CS 383 Final Project

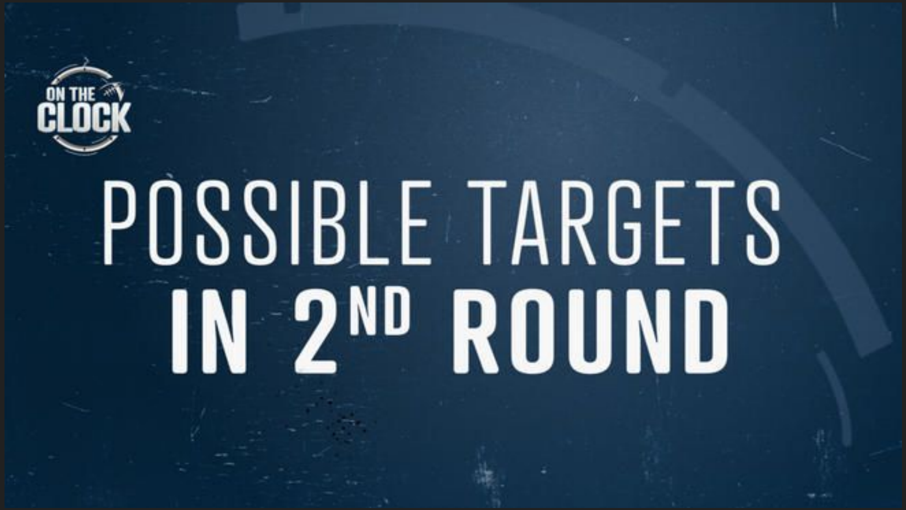
College QB Draft Predictions

Robert Roche

Purpose

Investing in Young Talent

What has changed?



The Dataset

kaggle

av8ramit

	DraftYear	Round	Pick	Age	GamesPlay	Completio	Attempts	Yards	Touchdow	Interceptic	RushAttem	RushYards	RushTouch	Player	College	Conferenc	Team	Heisman	Verdict
2	2015	1	1	21	27	562	851	7964	65	18	145	284	7	Jameis Wi	Florida St.	Atlantic Co	TAM	1	1
3	2015	3	75	24	37	688	1113	9190	64	27	212	283	4	Garrett Gr	Colorado S	Mountain	NOR	0	0
4	2015	3	89	23	47	1187	1838	13600	83	54	132	-804	2	Sean Mann	Oregon St.	Pac-12	STL	0	0
5	2015	4	103	24	37	530	845	8195	62	10	192	338	21	Bryce Pett	Baylor	Big 12	NYJ	0	0
6	2015	5	147	22	40	837	1241	9966	75	25	479	1747	30	Brett Hund	UCLA	Pac-12	GNB	0	0
7	2015	7	250	22	44	550	934	5931	27	24	142	-23	6	Trevor Sier	Northwest	Big Ten	DEN	0	1
8	2014	1	3	22	37	585	891	7598	56	19	195	561	15	Blake Bort	Central Flo	American A	JAX	0	1
9	2014	1	22	21	26	595	863	7820	63	22	345	2169	30	Johnny Ma	Texas A&M	Southeaste	CLE	1	0
10	2014	1	32	21	39	781	1142	9817	72	24	226	170	6	Teddy Brid	Louisville	Atlantic Co	MIN	0	1
11	2014	2	36	23	44	1087	1630	12843	113	24	164	190	5	Derek Carr	Fresno St.	Mountain	OAK	0	1
12	2014	2	62	22	45	1047	1668	13156	118	51	260	-67	8	Jimmy Gar	East. Illino	FCS	NWE	0	0
13	2014	4	120	23	47	693	1248	9003	52	39	495	1359	24	Logan Tho	Virginia Te	Atlantic Co	ARI	0	0
14	2014	4	135	24	31	430	757	5690	37	19	167	-319	4	Tom Savag	Pittsburgh	Atlantic Co	HOU	0	0
15	2014	5	164	24	53	686	1026	9019	77	15	119	-50	3	A.J. McCar	Alabama	Southeaste	CIN	0	0
16	2014	6	178	23	30	407	659	5783	35	15	83	-313	0	Zach Mett	LSU	Southeaste	TEN	0	0
17	2018	1	1	21	37	606	1055	8872	67	23	631	3974	49	Lamar Jack	Louisville	Atlantic Co	CLE	1	1
18	2018	1	2	21	30	711	1169	9301	59	26	109	-154	6	Josh Rosen	UCLA	Pac-12	NYG	0	1
19	2018	1	9	21	26	523	801	6873	57	21	129	340	7	Sam Darno	USC	Pac-12	CIN	0	1
20	2018	1	12	22	41	894	1415	13267	90	26	220	25	17	Mason Ru	Oklahoma	Big 12	WAS	0	1

	Player	Age	GamesPlay	Completio	Attempts	Yards	Touchdow	Interceptic	RushAtter	RushYards	RushTouch	Heisman	Round
2	EJ Manuel	23	43	600	897	7741	47	28	298	827	11	0	1
3	Geno Smit	22	44	988	1465	11662	98	21	245	342	4	0	2
4	Mike Glen	23	36	646	1069	7411	63	31	111	-281	3	0	3
5	Matt Barkl	23	47	1001	1562	12327	116	48	132	-113	6	0	4
6	Ryan Nass	23	47	791	1312	9190	70	28	242	168	5	0	4
7	Sean Renfr	23	42	898	1389	9465	51	40	153	-167	9	0	7
8	Andrew Lu	23	38	713	1064	9430	82	22	163	957	7	0	1
9	Robert Gri	22	41	800	1192	10366	78	17	528	2254	33	1	1
10	Ryan Tann	24	50	484	774	5450	42	21	115	369	5	0	1
11	Brandon W	28	30	767	1103	9260	75	27	39	-150	1	0	1
12	Brock Osw	21	25	412	680	5082	33	15	137	221	4	0	2
13	Russell Wil	23	50	907	1489	11720	109	30	441	1421	23	0	3
14	Nick Foles	23	36	938	1404	10068	67	33	106	-290	4	0	3
15	Kirk Cousir	24	45	723	1128	9131	66	30	111	-127	1	0	4
16	Ryan Lindl	23	49	961	1732	12690	90	47	90	-275	2	0	6
17	Cam Newt	22	20	191	292	2908	30	7	285	1586	24	1	1
18	Jake Locke	23	40	619	1147	7639	53	35	454	1939	29	0	1
19	Blaine Gab	21	31	568	933	6822	40	18	221	458	8	0	1
20	Christian P	23	35	596	965	6872	49	30	296	833	10	0	1
21	Andy Daltc	23	50	812	1317	10314	71	30	413	1611	22	0	2
22	Colin Kaep	23	51	740	1271	10098	82	24	600	4112	59	0	2
23	Ryan Malle	23	37	552	955	8385	69	24	135	-141	7	0	3
24	T.J. Yates	24	45	795	1277	9377	58	46	220	-333	7	0	5
25	Tyrod Tayl	22	50	495	865	7017	44	20	501	2196	23	0	6
26	Greg McEl	23	35	436	658	5691	39	10	114	71	2	0	7
27	Sam Bradf	22	31	604	893	8403	88	16	77	36	5	1	1
28	Tim Tebow	23	55	661	995	9285	88	16	692	2947	57	1	1
29	Jimmy Clau	23	35	695	1110	8148	60	27	175	-355	5	0	2

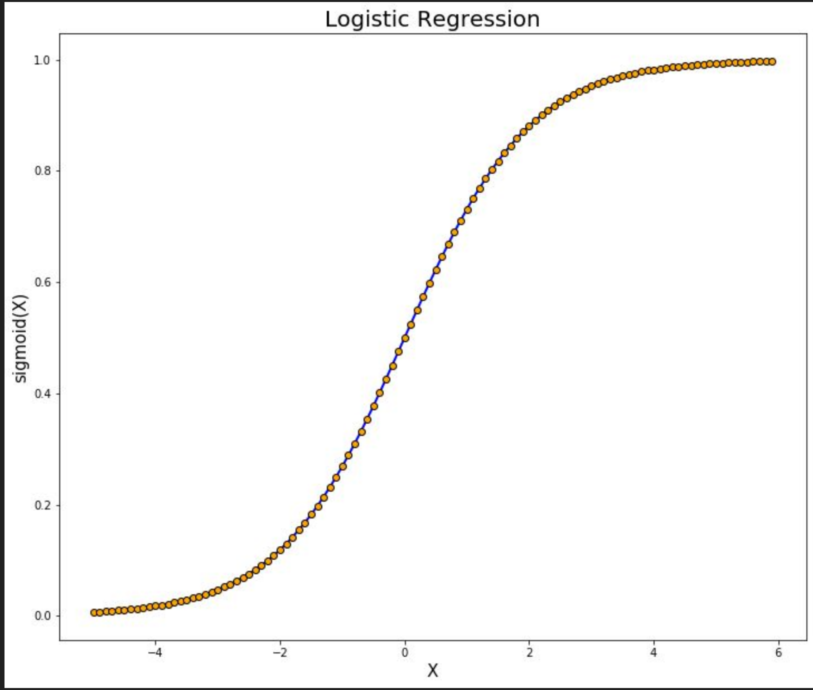
quarterback_training

Methodology

python™

Logistic Regression

7 models, 7 sets of thetas



Results

Upon Further Inspection

Code Review

st:8888/notebooks/Desktop/cs383_final_project/cs383_Final_Project.ipynb

jupyter cs383_Final_Project Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

Imports

In [465]:

```
import numpy as np
import csv
import sys
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from math import *
import pandas as pd
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
import random
```

Sigmoid and Prediction Methods

In [466]:

```
#sigmoid
def g(x):
    return (1/(1+np.exp(-x)))

def prediction(X,thetas):
    i = 0
    y_list = []
    while i < len(thetas):
        y_list.extend(g(X@thetas[i].T))
        i+=1
    return((y_list.index(max(y_list)))+1)
```

jupyter cs383_Final_Project Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

Retrieve and Setup Data

In [467]:

```
f = open("quarterback_total.csv")
length = len(f.readline().split(','))

#getting the data from our csv ignoring strings
data=np.loadtxt("quarterback_total.csv", delimiter=",", skiprows = 1, usecols=range(1,length))

#adding the names of each player to a list so we know can know who exactly we're ranking
labels = []
reader = csv.reader(f, delimiter=",")
for i in reader:
    labels.append(i[0])

f.close()

y = np.array(data[0:,11])
X = np.array(data[:,0:11])

y1 = []
y2 = []
y3 = []
y4 = []
y5 = []
y6 = []
y7 = []

#construct a 1 versus all list for each potential target value
for i in y:
    if i == 1:
        y1.append(1)
        y2.append(0)
        y3.append(0)
        y4.append(0)
```

jupyter cs383_Final_Project Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

Logistic Regression Learning

In [469]:

```
LogReg1 = LogisticRegression(random_state=0)
LogReg2 = LogisticRegression(random_state=0)
LogReg3 = LogisticRegression(random_state=0)
LogReg4 = LogisticRegression(random_state=0)
LogReg5 = LogisticRegression(random_state=0)
LogReg6 = LogisticRegression(random_state=0)
LogReg7 = LogisticRegression(random_state=0)
coeff1 = LogReg1.fit(X_train1,y_train1)
coeff2 = LogReg2.fit(X_train2,y_train2)
coeff3 = LogReg3.fit(X_train3,y_train3)
coeff4 = LogReg4.fit(X_train4,y_train4)
coeff5 = LogReg5.fit(X_train5,y_train5)
coeff6 = LogReg6.fit(X_train6,y_train6)
coeff7 = LogReg7.fit(X_train7,y_train7)

thetas_list = []

#condense all the thetas into a list
thetas_list.append(coeff1.coef_)
thetas_list.append(coeff2.coef_)
thetas_list.append(coeff3.coef_)
thetas_list.append(coeff4.coef_)
thetas_list.append(coeff5.coef_)
thetas_list.append(coeff6.coef_)
thetas_list.append(coeff7.coef_)
```

Making Predictions

In [470]:
```
pred = []
for x in X_test:
    pred.append(prediction(x,thetas_list))
```

In [471]:
```
i=0
count = 0
while i < len(pred):
    if( pred[i] == y_test[i]):
        count+=1
    i+=1
print("Accuracy of the models: " + str(count/len(y_test)))
```

```
Accuracy of the models: 0.4897959183673469
```

Individual Model Observation

In [472]:
```
print(metrics.accuracy_score(y_test1,coeff1.predict(X_test1)))
print(metrics.accuracy_score(y_test2,coeff2.predict(X_test2)))
print(metrics.accuracy_score(y_test3,coeff3.predict(X_test3)))
print(metrics.accuracy_score(y_test4,coeff4.predict(X_test4)))
print(metrics.accuracy_score(y_test5,coeff5.predict(X_test5)))
print(metrics.accuracy_score(y_test6,coeff6.predict(X_test6)))
print(metrics.accuracy_score(y_test7,coeff7.predict(X_test7)))
```

```
0.9183673469387755
0.8979591836734694
0.9183673469387755
0.8367346938775511
0.8571428571428571
0.8979591836734694
```

Conclusion