i]))]
        away_bdor_noms.append(df_bdor['Nominations'][index[0]])
    else:
        away_bdor_noms.append(0)
df_train['Home Team'] = home_team; df_train['Away Team'] = away_team; df_train['Home Rank'] = home
_rank;
df_train['Away Rank'] = away_rank; df_train['Home Confederation'] = home_confed;
df_train['Away Confederation'] = away_confed; df_train['Home Rank Change'] = home_rank_move;
df_train['Away Rank Change'] = away_rank_move; df_train["Home Ballon d'Or Nominations"] = home_bd
or_noms;
df_train["Away Ballon d'Or Nominations"] = away_bdor_noms; df_train['Neutral Field'] = neutral_field;
df_train['Result'] = match_result
df_train = df_train.dropna()

# Turn Categorical Variables into Numerical Variables
X = df_train.drop(['Home Team', 'Away Team', 'Result'], axis=1)
X = pd.get_dummies(X)
y = df_train.Result
X.reset_index(drop=True,inplace=True)
y = y.to_frame()
y.reset_index(drop=True,inplace=True)
y = np.ravel(y)

# Scale data for model convergence
from sklearn import preprocessing
scaler = preprocessing.StandardScaler().fit(X)
X_scaled = scaler.transform(X)

# Define the multinomial logistic regression model
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(multi_class='multinomial', solver='lbfgs')

# Find best regularization parameter
from sklearn.model_selection import cross_val_score
C = np.linspace(1, 10, 10)
acc = []
for cs in C:
    model = LogisticRegression(multi_class='multinomial', C=cs, solver='lbfgs')
    acc.append(np.mean(cross_val_score(model, X_scaled, y, cv=10)))
acc = np.array(acc)
ind = np.where(acc == acc.max())
param = C[ind][0]

# Decision Tree
from sklearn.tree import DecisionTreeClassifier
```

```
dt = DecisionTreeClassifier(random_state=0)
dt_accuracy = np.mean(cross_val_score(dt, X_scaled, y, cv=10))

# Feature Selection
from sklearn.feature_selection import SequentialFeatureSelector
sfs = SequentialFeatureSelector(model).fit(X_scaled, y)
features = sfs.get_support()
X_transf = df_train.drop(['Home Team', 'Away Team', 'Home Confederation', 'Away Confederation', 'Resu
lt'], axis=1)
X_transf = pd.get_dummies(X_transf)
y_transf = df_train.Result
X_transf.reset_index(drop=True,inplace=True)
y_transf = y_transf.to_frame()
y_transf.reset_index(drop=True,inplace=True)
y_transf = np.ravel(y_transf)
scaler_transf = preprocessing.StandardScaler().fit(X_transf)
X_scaled_transf = scaler_transf.transform(X_transf)
fs_accuracy = np.mean(cross_val_score(model, X_scaled_transf, y_transf, cv=10))
mlr_accuracy = np.mean(cross_val_score(model, X_scaled, y, cv=10))

# Cross Validation
from sklearn.model_selection import train_test_split
X_train, X_val, y_train, y_val = train_test_split(X_scaled, y, test_size=0.1, shuffle=True, random_state=0)
model = LogisticRegression(multi_class='multinomial', solver='lbfgs')
model.fit(X_train,y_train)
accuracy_train = model.score(X_train,y_train)
model.fit(X_val,y_val)
accuracy_val = model.score(X_val,y_val)

# Fit the model on the whole dataset
model.fit(X_scaled, y)
model_coefs = model.coef_
from sklearn.metrics import confusion_matrix
prediction = model.predict(X_scaled)
cm = confusion_matrix(y, prediction)

# Pull in table of groupstage matches (test data)
df_groupstage = pd.read_csv('test_data1.csv')
X_test = df_groupstage.drop(['Home Team', 'Away Team'], axis=1)
X_test = pd.get_dummies(X_test)
X_test_scaled = scaler.transform(X_test)
groupstage_results = model.predict(X_test_scaled)

# No ties in round of 16, so create new regression model with new scaled data
y = df_train.Result
y = y.to_frame()
y.reset_index(drop=True,inplace=True)
i_tie = y.index[y['Result']=="Tie"]
X.drop(X.index[np.array(i_tie)], inplace=True)
y.drop(y.index[np.array(i_tie)], inplace=True)
scaler = preprocessing.StandardScaler().fit(X)
X_scaled = scaler.transform(X)
model = LogisticRegression()
y = np.ravel(y)
model.fit(X_scaled, y)
```

```
X_train, X_val, y_train, y_val = train_test_split(X_scaled, y, test_size=0.1, shuffle=True, random_state=0)
model.fit(X_train,y_train)
accuracy_train2 = model.score(X_train,y_train)
model.fit(X_val,y_val)
accuracy_val2 = model.score(X_val,y_val)

# Pull in table of round of 16 matches
df_ro16 = pd.read_csv('round_of_16.csv')
X_test2 = df_ro16.drop(['Home Team', 'Away Team'], axis=1)
X_test2 = pd.get_dummies(X_test2)
# Ensure df matches df of model
X_test2['Home Confederation_CAF'] = np.full(len(X_test2),0)
X_test2['Home Confederation_AFC'] = np.full(len(X_test2),0)
X_test2['Home Confederation_CONCACAF'] = np.full(len(X_test2),0)
X_test2['Away Confederation_CAF'] = np.full(len(X_test2),0)
X_test2 = X_test2[X_test.columns]
X_test2_scaled = scaler.transform(X_test2)
ro16_results = model.predict(X_test2_scaled)

# Pull in table of quarterfinals matches
df_qf = pd.read_csv('quarter_final.csv')
X_test3 = df_qf.drop(['Home Team', 'Away Team'], axis=1)
X_test3 = pd.get_dummies(X_test3)
# Ensure df matches df of model
X_test3['Home Confederation_CAF'] = np.full(len(X_test3),0)
X_test3['Home Confederation_AFC'] = np.full(len(X_test3),0)
X_test3['Home Confederation_CONCACAF'] = np.full(len(X_test3),0)
X_test3['Home Confederation_CONMEBOL'] = np.full(len(X_test3),0)
X_test3['Away Confederation_CAF'] = np.full(len(X_test3),0)
X_test3['Away Confederation_AFC'] = np.full(len(X_test3),0)
X_test3['Away Confederation_CONCACAF'] = np.full(len(X_test3),0)
X_test3 = X_test3[X_test.columns]
X_test3_scaled = scaler.transform(X_test3)
qf_results = model.predict(X_test3_scaled)

# Pull in table of semifinals matches
df_sf = pd.read_csv('semi_final.csv')
X_test4 = df_sf.drop(['Home Team', 'Away Team'], axis=1)
X_test4 = pd.get_dummies(X_test4)
# Ensure df matches df of model
X_test4['Home Confederation_CAF'] = np.full(len(X_test4),0)
X_test4['Home Confederation_AFC'] = np.full(len(X_test4),0)
X_test4['Home Confederation_CONCACAF'] = np.full(len(X_test4),0)
X_test4['Home Confederation_CONMEBOL'] = np.full(len(X_test4),0)
X_test4['Away Confederation_CAF'] = np.full(len(X_test4),0)
X_test4['Away Confederation_AFC'] = np.full(len(X_test4),0)
X_test4['Away Confederation_CONCACAF'] = np.full(len(X_test4),0)
X_test4 = X_test4[X_test.columns]
X_test4_scaled = scaler.transform(X_test4)
sf_results = model.predict(X_test4_scaled)

# Pull in table of finals match
df_final = pd.read_csv('final.csv')
X_test5 = df_final.drop(['Home Team', 'Away Team'], axis=1)
X_test5 = pd.get_dummies(X_test5)
```

```
# Ensure df matches df of model
X_test5['Home Confederation_CAF'] = np.full(len(X_test5),0)
X_test5['Home Confederation_AFC'] = np.full(len(X_test5),0)
X_test5['Home Confederation_CONCACAF'] = np.full(len(X_test5),0)
X_test5['Home Confederation_UEFA'] = np.full(len(X_test5),0)
X_test5['Away Confederation_CONMEBOL'] = np.full(len(X_test5),0)
X_test5['Away Confederation_CAF'] = np.full(len(X_test5),0)
X_test5['Away Confederation_AFC'] = np.full(len(X_test5),0)
X_test5['Away Confederation_CONCACAF'] = np.full(len(X_test5),0)
X_test5 = X_test5[X_test.columns]
X_test5_scaled = scaler.transform(X_test5)
final_results = model.predict(X_test5_scaled)
```