NFL Draft Assistant

With the right kind of coaching and determination you can accomplish anything

MEMBER

Suphanat Thaiprasit	6310422034
Anuwat Malinee	6310422036
Kavin Singhakhet	6310422040

Agenda

01 Problem

02 Data Collection

03 Analysis

04 Result

PROBLEM

When you dream to play in the NFL but you are not convinced that you are ready to be the chosen one.

When you have confidence that you will be the chosen one But you still don't know who you should take as a role model.

If you are not selected. How do you know which skills you should improve?

DATA COLLECTION

1. Combine stat table

pro-football-reference.com/draft/2019-combine.htm

Draft Encyclopedia | Draft Finder | Combine Results | Yearly Totals | Draft Trade Value Chart | Back to top ▲

« 2018 Combine Results | 2020 Combine Results »

Find custom combine results using our NFL Combine Results tool

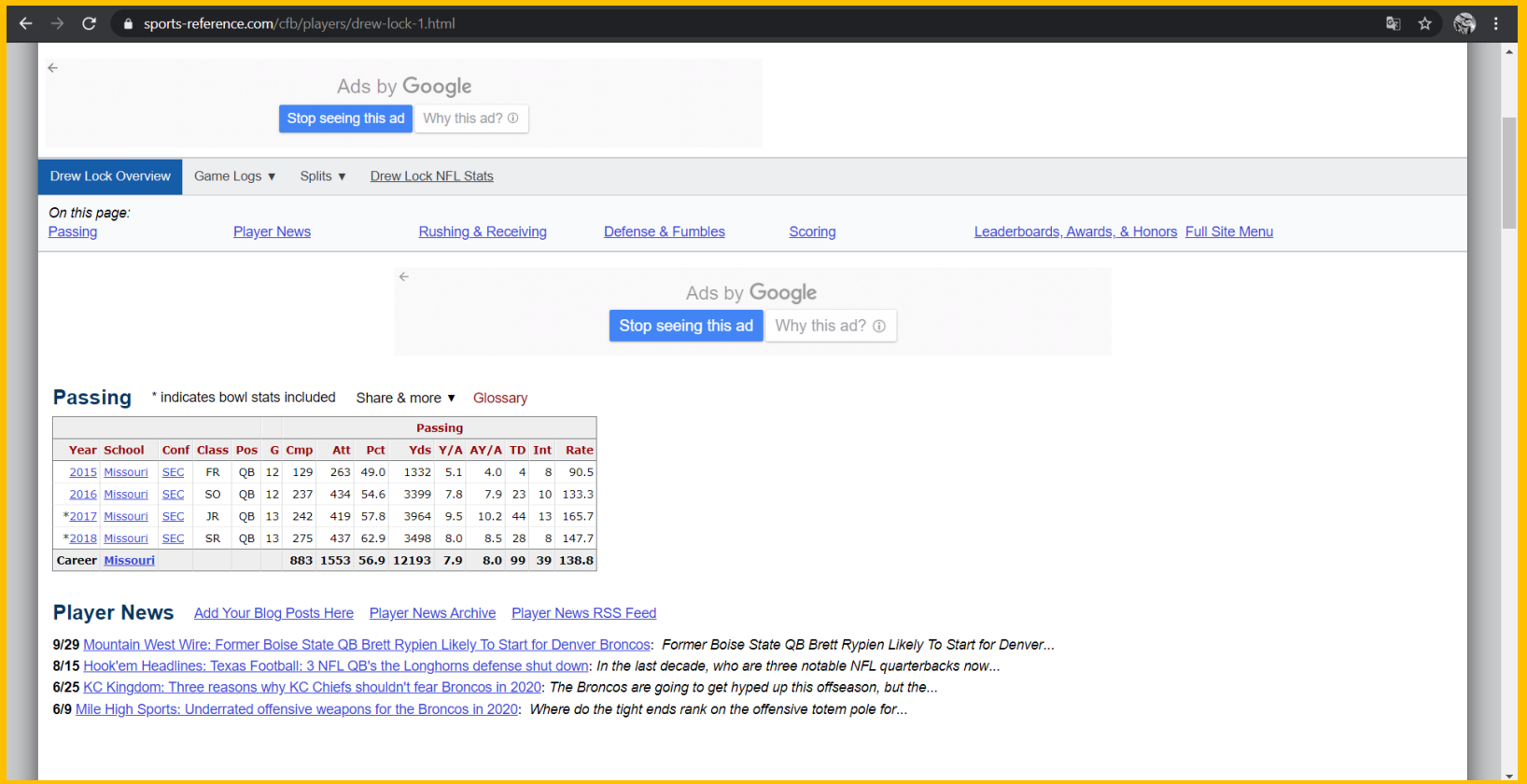
Combine Results Share & more ▾ Glossary

Player	Pos	School	College	Ht	Wt	40yd	Vertical	Bench	Broad Jump	3Cone	Shuttle	Drafted (tm/rnd/yr)
Johnathan Abram	S	Mississippi State	College Stats	5-11	205	4.45			116			Oakland Raiders / 1st / 27th pick / 2019
Paul Adams	OT	Missouri	College Stats	6-6	317	5.18	27.0	16	103	7.68	4.74	
Nasir Adderley	S	Delaware		6-0	206							
Azeez Al-Shaair	LB	Florida Atlantic	College Stats	6-1	234			16				
Otaro Alaka	LB	Texas A&M	College Stats	6-3	239	4.82	36.0	20	131			
Dakota Allen	LB	Texas Tech	College Stats	6-1	232	4.77	31.5	23	116	6.88	4.04	Los Angeles Rams / 7th / 251st pick / 2019
Josh Allen	EDGE	Kentucky	College Stats	6-5	262	4.63		28	118	7.15	4.23	Jacksonville Jaguars / 1st / 7th pick / 2019
Zach Allen	DL	Boston Col.	College Stats	6-4	281	5.00	32.0	24	112	7.34	4.36	Arizona Cardinals / 3rd / 65th pick / 2019
Bryson Allen-Williams	LB	South Carolina	College Stats	6-1	236	4.88		14		7.40		
Jeff Allison	LB	Fresno State	College Stats	5-11	228	4.82	31.0	12	115	7.20	4.45	
Ugo Amadi	S	Oregon	College Stats	5-9	199	4.51	32.5	18	115	7.21	4.19	
Rodney Anderson	RB	Oklahoma	College Stats	6-0	224			25				Cincinnati Bengals / 6th / 211th pick / 2019
J.J. Arcega-Whiteside	WR	Stanford	College Stats	6-2	225							Philadelphia Eagles / 2nd / 57th pick / 2019
Ryquell Armstead	RB	Temple	College Stats	5-11	220	4.45	30.0	22	114	7.02	4.29	Jacksonville Jaguars / 5th / 140th pick / 2019
Blessuan Austin	CB	Rutgers	College Stats	6-1	198			15				New York Jets / 6th / 196th pick / 2019
Jake Bailey	P	Stanford	College Stats	6-1	200	4.72	33.0		117			New England Patriots / 5th / 163rd pick / 2019
Zack Bailey	OL	South Carolina	College Stats	6-5	299		28.0	24	103			
Derrick Baity	CB	Kentucky	College Stats	6-2	197		38.5	10	127	7.06	4.31	
Deandre Baker	CB	Georgia	College Stats	5-11	193	4.52		14	118			New York Giants / 1st / 30th pick / 2019
Corey Ballentine	CB	Washburn		5-11	196	4.47	39.5	15	135	6.82	4.14	
Ben Banogu	EDGE	TCU	College Stats	6-3	250	4.62	40.0	23	134	7.02	4.27	Indianapolis Colts / 2nd / 49th pick / 2019
Alex Barnes	RB	Kansas State	College Stats	6-0	226	4.59	38.5	34	126	6.95	4.10	

https://www.pro-football-reference.com/draft/2019-combine.htm

DATA COLLECTION

2. College stat table



sports-reference.com/cfb/players/drew-lock-1.html

Ads by Google

Stop seeing this ad | Why this ad?

Drew Lock Overview | Game Logs ▾ | Splits ▾ | Drew Lock NFL Stats

On this page:
Passing | Player News | Rushing & Receiving | Defense & Fumbles | Scoring | Leaderboards, Awards, & Honors | Full Site Menu

Ads by Google

Stop seeing this ad | Why this ad?

Passing * indicates bowl stats included | Share & more ▾ | Glossary

						Passing								
Year	School	Conf	Class	Pos	G	Cmp	Att	Pct	Yds	Y/A	AY/A	TD	Int	Rate
2015	Missouri	SEC	FR	QB	12	129	263	49.0	1332	5.1	4.0	4	8	90.5
2016	Missouri	SEC	SO	QB	12	237	434	54.6	3399	7.8	7.9	23	10	133.3
*2017	Missouri	SEC	JR	QB	13	242	419	57.8	3964	9.5	10.2	44	13	165.7
*2018	Missouri	SEC	SR	QB	13	275	437	62.9	3498	8.0	8.5	28	8	147.7
Career	**Missouri**					**883**	**1553**	**56.9**	**12193**	**7.9**	**8.0**	**99**	**39**	**138.8**

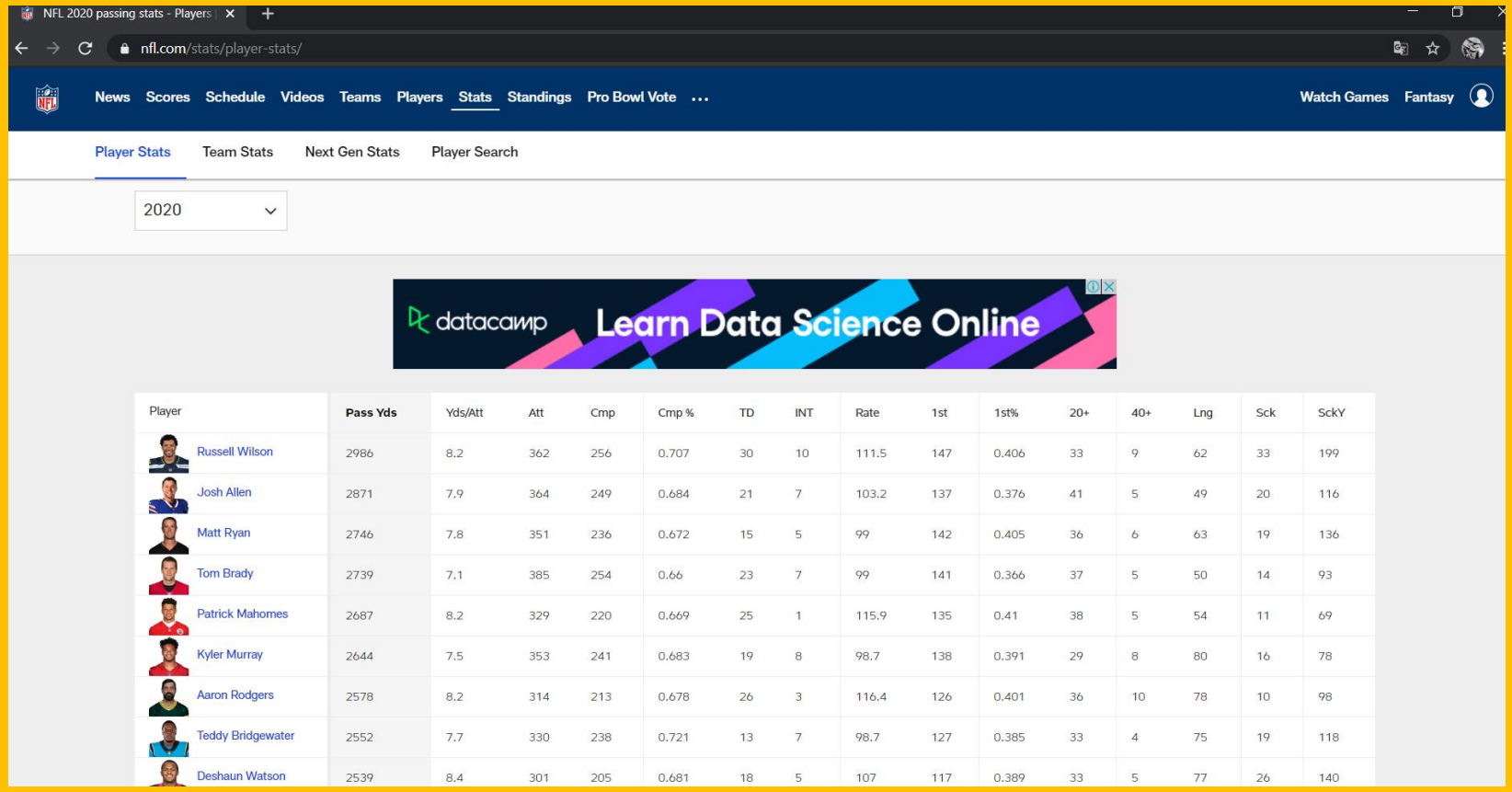
Player News Add Your Blog Posts Here | Player News Archive | Player News RSS Feed

9/29 Mountain West Wire: Former Boise State QB Brett Rypien Likely To Start for Denver Broncos: *Former Boise State QB Brett Rypien Likely To Start for Denver...*
8/15 Hook'em Headlines: Texas Football: 3 NFL QB's the Longhorns defense shut down: *In the last decade, who are three notable NFL quarterbacks now...*
6/25 KC Kingdom: Three reasons why KC Chiefs shouldn't fear Broncos in 2020: *The Broncos are going to get hyped up this offseason, but the...*
6/9 Mile High Sports: Underrated offensive weapons for the Broncos in 2020: *Where do the tight ends rank on the offensive totem pole for...*

https://www.sports-reference.com/cfb/players/drew-lock-1.html

DATA COLLECTION

3. Image data



Player	Pass Yds	Yds/Att	Att	Cmp	Cmp %	TD	INT	Rate	1st	1st%	20+	40+	Lng	Sck	SckY
Russell Wilson	2986	8.2	362	256	0.707	30	10	111.5	147	0.406	33	9	62	33	199
Josh Allen	2871	7.9	364	249	0.684	21	7	103.2	137	0.376	41	5	49	20	116
Matt Ryan	2746	7.8	351	236	0.672	15	5	99	142	0.405	36	6	63	19	136
Tom Brady	2739	7.1	385	254	0.66	23	7	99	141	0.366	37	5	50	14	93
Patrick Mahomes	2687	8.2	329	220	0.669	25	1	115.9	135	0.41	38	5	54	11	69
Kyler Murray	2644	7.5	353	241	0.683	19	8	98.7	138	0.391	29	8	80	16	78
Aaron Rodgers	2578	8.2	314	213	0.678	26	3	116.4	126	0.401	36	10	78	10	98
Teddy Bridgewater	2552	7.7	330	238	0.721	13	7	98.7	127	0.385	33	4	75	19	118
Deshaun Watson	2539	8.4	301	205	0.681	18	5	107	117	0.389	33	5	77	26	140

https://www.nfl.com/

ANALYSIS

1. Classification Model

Web Scraping Data

Clean Data

Join Table

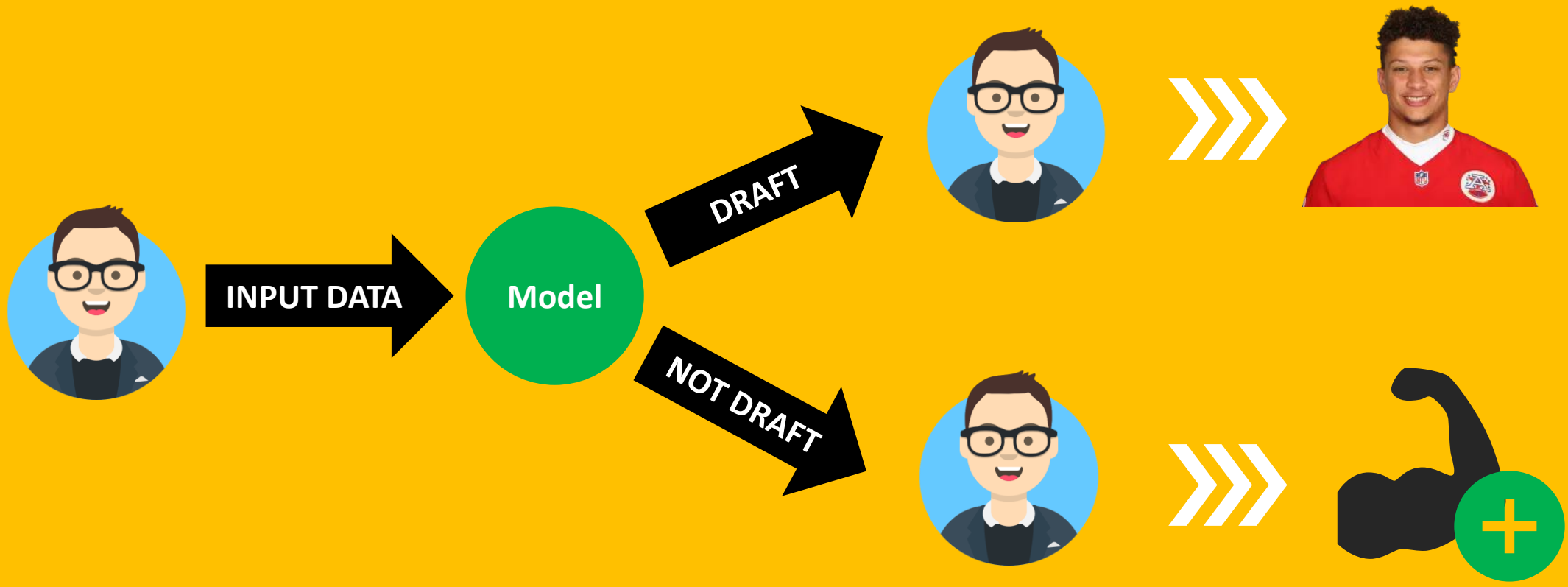
Prepare data for model

XGBoost Classifier

Model

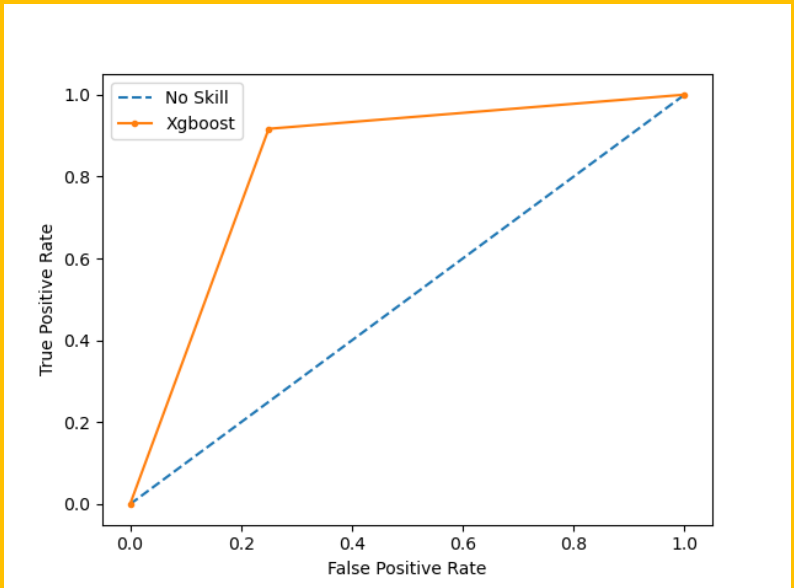
ANALYSIS

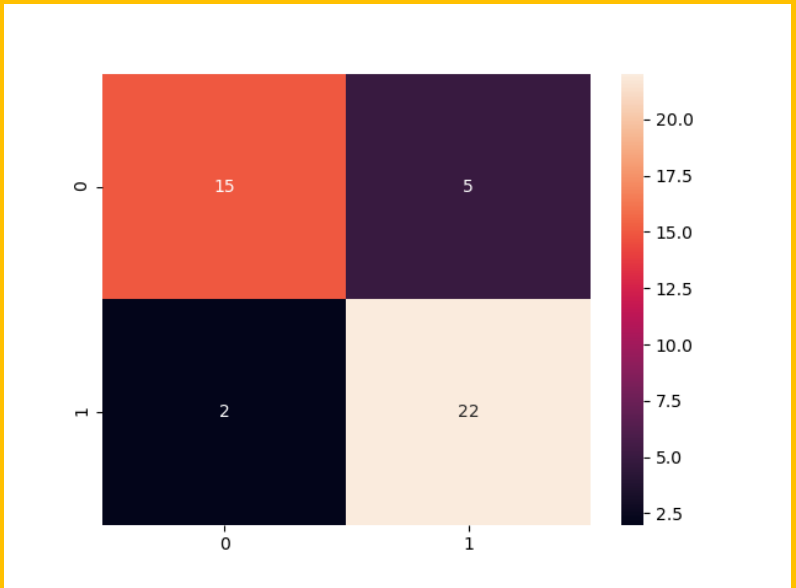
2. Similarity Model & Recommendation



RESULT

1. Classification Model

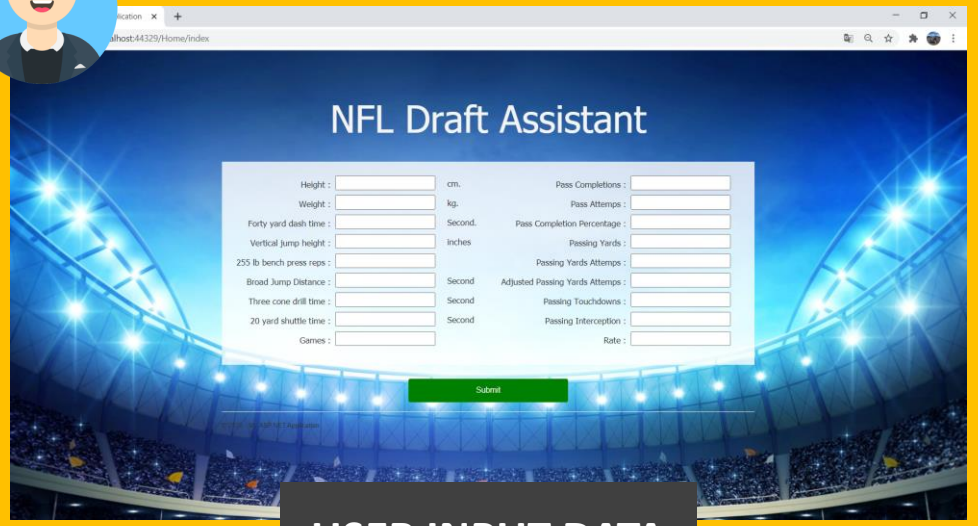




F-score = 0.863
No Skill: ROC AUC=0.500
Xgboost: ROC AUC=0.833

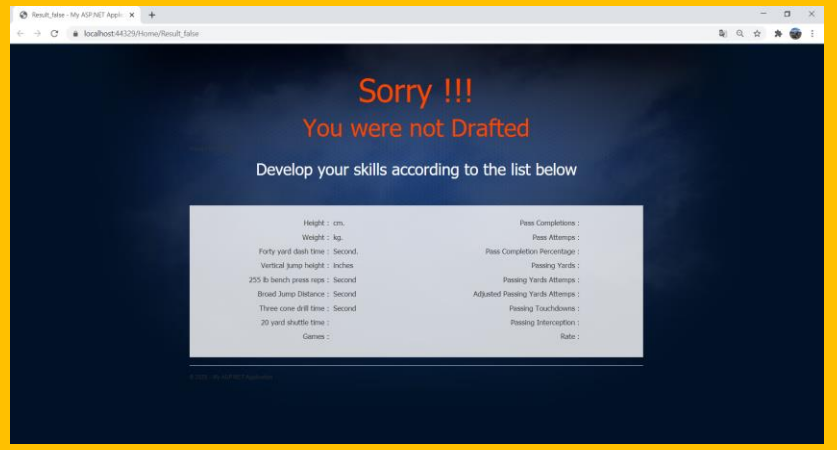
RESULT

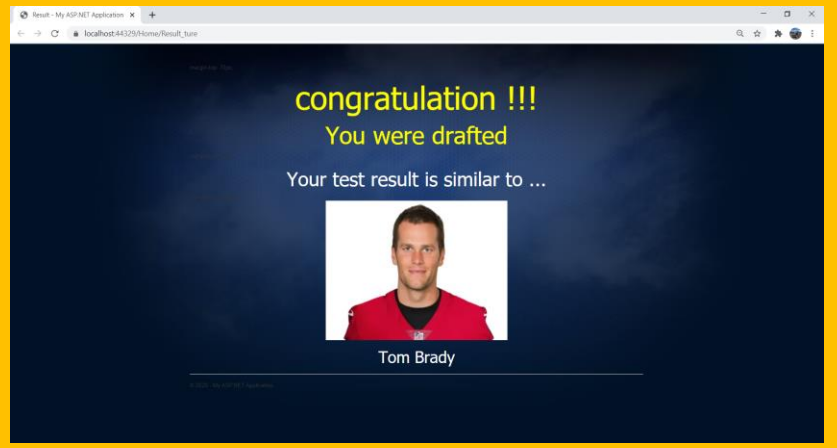
2. Similarity Model & Recommendation



USER INPUT DATA

RESULT





APPENDIX

Combine Table Scraping (I)

```
#Combine Scraping Data to CSV
import pandas as pd

s = 'https://www.pro-football-reference.com/draft/{}-combine.htm'
table = {}
df_combine = {}

for year in range(2000,2020):
    table['table{}'.format(year)] = pd.read_html(s.format(year))
    df_combine['combine{}'.format(year)] = table['table{}'.format(year)][0]

lst = [df_combine['combine{}'.format(year)] for year in range(2000,2020)]
combine1 = pd.concat(lst,ignore_index=True)

url = 'https://www.pro-football-reference.com/draft/2020-combine.htm'
table1 = pd.read_html(url)
combine2 = table1[0]
```

Combine Table Scraping (II)

```
18
19  def format_df(combine):
20      #Drop row header and Colledge column
21      combine.drop(combine[combine['Player'] == 'Player'].index, inplace = True)
22
23      #Change Ht from foot-inche to cm.
24      height_cm = []
25      for k in combine['Ht']:
26          a = k.split('-')
27          height_cm.append((int(a[0])*30)+(int(a[1])*2.5))
28
29      combine.insert(loc=3, column='Ht_cm', value=height_cm)
30      combine.drop(columns = 'Ht', inplace = True)
31
32      #Change Wt from Ib to Kg.
33      weight_kg = []
34      for b in combine['Wt']:
35          weight_kg.append(int(b)*0.45)
36
37      combine.insert(loc=4, column='Wt_kg', value=weight_kg)
38      combine.drop(columns = 'Wt', inplace = True)
39
40      return combine
41
42  #for train data
43  combine_train = format_df(combine1)
44  combine_train.to_csv(r"D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001/combine 2000 - 2019.csv", index = False)
45
46  #for test data
47  combine_test = format_df(combine2)
48  combine_test.to_csv(r"D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001/combine 2020.csv", index = False)
```

Combine Table Scraping (DATA)

AutoSave Off | combine 2000 - 2019 | Search | Kavin Sin

File Home Insert Page Layout Formulas Data Review View Help

Paste Cut Copy Format Painter | Calibri 11 | Wrap Text | Merge & Center | General | Conditional Formatting | Format as Table | Cell Styles | Insert Delete Format | AutoSum Fill Clear

Clipboard Font Alignment Number Styles Cells Editing

O4

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Player	Pos	School	Ht_cm	Wt_kg	College	40yd	Vertical	Bench	Broad Jump	3Cone	Shuttle	Drafted (tm/rnd/yr)
2	John Abraham	OLB	South Carolina	190	113.4		4.55						New York Jets / 1st / 13th pick / 2000
3	Shaun Alexander	RB	Alabama	180	98.1	College Stats	4.58						Seattle Seahawks / 1st / 19th pick / 2000
4	Darnell Alford	OT	Boston Col.	190	150.3		5.56	25	23	94	8.48	4.98	Kansas City Chiefs / 6th / 188th pick / 2000
5	Kyle Allamon	TE	Texas Tech	185	113.85		4.97	29		104	7.29	4.49	
6	Rashard Anderson	CB	Jackson State	185	92.7		4.55	34		123	7.18	4.15	Carolina Panthers / 1st / 23rd pick / 2000
7	Jake Arians	K	Ala-Birmingham	175	90.9								
8	LaVar Arrington	OLB	Penn State	187.5	112.5	College Stats	4.53						Washington Redskins / 1st / 2nd pick / 2000
9	Corey Atkins	OLB	South Carolina	180	106.65		4.72	31	21	112	7.96	4.39	
10	Kyle Atteberry	K	Baylor	180	75.15								
11	Reggie Austin	CB	Wake Forest	172.5	78.75		4.44	35	17	119	7.03	4.14	Chicago Bears / 4th / 125th pick / 2000
12	John Baker	P	North Texas	187.5	102.15	College Stats							
13	Mark Baniewicz	OT	Syracuse	195	140.4		5.34	28	20	96	7.72	4.73	Jacksonville Jaguars / 7th / 247th pick / 2000
14	Rashidi Barnes	S	Colorado	180	93.6		4.62	35	10	114	6.92	4.32	Cleveland Browns / 7th / 225th pick / 2000
15	David Barrett	CB	Arkansas	175	89.55		4.44	37.5	16	116	6.81	4.04	Arizona Cardinals / 4th / 102nd pick / 2000
16	William Bartee	CB	Oklahoma	182.5	86.4		4.43	38.5		124	6.7	3.99	Kansas City Chiefs / 2nd / 54th pick / 2000
17	Andrew Bayes	P	East Carolina	185	90								
18	Terrance Beadles	OG	Ark-Pine Bluff	187.5	140.4		5.19		29				
19	Robert Bean	CB	Mississippi State	177.5	80.1		4.5	34.5		122	6.87	4.2	Cincinnati Bengals / 5th / 133rd pick / 2000
20	Anthony Becht	TE	West Virginia	195	121.5	College Stats	4.78	33.5		123	6.94	4.08	New York Jets / 1st / 27th pick / 2000
21	Matt Beck	ILB	California	187.5	105.3		4.65						
22	Rogers Beckett	S	Marshall	182.5	93.15		4.62	39.5	15	119	6.48	4.29	San Diego Chargers / 2nd / 43rd pick / 2000
23	Brad Bedell	OG	Colorado	190	135.9		5.07	31.5	17	103	7.76	4.58	Cleveland Browns / 6th / 206th pick / 2000
24	Marcus Bell	ILB	Arizona	185	106.65		4.78	31.5	21	111	7.17	4.33	Seattle Seahawks / 4th / 116th pick / 2000
25	Gary Berry	S	Ohio State	177.5	91.8		4.55						Green Bay Packers / 4th / 126th pick / 2000
26	Michael Boireau	DE	Miami (FL)	190	123.3		5.09	29	26	105	7.68	4.49	Minnesota Vikings / 2nd / 56th pick / 2000
27	Matt Bowen	S	Iowa	182.5	90.45		4.49	33	12	115	7.06	4.09	St. Louis Rams / 6th / 198th pick / 2000
28	Carl Bradley	DT	Virginia Tech	185	135		5.03						
29	Tom Brady	QB	Michigan	190	94.95	College Stats	5.28	24.5		99	7.2	4.38	New England Patriots / 6th / 199th pick / 2000

combine 2000 - 2019

College Table Scraping (I)

```python
#sort QB Data from Combine 2010 - 2019
import pandas as pd
import numpy as np
combine = pd.read_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\combine 2000 - 2019.csv',encoding= 'UTF-8')
combine2020 = pd.read_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\combine 2020.csv',encoding= 'UTF-8')

# Code for scraping colledge data QB
def scrap_college(combine):
  qb = combine["Pos"].isin(["QB"])
  position_QB = combine[qb]
  n = [i.lower().replace('.', '').replace("'",'').replace(' ', '-') for i in position_QB.Player] #clean name for web scraping
  x = {}
  college = {}
  cl = 'https://www.sports-reference.com/cfb/players/{}-{}.html'

  for i in n:
    for c in range(1,3):
      try:
        x['{}-{}'.format(i,c)] = pd.read_html(cl.format(i,c))
        college['{}-{}'.format(i,c)] = x['{}-{}'.format(i,c)][0]
      except:
        break

  keys = list(college.keys())
  for a in keys:
    df = college[a]
    position = df['Unnamed: 4_level_0']
    if position['Pos'].iloc[1] != 'QB':
      college.pop(a)

  keys = list(college.keys())
  for z in keys:
    if len(college[z].columns) == 15:
      college[z].columns = ['Year','School','Conf','Class','Pos','G','Cmp','Att','Pct','Yds','Y/A','AY/A','TD','Int','Rate']
    else:
      college.pop(z)
```

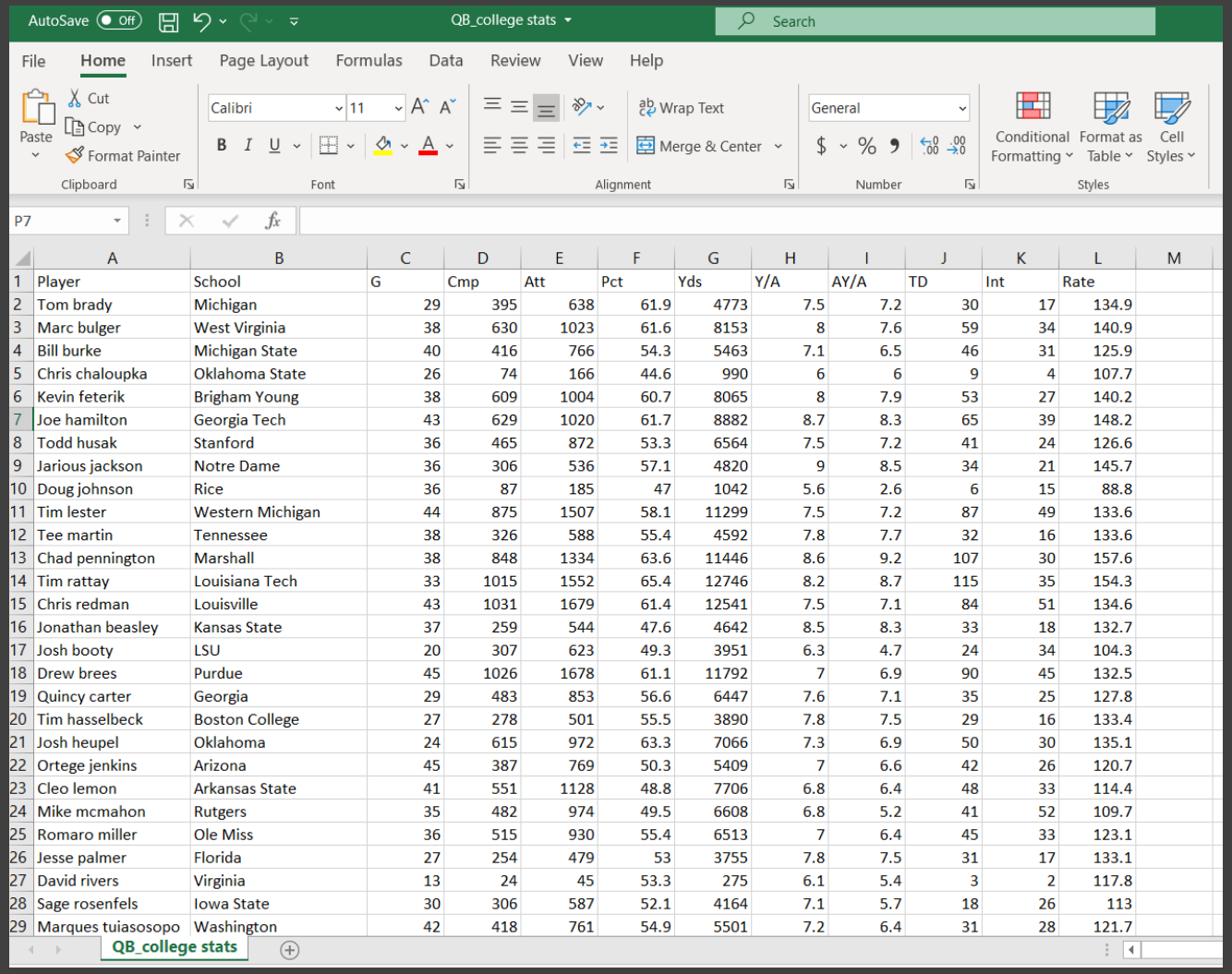
College Table Scraping (II)

```
    keys = list(college.keys())
    for z in keys:
        if len(college[z].columns) == 15:
            college[z].columns = ['Year','School','Conf','Class','Pos','G','Cmp','Att','Pct','Yds','Y/A','AY/A','TD','Int','Rate']
        else:
            college.pop(z)

    for qb1 in college.keys():
        df1 = college[qb1]
        g = df1['G'].sum()
        df2 = df1['Year']
        new_df = df1[df2 == 'Career']
        name = qb1[:-1].replace('-', ' ').rstrip()
        new_df.insert(loc = 0, column = 'Player', value = name.capitalize() )
        new_df.replace({'G':np.nan},g,inplace = True)
        college[qb1] = new_df

    return college

# concad DataFrame QB_college stats
def concat(college):
    lst = [college[qb_data] for qb_data in college.keys()]
    college_stats = pd.concat(lst,ignore_index=True)
    college_stats.drop(columns = ['Year','Conf','Class','Pos'], inplace = True)

    return college_stats

```

College Table Scraping (III)

```
59  #train data
60  train = scrap_college(combine)
61  train_df = concat(train)
62  train_df.to_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\QB_college stats.csv', index = False)
63  qb1 = combine["Pos"].isin(["QB"])
64  position_QB1 = combine[qb1]
65  position_QB1.to_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\NameQB.csv', index = False)
66
67  #test data
68  test = scrap_college(combine2020)
69  test_df = concat(test)
70  test_df.to_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\QB_college stats(test).csv', index = False)
71  qb2 = combine2020["Pos"].isin(["QB"])
72  position_QB2 = combine2020[qb2]
73  position_QB2.to_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\NameQB(test).csv', index = False)
```

College Table Scraping (DATA)



AutoSave Off | QB_college stats | Search

File Home Insert Page Layout Formulas Data Review View Help

Cut Copy Format Painter | Paste | Clipboard | Calibri 11 | Font | Wrap Text | Merge & Center | Alignment | General | Number | Conditional Formatting | Format as Table | Cell Styles | Styles

P7

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Player	School	G	Cmp	Att	Pct	Yds	Y/A	AY/A	TD	Int	Rate	
2	Tom brady	Michigan	29	395	638	61.9	4773	7.5	7.2	30	17	134.9	
3	Marc bulger	West Virginia	38	630	1023	61.6	8153	8	7.6	59	34	140.9	
4	Bill burke	Michigan State	40	416	766	54.3	5463	7.1	6.5	46	31	125.9	
5	Chris chaloupka	Oklahoma State	26	74	166	44.6	990	6	6	9	4	107.7	
6	Kevin feterik	Brigham Young	38	609	1004	60.7	8065	8	7.9	53	27	140.2	
7	Joe hamilton	Georgia Tech	43	629	1020	61.7	8882	8.7	8.3	65	39	148.2	
8	Todd husak	Stanford	36	465	872	53.3	6564	7.5	7.2	41	24	126.6	
9	Jarious jackson	Notre Dame	36	306	536	57.1	4820	9	8.5	34	21	145.7	
10	Doug johnson	Rice	36	87	185	47	1042	5.6	2.6	6	15	88.8	
11	Tim lester	Western Michigan	44	875	1507	58.1	11299	7.5	7.2	87	49	133.6	
12	Tee martin	Tennessee	38	326	588	55.4	4592	7.8	7.7	32	16	133.6	
13	Chad pennington	Marshall	38	848	1334	63.6	11446	8.6	9.2	107	30	157.6	
14	Tim rattay	Louisiana Tech	33	1015	1552	65.4	12746	8.2	8.7	115	35	154.3	
15	Chris redman	Louisville	43	1031	1679	61.4	12541	7.5	7.1	84	51	134.6	
16	Jonathan beasley	Kansas State	37	259	544	47.6	4642	8.5	8.3	33	18	132.7	
17	Josh booty	LSU	20	307	623	49.3	3951	6.3	4.7	24	34	104.3	
18	Drew brees	Purdue	45	1026	1678	61.1	11792	7	6.9	90	45	132.5	
19	Quincy carter	Georgia	29	483	853	56.6	6447	7.6	7.1	35	25	127.8	
20	Tim hasselbeck	Boston College	27	278	501	55.5	3890	7.8	7.5	29	16	133.4	
21	Josh heupel	Oklahoma	24	615	972	63.3	7066	7.3	6.9	50	30	135.1	
22	Ortege jenkins	Arizona	45	387	769	50.3	5409	7	6.6	42	26	120.7	
23	Cleo lemon	Arkansas State	41	551	1128	48.8	7706	6.8	6.4	48	33	114.4	
24	Mike mcmahon	Rutgers	35	482	974	49.5	6608	6.8	5.2	41	52	109.7	
25	Romaro miller	Ole Miss	36	515	930	55.4	6513	7	6.4	45	33	123.1	
26	Jesse palmer	Florida	27	254	479	53	3755	7.8	7.5	31	17	133.1	
27	David rivers	Virginia	13	24	45	53.3	275	6.1	5.4	3	2	117.8	
28	Sage rosenfels	Iowa State	30	306	587	52.1	4164	7.1	5.7	18	26	113	
29	Marques tuiasosopo	Washington	42	418	761	54.9	5501	7.2	6.4	31	28	121.7	

QB_college stats

Image Scraping (I)

In [1]:
```
import pandas as pd
import requests
from bs4 import BeautifulSoup, SoupStrainer
from PIL import Image
import io
```

In [2]:
```
nfl = pd.read_csv('NameQB.csv')
```

In [3]:
```
nn = []
name = nfl['Player']
for i in name:
    nn.append(i)
```

In [4]:
```
ww = {}
for i in nn:
    ww[i] = i.split()[1][:4]+i.split()[0][:2]
```

In [*]:
```
img = {}
number = ['00','01','02','03','04','05','06']
j = 0

web = 'https://www.pro-football-reference.com/players/xxx/{}{}.htm'

for i in ww:
    x = web.replace('xxx',ww[i][0])
    for c in number:
        if j == 0:
            try:
                e = x.format(ww[i],c)
                r = requests.get(e)
                s = BeautifulSoup(r.text, 'lxml')
                d = s.find('div',{'class':'media-item'})
                img[i] = [d]
            except:
                break
        else:
            j += 1
            if img.get(i) != [None]:
                pass
            else:
                try:
                    e = x.format(ww[i],c)
                    r = requests.get(e)
                    s = BeautifulSoup(r.text, 'lxml')
                    d = s.find('div',{'class':'media-item'})
                    img[i] = [d]
                except:
                    break
```

Clean/Join/Prepare Data(I)

```python
import pandas as pd
import numpy as np

#Load data from combine & college stats
combine_train = pd.read_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\NameQB.csv',encoding= 'UTF-8')
college_train = pd.read_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\QB_college stats.csv',encoding= 'UTF-8')
combine_test = pd.read_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\NameQB(test).csv',encoding= 'UTF-8')
college_test = pd.read_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\QB_college stats(test).csv',encoding= 'UTF-8')

#Clean combine data
def clean_df(combine):
        combine.drop(columns = ['Pos','School','College'],inplace = True)
        combine['y'] = combine['Drafted (tm/rnd/yr)'].fillna(0)
        combine['y'] = np.where(combine['y'] == 0,0,1)
        combine.drop(columns = ['Drafted (tm/rnd/yr)'],inplace = True)
        name = [i.lower().replace('.', '').replace("'",'').capitalize() for i in combine.Player]
        combine['Player'] = name

        return combine

#join data
def join_df(combine,college):
        qb_stats = combine.set_index('Player').join(college.set_index('Player'))
        qb_stats = qb_stats[['School','Ht_cm','Wt_kg','40yd','Vertical','Bench','Broad Jump','3Cone','Shuttle','G','Cmp', 'Att', 'Pct', 'Yds', 'Y/A',
                'AY/A', 'TD', 'Int', 'Rate','y']]
        qb_stats.drop(columns = ['School'],inplace = True)

        return qb_stats
```

Clean/Join/Prepare Data(II)

```
30  #Over Sampling
31  def overs(data):
32          max_size = data['y'].value_counts().max()
33          lst = [data]
34          for class_index, group in data.groupby('y'):
35                  lst.append(group.sample(max_size-len(group), replace=True))
36          frame_new = pd.concat(lst)
37          data = frame_new
38
39          return data
40
41  #train data
42  train = clean_df(combine_train)
43  train_df = join_df(train,college_train)
44  final_train = overs(train_df)
45  final_train.to_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\Train_data.csv',index=False)
46
47  #test data
48  test = clean_df(combine_test)
49  test_df = join_df(test,college_test)
50  final_test = overs(test_df)
51  final_test.to_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\Test_data.csv',index=False)
```

Clean/Join/Prepare Data(I)

AutoSave Off | Train_data | Search

File Home Insert Page Layout Formulas Data Review View Help

Paste Cut Copy Format Painter | Calibri 11 | Wrap Text | Merge & Center | General | Conditional Formatting | Format as Table | Cell Styles | Insert Delete Format

Clipboard Font Alignment Number Styles Cells

A1 Ht_cm

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Ht_cm	Wt_kg	40yd	Vertical	Bench	Broad Jump	3Cone	Shuttle	G	Cmp	Att	Pct	Yds	Y/A	AY/A	TD	Int	Rate	y	
2	187.5	96.75	4.61	31.5		106	7.03	4.3	8	19	30	63.3	169	5.6	4.8	1	1	115	0	
3	180	93.15	4.84						52	921	1478	62.3	13166	8.9	9.3	121	41	158.6	1	
4	185	100.35	4.71	34.5		110	7.38		25	424	665	63.8	5469	8.2	8.6	43	13	150.3	1	
5	182.5	99.45	4.78	32		108	7.14	4.33	45	603	1027	58.7	7548	7.3	7.1	52	29	131.5	0	
6	187.5	98.1	4.71	34.5	22	128	6.93	4.08	18	98	192	51	1215	6.3	7.6	14	1	127.2	1	
7	187.5	99	4.94	28		99	7.18	4.34	53	686	1026	66.9	9019	8.8	9.6	77	15	162.5	1	
8	185	94.95	4.96	28		112	7.22	4.19	46	848	1451	58.4	10913	7.5	7.2	76	43	133	1	
9	190	97.65	4.71	32		113	6.82	3.96											1	
10	190	103.05	4.85						44	791	1278	61.9	9360	7.3	7.7	79	25	139.9	1	
11	190	105.3	4.59	36		124	6.8	4.28	38	713	1064	67	9430	8.9	9.5	82	22	162.8	1	
12	195	104.85	4.86						48	777	1416	54.9	10617	7.5	7.6	85	36	132.6	1	
13	185	96.75	4.83	29.5		106	6.93	4.27	50	812	1317	61.7	10314	7.8	7.9	71	30	140.7	1	
14	180	103.95	5.03	26.5		100	7.47	4.62	46	540	923	58.5	5789	6.3	5.9	38	24	119.6	0	
15	190	103.95	5.07	26.5	16	108	7.43	4.6	37	460	821	56	5275	6.4	6.1	31	19	117.8	0	
16	180	94.5	4.81	29.5		112	7.18	4.48	27	359	607	59.1	5045	8.3	7.9	36	21	141.6	0	
17	185	98.55	4.69	31		109	6.73	4.11	46	933	1527	61.1	10892	7.1	7.4	83	27	135.4	0	
18	182.5	96.75	4.84	29		111	7	4.28	48	1026	1497	68.5	14607	9.8	10.6	131	30	175.4	1	
19	190	101.7	4.57	32.5		114	6.94	4.26											0	
20	185	99.9	4.72	33		108	7.41	4.34											0	
21	192.5	108.45	4.75						38	854	1304	65.5	10829	8.3	8.4	84	34	151.3	1	
22	187.5	100.35	4.68	24.5	18	109			35	483	803	60.1	5616	7	6.2	31	28	124.7	0	
23	190	92.7	5.03	28.5		107	7.46	4.42	40	416	766	54.3	5463	7.1	6.5	46	31	125.9	0	
24	187.5	104.85	4.94			109	7.07	4.38											1	
25	180	94.5	4.9						31	529	822	64.4	6378	7.8	7.8	59	25	147.1	1	
26	190	105.3	4.61	33.5		120	6.84	4.26	31	568	933	60.9	6822	7.3	7.3	40	18	132.6	1	
27	192.5	104.4	4.93	32.5		115	7.08	4.21	37	585	891	65.7	7598	8.5	8.8	56	19	153.8	1	
28	177.5	98.1	4.57	30.5		115			37	275	430	64	3731	8.7	9	30	10	155.2	0	
29	177.5	90.45	4.67	33.5		112			23	211	362	58.3	3155	8.7	9.5	30	7	155	0	

Train_data

Train Classification Model (I)

```python
#load data
import pandas as pd
import numpy as np
qb_stats = pd.read_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\Train_data.csv',encoding= 'UTF-8')

#Model
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from matplotlib import pyplot

x = qb_stats.drop(columns = ['y'])
y = qb_stats['y']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=22)
ns_probs = [0 for _ in range(len(y_test))]
clf = xgb.XGBClassifier(missing=np.nan,n_estimators=1000, learning_rate=0.2)
clf.fit(X_train, y_train)
lr_probs = clf.predict(X_test)
ns_auc = roc_auc_score(y_test, ns_probs)
lr_auc = roc_auc_score(y_test, lr_probs)

#Confusion Metrix
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
import seaborn as sns

cfm = confusion_matrix(y_test, lr_probs)
fscore = f1_score(y_test, lr_probs)
print('F-score = %.3f'% fscore)
sns.heatmap(cfm, annot=True)
pyplot.show()
```

Train Classification Model (II)

```
34  #plot AUC
35  print('No Skill: ROC AUC=%.3f' % (ns_auc))
36  print('Xgboost: ROC AUC=%.3f' % (lr_auc))
37  ns_fpr, ns_tpr, _ = roc_curve(y_test, ns_probs)
38  lr_fpr, lr_tpr, _ = roc_curve(y_test, lr_probs)
39
40  pyplot.plot(ns_fpr, ns_tpr, linestyle='--', label='No Skill')
41  pyplot.plot(lr_fpr, lr_tpr, marker='.', label='Xgboost')
42  pyplot.xlabel('False Positive Rate')
43  pyplot.ylabel('True Positive Rate')
44  pyplot.legend()
45  pyplot.show()
46
47  clf.save_model('nfl.model')
```

Similarity Model

```python
#load data
import pandas as pd
import numpy as np
qb_stats = pd.read_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\Train_data.csv',encoding= 'UTF-8')

#Similarity
from scipy.spatial.distance import euclidean,pdist,squareform
user_input = {'userinput from website'}
qb_drafted = qb_stats[qb_stats['y']==1].fillna(0)
euclidean_max = []

for i in range(0,len(qb_drafted.index)):
    euclidean_max.append(euclidean(user_input,i))

max(euclidean_max)
```

Recommendation Model

```
#load data
import pandas as pd
import numpy as np
qb_stats = pd.read_csv(r'D:\NIDA\Intro BADS\Project BADS6001\Project-NFL-BADS6001\Train_data.csv',encoding= 'UTF-8')

#Recomendation
user_input = {'userinput from website'}
qb_drafted = qb_stats[qb_stats['y']==1]

mean_stat = {}
for i in qb_drafted.columns:
    x = qb_drafted[[i]].mean(axis=1)
    qb_drafted[i] = x

for a in mean_stat.keys:
    if a in ['40yd','3Cone'.'Shuttle']:
        if user_input[a] > mean_stat[a]:
            print('Your {} status is too high'.format(a))
        if user_input[a] < mean_stat[a]:
            print('Your {} status is too low'.format(a))

```

HTML for USER

```
@{
    ViewBag.Title = "index";
}

<body style="background-image: url('../Image/79eae60f1e4b247c88219f8caae89a42.jpg');
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-size: cover;">

    <div style="text-align:center"><H1 style="color:whitesmoke;font-size:80px;margin-top:50px;font-family: indie-flower, Tahoma;">NFL Draft Assistant</H1></div>
    @using (Html.BeginForm("ResultComput", "Home", FormMethod.Post, new { enctype = "multipart/form-data" }))
    {
            <div class="grid-container">
        <div margin - top: 50px;
            class="item1">
            <p class="textp">Height :</p>
            <p class="textp">Weight :</p>
            <p class="textp">Forty yard dash time :</p>
            <p class="textp">Vertical jump height :</p>
            <p class="textp">255 lb bench press reps :</p>
            <p class="textp">Broad Jump Distance :</p>
            <p class="textp">Three cone drill time :</p>
            <p class="textp">20 yard shuttle time :</p>
            <p class="textp">Games :</p>
        </div>
        <div class="item2">
            <p class="textl"><input type="number" name="ht" /> cm.</p>
            <p class="textl"><input type="number" name="Wt" /> kg.</p>
            <p class="textl"><input type="number" name="fortyyd" /> Second.</p>
            <p class="textl"><input type="number" name="Vertical" /> inches</p>
            <p class="textl"><input type="number" name="Bench" /></p>
            <p class="textl"><input type="number" name="BroadJump" /> Second</p>
            <p class="textl"><input type="number" name="threeCone" /> Second</p>
            <p class="textl"><input type="number" name="Shuttle" /> Second</p>
            <p class="textl"><input type="number" name="G" /> </p>
        </div>
        <div class="item3">
            <p class="textp">Pass Completions :</p>
            <p class="textp">Pass Attemps :</p>
            <p class="textp">Pass Completion Percentage :</p>s
            <p class="textp">Passing Yards :</p>
            <p class="textp">Passing Yards Attemps :</p>
            <p class="textp">Adjusted Passing Yards Attemps :</p>
            <p class="textp">Passing Touchdowns :</p>
            <p class="textp">Passing Interception :</p>
            <p class="textp">Rate :</p>
        </div>
        <div class="item4">
            <p class="textl"><input type="number" name="Cmp" /></p>
            <p class="textl"><input type="number" name="Att" /></p>
            <p class="textl"><input type="number" name="number" /> </p>
            <p class="textl"><input type="number" name="Yds" /></p>
            <p class="textl"><input type="number" name="YA" /></p>
            <p class="textl"><input type="number" name="AYA" /></p>
            <p class="textl"><input type="number" name="TD" /></p>
            <p class="textl"><input type="number" name="Int" /> </p>
            <p class="textl"><input type="number" name="Rate" /> </p>
        </div>
    </div>
    <div style="width:100%;text-align:center;">

        <button id="button" class="btn"
                type="Submit">
            Submit
        </button>

    </div>
    }
</body>
```