2021 한국외대 데이터 청년 캠퍼스

2020 도쿄 올림픽
축구 우승 예측 프로젝트

INDEX

프로젝트 소개

FIFA

1. 과거의 매치 데이터와 FIFA 랭킹을 토대로
   도쿄 올림픽 축구 경기의 **최종 우승 국가 예측**

2. 팀원 대부분의 관심사가 **축구**로 모아질 뿐더러
   현재 올림픽이 진행되는 기간으로 우승 국가를
   직접 예측해보는 기회를 가져보고자 기획

TOKYO 2020

프로젝트 내용

1. FIFA 랭킹과 국제 경기 기록을 바탕으로 데이터셋 구축
2. Target 국가는 home, 상대 국가는 away로 구분하여 데이터 전처리 진행
3. Simulation은 조별 리그 예측까지 포함한 결과와 이번 조별 리그의 결과를 반영한 두가지로 진행

Datasets

rankings FIFA Ranking with Olympic Medals

	rank	country_full	country_abrv	rank_date	accumulate_gold	accumulate_silver	accumulate_bronze	accumulate_forth
0	199	Montenegro	MNE	2007-06-13	0	0	0	0
1	199	Montenegro	MNE	2007-07-18	0	0	0	0
2	199	Montenegro	MNE	2007-08-22	0	0	0	0
3	186	Montenegro	MNE	2007-09-19	0	0	0	0
4	171	Montenegro	MNE	2007-10-24	0	0	0	0
...	...	...	...	...	...	...	...	...
63049	117	Kosovo	KVX	2020-11-26	0	0	0	0
63050	117	Kosovo	KVX	2020-12-10	0	0	0	0
63051	117	Kosovo	KVX	2021-02-18	0	0	0	0
63052	120	Kosovo	KVX	2021-04-07	0	0	0	0
63053	120	Kosovo	KVX	2021-05-27	0	0	0	0

63054 rows × 8 columns

matches Results(1872-2021)

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	False
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	False
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	False
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	False
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	False
...	...	...	...	...	...	...	...	...	...
42423	2021-07-06	Trinidad and Tobago	French Guiana	1	1	Gold Cup qualification	Fort Lauderdale	United States	True
42424	2021-07-07	England	Denmark	2	1	UEFA Euro	London	England	False
42425	2021-07-09	Peru	Colombia	2	3	Copa América	Brasília	Brazil	True
42426	2021-07-10	Brazil	Argentina	0	1	Copa América	Rio de Janeiro	Brazil	False
42427	2021-07-11	England	Italy	1	1	UEFA Euro	London	England	False

42428 rows × 9 columns

olympic Olympic 2020 Dataset

Team	Group	First match against	Second match against	Third match against	Advance to 8
Mexico	A	France	Japan	South Africa	Y
France	A	Mexico	South Africa	Japan	NaN
South Africa	A	Japan	France	Mexico	NaN
Japan	A	South Africa	Mexico	France	Y
Korea Republic	B	New Zealand	Romania	Honduras	Y
Honduras	B	Romania	New Zealand	Korea Republic	NaN
New Zealand	B	Korea Republic	Honduras	Romania	Y
Romania	B	Honduras	Korea Republic	New Zealand	NaN
Spain	C	Egypt	Australia	Argentina	Y
Australia	C	Argentina	Spain	Egypt	NaN
Egypt	C	Spain	Argentina	Australia	Y
Argentina	C	Australia	Egypt	Spain	NaN
Ivory Coast	D	Saudi Arabia	Brazil	Germany	Y
Brazil	D	Germany	Ivory Coast	Saudi Arabia	Y
Germany	D	Brazil	Saudi Arabia	Iovry Coast	NaN
Saudi Arabia	D	Ivory Coast	Germany	Brazil	NaN

Features

In [2]:
```
#메달에 따른 가산점 추가
rankings['medal_points'] = (rankings['accumulate_gold']*4)+(rankings['accumulate_silver']*3)+\
                           (rankings['accumulate_bronze']*2)+rankings['accumulate_forth']
```

In [3]:
```
#matches와 merge하기 위해 rankings를 시계열 데이터로 늘이기
rankings = rankings.set_index(['rank_date'])\
            .groupby(['country_full'], group_keys=False)\
            .resample('D').first()\
            .fillna(method='ffill')\
            .reset_index()
```

- 각 메달에 따른 가중치 추가(금메달은 4점, 은메달은 3점, 동메달은 2점, 4위는 1점)
- Matches와 merge하기 위해 rankings를 시계열 데이터화

Features

In [4]:
```
# rankings와 join
matches = matches.merge(rankings,
                        left_on=['date', 'home_team'],
                        right_on=['rank_date', 'country_full'])
matches = matches.merge(rankings,
                        left_on=['date', 'away_team'],
                        right_on=['rank_date', 'country_full'],
                        suffixes=('_home', '_away'))
```

In [5]:
```
#feture 추가
matches['rank_difference'] = matches['rank_home'] - matches['rank_away']
matches['average_rank'] = (matches['rank_home'] + matches['rank_away'])/2
matches['score_difference'] = matches['home_score'] - matches['away_score']
matches['medal_difference'] = matches['medal_points_home'] - matches['medal_points_away']
matches['is_won'] = matches['score_difference'] > 0 #무승부는 패배 판정
matches['is_stake'] = matches['tournament'] != 'Friendly'

#올림픽 참가 여부 추가
matches['op_participant'] = matches['home_team'] * matches['home_team'].isin(olympic.index.tolist())
matches['op_participant'] = matches['op_participant'].replace({'':'Other'})
matches = matches.join(pd.get_dummies(matches['op_participant']))
```

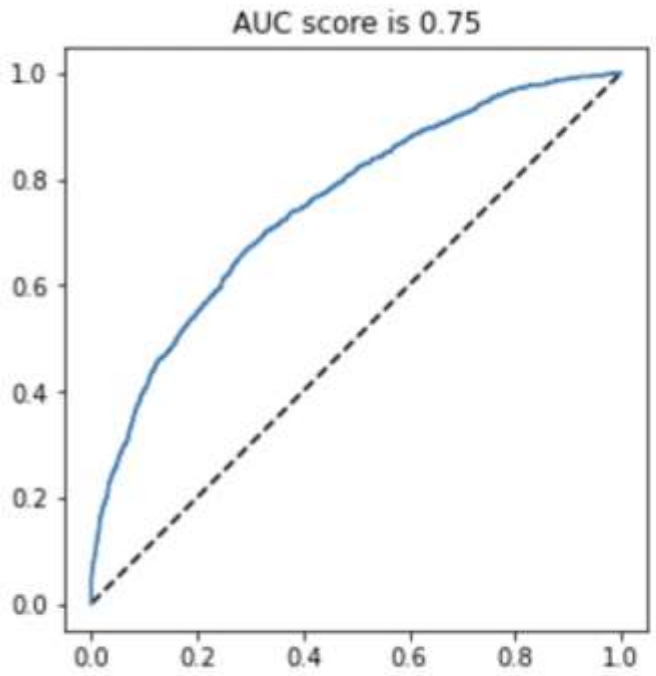
- rankings의 column들을 home, away으로 각각 구분하여 matches와 join
- 양팀의 FIFA 랭크, 점수, 메달 수를 이용해 새로운 feature들 추가
- 무승부는 패배 판정, 각 나라의 이번 올림픽 참가 여부 추가

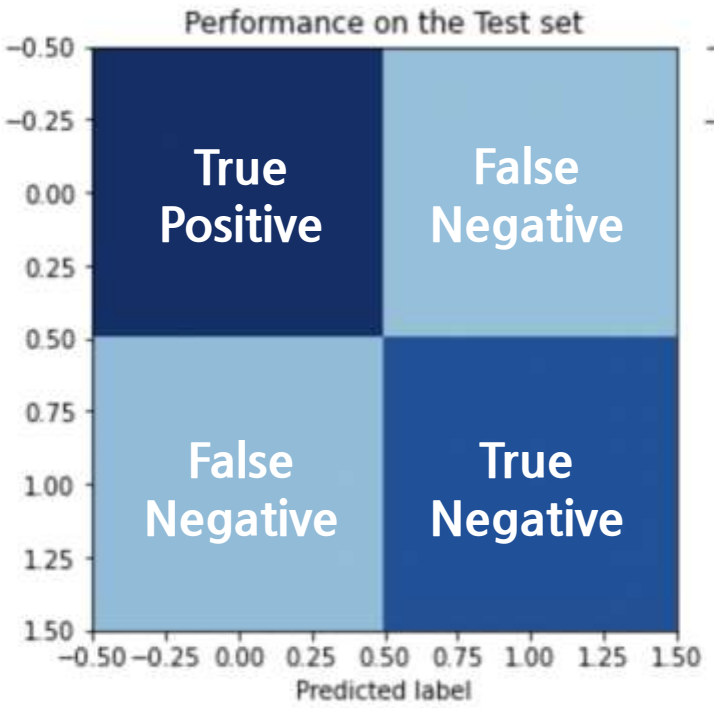
Modeling

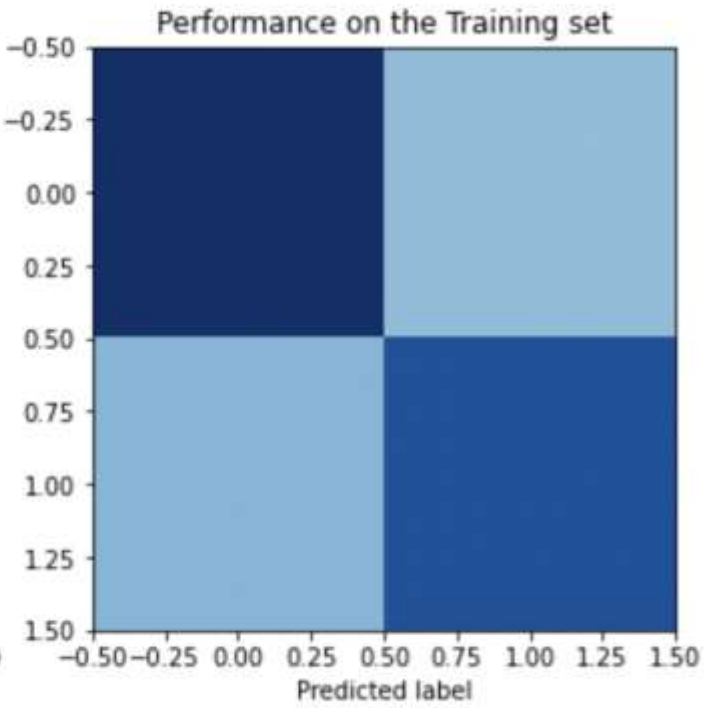
```
from sklearn import linear_model
from sklearn import ensemble
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, roc_curve, roc_auc_score
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures

X, y = matches.loc[:,['average_rank', 'rank_difference', 'medal_difference', 'is_stake' ]], matches['is_won']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

logreg = linear_model.LogisticRegression(C=1e-5) #강도를 낮춰주는 파라미터, C값이 낮을수록 계수를 0으로 근사하므로 정규화가 강화된다.
features = PolynomialFeatures(degree=2)

model = Pipeline([  #서로 다른 매개변수를 설정하면서 함께 교차로 검증할 수 있는 단계를 종합
    ('polynomial_features', features),
    ('logistic_regression', logreg)])

model = model.fit(X_train, y_train)
```

- 로지스틱 회귀 모델과 다항 회귀 사용

Figures

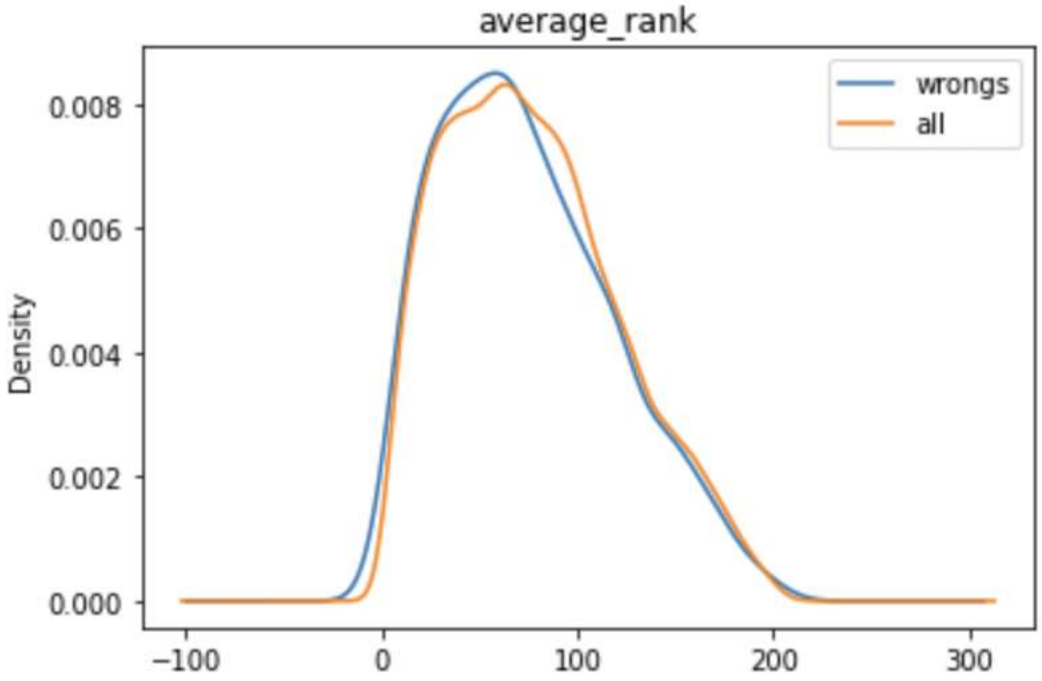


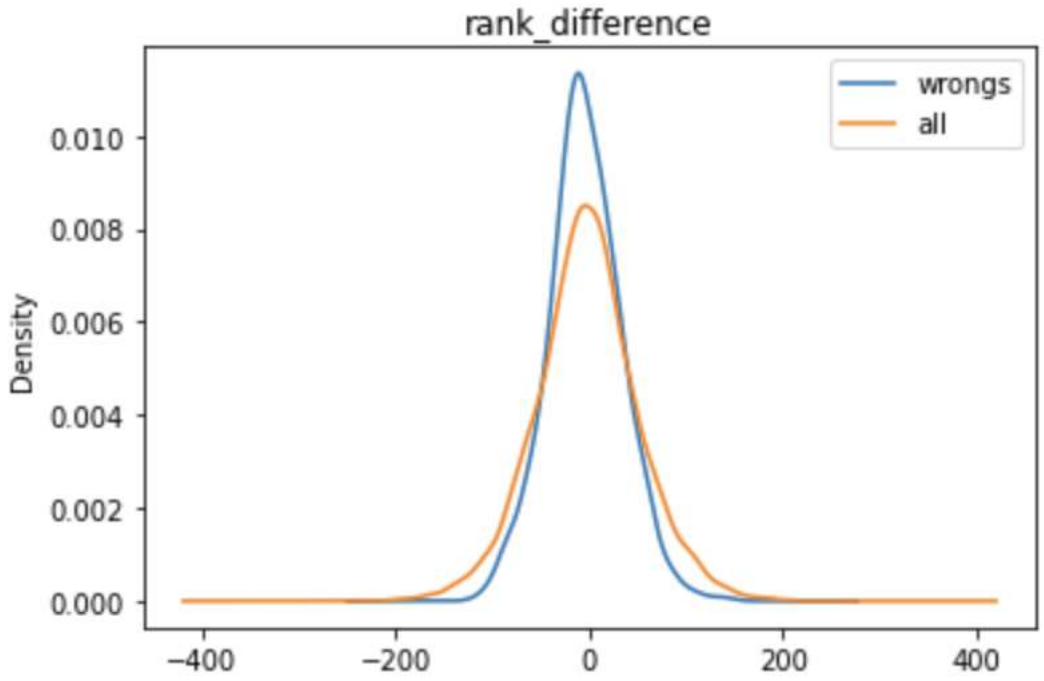




- 위의 파란 곡선 아래 면적의 넓이는 AUC
- 양성으로 잘 예측한 경우가 많으면(민감도가 높을수록) 좋은 모델(그래프의 왼쪽 위)
- True 영역이 진할수록 정확도가 높음
- 모델 간의 차이는 크지 않았음

Figures





- 양팀의 FIFA 랭크 평균의 특정 구간에서 예측 정확도가 떨어짐
- 양팀의 FIFA 랭크가 비슷할 경우 예측 정확도가 떨어짐

Simulation

```
margin = 0.05

olympic_rankings = rankings.loc[(rankings['rank_date'] == rankings['rank_date'].max()) &
                                rankings['country_full'].isin(olympic.index.unique()) ]
olympic_rankings = olympic_rankings.set_index(['country_full'])

#Home Team Advantage(개최국에 2배 가중치)
olympic_rankings.loc[['Japan'],['medal_points']] = olympic_rankings.loc[['Japan'],['medal_points']]*2

#가장 최근 랭킹에서 올림픽에 참가하는 팀 선별
```

- 올림픽 참가팀의 FIFA 랭킹은 최신 랭킹 반영
- 개최국은 medal_points에 두배의 가산점 부여

Simulation

country_full	rank_date	rank	country_abrv	accumulate_gold	accumulate_silver	accumulate_bronze	accumulate_forth	medal_points
Argentina	2021-05-27	8.0	ARG	2.0	2.0	0.0	0.0	14.0
Australia	2021-05-27	41.0	AUS	0.0	0.0	0.0	1.0	1.0
Brazil	2021-05-27	3.0	BRA	1.0	3.0	2.0	1.0	18.0
Egypt	2021-05-27	46.0	EGY	0.0	0.0	0.0	2.0	2.0
France	2021-05-27	2.0	FRA	1.0	1.0	0.0	0.0	7.0
Germany	2021-05-27	12.0	GER	0.0	1.0	2.0	1.0	8.0
Honduras	2021-05-27	67.0	HON	0.0	0.0	0.0	1.0	1.0
Ivory Coast	2021-05-27	59.0	CIV	0.0	0.0	0.0	0.0	0.0
Japan	2021-05-27	28.0	JPN	0.0	0.0	1.0	1.0	6.0
Korea Republic	2021-05-27	39.0	KOR	0.0	0.0	1.0	0.0	2.0
Mexico	2021-05-27	11.0	MEX	1.0	0.0	0.0	1.0	5.0
New Zealand	2021-05-27	122.0	NZL	0.0	0.0	0.0	0.0	0.0
Romania	2021-05-27	43.0	ROU	0.0	0.0	0.0	0.0	0.0
Saudi Arabia	2021-05-27	65.0	KSA	0.0	0.0	0.0	0.0	0.0
South Africa	2021-05-27	75.0	RSA	0.0	0.0	0.0	0.0	0.0
Spain	2021-05-27	6.0	ESP	1.0	2.0	0.0	0.0	10.0

Simulation 조별 경기

```
for group in sorted(set(olympic['Group'])):
    print('---Group {}---'.format(group))
    for home, away in combinations(olympic.query('Group == "{}"'.format(group)).index, 2):
        print("{} vs. {}: ".format(home, away), end='')
        row = pd.DataFrame(np.array([[np.nan, np.nan, np.nan, True]]), columns=X_test.columns)
        home_rank = olympic_rankings.loc[home, 'rank']
        home_points = olympic_rankings.loc[home, 'medal_points']
        opp_rank = olympic_rankings.loc[away, 'rank']
        opp_points = olympic_rankings.loc[away, 'medal_points']

        row['average_rank'] = (home_rank + opp_rank) / 2
        row['rank_difference'] = home_rank - opp_rank
        row['medal_difference'] = home_points - opp_points

        home_win_prob = model.predict_proba(row)[:,1][0]
        olympic.loc[home, 'total_prob'] += home_win_prob
        olympic.loc[away, 'total_prob'] += 1-home_win_prob

        points = 0
```

```
if home_win_prob <= 0.5 - margin:
    print('\033[31m' + "{} Wins with {:.2f}".format(away, 1-home_win_prob) + '\033[0m')
    olympic.loc[away, 'points'] += 3
if home_win_prob > 0.5 - margin:
    points = 1
if home_win_prob >= 0.5 + margin:
    points = 3
    olympic.loc[home, 'points'] += 3
    print('\033[31m' + "{} Wins with {:.2f}".format(home, home_win_prob) + '\033[0m')
if points == 1:
    print('\033[31m' + "Draw" + '\033[0m')
    olympic.loc[home, 'points'] += 1
    olympic.loc[away, 'points'] += 1
```

Simulation 조별 경기

```
---Group A---
Mexico vs. France: France Wins with 0.56
Mexico vs. South Africa: Mexico Wins with 0.79
Mexico vs. Japan: Mexico Wins with 0.57
France vs. South Africa: France Wins with 0.81
France vs. Japan: France Wins with 0.62
South Africa vs. Japan: Japan Wins with 0.82
```

```
---Group B---
Korea Republic vs. Honduras: Korea Republic Wins with 0.62
Korea Republic vs. New Zealand: Korea Republic Wins with 0.85
Korea Republic vs. Romania: Draw
Honduras vs. New Zealand: Honduras Wins with 0.75
Honduras vs. Romania: Romania Wins with 0.64
New Zealand vs. Romania: Romania Wins with 0.85
```

```
---Group C---
Spain vs. Australia: Spain Wins with 0.65
Spain vs. Egypt: Spain Wins with 0.68
Spain vs. Argentina: Draw
Australia vs. Egypt: Draw
Australia vs. Argentina: Argentina Wins with 0.81
Egypt vs. Argentina: Argentina Wins with 0.83
```

```
---Group D---
Ivory Coast vs. Brazil: Brazil Wins with 0.93
Ivory Coast vs. Germany: Germany Wins with 0.82
Ivory Coast vs. Saudi Arabia: Draw
Brazil vs. Germany: Draw
Brazil vs. Saudi Arabia: Brazil Wins with 0.72
Germany vs. Saudi Arabia: Germany Wins with 0.74
```

Simulation 실제 조별 경기 결과 반영

```
#대진 순 8강 진출 팀 입력
next_round_olympic = olympic.loc[olympic['Advance to 8'].isnull() == False]
next_round_olympic = next_round_olympic.reset_index()
next_round_olympic
```

	Team	Group	First match against	Second match against	Third match against	Advance to 8	points	total_prob
0	Mexico	A	France	Japan	South Africa	Y	6	1.793164
1	Japan	A	South Africa	Mexico	France	Y	3	1.635610
2	Korea Republic	B	New Zealand	Romania	Honduras	Y	7	1.971487
3	New Zealand	B	Korea Republic	Honduras	Romania	Y	0	0.548857
4	Spain	C	Egypt	Australia	Argentina	Y	7	1.822281
5	Egypt	C	Spain	Argentina	Australia	Y	1	1.012835
6	Ivory Coast	D	Saudi Arabia	Brazil	Germany	Y	1	0.734709
7	Brazil	D	Germany	Ivory Coast	Saudi Arabia	Y	7	2.161336

Simulation 8강, 준결승, 3위 결정전, 결승

```
    # 다음 경기 진출팀, 3위 결정전 진출팀, 메달 결정
    if f == 'Semifinal':
        third_place_match.append(lose) # 준결승전이면 패자를 3위 결정전 진출팀으로 추가
    if f != 'Third_Place_Match':
        winners.append(win) # 3위 결정전을 제외한 모든 경기는 승자를 winners에 추가
    else:
        medals.append(win) # Bronze
    if f == 'Final':
        medals.append(lose) # Sliver
        medals.append(win) # Gold

    labels.append("{}({:.2f}) vs. {}({:.2f})".format(olympic_rankings.loc[home, 'country_abrv'], 1/home_win_prob,
                                                     olympic_rankings.loc[away, 'country_abrv'], 1/(1-home_win_prob
    odds.append([home_win_prob, 1-home_win_prob])

if f != 'Third_Place_Match': # 3위 결정전일때는 next_round_olympic에 승자를 반영하지 않음
    next_round_olympic_ = next_round_olympic_.loc[winners]
```

Simulation 8강, 준결승, 3위 결정전, 결승

```
---Round_of_8---
France vs. Korea Republic: France Wins with 0.67
Argentina vs. Germany: Argentina Wins with 0.51
Romania vs. Spain: Spain Wins with 0.79
Mexico vs. Brazil: Brazil Wins with 0.65


---Semifinal---
France vs. Argentina: Argentina Wins with 0.50
Spain vs. Brazil: Brazil Wins with 0.56


---Third_Place_Match---
France vs. Spain: France Wins with 0.51


---Final---
Argentina vs. Brazil: Brazil Wins with 0.55


Gold   : Brazil
Silver : Argentina
Bronze : France
```

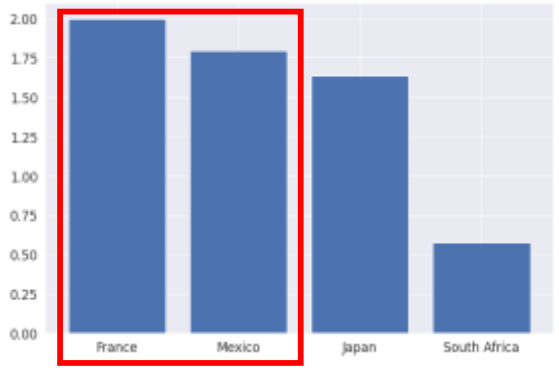
Simulation 실제 조별 경기 결과 반영

```
---Round_of_8---
Mexico vs. Korea Republic: Mexico Wins with 0.63
Brazil vs. Egypt: Brazil Wins with 0.64
Japan vs. New Zealand: Japan Wins with 0.89
Spain vs. Ivory Coast: Spain Wins with 0.73


---Semifinal---
Mexico vs. Brazil: Brazil Wins with 0.65
Japan vs. Spain: Spain Wins with 0.65


---Third_Place_Match---
Mexico vs. Japan: Mexico Wins with 0.57


---Final---
Brazil vs. Spain: Spain Wins with 0.51


Gold   : Spain
Silver : Brazil
Bronze : Mexico
```

Visualization 예선전 그룹별 결과

Pipeline

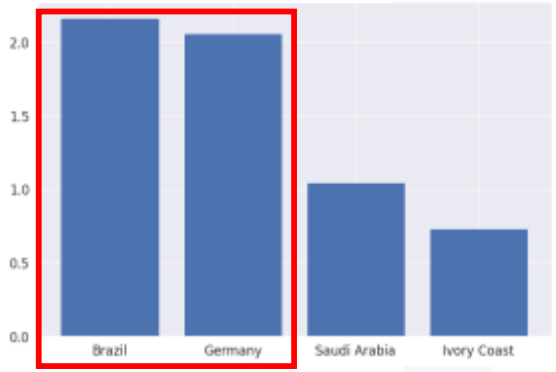
A group total prob



B group total prob

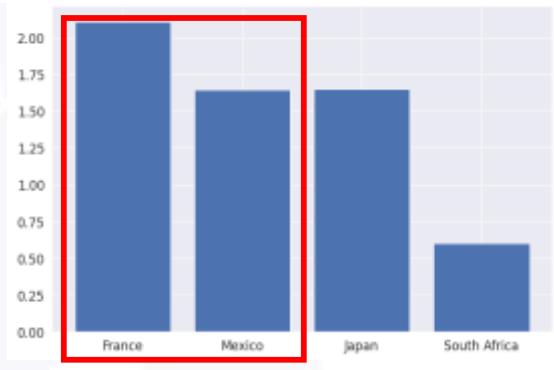
C group total prob

D group total prob

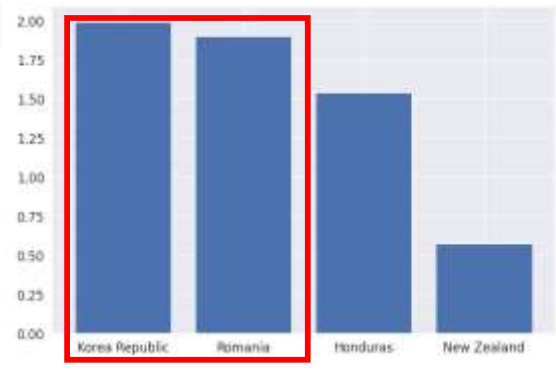


RandomForest

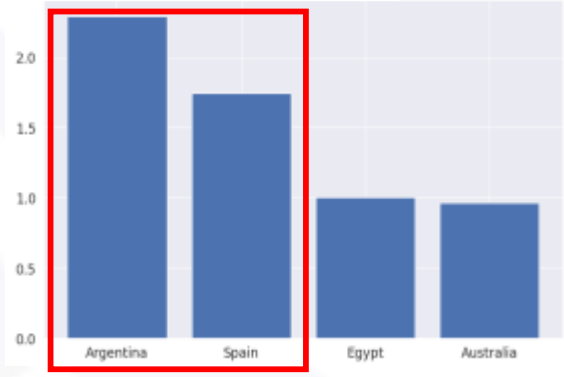
A group total prob



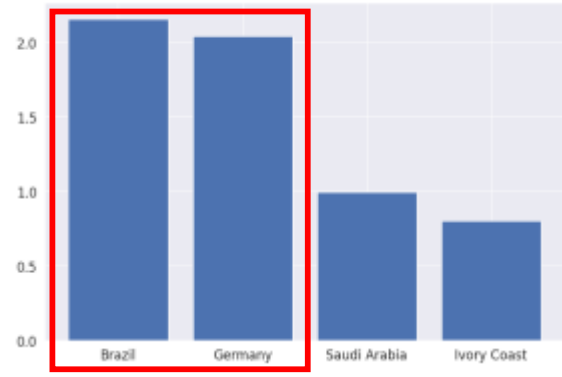
B group total prob



C group total prob

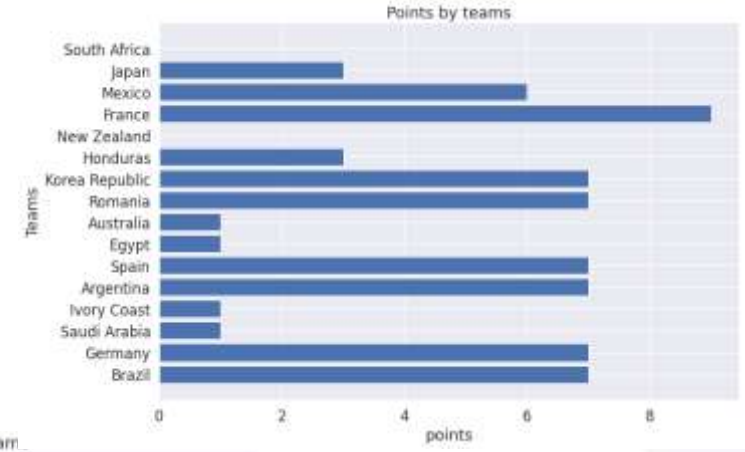


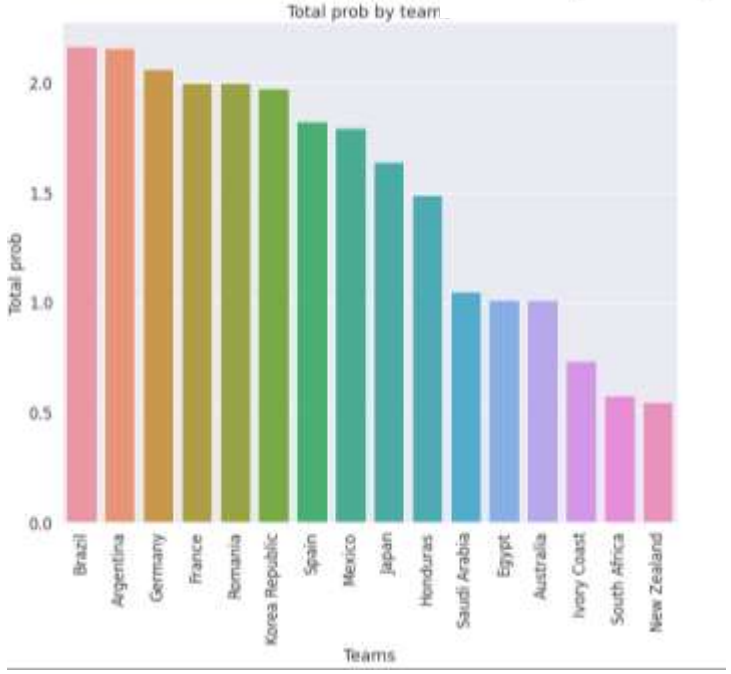
D group total prob



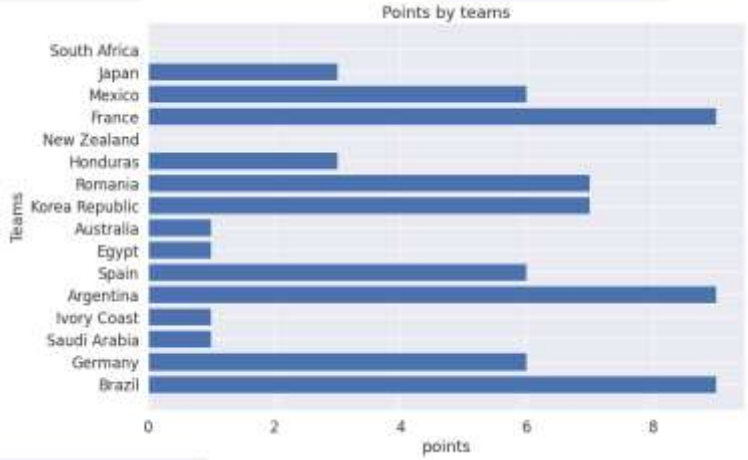
Visualization 국가별 total prob, points

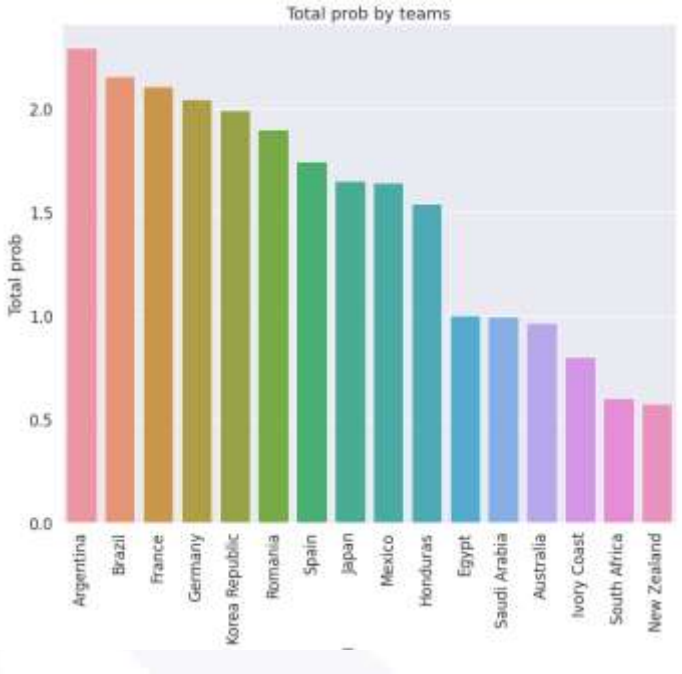
Pipeline





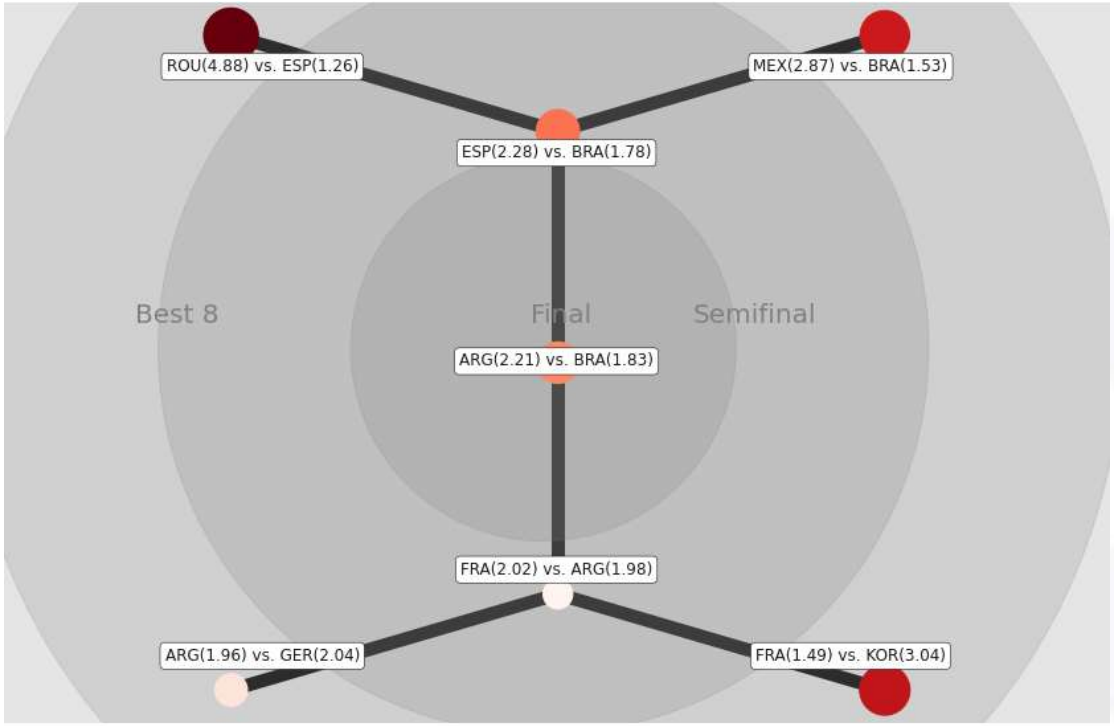
RandomForest



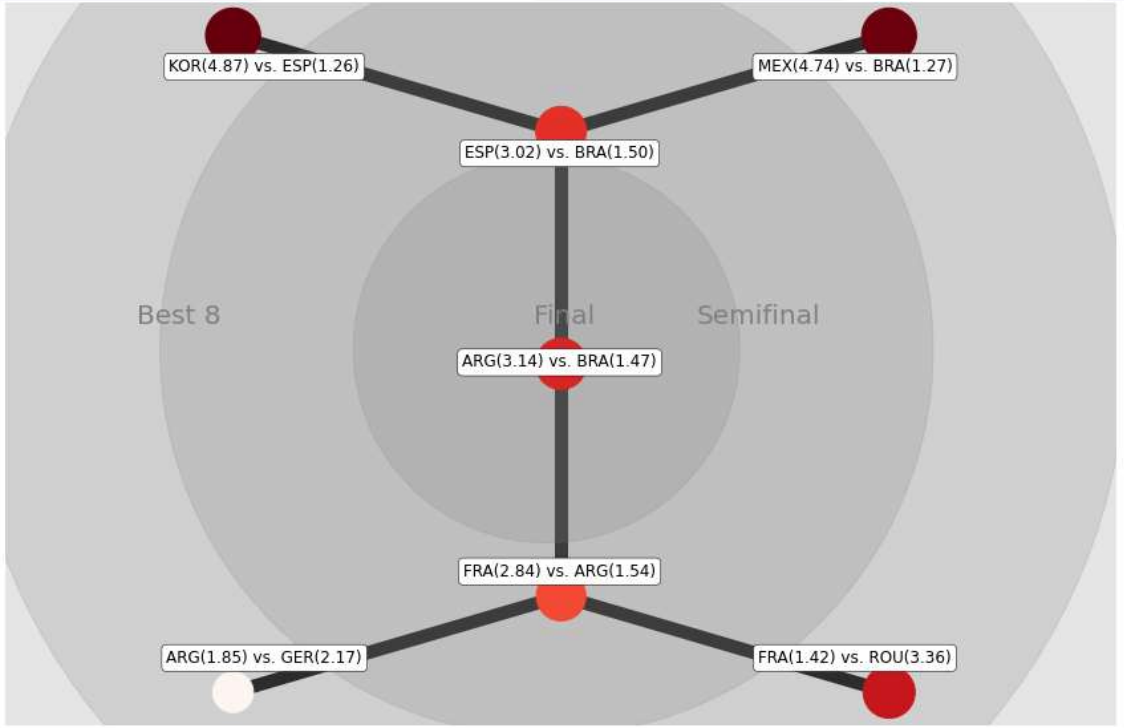


Visualization 8강, 준결승전, 결승전

Pipeline



RandomForest



Result

	Pipeline	RandomForest
1	 스페인	 브라질
2	 브라질	 스페인
3	 멕시코	 멕시코

마치며

결론

우승은
브라질 or 스페인

개선점

Dataset 부족

올림픽 경기에 대한
자세한 데이터셋 확보 미흡

FIFA 랭킹에 치우쳐진 성능

승자 대부분이
FIFA 랭킹 상위권 국가들

아쉬운 시각화

토너먼트 식의
그래프 구현 실패

감사합니다

4조 오공이문