Classifying NFL Over/Unders

By: Mike Messina

I. Business Understanding

With sports gambling becoming legalized across the United States at a rapid pace, more people than ever are gambling on sports because well, *sports*. Everyone watches sports. Whether you grew up watching with your dad on the couch as he yelled at the tv about his team. Or picked it up later in life when it was shoved down your throat on every commercial on television. One doesn't need to go to school to "know sports" which is why everyone thinks they can be pretty good at gambling on it.

Throughout the industry, the magic number in sports betting is 52.4%. This is the golden win rate that all sports bettors strive for. You might ask: why not 51%? Well, on a typical NFL bet, the sportsbook factor in a vig which is essentially a 10% deposit on the amount you bet that you will get refunded if you win your bet, but you will lose if you lose your bet. Because of the vig, one must strive for a 52.4% win rate. This is the metric my model will be striving for in terms of accuracy.

There are multiple ways to bet on an NFL football game. You can bet on each team winning straight up. You can use the lines set by oddsmakers to choose a team to win or lose within a certain spread, and you can bet on the total amount of points both team will score in the game, known as the over/under total (O/U). Before the start of each game, oddsmakers will set a total amount of points that you can choose to bet if the actual total amount of points will be over or under that threshold. This is the bet I will be focusing on here.

The goal of this model is to take in as much pregame information as I can regarding the game over the last 20 years, and see if I can identify if certain pregame factors can tell me which side of the O/U is more likely to hit. I will be using a classification model as I am are trying to predict the non-numeric result of if I should bet over, or under the O/U total for the game.

II. Data Preparation

I began my search for a dataset that would incorporate the most pregame variables we could find. Thanks to our friends over at *Spreadspoke*, who specialize in analyzing sports odds information, I was able to obtain a data set that included 17 different variables about every NFL football game played since 1966 which you can find on Kaggle here: https://www.kaggle.com/tobycrabtree/nfl-scores-and-betting-data#spreadspoke_scores.csv. This comes in at 12,667 games.

To begin preparing this data, I first cut the amount of games by 7,351 to only encompass the games from the seasons 2000 through 2018 which is 5,316 games. I eliminated the years prior to 2000 as the game of football was played drastically different than it is today. Running plays were much more common than passing so weather might not have had as big of an impact back then as it does today as it is harder to throw the ball in poor weather. Scoring was also lower in these earlier years so the O/U line set for these games was most likely a lot lower than games played within the past 20 years. Oddsmakers, with the help of computers, have also gotten incredibly more accurate at setting these O/U lines so I wanted to be sure to factor this in. I additionally eliminated the 2019 season as the data for these games was incomplete.

While then turning my attention to the variables that were included in this data set, I first tried to identify which variables were essentially repeated, which variables I believed did not have a material effect on what we are trying to predict, and which variables, if any, might be missing that could help our model.

I first removed the “weather_humidity” variable due to what I believe was a lack of relevance to the model. I also removed “game_id” for lack of relevance as well. These were the only two variables which I believed had little to no effect on how many points were scored in a game.

I then wanted to add in a “result” variable which would actually tell us if the over or the under hit for the game. This will be the variable my model is trying to predict. I simply added this by taking the sum of the “score_home” and “score_away” variables and if this sum was greater than the variable “over_under_line” then “over” would be the result and vice versa. After adding this variable, I removed the variables “score_home” and “score_away” as these would completely give away what the model is trying to predict. This is also postgame information rather than information that is known before the game which our model must only use.

Moving onto the variables that were repeated. I then removed “weather_detail” as this is a repeated variable with “weather_temp” which is a much more consistent variable in terms of the data within this column.

I then wanted to think about variables that could still be missing that could be important to how many points are scored in a football game. I believe that knowing how good each team playing was would be a great indicator. For example, games with one team that is a lot better than the other team may be more likely to score more total points. I decided I would add three variables here. “home_rank”, “away_rank”, and “rank_diff”. To start, I took the count of how many times each team appeared in the “team_favorite” column of the data. Each team was then ranked 1 – 32 (32 being the worst). I then used vlookup with the ranks of each team and whenever each team appeared in the “team_home” and “team_away” columns to insert the “away_rank” and “home_rank” column data. I then took the difference of these two columns to populate the “rank_diff” variable which would show that a larger differential meant that two uneven teams were playing which might result in more points. This could be thought of as a closely repeated variable with “spread_favorite” variable but I believed that this was important enough information to rank every team from the last 20 years.

After running a summary of the data at this point in R, I then identified that there were 120 missing values for “weather_temp” and “weather_wind_mph”. You can view this in *figure 1* below:

Figure 1.

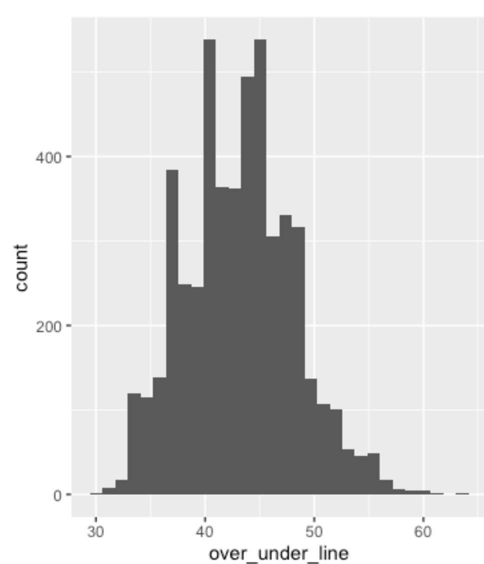
```
schedule_date  schedule_season schedule_week  schedule_playoff
1/1/12 :  16   Min.   :2000    14      : 302  Mode :logical
1/1/17 :  16   1st Qu.:2004    15      : 302  FALSE:4848
1/2/05 :  16   Median :2009    16      : 302  TRUE :209
1/2/11 :  16   Mean   :2009    17      : 302
1/3/10 :  16   3rd Qu.:2014    1       : 301
1/3/16 :  16   Max.   :2018    13      : 301
(Other):4961                   (Other):3247
               team_home                     team_away
New England Patriots: 181   Baltimore Ravens    : 169
Philadelphia Eagles : 166   Indianapolis Colts  : 165
Pittsburgh Steelers : 166   Seattle Seahawks    : 165
Indianapolis Colts  : 165   Green Bay Packers   : 164
Green Bay Packers   : 164   New England Patriots: 163
Denver Broncos      : 163   New York Jets       : 163
(Other)             :4052   (Other)             :4068
team_favorite_id spread_favorite   over_under_line
NE     : 271     Min.   :-26.500   Min.   :30.00
PIT    : 236     1st Qu.: -7.000   1st Qu.:39.50
IND    : 218     Median : -4.500   Median :43.00
PHI    : 216     Mean   : -5.385   Mean   :43.16
GB     : 215     3rd Qu.: -3.000   3rd Qu.:46.50
DEN    : 207     Max.   :  0.000   Max.   :63.50
(Other):3694
                   stadium     stadium_neutral
Giants Stadium         : 166   Mode :logical
Lambeau Field          : 163   FALSE:5000
Bank of America Stadium: 158   TRUE :57
Gillette Stadium       : 158
M&T Bank Stadium       : 158
Arrowhead Stadium      : 157
(Other)                :4097
weather_temperature weather_wind_mph          weather_detail
Min.   :-6.0        Min.   : 0.000   Clear           :3658
1st Qu.:50.0        1st Qu.: 0.000   DOME            :1167
Median :64.0        Median : 6.000   Rain            : 106
Mean   :60.4        Mean   : 6.359   DOME (Open Roof):  56
3rd Qu.:72.0        3rd Qu.:10.000   Fog             :  28
```

```
Max.   :97.0        Max.   :40.000   Rain | Fog      :  22
NA's   :120         NA's   :120      (Other)         :  20
  home_rank        away_rank        rank_diff       result
Min.   : 1.00   Min.   : 1.00   Min.   : 0.00   Over :2538
1st Qu.: 8.00   1st Qu.: 8.00   1st Qu.: 5.00   Under:2519
Median :16.00   Median :16.00   Median : 9.00
Mean   :16.13   Mean   :16.19   Mean   :11.28
3rd Qu.:24.00   3rd Qu.:24.00   3rd Qu.:17.00
Max.   :32.00   Max.   :32.00   Max.   :31.00
```

Although this data that is missing is only 2.37%, it is still very important to my model. As it is weather data and the games vary from city to city across the United States with different climates, and the games vary at different times of the year where it gets colder as the season goes on, I did not believe that replacing these values with averages would be accurate. Instead, one can deduct that these missing variables can be related to the name of the stadium in the "stadium" column (for location purposes) and the "schedule_season" column that gives us the week of the season (for the time of the year that the weather is like). I first made a separate table of all the stadiums included in our dataset and pulled the average of these temperatures and wind speeds from the applicable columns of the data for each week of the season. I then ran a vlookup to identify the stadiums that were missing this weather data and inserted the average temperature and wind speed for the given week in the season in these locations. Our data is now 100% complete.

Some interesting variables of note when looking at the summary of data was the average O/U line and the range that these fell in See *figure 2* below for a histogram of this variable

Figure 2.



The median O/U for the last 20 years was noted to be 43. This seems very moderate to me and tells me that O/U lines are typically not very drastic. Not enough at least to sway the median one way or the other. *Figure 2* also tells us that the vast majority of these lines all fall within 10 points of each other (40 - 50) which is quite small when you consider one touchdown from one team alone is worth 6 points.

Another interesting thing to note from *Figure 1* is that 92% of the games from this past year occurred in non-inclement weather conditions. I did not expect this number to be this high so my assumption that weather related variables having a major effect on this model could prove to be false.

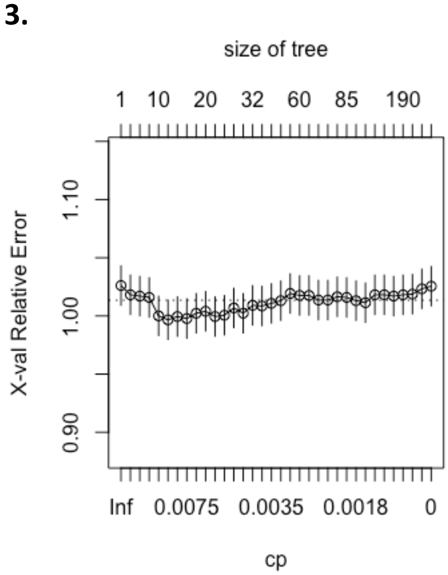
As we set our sights on modeling, I used a 70/30 train/test split of the data which I believe was an optimal balance of introducing new data to the model given the large size of the data set.

III. Modeling and Evaluation

Decision Tree

The first model I built to predict the result of the O/U was a decision tree. Due to the size of the tree given the amount of variables in the dataset and the length of variable names, an illustration of the tree was not useful to view. However, as we built the model it was determined that 10 branches would be optimal as you can see below in *figure 3.* A decision tree was chosen to model our data as I felt it would be the quickest way to get a general idea of if what I was trying to classify was going to be remotely possible and to get a baseline for variable importance.

Figure 3.



A 70/30 train/test split was used in the model which I thought was a fair spread to find the right balance between an overfitted and underfitted model given the dataset included 5,316 rows. The accuracy of this general decision tree came out to 49.41% (see *figure* 4) which is not exceptional, but not as bad as I thought it was going to be. Remember, the target accuracy we're shooting for is 52.4% so this is a good first step. It's concerning that the sensitivity (50.72%) is higher than the specificity (48.08%) as I would much prefer to predict the result correctly than to be wrong.

Figure 4.

```
Confusion Matrix and Statistics

          Reference
Prediction Over Under
     Over   386   392
     Under  375   363

               Accuracy : 0.4941
                 95% CI : (0.4686, 0.5196)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : 0.7396

                  Kappa : -0.012

 Mcnemar's Test P-Value : 0.5634

            Sensitivity : 0.5072
            Specificity : 0.4808
         Pos Pred Value : 0.4961
         Neg Pred Value : 0.4919
             Prevalence : 0.5020
         Detection Rate : 0.2546
   Detection Prevalence : 0.5132
      Balanced Accuracy : 0.4940

       'Positive' Class : Over
```

After viewing the variable importance of the general decision tree in *figure 5,* I didn't expect to see the schedule date variable ranked as the most important variable. Although the date of the game is directly related to the temperature and weather of that game which I believe have the greatest effect on the amount of points scored in a game, it appears reasonable to me that this would have some importance to the model. But with the wind speed and weather detail variables ranked so low, combined with the low accuracy metric, I am thinking that this general decision tree won't be a very useful model as we move forward.

Figure 5.

```
> tree$variable.importance
    schedule_date           team_home           team_away
       1176.31243           452.96874           420.00624
 team_favorite_id             stadium       schedule_week
        399.77922           313.11607           285.63020
  schedule_season weather_temperature      over_under_line
         87.11938            85.71070            78.22390
  spread_favorite           home_rank           away_rank
         77.64700            76.25654            75.48929
        rank_diff     weather_wind_mph      weather_detail
         51.46295            46.74846            25.91582
 schedule_playoff
         16.88887
```

Given these poor metrics of our initial decision tree, I believe that pruning the tree will be useful to hopefully eliminate some noise as we can reduce some complexity of the tree. After pruning the tree, the accuracy of our decision tree model actually decreased to 48.15% from 49.41% as seen in *figure 6.* The specificity value at least became higher than the sensitivity which was good to see in the model. If we can take one positive thing away here, it's that our model doesn't seem to be overfitted since pruning the tree did not improve the accuracy of the model.

Figure 6.

```
Confusion Matrix and Statistics

          Reference
Prediction Over Under
     Over   361   386
     Under  400   369

               Accuracy : 0.4815
                 95% CI : (0.4561, 0.507)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : 0.9472

                  Kappa : -0.0369

 Mcnemar's Test P-Value : 0.6429

            Sensitivity : 0.4744
            Specificity : 0.4887
         Pos Pred Value : 0.4833
         Neg Pred Value : 0.4798
             Prevalence : 0.5020
         Detection Rate : 0.2381
   Detection Prevalence : 0.4927
      Balanced Accuracy : 0.4816

       'Positive' Class : Over
```

As I stated previously on pg.3, the home and away rank and rank differential variables that I added to the dataset could actually just be adding noise to the data since they are so closely related to the spread variable. To see if this was true, I removed these three variables which in turn, increased the accuracy of our decision tree model to 49.87%. Although this is only a .04% increase, it still proves that these variables were not helping the model and were simply adding to the already high complexity of the data. See *figure 7* for the confusion matrix of the decision tree without these rank variables.

Figure 7.

```
Confusion Matrix and Statistics

          Reference
Prediction Over Under
     Over   375   374
     Under  386   381

               Accuracy : 0.4987
                 95% CI : (0.4732, 0.5242)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : 0.6112

                  Kappa : -0.0026

 Mcnemar's Test P-Value : 0.6899

            Sensitivity : 0.4928
            Specificity : 0.5046
         Pos Pred Value : 0.5007
         Neg Pred Value : 0.4967
             Prevalence : 0.5020
         Detection Rate : 0.2474
   Detection Prevalence : 0.4941
      Balanced Accuracy : 0.4987

       'Positive' Class : Over
```

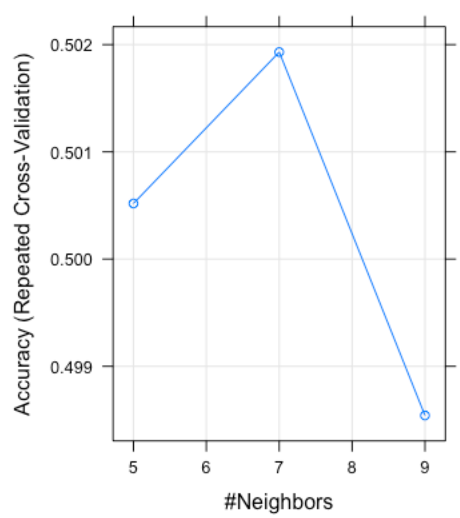
To conclude, my three Decision Tree models had an average accuracy of 49.14% with no model surpassing even 50% which is a losing model if we are betting on O/U spreads over time.

K-Nearest Neighbor

As the Decision Trees didn't show us optimal results, I decided to run the data through a K-Nearest Neighbor model. Due to the near zero variance of the "schedule_date" variable causing errors in the model, we simply eliminated this variable as it is expressed in the "schedule_season" and "schedule_week" columns. It was decided that 7 neighbors was optimal for our data (see *figure 8*).

Figure 8.

```
k-Nearest Neighbors

3541 samples
 210 predictor
   2 classes: 'Over', 'Under'

No pre-processing
Resampling: Cross-Validated (3 fold, repeated 3 times)
Summary of sample sizes: 2360, 2361, 2361, 2361, 2361, 2360, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.5005184   0.001054670
  7  0.5019302   0.003875993
  9  0.4985419  -0.002860663

Accuracy was used to select the optimal model using the
 largest value.
The final value used for the model was k = 7.
```



When running variable importance for the K-Nearest Neighbor model in *figure 9*, these variables seemed a lot more reasonable than what was illustrated in the Decision Tree variable Importance. With weather related variables being three of the four most important variables to the model and even the Carolina Panthers team and stadium being in the top 20 most important variables (Per tripsavvy.com, Raleigh, NC is in the top 14 for wettest cities in the USA), this showed me that the K-Nearest Neighbor model was probably going to be a better model for predicting the O/U result than a Decision tree.

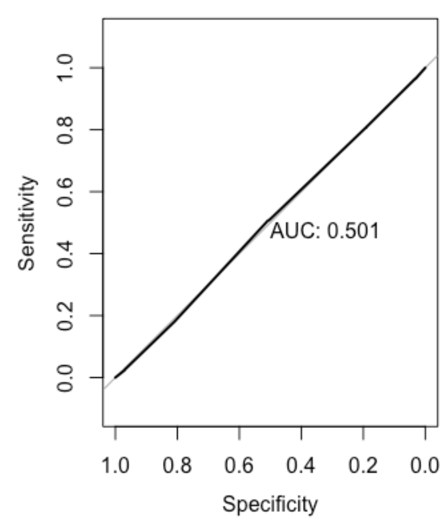
Figure 9.

```
ROC curve variable importance

  only 20 most important variables shown (out of 210)

                                Importance
weather_wind_mph                    100.00
away_rank                            79.76
weather_detail.Clear                 52.41
weather_detail.DOME                  51.98
schedule_week.5                      49.32
over_under_line                      46.29
team_away.Indianapolis Colts         24.60
schedule_week.4                      23.33
schedule_week.1                      22.79
team_favorite_id.CAR                 21.44
schedule_week.12                     20.76
schedule_week.11                     20.72
team_away.Green Bay Packers          20.59
team_home.Carolina Panthers          19.44
stadium.Bank of America Stadium      19.44
stadium.FirstEnergy Stadium          19.42
team_favorite_id.NO                  18.49
team_home.Cleveland Browns           18.42
team_favorite_id.MIA                 18.33
stadium.Lucas Oil Stadium            18.23
```

As seen from the confusion matrix of the K-Nearest Neighbor model in *figure 10,* the accuracy of our K-Nearest Neighbor model still falls short of our 52.4% accuracy target at 48.28% with even lower sensitivity and specificity metrics as our Decision Tree model. This is surprising given how I believed the K-Nearest Neighbor model was using more predictive variables than the Decision Tree yet we saw no improvement in accuracy. As the accuracy measure falls using KNN, we would most likely prefer the Decision Tree model at this point due to the cost of running each model. Although, the AUC value of our ROC curve shows a favorable .501 value as it is able to separate the over and under classes about half the time. Again, still short about 2 percentage points of optimal for what we're trying to do.

Figure 10.

```
Confusion Matrix and Statistics

          Reference
Prediction Over Under
     Over   364   387
     Under  397   368

               Accuracy : 0.4828
                 95% CI : (0.4574, 0.5083)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : 0.9352

                  Kappa : -0.0343

 Mcnemar's Test P-Value : 0.7479

            Sensitivity : 0.4783
            Specificity : 0.4874
         Pos Pred Value : 0.4847
         Neg Pred Value : 0.4810
             Prevalence : 0.5020
         Detection Rate : 0.2401
   Detection Prevalence : 0.4954
      Balanced Accuracy : 0.4829

       'Positive' Class : Over
```



Naïve Bayes

As we still have not seen adequate results from our previous two models, I decided to at least see what a Naïve Bayes model could do with the data. Although this model assumes cold game temperatures to be completely unrelated to the schedule date of the game, for example, maybe this is the kind of simplicity our data needs to make a more accurate prediction of the O/U spread.

After running the data through a Naïve Bayes classifier (as seen in *figure 11),* we found our most accurate classifier yet at 51.45%. However, this mark is still lower than 52.4% and the severe decline in specificity leaves a lot to be desired for our purposes.

Figure 11.

```
Naive Bayes

3541 samples
 154 predictor
   2 classes: 'Over', 'Under'

No pre-processing
Resampling: Cross-Validated (3 fold, repeated 3 times)
Summary of sample sizes: 2360, 2361, 2361, 2361, 2361, 2360,
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.5028699  0.005598662
   TRUE      0.5061657  0.011061945

Tuning parameter 'fL' was held constant at a value of 0

Tuning parameter 'adjust' was held constant at a value of 1
Accuracy was used to select the optimal model using the
 largest value.
The final values used for the model were fL = 0, usekernel
 = TRUE and adjust = 1.
```

```
Confusion Matrix and Statistics

          Reference
Prediction Over Under
     Over   540   515
     Under  221   240

               Accuracy : 0.5145
                 95% CI : (0.489, 0.54)
    No Information Rate : 0.502
    P-Value [Acc > NIR] : 0.171

                  Kappa : 0.0275

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7096
            Specificity : 0.3179
         Pos Pred Value : 0.5118
         Neg Pred Value : 0.5206
             Prevalence : 0.5020
         Detection Rate : 0.3562
   Detection Prevalence : 0.6959
      Balanced Accuracy : 0.5137

       'Positive' Class : Over
```

IV. Conclusion

To summarize, none of my models performed up to our positive expected value benchmark of 52.4% accuracy. Looking back, this is most likely due to the complexity of the variables in the dataset. However, the fact that each model was within 2% of 50% accuracy is impressive to me and illustrates that there is something to predicting O/U totals in NFL games. Perhaps weather does not have as big of an impact on point totals in football games as I initially believed. If we were to revisit this in the future, I would actually like to incorporate some regression analysis into the specific O/U totals set for each game to see if we can identify certain totals that lean one way or the other more often that we can try and narrow our classifier on. Perhaps we could look more into the variable importance for each model and cut the amount of variables used in each model in half.

Model	Accuracy
Decision Tree	49.41%
Decision Tree (Pruned)	48.15%
K-Nearest Neighbor	48.28%
Naïve Bayes	51.45%

If we actually take a closer look at each O/U total from every NFL game from the past 20 years in *figure 12,* it's incredible to note that the Over has hit in 49% of all games and the Under has hit in 51% of all games as seen in the grand total row of the graph. This showcases incredible precision by oddsmakers to set these lines so as to not give an edge to one side or the other and partially explains why each of our models were hitting around the 50% mark. Only near the outliers of O/U totals do you see any major lean one way or the other. So, if we can take anything away from this project, it's that you just can't beat Vegas.

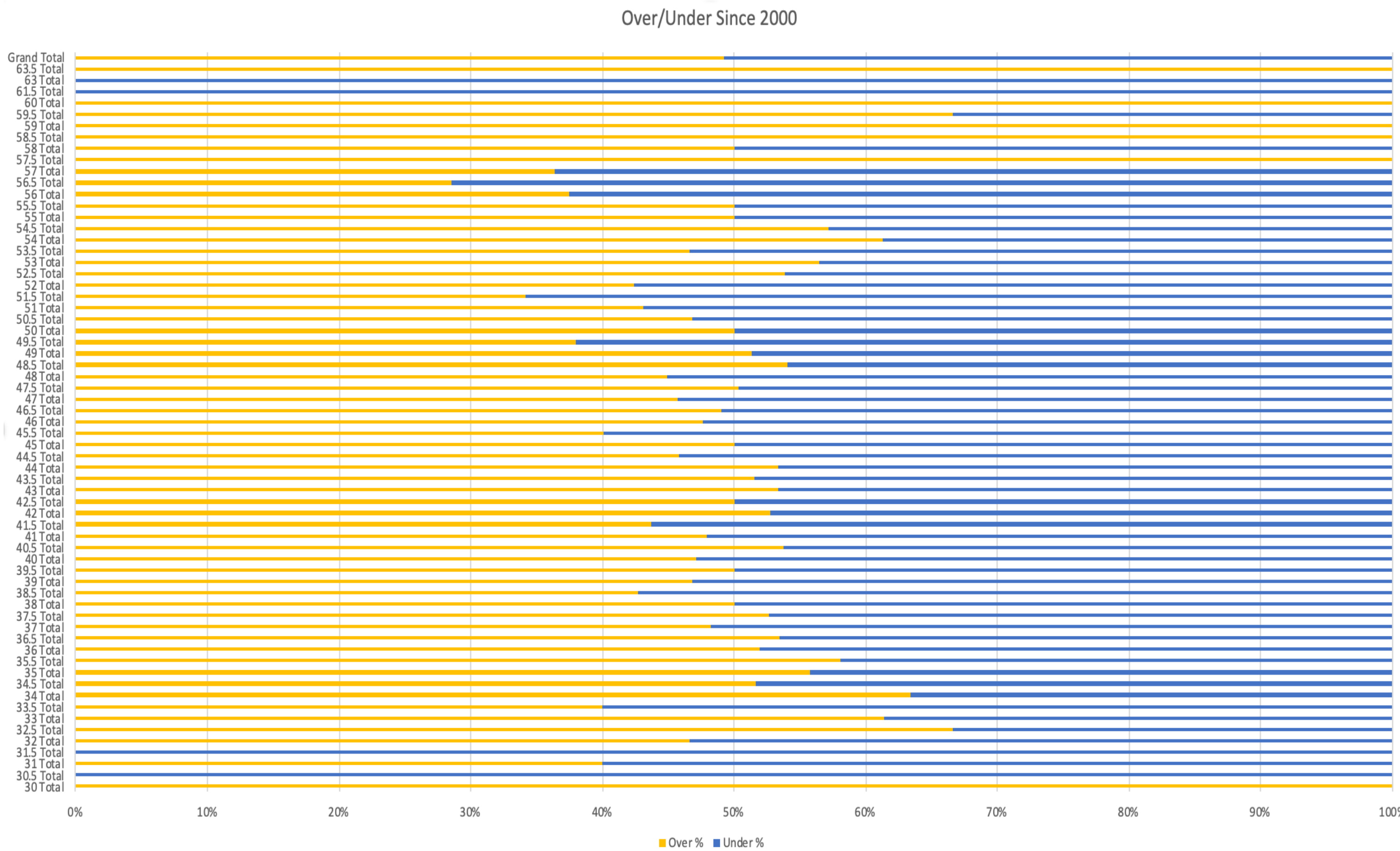

Over/Under Since 2000
Grand Total
63.5 Total
63 Total
61.5 Total
60 Total
59.5 Total
59 Total
58.5 Total
58 Total
57.5 Total
57 Total
56.5 Total
56 Total
55.5 Total
55 Total
54.5 Total
54 Total
53.5 Total
53 Total
52.5 Total
52 Total
51.5 Total
51 Total
50.5 Total
50 Total
49.5 Total
49 Total
48.5 Total
48 Total
47.5 Total
47 Total
46.5 Total
46 Total
45.5 Total
45 Total
44.5 Total
44 Total
43.5 Total
43 Total
42.5 Total
42 Total
41.5 Total
41 Total
40.5 Total
40 Total
39.5 Total
39 Total
38.5 Total
38 Total
37.5 Total
37 Total
36.5 Total
36 Total
35.5 Total
35 Total
34.5 Total
34 Total
33.5 Total
33 Total
32.5 Total
32 Total
31.5 Total
31 Total
30.5 Total
30 Total
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Over %
Under %

Figure 12.