|  |  |  |
| --- | --- | --- |
| Open Source Develop Project Final Project 2022 World cup Qatar Winner Prediction | 학과 | 컴퓨터공학과 |
| 학번 | 2019037004 |
| 이름 | 김남현 |

1. Data Collection

This is a project to predict the winning country using machine learning techniques for the Soccer World Cup 2022 Prediction category in the sports field of Keggle (https://www.kaggle.com/datasets).

2. Data Analysis

1) Simple statistical analysis after data grouping using the groupby() function

텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명텍스트, 모니터, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

↑raw data ↑ Grouping by using groupby() function

2) Draw two graphs using matplotlib

텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명

example of Korea Republic’s Win, Lose, Draw since 1800.

텍스트, 스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

Representation of all-time win rate of all countries in Group H as a pie chart

3) Calculate model training and model evaluation metrics using one or more machine learning techniques

# test, train값 산출

logreg = LogisticRegression()

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

score = logreg.score(X\_train, y\_train)

score2 = logreg.score(X\_test, y\_test)

print("Training set accuracy: ", '%.3f'%(score))

print("Test set accuracy: ", '%.3f'%(score2))

def clean\_and\_predict(matches, df1, final, logreg):

    # Initialization of auxiliary list for data cleaning

    positions = []

    # Loop to retrieve each team's position according to FIFA ranking

    for match in matches:

        positions.append(df1.loc[df1['Team'] == match[1],'FIFA Ranking'].iloc[0])

        positions.append(df1.loc[df1['Team'] == match[1],'FIFA Ranking'].iloc[0])

    # Creating the DataFrame for prediction

    pred\_set = []

    # Initializing iterators for while loop

    i = 0

    j = 0

    # 'i' will be the iterator for the 'positions' list, and 'j' for the list of matches (list of tuples)

    while i < len(positions):

        dict1 = {}

        # If position of first team is better, he will be the 'home' team, and vice-versa

        if positions[i] < positions[i + 1]:

            dict1.update({'Home Team': matches[j][0], 'Away Team': matches[j][1]})

        else:

            dict1.update({'Home Team': matches[j][1], 'Away Team': matches[j][0]})

        # Append updated dictionary to the list, that will later be converted into a DataFrame

        pred\_set.append(dict1)

        i += 2

        j += 1

    # Convert list into DataFrame

    pred\_set = pd.DataFrame(pred\_set)

    backup\_pred\_set = pred\_set

    # Get dummy variables and drop winning\_team column

    pred\_set = pd.get\_dummies(pred\_set, prefix=['Home Team', 'Away Team'], columns=['Home Team', 'Away Team'])

    # Add missing columns compared to the model's training dataset

    missing\_cols2 = set(final.columns) - set(pred\_set.columns)

    for c in missing\_cols2:

        pred\_set[c] = 0

    pred\_set = pred\_set[final.columns]

    # Remove winning team column

    pred\_set = pred\_set.drop(['Winning Team'], axis=1)

    # Predict!

    predictions = logreg.predict(pred\_set)

    for i in range(len(pred\_set)):

        print(backup\_pred\_set.iloc[i, 1] + " and " + backup\_pred\_set.iloc[i, 0])

        if predictions[i] == 2:

            print("Winner: " + backup\_pred\_set.iloc[i, 1])

        elif predictions[i] == 1:

            print("Draw")

        elif predictions[i] == 0:

            print("Winner: " + backup\_pred\_set.iloc[i, 0])

        print('Probability of ' + backup\_pred\_set.iloc[i, 1] + ' winning: ' , '%.3f'%(logreg.predict\_proba(pred\_set)[i][2]))

        print('Probability of Draw: ', '%.3f'%(logreg.predict\_proba(pred\_set)[i][1]))

        print('Probability of ' + backup\_pred\_set.iloc[i, 0] + ' winning: ', '%.3f'%(logreg.predict\_proba(pred\_set)[i][0]))

        print("")

clean\_and\_predict(Round\_of\_16, df1, final, logreg)

3. Github repository URL

<https://github.com/kimnamhyeon0112/2022-2_OSP_Final_Project>

4. Result interpretation and application direction design

The final result, the World Cup winner, is the Netherlands, and the runner-up is England.

Looking at the data analysis part 2, select the country to draw the graph in the international\_matches.csv file, calculate the number of all matches from the beginning of the country to September 2022 based on the date, calculate the win/loss/draw, and create a bar graph It was made and the result was derived. Using the groupby() function, the all-time win rates of the countries in each group were expressed as a pie chart, and the graph was drawn as shown in the figure above. However, there is no guarantee that you will win unconditionally with a high win rate in soccer, and since there are differences in the number of wins/losses and the probability depending on which country the opponent is, it is better to view it as sub-contents.

Looking at the prediction part, in the 2022\_world\_cup\_groups.csv file, the countries participating in this World Cup are substituted into international\_matches.csv to measure the total number of matches, calculate and compare the test and train values, and match the actual Qatar World Cup match schedule and confrontation country Predict the outcome of the match and predict the final winning country.

In this project, I analyzed only which country was the winner, but it would be nice to design a direction to predict what the final score would be if an arbitrary World Cup participating country was input. In addition, it can be used as an event in which a small reward is paid if the correct answer is obtained by conducting a survey on the final results of the winning country and an arbitrary country in cooperation with a game company or event company related to soccer. If an app is created by supplementing the program, it would be nice to make it so that users can make their own matches and share them with their friends through a messenger app.

5. Report

This figure is a bar graph representation of Korea's all-time wins/losses/draws.

텍스트, 스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명

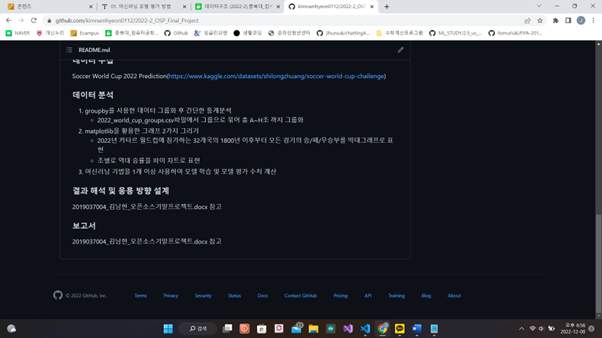
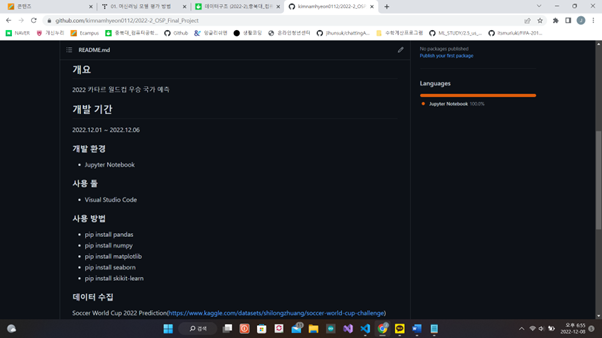
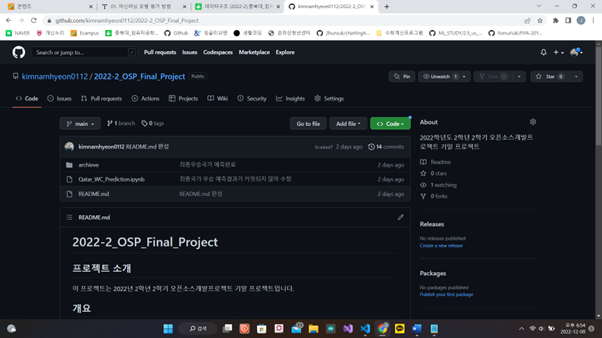
In addition to this, all 32 countries participating in the World Cup in Qatar have been represented in a bar graph of all wins and losses in all-time matches.

This figure is a pie chart representation of the all-time win rates of the four countries in Group H of the Qatar World Cup.

텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

In addition, the win rates of all countries from Group A to Group H were expressed in a pie chart after dividing by group using the groupby() function.



This figure shows the history of uploading to my personal repository after completing the project.

According to the Alan Turing Institute, a British data science and artificial intelligence research institute, the winner of the 2022 Qatar World Cup is Brazil. did Therefore, based on Keggle's data and the machine learning technique I used, I was curious about whether Brazil would really win the championship, or whether Brazil would be the actual winner, so I adopted this topic.

http://www.newsmc.net/news/articleView.html?idxno=56170 As shown in the article, it is highly likely that the analysis was quite accurate as the Alan Turing Institute is called the world's top three artificial intelligence research institutes. Contrary to the results, the result was that the Netherlands won, so I will see if the Netherlands will actually win. Of course, I think that Brazil has a high chance of winning, and no one knows which country will win the World Cup trophy, but if the Netherlands, the predicted winner of my project, wins, I think I can get a satisfactory result.