NFL Wide Receiver Touchdown Analysis

Clay Jackson, cjackson@bellarmine.edu

Abstract (Project Summary)

For this project, we will be analyzing data and statistics of NFL wide receivers. Football is the biggest sport in America and some of the biggest stars in the league are wide receivers. Receivers are the players who catch the passes the quarterback throws. They are the ones who score the majority of the touchdowns in the NFL. Due to the NFL's huge success, millions of people now play fantasy football. This is a game where you draft NFL players, and you earn points based off their performance (reception = 1pt, 10 yards= 1pt, touchdown= 6 points). As you can see, touchdowns are worth a lot of points in fantasy football! So, we want players on our fantasy team who will score touchdowns. However, touchdowns are very volatile and vary year to year. So, how do we know what receivers will have the most touchdowns? Is there a way we can predict a receivers touchdowns based on his underlying stats and metrics? Using exploratory analysis and modeling, I will attempt to do just that in this project.

I. Introduction

I retrieved my data from *FantasyPros*. *FantasyPros* is an online media company that specializes in sports data collection and aggregate expert evaluations on an array of fantasy sports (football, baseball, basketball, etc). The data set displays the season statistics of every NFL wide receiver who caught a pass in the last three years. Each row represents the stats of a specific player during a specific season (Ex. 2022, Tyreek Hill). It is comprised of 25 different categories of player tracking data (Ex. receptions, dropped passes, percent of team targets, yards after contact, games played, etc). Our goal is to use these variables to predict how many touchdowns a wide receiver should have scored. Touchdowns are a high variance statistic and cannot be predicted well just by looking back at how many touchdowns the player scored last year. However, it may be able to be predicted by the players underlying metrics. Once I create an accurate model, I will then use it to try and predict player touchdown performance for the 2024 NFL season. Thus, helping me decide who I want to draft on my fantasy team.

II. Background

So, why was this data collected in the first place? While the NFL is full of incredible athletes and jocks, the "nerds" have found a home in football over the past five years. NFL teams have been using advanced metrics to decide what players to draft and plays to run. With the growing popularity of fantasy football (and an increase in sports and fantasy betting), individual player statistics have been tracked by various platforms, like FantasyPros. This data is being tracked by fantasy experts to personally make money in their own fantasy leagues, and also to create content and become influencers. These stats and underlying metrics could be the key to determining who will be the most efficient players in the next fantasy football season. The experts want to have the edge on the people they are going against.

III. Exploratory Analysis: Variable Summary

My data contains 590 NFL wide receiver seasons. There are thirty columns with various data types. **Table 1**, lists each variable in the dataset along with the data type (nominal, ordinal, discrete, or continuous) and its data class in Python. Our dependent variable in this study is touchdowns. This variable's output is how many touchdowns the receiver had in the recorded NFL season. There were no missing values in the dataset. However, the dataset did not include the player's touchdowns scored. So, I had to manually enter them. I also created Player ID, to ensure each row was unique (due to there being repeat player names. Ex. Tyreek Hill has season stats for 2021, 2022, and 2023).

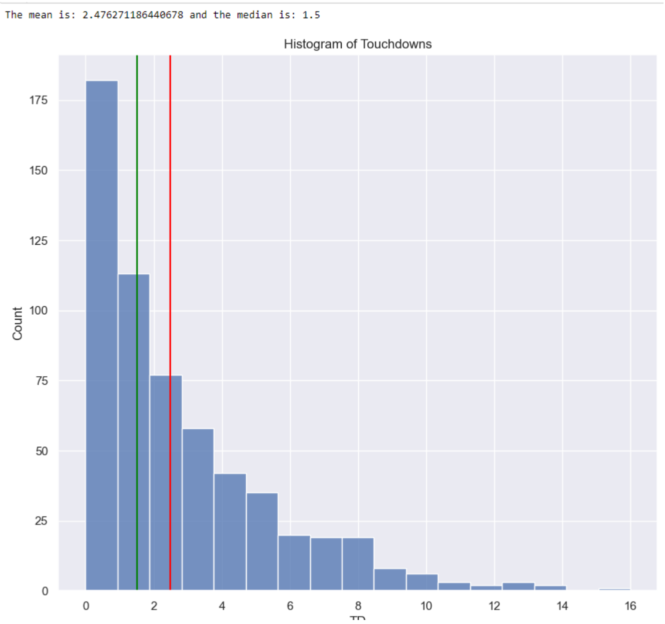
Table 1: Data Types and Python class

Variable Name	Data Type	Python Class
Year	discrete	integer
Player	nominal	object
Player ID	nominal	integer
Receptions (REC)	discrete	integer
Yards (YDS)	discrete	integer
Yards per Reception (Y/R)	continuous	float
Yards before catch (YBC)	discrete	integer
YBC per reception (YBC/R)	continuous	float
Air yards (AIR)	discrete	integer
AIR per reception (AIR/R)	continuous	float
Yards after catch (YAC)	discrete	integer
YAC per reception (YAC/R)	continuous	float
Yards after contact (YACON)	discrete	integer
YACON per reception (YACON/R)	continuous	float
Broken Tackles (BRKTKL)	discrete	integer
Targets (TGT)	discrete	integer
% of team targets (% TM)	continuous	float
Catchable passes (CATCHABLE)	discrete	integer
Dropped passes (DROP)	discrete	integer
Red zone targets (RZ TGT)	discrete	integer

10 + yard reception (10+ YDS)	discrete	integer
20 + yard reception (20+ YDS)	discrete	integer
30 + yard reception (30+ YDS)	discrete	integer
40 + yard reception (40+ YDS)	discrete	integer
50 + yard reception (50+ YDS)	discrete	integer
Longest reception (LNG)	discrete	integer
Touchdowns (TD)	discrete	integer

IV. Exploratory Analysis: Summary Statistics and Distribution

There are too many numerical variables in the dataset to display all of their descriptive statistics (see Jupiter notebook for these stats). However, there is one main point I want to discuss when it comes to the variable's summary statistics and distribution. While doing my analysis, I noticed that every single receiver metric had a larger mean than median. Recall that the median is not affected by outliers, but the mean is. This tells me that there are some positive outliers in the data. These positive outliers are representing the elite, star receivers in the NFL. The vast majority of all NFL receivers are quite average. They catch a couple dozen passes, gain a couple hundred yards, and score 0-2 touchdowns. However, there are a handful of star receivers that are catching over 100 passes, gaining over one thousand yards, and scoring ten touchdowns. These star receivers are positively skewing this data's distribution. Below are some figures that show this trend. Every single variable looks like the histograms/boxplots below (a large cluster near the low numbers, and then a positive skew to the right of the mean and median). Here are just a few (For more graphs, distributions, and statistics, refer to the Jupiter notebook):



```
The mean is: 2.476271186440678 and the median is: 1.5
```

Figure 1: Histogram of Touchdowns Scored (Above)

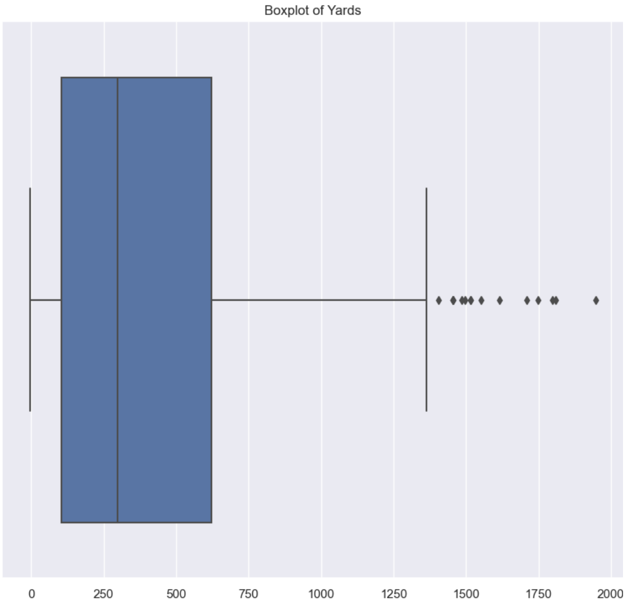


Figure 2: Boxplot of Yards Gained

```
The mean is: 33.52542372881356 and the median is: 24.0
```

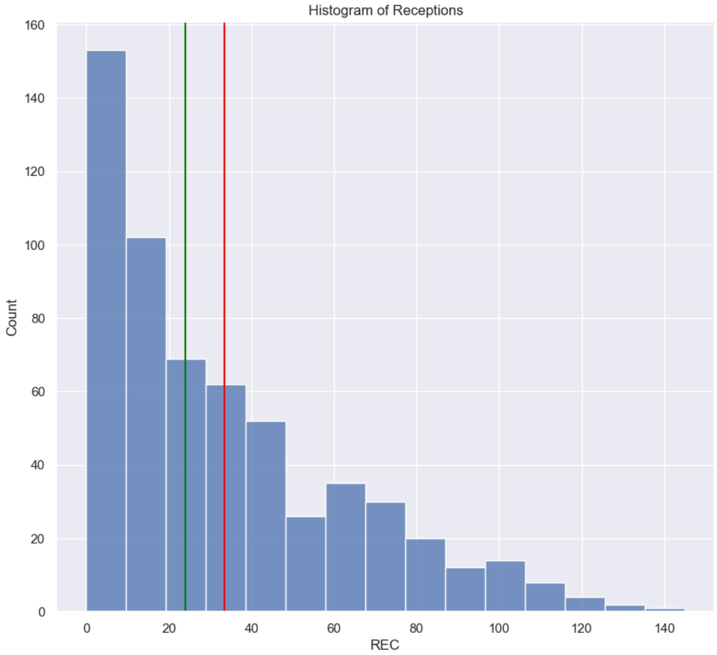


Figure 3: Histogram of Receptions

V. Exploratory Analysis: Correlation

My next step in my exploratory analysis was to find correlation between the independent variables and the dependent variable, touchdowns scored. First, I created a heatmap so I could quickly spot variables that had a strong positive correlation with touchdowns scored (See **Figure 4**). From there, I looked at each of these variables individually. I calculated their R-squared score and created a scatter plot, comparing them to touchdowns scored (See **Figure 5).** I found 14 variables that had moderate to very strong positive correlation with touchdowns. To be brief, I will only show you one scatter plot. To see all ten scatter plots I created, please refer to the Jupiter notebook. Next, I created another heat map. This one focused in on the correlation between 10 to 50 yard receptions and touchdowns (See **Figure 6**).

Several discoveries were made during this correlation analysis. First, I discovered what variables had the strongest correlation to touchdowns. These were: REC, YDS, YBC, YAC, TGT, % TM, CATCHABLE, BRKTKL, RZ TGT, YACON, DROP, 10+ YDS, 20+ YDS, 30+ YDS, and 40+ YDS. Second, I notice that the "per reception" variables have much less correlation to touchdowns scored than overall production (ex. yards has a strong correlation to TD's but yards per reception does not). This could be for a couple reasons. First, it appears that the probability of scoring touchdowns depends more on the volume a receiver gets than his productivity. Second, "per reception" stats could be skewed by small volume. For example, a bad or mediocre receiver could have 1 catch for 20 yards for the whole season. His "per reception" stats will be incredible! But that does not mean he is a good receiver. My last observation is that the correlation gets worse as the yardage goes up in **Figure 6**. This tells me that receivers who catch a lot of 10-30 yard passes may be more consistent. This is a better indicator of if they will score a touchdown. Whereas 40-50 yard plays are more random, and may not result in touchdowns.

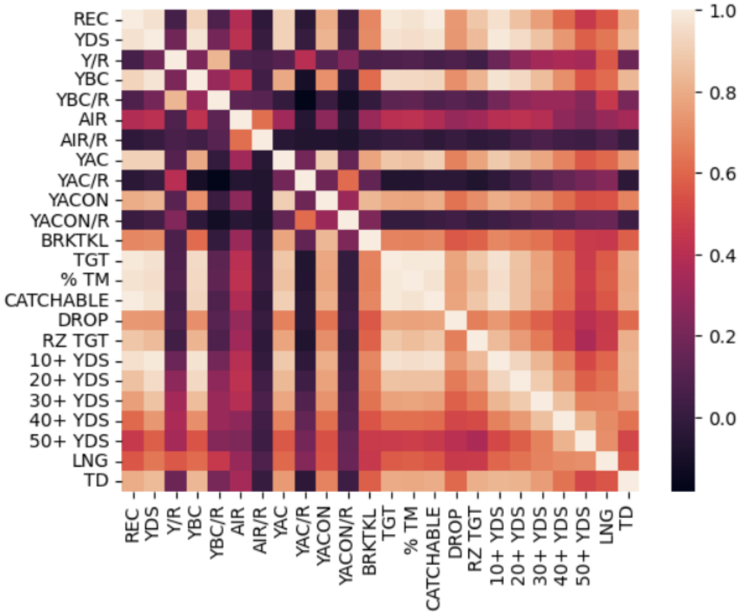
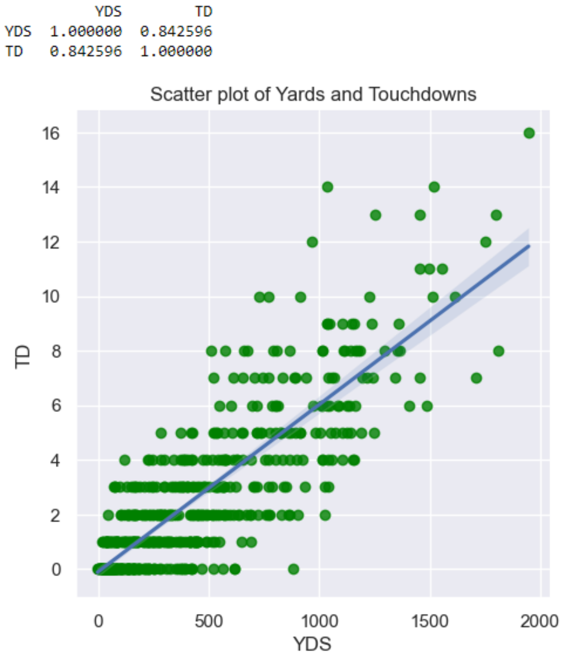
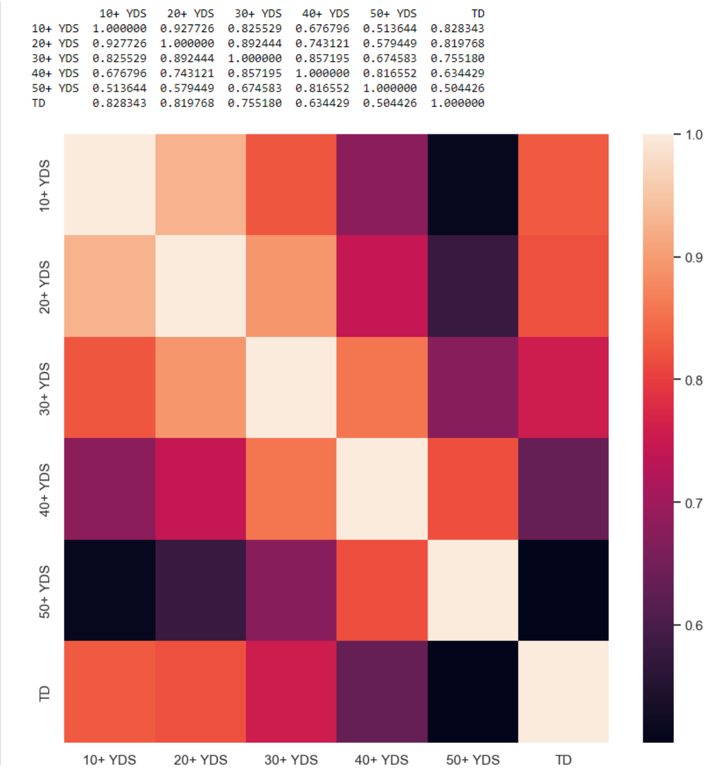


Figure 4: Correlation Heat Map (Above)



```
          YDS        TD
YDS  1.000000  0.842596
TD   0.842596  1.000000
```

Figure 5: Scatterplot of Yards and Touchdowns



```
         10+ YDS   20+ YDS   30+ YDS   40+ YDS   50+ YDS        TD
10+ YDS  1.000000  0.927726  0.825529  0.676796  0.513644  0.828343
20+ YDS  0.927726  1.000000  0.892444  0.743121  0.579449  0.819768
30+ YDS  0.825529  0.892444  1.000000  0.857195  0.674583  0.755180
40+ YDS  0.676796  0.743121  0.857195  1.000000  0.816552  0.634429
50+ YDS  0.513644  0.579449  0.674583  0.816552  1.000000  0.504426
TD       0.828343  0.819768  0.755180  0.634429  0.504426  1.000000
```

Figure 6: Correlation Heat Map (10-50 yard receptions & TD's) (Above)

VI. Methods: Model Chosen & Data Preparation

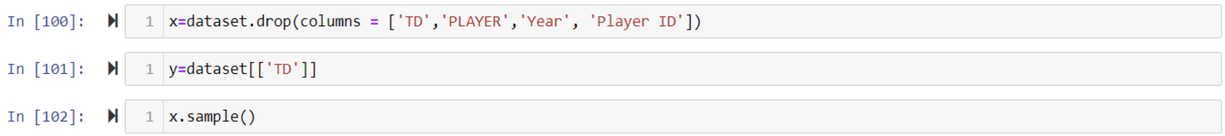
I used a multiple linear regression model rather than a logistic regression model. While touchdowns scored is technically a discrete variable., I want my models' predictions to have decimals. This will help me to see if the player was predicted to have closer to one whole number or the other, rather than just seeing the whole number (ex. If the model predicts a player to have scored 5.65 touchdowns, that .65 is valuable information to me, rather than it just rounding). In fantasy football, many prediction software's will say "your player is projected .78 touchdowns this game." Obviously, this is not possible, but it sheds light on the likelihood of it happening. Since they model touchdowns as a continuous variable, so will I.

For data preparation, I had to decide what variables to keep, and what variables to drop. Even in my first two models, using raw features, I dropped some variables. First, I dropped the year. This is a numerical value but not s statistic, I did not want it impacting my model. Second, I also dropped player ID for the same reasons I dropped the year. Lastly, I dropped the player's name. It would be too tedious and unnecessary to try and convert 590 player names to numeric values. It is also not a statistic, so it is irrelevant to my model.

VII. Methods: Experimental Design & Accuracy

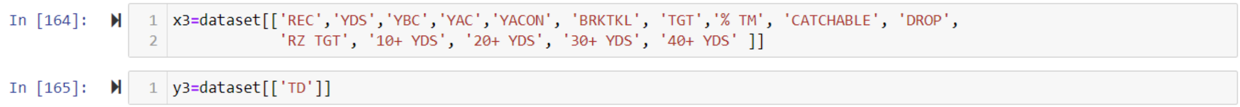
I created four different models to train and test in hopes of predicting a receivers touchdowns scored. The first two models contained all of the receiver metrics in the data set (Refer to **Figure 7** to see how these were split up). The second two, however were specifically chosen variables that had a high correlation with touchdowns scored (Refer to **Figure 8** to see the chosen independent variables for Model 3 & 4). I split and trained each model by importing train_test_split and LinearRegression (see **Figure 8**). Since I am using a multiple linear regression model, I used R-square, MSE, and RMSE to determine the accuracy of my model. To my surprise, the raw feature models performed slightly better than the selected feature models. **Table 2** shows each models parameters and accuracy scores. Model 2 was slightly the most accurate and what I ended up using in the "Application" section of this paper.

Model Number	Parameters	R-squared	MSE	RMSE
1	All raw features with 80/20 split for train, and test	0.708	1.752	1.32
2	All raw features with 70/30 split for train, and test	0.728	1.88	1.37
3	Selected features with 80/20 split for train, and test	0.707	1.756	1.325
4	Selected features with 70/30 split for train, and test	0.726	1.890	1.374



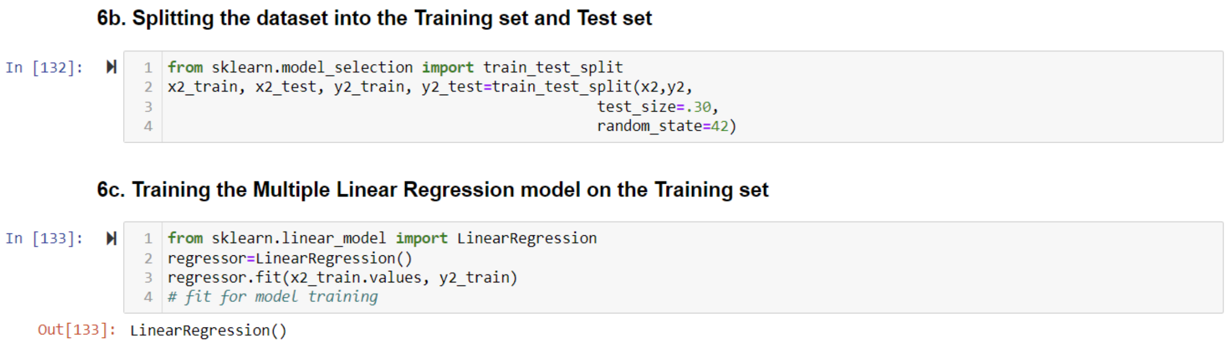
```
In [100]:  x=dataset.drop(columns = ['TD','PLAYER','Year', 'Player ID'])

In [101]:  y=dataset[['TD']]

In [102]:  x.sample()
```

Figure 7: X & Y split for Model 1 & 2



```
In [164]:  x3=dataset[['REC','YDS','YBC','YAC','YACON', 'BRKTKL', 'TGT','% TM', 'CATCHABLE', 'DROP',
                      'RZ TGT', '10+ YDS', '20+ YDS', '30+ YDS', '40+ YDS' ]]

In [165]:  y3=dataset[['TD']]
```

Figure 8: X & Y split for Model 3 & 4



6b. Splitting the dataset into the Training set and Test set

```
In [132]:  from sklearn.model_selection import train_test_split
           x2_train, x2_test, y2_train, y2_test=train_test_split(x2,y2,
                                                                 test_size=.30,
                                                                 random_state=42)
```

6c. Training the Multiple Linear Regression model on the Training set

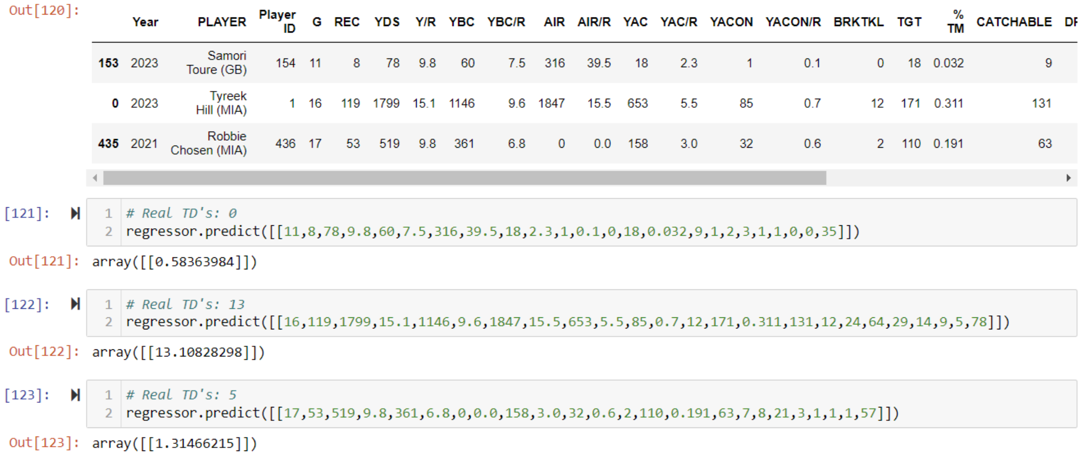
```
In [133]:  from sklearn.linear_model import LinearRegression
           regressor=LinearRegression()
           regressor.fit(x2_train.values, y2_train)
           # fit for model training

Out[133]: LinearRegression()
```

Figure 9: Splitting and Training the models

VIII. Results & Application

All four of my models were nearly identical (within 2 percent R-square of each other). Model 2 had the best R-square score. I think this is due to the large amounts of independent variables it had compared to the selected feature models. Along with the 70/30 split being slightly more advantageous than the 80/20. I then tested Model 2 on 75 receivers' 2023 season (See *2023 TD's Model Predictions* notebook for all 75). I was quite satisfied with my models predictions (See **Figure 10** for an example). The model was quite accurate in predicting a receivers touchdowns. There were a few large variances between actual touchdowns scored and the model's prediction. However, I am not discouraged by this, I am rather intrigued. I do not think these specific variances are due to my model's inaccuracies. Rather, I think it is due to the volatility of touchdowns.

This variance between my model and actual touchdowns scored is where I can apply this knowledge to fantasy football. For the majority of the receivers, the model prediction and the actual touchdowns were very close to each other. However, there are a few receivers that drastically over/under performed my models' expectation. For example, based on his underlying metrics, my model predicted Drake London to score seven touchdowns last season. However, he only scored two. This tells me that, based on his performance, Drake London should have scored many more touchdowns than he did. He was just unlucky and should positively regress to the mean. He will likely be undervalued in drafts next year. So, I should target him in drafts! On the other hand, there are some receives who outperformed the models expectation. Based on his output, my model projected Mike Evans to score eight touchdowns last season. However, he scored 13! If my model proves to be accurate, with the same production, Evans likely got a little lucky and will regress to the mean next year. I may want to avoid him in drafts seeing he will be overvalued.



Out[120]:

	Year	PLAYER	Player ID	G	REC	YDS	Y/R	YBC	YBC/R	AIR	AIR/R	YAC	YAC/R	YACON	YACON/R	BRKTKL	TGT	% TM	CATCHABLE	DF
153	2023	Samori Toure (GB)	154	11	8	78	9.8	60	7.5	316	39.5	18	2.3	1	0.1	0	18	0.032	9	
0	2023	Tyreek Hill (MIA)	1	16	119	1799	15.1	1146	9.6	1847	15.5	653	5.5	85	0.7	12	171	0.311	131	
435	2021	Robbie Chosen (MIA)	436	17	53	519	9.8	361	6.8	0	0.0	158	3.0	32	0.6	2	110	0.191	63	

```
In [121]:
# Real TD's: 0
regressor.predict([[11,8,78,9.8,60,7.5,316,39.5,18,2.3,1,0.1,0,18,0.032,9,1,2,3,1,1,0,0,35]])

Out[121]: array([[0.58363984]])

In [122]:
# Real TD's: 13
regressor.predict([[16,119,1799,15.1,1146,9.6,1847,15.5,653,5.5,85,0.7,12,171,0.311,131,12,24,64,29,14,9,5,78]])

Out[122]: array([[13.10828298]])

In [123]:
# Real TD's: 5
regressor.predict([[17,53,519,9.8,361,6.8,0,0.0,158,3.0,32,0.6,2,110,0.191,63,7,8,21,3,1,1,1,57]])

Out[123]: array([[1.31466215]])
```

Figure 10: Predicting touchdowns scored compare to actual touchdowns scored

IX. Problems, Limitations, & Improvements

The greatest problem I faced was the time it took to 1). pull the data and 2.) compare my model to the actual wide receivers. First, while the FantasyPros database did a great job providing me all of the advanced metrics, I had to manually look up all 590 players touchdowns scored and enter them into the data. It also took a lot of time to see what my model predicted for 75 receivers. I wanted to do this so I could actually apply my model and find the wide receivers I want to draft and the ones I want to avoid in fantasy football next season (I would say I went above and beyond the expectation). The biggest limitation of my model is the factors that I couldn't quantify. I could not account for the skill level of the receiver's quarterback, the team's success, overall sheer talent, etc. These are all factors that will drastically affect a player's ability to score that aren't shown in these stats. If I could improve this model, I would add team stats to the independent variables. Wide receivers that are on teams with better offenses have more opportunities to score. Maybe I could have created a "QB ranking" column or find a quantifiable way to show the quality of the receiver's quarterback.

X. Tools

I used the following tools for this analysis & model: Python v3.5.2 running the Anaconda 4.3.22 on a Samsung V3 computer was used for all analysis and implementation. The following libraries were also used in Python: Pandas 0.18.1, Numpy 1.11.3, Matplotlib 1.5.3, Seaborn 0.7.1, SKLearn 0.18.1, and LinearRegression. I also used Microsoft Excel to export my data, create a CSV file, and get it into Python. I chose these tools because we have learned how to use them efficiently in DS-160 to create insightful analysis and accurate models.

XI. Conclusion

In conclusion, I would say this analysis was a success. I was able to see that wide receiver statistics' distribution are positively skewed due to outliers (the elite wide receivers). I found over a dozen variables that have moderate to strong positive correlation with touchdowns scored. I created four successful models that did a quite accurate job at predicting a player's touchdowns scored by analyzing his underlying metrics. I was able to use this model to determine regression candidates for the next NFL season. From this, I have a better idea of who I should target (Drake London, Amari Copper, etc.) and who I should avoid (Mike Evans, Jordan Addison, etc) in fantasy football drafts next season.