CS 105 Final Project

March 17, 2025

1 PREDICTING STANDINGS OF THE FORMULA 1 DRIVER'S CHAMPIONSHIP

Group Members: Dmitry Sorokin, Justin Shiu, Kyle Chahal, Justin An

2 Phase 2: Project Outline

For our project, we decided to try and predict the standings of the Driver's Championship in Formula 1 (F1). Our data set consists of F1 race results, drivers, and teams from the 2019-2024 seasons. Our goal is to use the driver's team, their number of podium finishes, their fastest laps for races, their finishing position and starting position, and their podium percentage to determine their final Driver's Championship ranking. Furthermore, we plan to use a model to partition the drivers into 3 different tiers: lower tier, middle tier, and high tier drivers. This is to see the caliber of drivers that each team and country has.

3 Features in the Dataset

3.1 All_drivers Variables:

- **Number** - A driver's personal identifying number, unique to each driver
- **Team** - The team that the driver races for
- **Country** - The country where the driver originates from
- **Podiums** - The number of podiums that a driver has achieved for the season. A podium is classified as achieving first, second, or third place in a race.
- **Points** - The total number of points that a driver has accumulated in their career. Points are awarded to the top 10 places in a Grand Prix, with 1st place receiving 25 points and decreasing to 10th place receiving 1 point.
- **Grand Prix Entered** - The number of races that a driver has participated in during a season.
- **World Championships** - The number of World Championships a driver has won.
- **Highest Grid Position** - The highest starting position that a driver is granted in a race.
- **Podium Percentage** - A percentage calculated as podiums divided by Grand Prix entered.
- **Has WC** - This is a binary variable. `1` is granted to drivers who have won a world championship, while `0` is granted to those who have not.

3.2 All_races Variables:

- **Track** - The location of the race.

- **Position** - The final placement of the driver in this race.
- **No** - A driver's personal identifying number, unique to each driver.
- **Team** - The team that the driver races for.
- **Starting Grid** - The starting position of the driver in this race.
- **Points** - Number of points accumulated by a driver in this race.
- **Year** - The year of the race.
- **Net Position** - The difference between a driver's starting grid position and final position for this race.
- **Time (seconds)** - The recorded time to complete the race.

3.3 Standings Variables:

- **Year** - The year of the season.
- **Number** - A driver's personal identifying number, unique to each driver.
- **Position** - The place the driver finished in at the end of the season, in terms of total points.
- **Points** - Total points earned by a driver in that season.

Slides Link https://docs.google.com/presentation/d/1Dpz5lbLk-aITkDaxrO2_3_PTt-GS_-aODBaSxiGjrAE/edit?usp=sharing

Adding Libraries

Getting the Data frames ready:

Creating a dataframe that contians all of the information on the drivers across years

```
[2]:   Number             Team         Country  Podiums  Points  \
     0    44.0         Mercedes  United Kingdom      151  3431.0
     1    77.0         Mercedes         Finland       45  1289.0
     2    33.0  Red Bull Racing     Netherlands       31   948.0
     3    16.0          Ferrari          Monaco       10   303.0
     4     5.0          Ferrari         Germany      120  2985.0

        Grands Prix Entered  World Championships  Highest Grid Position  Year  \
     0                  250                    6                      1  2019
     1                  140                    0                      1  2019
     2                  102                    0                      1  2019
     3                   42                    0                      1  2019
     4                  241                    4                      1  2019

        Podium Percentage  Has WC
     0           0.604000       1
     1           0.321429       0
     2           0.303922       0
     3           0.238095       0
     4           0.497925       1
```

Now we will create a Data Frame that contains wanted information on all of the races across the years

```
[3]:       Track  Position  No                   Team  Starting Grid  Points  \
     0  Australia         1  77               Mercedes              2    26.0
     1  Australia         2  44               Mercedes              1    18.0
     2  Australia         3  33  Red Bull Racing Honda              4    15.0
     3  Australia         4   5                Ferrari              3    12.0
     4  Australia         5  16                Ferrari              5    10.0

       Fastest Lap  Year Time/Retired  Net Position
     0         Yes  2019  1:25:27.325             1
     1          No  2019      +20.886            -1
     2          No  2019      +22.520             1
     3          No  2019      +57.109            -1
     4          No  2019      +58.230             0
```

Since we do not have uniform time for our 'Time/Retired' variable, we will convert the race times to all be in seconds.

For each time required that has a value of + X number of laps, we add X * 90 seconds to the fastest time. We chose 90 seconds because that is around the average time it takes for a driver to complete a lap on any circuit.

```
[4]:          Track  Position  No                       Team  Starting Grid  \
     0    Australia         1  77                   Mercedes              2
     1    Australia         2  44                   Mercedes              1
     2    Australia         3  33      Red Bull Racing Honda              4
     3    Australia         4   5                    Ferrari              3
     4    Australia         5  16                    Ferrari              5
     ..         …        …  ..                        …            …
     107     Monaco         8  23  Scuderia Toro Rosso Honda             10
     108     Monaco         9   3                    Renault              6
     109     Monaco        10   8               Haas Ferrari             13
     110     Monaco        11   4            McLaren Renault             12
     111     Monaco        12  11  Racing Point BWT Mercedes             16

          Points Fastest Lap  Year Time/Retired  Net Position  Time (seconds)
     0      26.0         Yes  2019  1:25:27.325             1        5127.325
     1      18.0          No  2019      +20.886            -1        5148.211
     2      15.0          No  2019      +22.520             1        5149.845
     3      12.0          No  2019      +57.109            -1        5184.434
     4      10.0          No  2019      +58.230             0        5185.555
     ..      …         …   …          …           …           …
     107     4.0          No  2019      +55.200             2        5182.525
     108     2.0          No  2019      +60.894            -3        5188.219
     109     1.0          No  2019      +61.034             3        5188.359
     110     0.0          No  2019      +66.801             1        5194.126
     111     0.0          No  2019       +1 lap             4        5217.325

     [100 rows x 11 columns]
```

The output above shows our complete and modified ‘all_races’ data frame.

Let’s also import in a csv that has the driver standings by year.

No modifications were required as the data was clean when we imported it.

EDA

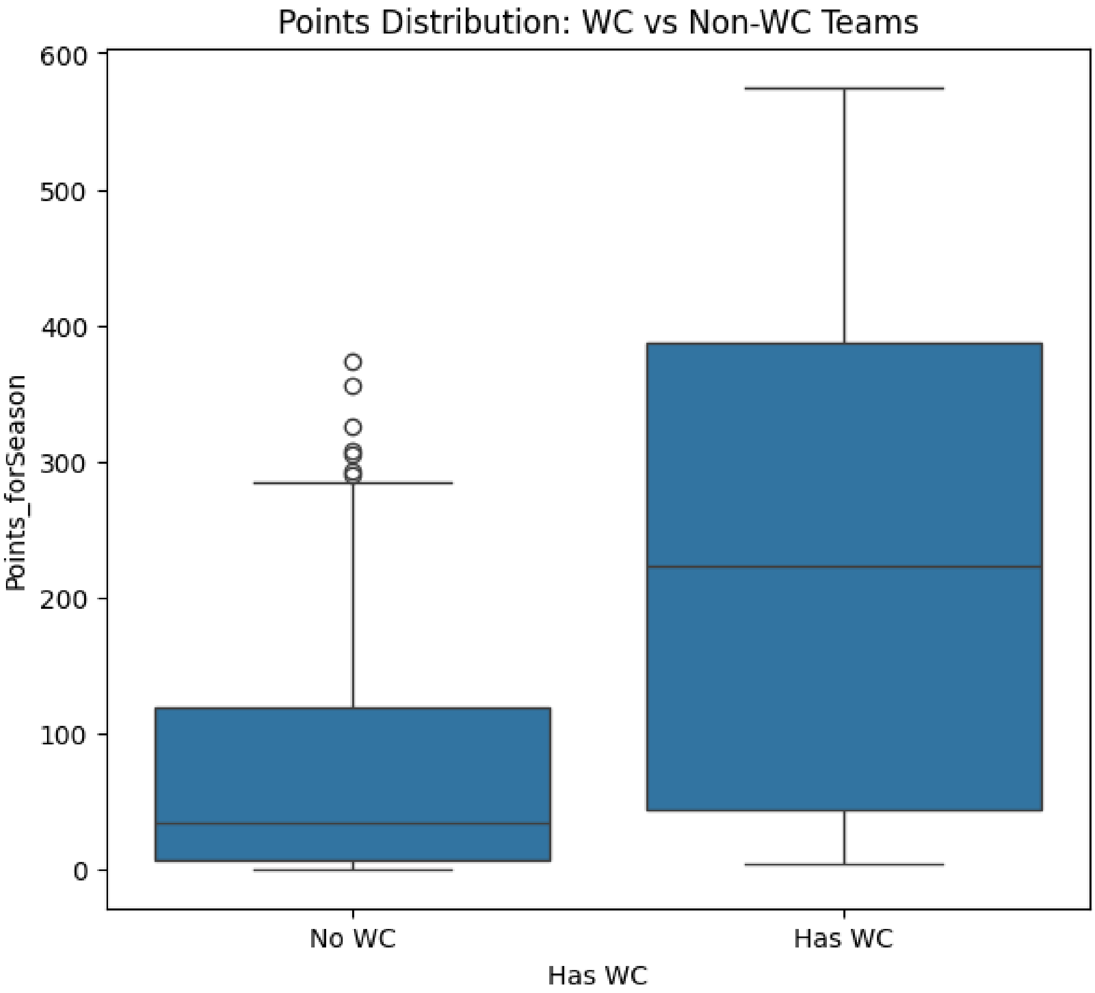
```
[6]:    Year  Position  Number  Points
     0  2019         1      44   413.0
     1  2019         2      77   326.0
     2  2019         3      33   278.0
     3  2019         4      16   264.0
     4  2019         5       5   240.0
```

To perform EDA we have created a merged data frame between the standings data frame and the all_drivers data frame.

```
[7]:    Number             Team         Country  Podiums  Points_driverTotal  \
     0    44.0         Mercedes  United Kingdom      151              3431.0
     1    77.0         Mercedes         Finland       45              1289.0
     2    33.0  Red Bull Racing     Netherlands       31               948.0
     3    16.0          Ferrari          Monaco       10               303.0
     4     5.0          Ferrari         Germany      120              2985.0

        Grands Prix Entered  World Championships  Highest Grid Position  Year  \
     0                  250                    6                      1  2019
     1                  140                    0                      1  2019
     2                  102                    0                      1  2019
     3                   42                    0                      1  2019
     4                  241                    4                      1  2019

        Podium Percentage  Has WC  Position  Points_forSeason
     0           0.604000       1         1             413.0
     1           0.321429       0         2             326.0
     2           0.303922       0         3             278.0
     3           0.238095       0         4             264.0
     4           0.497925       1         5             240.0
```

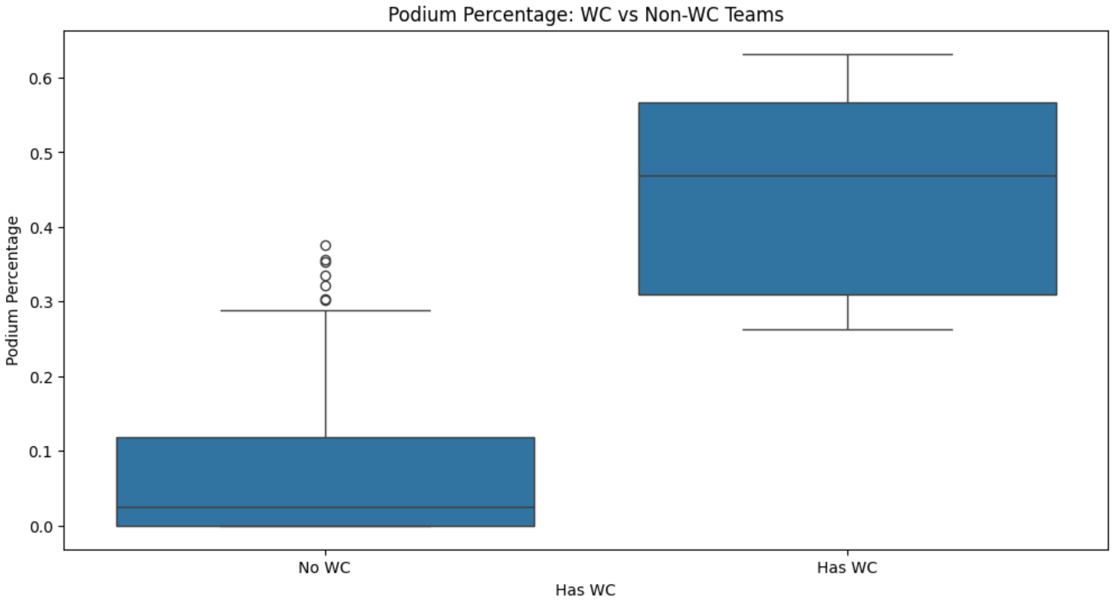
Now that we have our data cleaned and processed, we want to explore it and see what we can find.

To start, we were wondering if there would be any point difference between teams that have won a world championship vs teams who have not won a world championship.



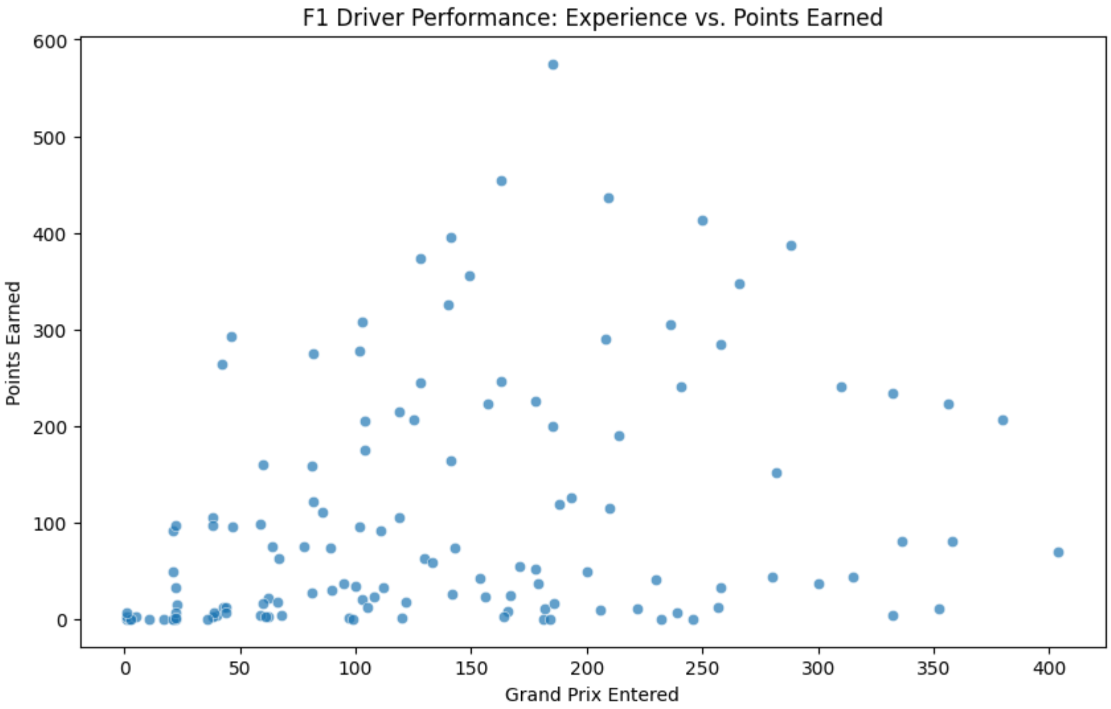
We can see here that generally those who have won world championships tend to have earned more points for the season. However, there are some instances where teams who haven't won a world championship have scored higher than those who have one a world championship which is demostrated by the outliers in the no world championship plot. We observe that many of these outliers for non world champions fall above the median points for world champions.

Next we were interested to see if there is any relationship between podium percentage and if a team has won a world championship or not.



We can see that teams that have won world championships have a substantially higher podium percentage. We can observe that the median podium percentage for non world champions is lower than 0.1 where as the median for world champions is over 0.5. There is no overlap between the upper quartile of non world champions and world champions, however, we do see some outliers here where some teams have higher podium percentages but just haven't won a world championship.

Continuing, we want to see the relationship between points earned and number of grand prix (races entered)

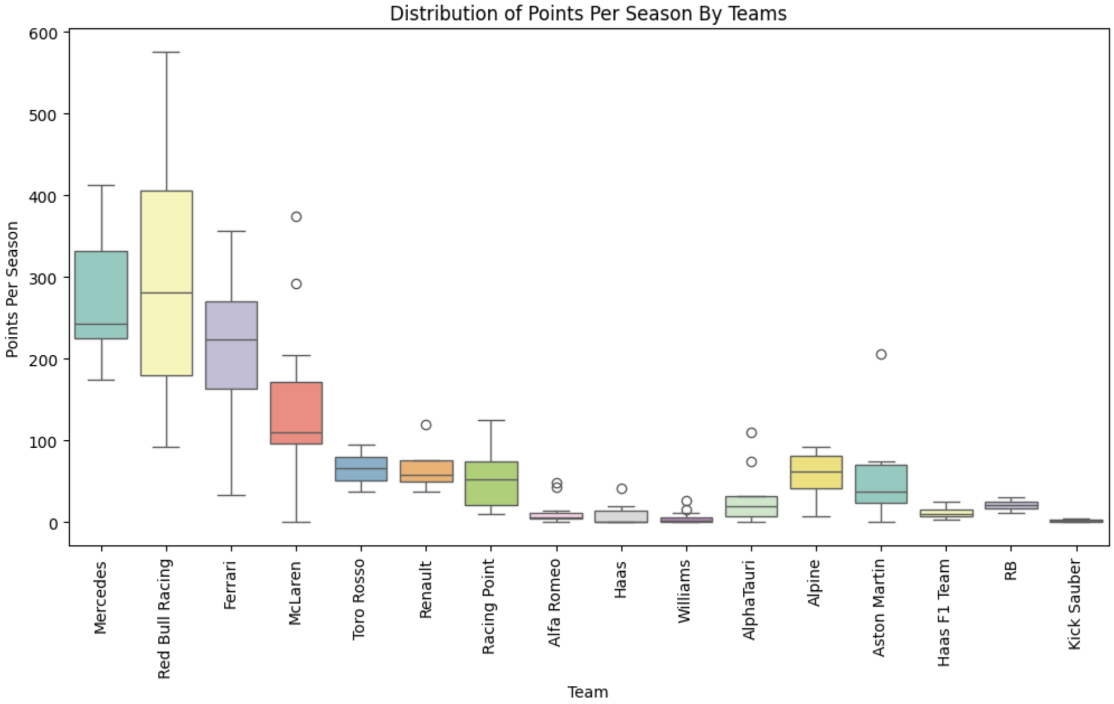


Just by looking at the visualization, there appears to be no trend between Grnad Prixed Entered and Points Earned in a Season. There is a slight positive trend in the middle of the plot but this trend dies down as the number of Grand Prixed Entered grows larger.

```
Pearson r: 0.8056, p-value: 2.4295e-31
```

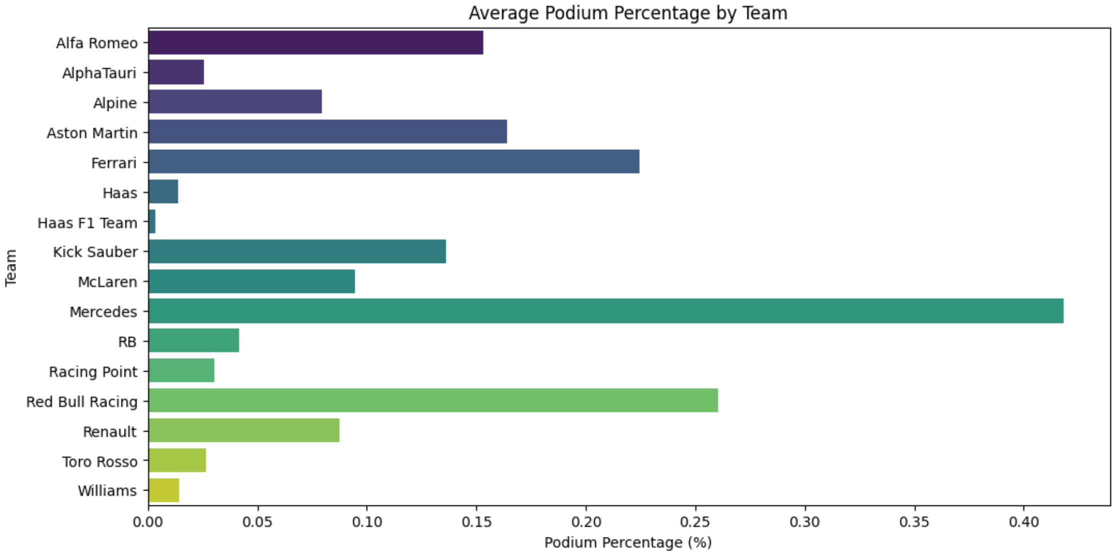
Here we have ran a pearson correlation test to see the correlation between the two variables. The calculated correlation coefficient is 0.2862 which is a low coefficient. This tells us there is little correlation between Grand Prix Entered and the points earned by a driver in a season.

Now we can see the point distribution across each team.



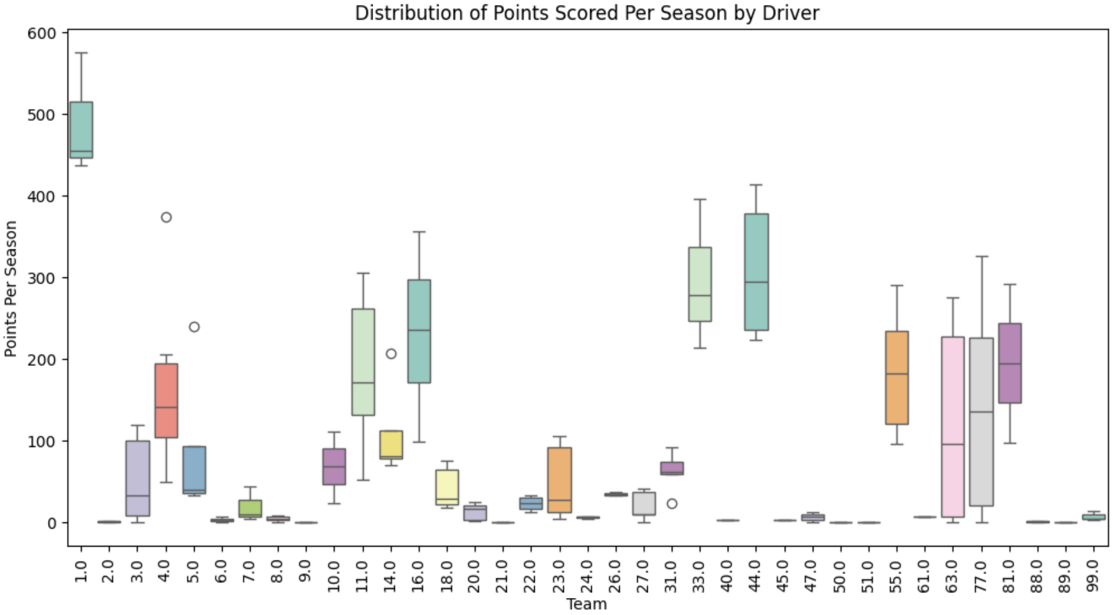
We can observe that Red Bull tends to be the dominant team, scoring more points on average than other teams. The next closest team is mercedes, but on average it has substantially less points that red bull does. This could show a trend of red bull being the most dominant team in this sport.

Now we want to see the podium percentage across each team.



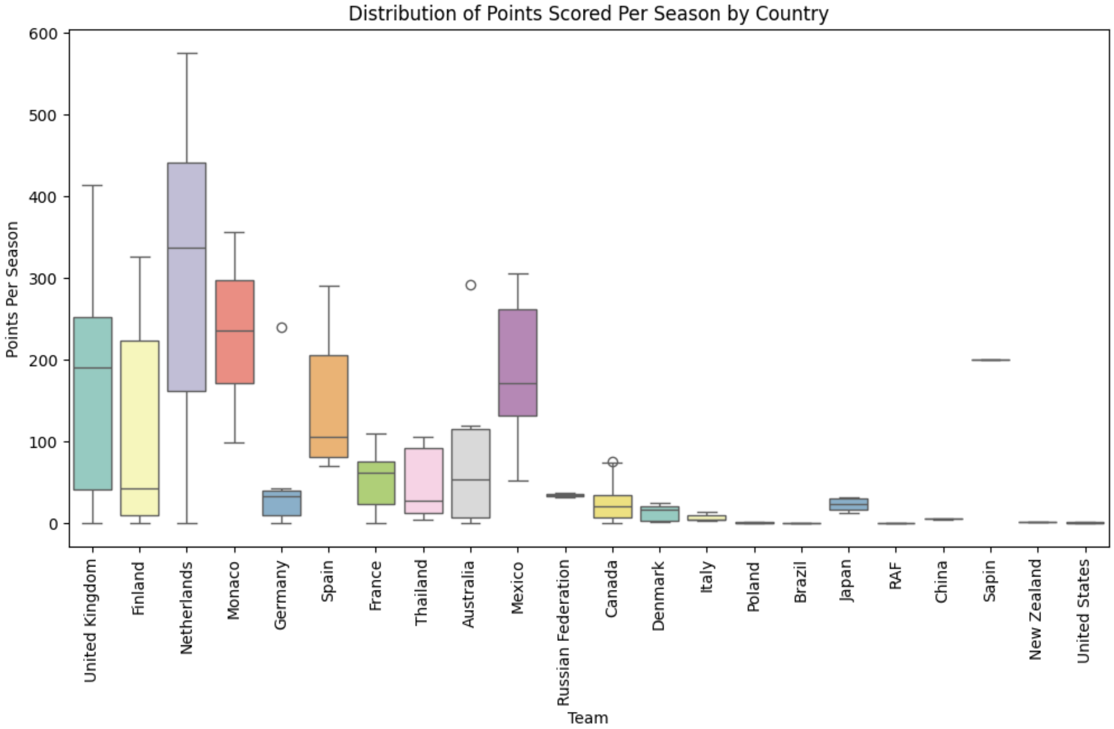
Here we observe that Mercedes actually has a much higher average podium percentage than red bull does. This is interesting because as we saw above, red bull racing had the highest points on average. Here, Ferrari also has a close podium percentage to red bull but Ferrari generally scores significantly less points than Red Bull does as we saw in the plot above.

Now we want to see the Distribution of Points scored per Season by Driver.



From this visualization we can observe that number 1, 44, 33, and 16 have been the most dominant drivers in the sport since 2019. We know that 1 and 33 are the same driver due to a number switch after winning the world championship. It makes sense that 1 has the highest points for a season because the number 1 is reserved for the world champion of the sport. Some other drivers that have shown significant improvement throughout the span of our data are number 16, 63, and 77.

Now we want to see the Distribution of Points scored per Season by Country to see which country has the best drivers.



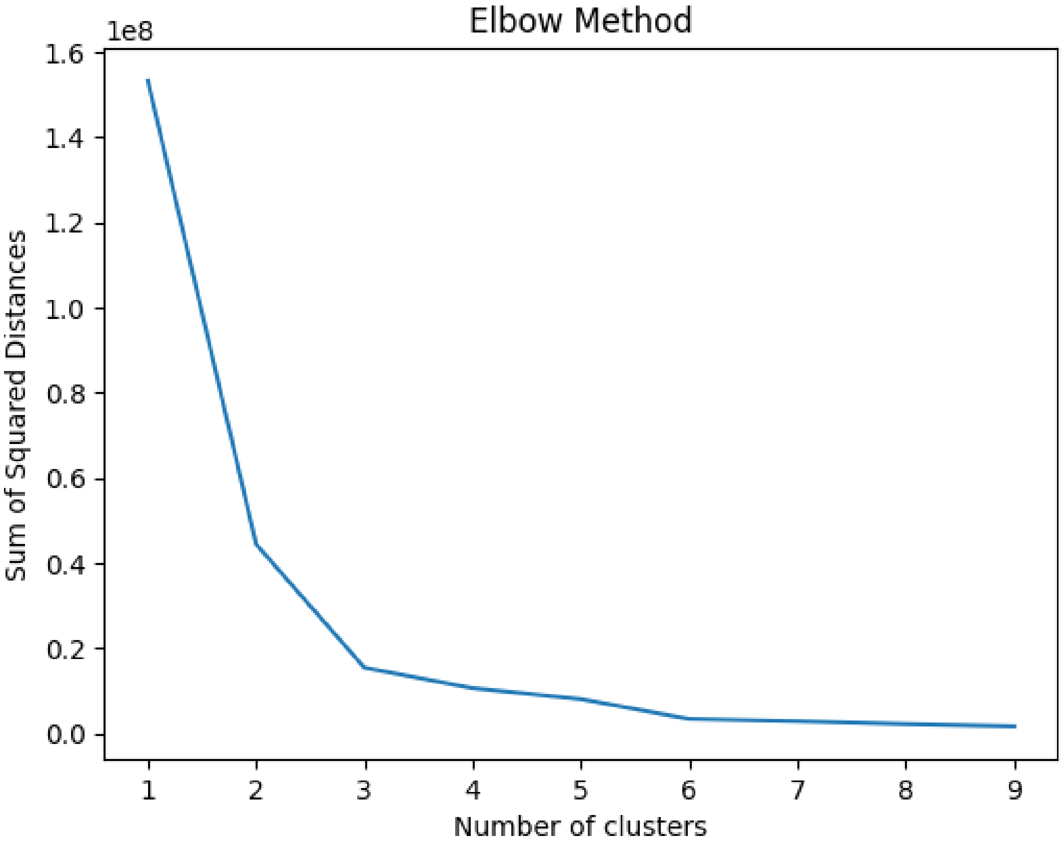
We can observe from the visualization that the Netherlands appears to be the dominant country with the best drivers. Some other notable countries are the United Kingdom, Monaco, and Mexico. Some countries that appear to be notoriously bad are the United States, New Zealand, Brazil, Poland, and RAF.

Starting K-Means Clustering to try and categorize the driver's into performance groups:

For our unsupervised Learning, we will use K-Means Clustering to categorize our drivers into different clusters based on the driver's podium percentage, their total number of points, and their historical highest grid position. We will use the K-Means function from the sklearn library.

To select the best number of clusters, we will use the elbow method to find our optimal k clusters.

For our clustering we will use the driver's average lap time, number of pole positions, and their podium percentage to try to cluster the drivers. We will try to find the best number of clusters using the Elbow method.



Based on the Elbow graph above, we see that a number of clusters equal to 3 seems to be ideal for us. This should make sure that we do not over fit our model.

Let's see how k-means with a k = 3 will cluster our data:

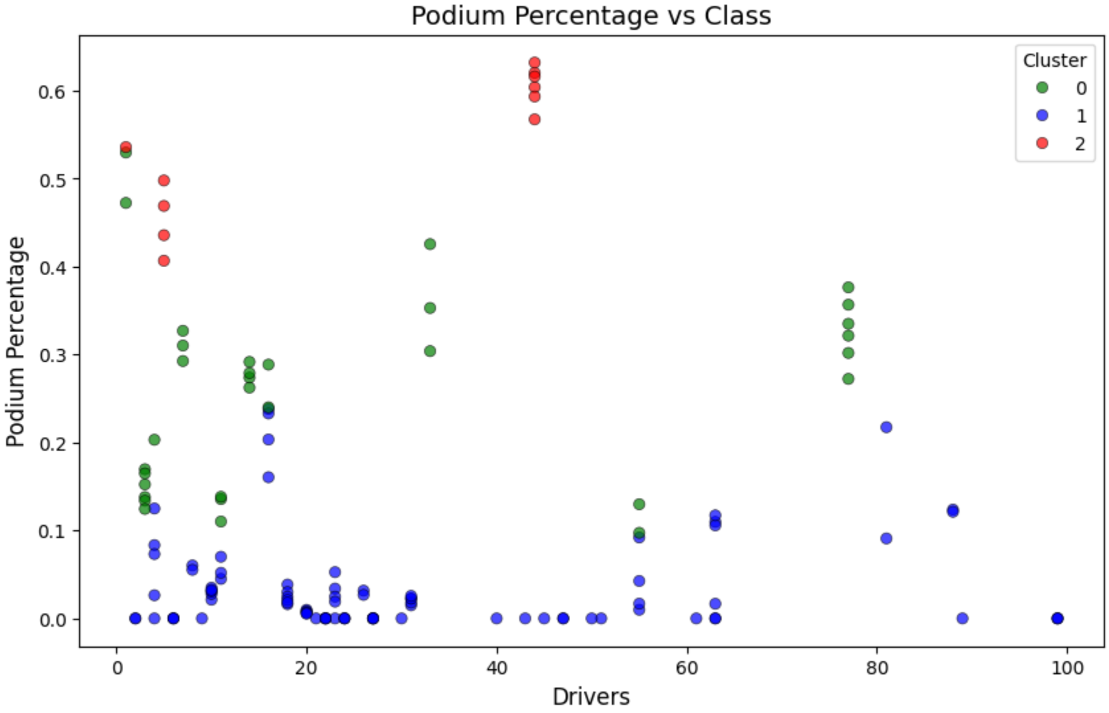
```
[17]:   Number             Team         Country  Podiums  Points  \
      0   44.0         Mercedes  United Kingdom      151  3431.0
      1   77.0         Mercedes         Finland       45  1289.0
      2   33.0  Red Bull Racing     Netherlands       31   948.0
      3   16.0          Ferrari          Monaco       10   303.0
      4    5.0          Ferrari         Germany      120  2985.0

         Grands Prix Entered  World Championships  Highest Grid Position  Year  \
      0                  250                    6                      1  2019
      1                  140                    0                      1  2019
      2                  102                    0                      1  2019
      3                   42                    0                      1  2019
      4                  241                    4                      1  2019

         Podium Percentage  Has WC  Cluster
      0           0.604000       1        2
```

```
1            0.321429       0         0
2            0.303922       0         0
3            0.238095       0         1
4            0.497925       1         2
```

Here we can see our kmeans algorithm has partitioned our clusters into 3 groups. We used the variables Podium Percentage, Points, and Highest Grid Position to partition the drivers into 3 seperate classes that would allow us to see how good of a driver they are: high class, medium class, and low class driver.

To understand how our data was clustered, we will explore the data further with the clusters in mind. We ended up with 3 clusters, labeled 0, 1, and 2 respectively.

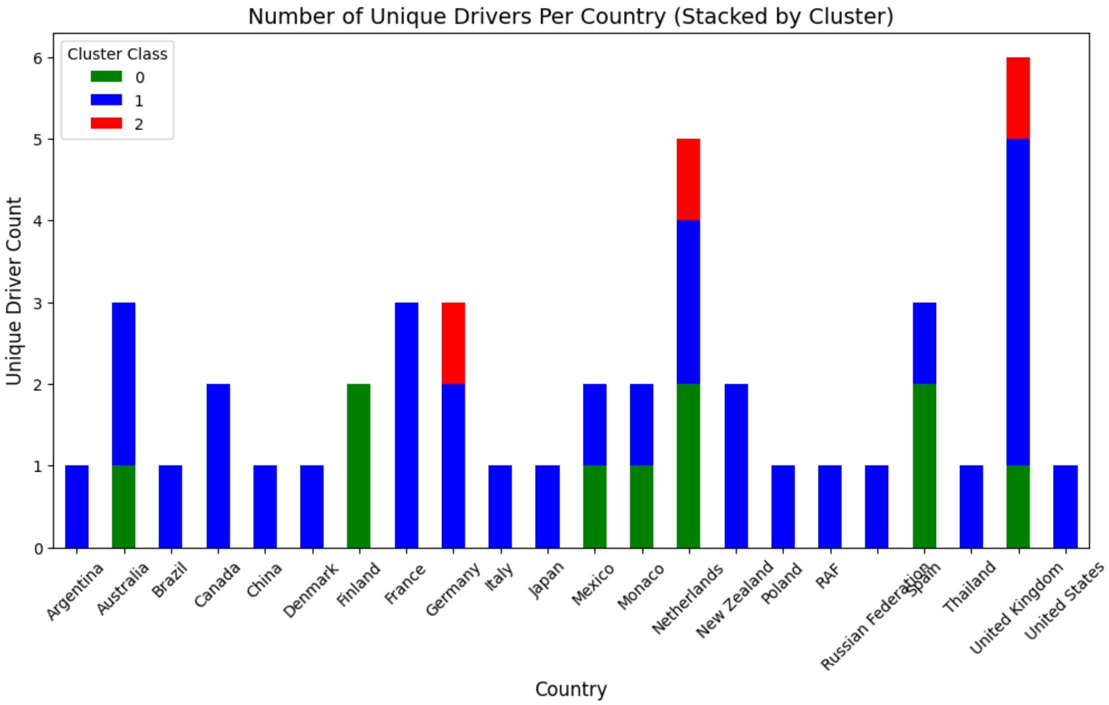
The figure below explores our drivers, their respective podium percentage, and the points are colored by class.



From this graph we can see that a cluster value of 0 corresponds to the lower class (color blue), 1 corresponds to the middle class (color green), 2 corresponds to the higher class (color red). We also notice that the number of drivers in each group decreases as we ge up in podium percentage. We can clearly see that around a 0 to 0.1 podium percentage, the majority of class 1 points are distributed. As we move up to a podium percentage 0.2 to around 0.4, we see a majority of the class 0 drivers. Lastly, with the fewest number of drivers in its class, class 2 contians drivers that have podium percentage of around 0.4 and higher.

Now, we will look at the relationship between driver and country, again with respect to their class as the coloring basis.

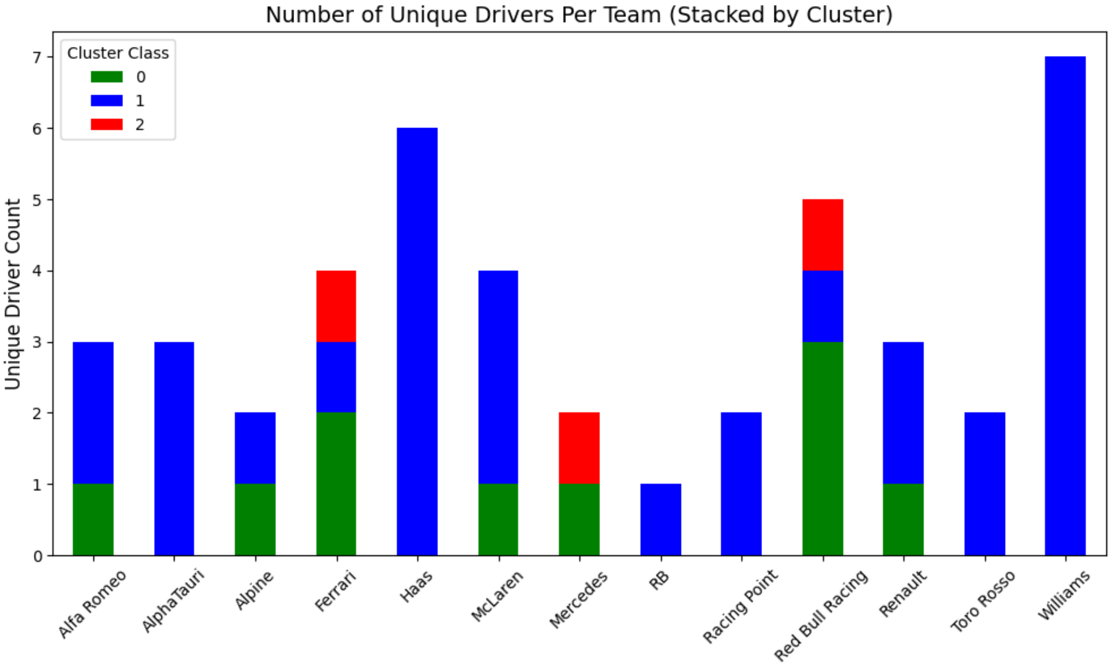
```
<Figure size 1200x600 with 0 Axes>
```



This here shows the distribtion of driver class across each country. We made sure to remove any duplicates where drivers were in the same class across multiple years. However, we included when the same driver appeared in multiple classes over different years. We can see that only Germany, Netherlands, and the United Kingdom has produced high class drivers. We also see that Finland has only ever produced low class drivers. Some other countries that produced middle class drivers are Australia, Canada, France, United Kingdom, and New Zealand.

Lastly, we will also explore the distribution of drivers across teams to see if there are any interesting findings here.

```
<Figure size 1200x600 with 0 Axes>
```



This graph shows the distribution of classes across the different teams in Formula 1. We made sure to remove any duplicates where drivers were in the same class across multiple years. However, we included when the same driver appeared in multiple classes over different years. Here we see that Williams and Haas have had the most drivers in our 6 year time frame for our data. They have also consistely produced middle class drivers, and have not produced any low class drivers. Red Bull Racing, has a high class driver but it also has the most low class drivers.

Lets see how much our variables influence the class prediction of our k-means model by looking at the correlation between class and each variable that we used to classify our drivers.

```
The correlation between Podium Percentage: 0.03535578966409758
The correlation between Previous Points Percentage: 0.10583439063861579
The correlation between Highest Grid Position Percentage: 0.18425136924366953
```

Next we wanted to see how lap times varied by each track, so we took the fastest lap times from each team for each track and compared them. As we see, out of the variables we used to classify our points, it seems that the variable 'Highest Grid Position' has about an 18.42% correlation with the classes asigned. It is followed by their total career points until that season (~10.58%), and lastly, their podium percentage (~3.54%).

Based on this, we can say that the variables we chose have a positive correlation, however a mild one, with the strongest correlation being a somewhate positive correlation, and with the smallest correlation being a pretty weak positive correlation.

KNN Regression – Supervised Learning

Now let's use K-Nearest-Neighbors to try and predict the driver's standings at the end of a season.

We will use multiple variables to help predict the standings. However, we want to modify some of our variables first. We will start by only selecting races that are in all of our years for consistency.

```
[22]:         Track  Position  No                          Team  Starting Grid  \
      0    Australia         1  77                      Mercedes              2
      1    Australia         2  44                      Mercedes              1
      2    Australia         3  33          Red Bull Racing Honda              4
      3    Australia         4   5                       Ferrari              3
      4    Australia         5  16                       Ferrari              5
      ..         …         …  ..                             …            …
      471  Abu Dhabi        13  24            Kick Sauber Ferrari             15
      472  Abu Dhabi        14  18  Aston Martin Aramco Mercedes             13
      473  Abu Dhabi        15  61                 Alpine Renault             17
      474  Abu Dhabi        16  20                   Haas Ferrari             14
      475  Abu Dhabi        17  30                   RB Honda RBPT             12

           Points Fastest Lap  Year Time/Retired  Net Position  Time (seconds)
      0      26.0         Yes  2019  1:25:27.325             1        5127.325
      1      18.0          No  2019      +20.886            -1        5148.211
      2      15.0          No  2019      +22.520             1        5149.845
      3      12.0          No  2019      +57.109            -1        5184.434
      4      10.0          No  2019      +58.230             0        5185.555
      ..      …          …   …          …            …              …
      471     0.0          No  2024       +1 lap             2        5217.325
      472     0.0          No  2024       +1 lap            -1        5217.325
      473     0.0          No  2024       +1 lap             2        5217.325
      474     0.0         Yes  2024       +1 lap            -2        5217.325
      475     0.0          No  2024          DNF            -5             NaN

      [1122 rows x 11 columns]
```

Now, after separating the consistent races, we will create a new column that will have the average fastest lap for each driver.

```
0      0.023364
1      0.004673
2      0.004673
3      0.009346
4      0.004673
         …
19     0.000000
20     0.000000
21     0.000000
22     0.000000
23     0.000000
Name: Avg Fastest Lap, Length: 132, dtype: float64
```

The output above shows us some of our average lap times, including some of the first and some of the last results in our data. We will also create a new variable called “Avg Net Position” which is contain the average net position of the drivers acrosss the years avaliable in our data.

```
0      0.014019
1      0.056075
2      0.046729
3     -0.023364
4     -0.168224
         …
19     0.060748
20     0.004673
21    -0.028037
22     0.000000
23     0.009346
Name: Avg Net Position, Length: 132, dtype: float64
```

Now that we have created our variabels, we can move onto creating and training our KNN Regression model.

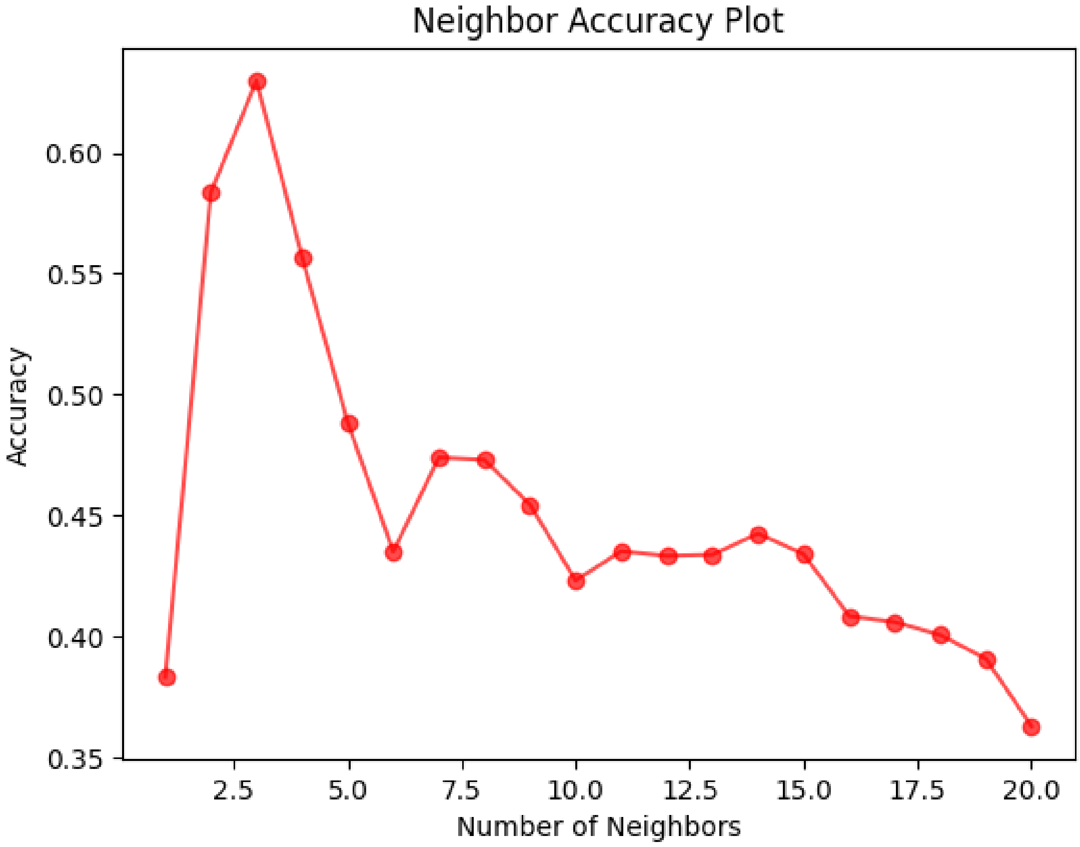
We will use a 70-30 split for our data, wehere 70% of the data will be used to train the mdoel and we will use the remaining 30% to test it. We use the variables "Avg Net Position", "Avg Fastest Lap", "Cluster", "Podium Percentage", "Has WC", "Points" from our unique_drivers data and we will be trying to predict the driver standings as found in the "Position" variables in the satandings data.

However, before we can use our model, we need to determine the optimal number of neighbors to compare to. To do this, we will chose the number of neighbors for which the model with the highest acuracy.

```
Score for k=1: 0.38325417994925515
Score for k=2: 0.5836315138897925
Score for k=3: 0.6298946790131489
Score for k=4: 0.556998568733329
Score for k=5: 0.4886734760262833
Score for k=6: 0.4351195252242679
Score for k=7: 0.47405596810319983
Score for k=8: 0.47305396525925447
Score for k=9: 0.4545203369179255
Score for k=10: 0.42321904885823947
Score for k=11: 0.43530561737220097
Score for k=12: 0.4333485134343895
Score for k=13: 0.43385665544776364
Score for k=14: 0.4426026216875404
Score for k=15: 0.4341292043458461
Score for k=16: 0.40852498210916666
Score for k=17: 0.4060243315334071
Score for k=18: 0.40067314511614394
Score for k=19: 0.3908646548317145
Score for k=20: 0.36292043458460754
```

To further look into how our model performes with a different k neighbors, we will plot our accuracy to the number of neighbors.

```
[26]: Text(0, 0.5, 'Accuracy')
```



From the graph above, we can see that for 3 neighbors, our model performes the best with around 63% accuracy, and afterwards we start losing accuracy relatively-consistently. This tells us that we want to use 3 neighbors for our model so that we get the optimal model we can.

Now that we have our model, lets see if it can accurately predict the top three drivers from the 2024 race year accurately. The correct standings, in order, would be Max Verstappen, Lando Norris, and Charles Leclerc.

```
Prediction for Max Verstappen: 10.0
Prediction for Lando Norris: 5.333333333333333
Prediction for Charles Leclerc: 2.3333333333333335

Actual for Max Verstappen: 1
Actual for Lando Norris: 2
Actual for Charles Leclerc: 3
```

From the output above, we see that our model is definitely not perfect. It predicted that Max would end up finishing in 10th place, Lando finishing around 5th, and Charles leading the group by ending in about 2nd place. Its closest prediction was Charles Leclerc, who actually finished in 3rd place that year, and its worst prediction would be Max Verstappen, who was predicted to be nine places away from his actual standings.

From the results we can definitely say that our model is not perfect as it does not predict accurately enough to be relied on. However, it can likely be improved with more data or maybe some different variables.

CONCLUSION:

Throughout the project, we have explored our F1 data, explored relationships between variables during our EDA, trained and optimized a K-Means clustering model, and used all of our data manipulation and knowledge to create, optimize, and run a K-Nearest-Neighbors regression model to try and predict F1 driver's standings.

We saw that our KNN regression model only got to around 63% accuracy after several tests to determine our best model. This means that arounrd 63% of our data will be correctly predicted into the right spot on the driver's standings.

Overall, we have learned that it is possible to create a KNN Regression model that will predict the drivers standings, however, with the data and resources we had, we were unable to create a better model. This does not mean, however, that a better model cannot be created. We believe that with more data, and some differnet models, such as gradient descent with linear regression, it is possible for us to create a better predictive model that will more accurately predict the drivers standings.

Contributions:

Dmitry Sorokin: (30%) - K-means Clustering - KNN Regression: Model and Analysis - Initial Data Cleaning and Pre-Processing - Presentation Cleaning - Proposal Document

Kyle Chahal: (30%) - KNN Feature Selection and Creation - Video Editing - Presentation - Proposal Document

Justin Shiu: (30%) - EDA - K-means Analysis - Project Presentation - Final Project Cleaning and Touch-Up

Justin An: (10%) - Project Slides