CS 105 Final Project Report

Introduction

Our project focuses on a video game called Valorant. It’s a first person shooting game run by Riot. Our goal is to predict players’ rank based on their headshot statistic.

Data

Here’s the data set we use.

region	name	tag	rating	damage_round	headshots	headshot_percent	aces	clutches	flawless	first_bloods	kills	deaths	assists
NA	ShimmyXD	#NA1	Radiant	135.8	992	24.9	0	140	80	161	1,506	1,408	703
NA	XSET Cryo	#cells	Radiant	170.3	879	28.3	2	122	94	316	1,608	1,187	206
NA	PuRelittleone	#yoruW	Radiant	147.5	720	24	3	117	59	216	1,115	1,064	267
NA	Boba	#0068	Radiant	178.2	856	37.3	3	83	49	235	1,134	812	157
NA	i love mina	#kelly	Radiant	149.8	534	24.4	2	71	38	137	869	781	213
NA	Decay	#GODK	Radiant	134.1	1,038	26	2	162	94	179	1,542	1,492	629
NA	Osias	#1212	Radiant	163.4	1,472	25.2	7	186	92	311	2,190	1,890	614
NA	Knights RIKU	#KRN	Radiant	153.3	510	17.5	2	112	64	215	1,246	1,066	341
NA	RaijuACE	#3131	Radiant	153.7	1,181	24.6	2	189	132	515	2,859	2,182	440
NA	dawn	#24k	Radiant	153.6	339	20.8	1	56	44	103	774	564	140
NA	100T Derrek	#100	Radiant	144.2	795	35.3	1	91	46	101	1,029	930	445
NA	acts017	#ttv	Radiant	137.8	868	29.2	0	146	80	226	1,397	1,358	379
NA	sam wow	#linda	Radiant	142.2	1,226	32.3	1	157	84	250	1,633	1,559	419
NA	kipp	#0002	Radiant	131.1	1,016	20.1	2	175	98	235	1,887	1,819	620
NA	Dinxx	#frog	Radiant	157	724	27.4	1	92	70	161	1,081	841	203
NA	svL	#DEMON	Radiant	151.9	681	22.4	1	68	60	104	1,046	875	534
NA	pnk	#drvn	Radiant	131.4	821	20.3	1	131	81	93	1,389	1,299	656
NA	jwr	#washd	Radiant	161.8	1,195	30.1	5	128	75	251	1,621	1,323	321
NA	j0HNL3	#wtF	Radiant	152	523	22.8	0	70	38	85	872	772	338
NA	LunatiK	#urdog	Radiant	130.1	1,702	19.7	3	294	205	458	2,902	2,979	952
NA	Summr	#TT7M	Radiant	143.7	953	28.5	4	123	64	232	1,290	1,323	336
NA	Ziehm	#2121	Radiant	142.3	1,830	21.6	3	275	157	452	3,195	2,988	947
NA	Darth Vader	#SVJ	Radiant	140.6	1,231	29.8	3	132	64	221	1,639	1,541	543
NA	chosen	#ari	Radiant	131.8	774	21.9	0	145	73	114	1,257	1,259	460
NA	krisspy	#fox	Radiant	133.4	888	28.4	0	117	74	119	1,230	1,182	545
NA	attach	#2uu	Radiant	127.7	690	19.1	0	126	72	154	1,182	1,300	657
NA	Hydra	#zzz	Immortal 3	136.5	1,273	22.7	1	138	133	196	1,973	1,737	779

Here’s the data set after cleaning. The first step was ensuring that we scaled the dataset down a bit as the original set contained over 80,000+ rows. We scaled this down to a sample of 5000. This dataset also contained a lot of features that weren't relevent to the point of this project and we dropped those columns from the dataset as well such as player name, region, and round scores. From there we assigned the categorical values of agent names and gun names to more easily perform statistical analysis on. We then cleaned up the data as a lot of columns were objects instead of ints as a way to use commas to signify 1000s.

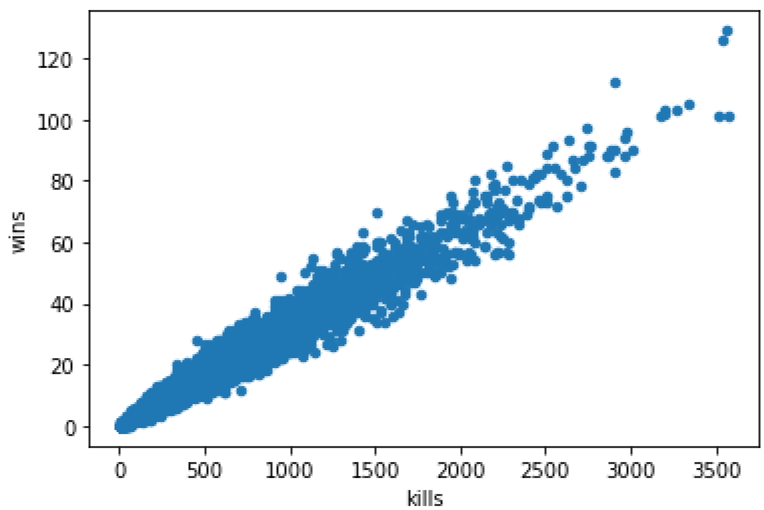
```
df.drop(df.columns[[0, 1, 2, 3, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 21, 22, 24, 25, 26, 29, 30, 31, 34, 35, 36]], axis=1, inpla
df.drop(df.index[5000:], axis = 0, inplace=True)
df.head()
```

	damage_round	headshots	headshot_percent	kills	wins	win_percent	agent_1	gun1_name	gun1_kills	gun2_name	gun2_kills	gun3_name	gun3_kil
0	135.8	992	24.9	1,506	59	59.6	Fade	Vandal	802	Phantom	220	Classic	14
1	170.3	879	28.3	1,608	52	65.8	Chamber	Vandal	689	Operator	226	Phantom	13
2	147.5	720	24.0	1,115	42	65.6	Yoru	Vandal	444	Phantom	231	Operator	10
3	178.2	856	37.3	1,134	32	62.8	Jett	Vandal	754	Sheriff	48	Phantom	3
4	149.8	534	24.4	869	32	62.8	Jett	Vandal	419	Spectre	65	Operator	6

EDA visualization

Here we have a scatterplot analyzing the amount of kills each player has gotten throughout the season compared to how many wins they have achieved throughout the season. We can see that there is a strong positive relationship between the amount of kills a player has and their win rate. From the graph we can conclude that the amount of games a player wins is heavily dictated by the amount of kills they are able to obtain during the game.

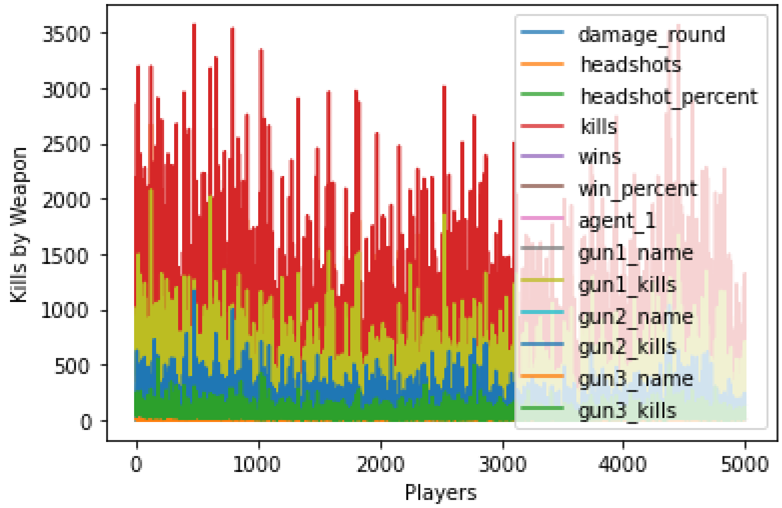
```
[5]: df.plot.scatter(x = 'kills', y = 'wins')

[5]: <AxesSubplot:xlabel='kills', ylabel='wins'>
```



Here we have a plot that shows the difference in the amount of kills a player gets as well as the subtype of weapons that tend to contribute the most to the their kill count. A players main gun (gun1) tends to contribute the most to their kill score while the players secondary weapons (gun2) come in second and their tertiary weapons (gun3) come in last.

```
df.plot(xlabel="Players", ylabel="Kills by Weapon")
```

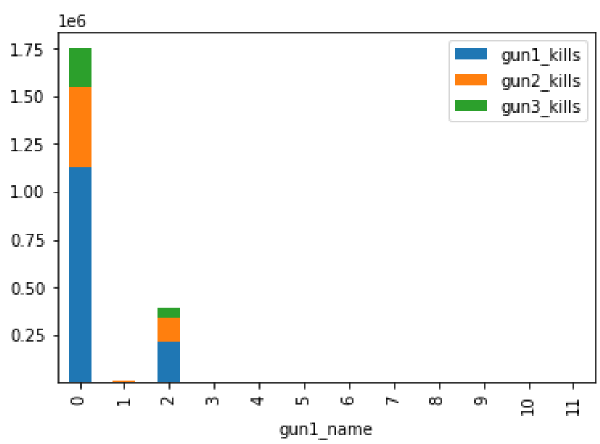
```
<AxesSubplot:xlabel='Players', ylabel='Kills by Weapon'>
```



Here we have another visualization to show how gun1 gets a majority of the kills, with gun2 second, and gun3 getting the least amount of kills.

```
df.groupby(["gun1_name"]).sum().plot(kind='bar',stacked=True,y=["gun1_kills","gun2_kills","gun3_kills"])
```

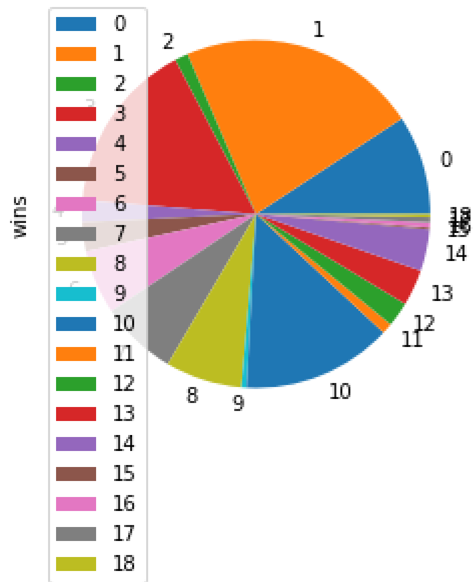
```
<AxesSubplot:xlabel='gun1_name'>
```



Here we analyze the amount of wins a player has alongside the agent they play that gets the most wins. Here it's easy to see that Chamber and Jett are easily the best availible agents to pick at the moment as the amount of wins they contribute to a players win rate dwarfs the other agents in the game with Reyna being the 3rd most winning agent pick.

```
df.groupby(['agent_1']).sum().plot(kind='pie',y='wins')
```

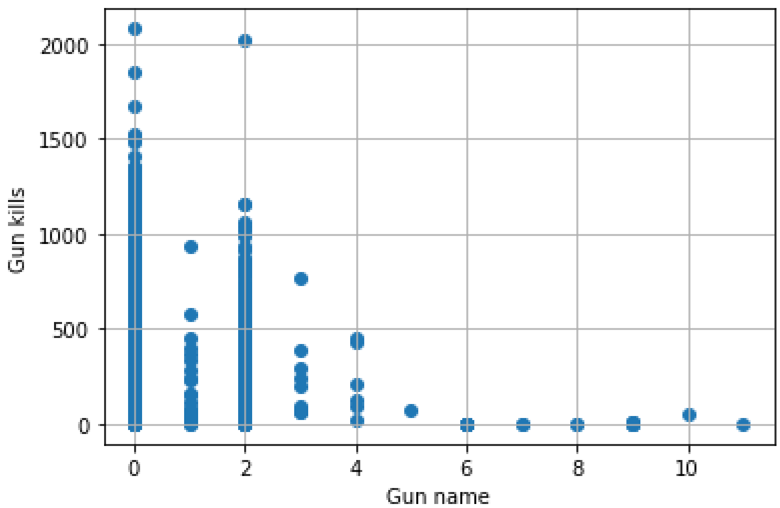
```
<AxesSubplot:ylabel='wins'>
```



Here we have a scatterplot comparing each individual weapon and the kills they get within the players main gun loadout. We can see from the data that the Vandal and the Phantom are by far the most popular weapons to use as well as the weapons that put up the most kills for most players. The Operator, the games primary sniper isn't far behind putting up the 3rd most amount of kills for players. This makes sense as they are the three most expensive weapons as well as the main weapons that the game intends players to use during most rounds. It also explains why the cheaper and weaker weapons tend to get less kills as they are only used on rounds where the team is trying to save money which is only during certain conditions of the game.

```python
import matplotlib.pyplot as plt

plt.scatter(df['gun1_name'], df['gun1_kills'])
plt.grid(True)
plt.xlabel("Gun name")
plt.ylabel("Gun kills")
```

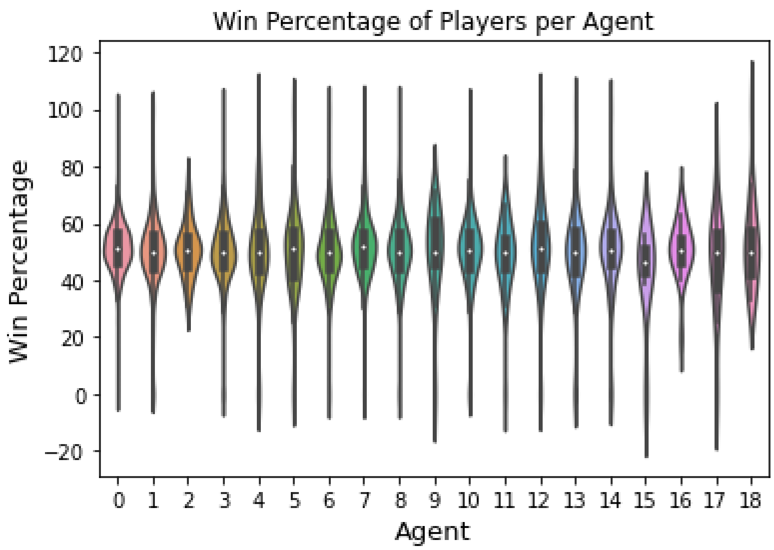
```
Text(0, 0.5, 'Gun kills')
```



Here we compare the win rate of players and the main agent they like to use. We can see that for the most part, agent win rates all fall within a similar range of each other with a few exceptions. The agents Killjoy and Phoenix have lower win rates that characters such as Cypher who has the highest win rate according to the graph. While its mostly similar enough to conclude that agent picks aren't as important when you get to a higher level of play, there are still agents that seem to perform better even in the toughest of competitions.

```
import seaborn as sns

violinPlot1 = sns.violinplot(x = df["agent_1"], y = df["win_percent"])
violinPlot1.set_xlabel("Agent", fontsize = 13)
violinPlot1.set_ylabel("Win Percentage", fontsize = 13)
plt.title("Win Percentage of Players per Agent")
```

```
Text(0.5, 1.0, 'Win Percentage of Players per Agent')
```



K-means clustering

Here our goal is to use the headshot statistics to create 4 main clusters to identify the best players in their rank. The intuition here is that as headshot ratings increase, the cluster will get tighter towards the top insintuating that less people are able to achieve such a high standard of play.

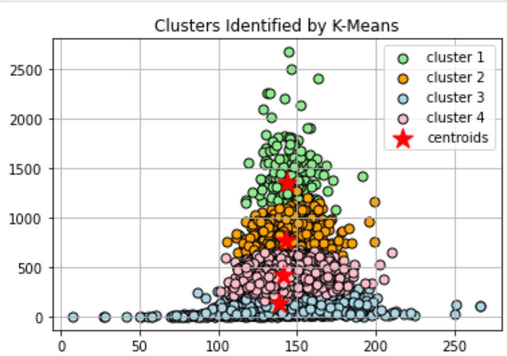
```
kmean_df=df
kmean_df.head()
```

	damage_round	headshots	headshot_percent	kills	wins	win_percent	agent_1	gun1_name	gun1_kills	gun2_name	gun2_kills	gun3_name	gun3_kills
0	135.8	992	24.9	1506	59	59.6	0	0	802	2	220	7	147
1	170.3	879	28.3	1608	52	65.8	1	0	689	1	226	2	137
2	147.5	720	24.0	1115	42	65.6	2	0	444	2	231	1	102
3	178.2	856	37.3	1134	32	62.8	3	0	754	9	48	2	36
4	149.8	534	24.4	869	32	62.8	3	0	419	6	65	1	64

```
km = KMeans(n_clusters=4, init='random', n_init=10, max_iter=300, tol=1e-04, random_state=0)
kmean_df=kmean_df.to_numpy()
y_km = km.fit_predict(kmean_df)
y=y_km
print(y)
print(km.labels_)
```

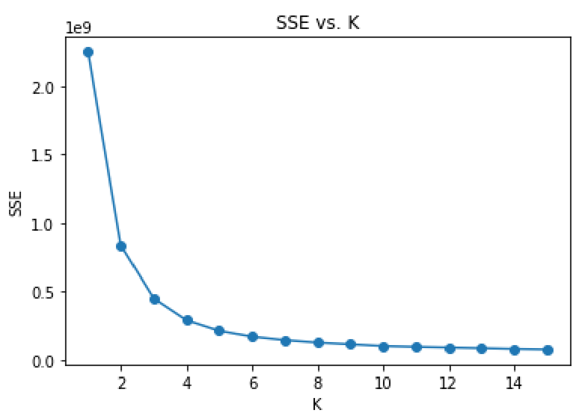
```
[1 1 1 ... 1 1 3]

[1 1 1 ... 1 1 3]
```

```
#plot of clusters & centroids
plt.title("Clusters Identified by K-Means")
plt.scatter(kmean_df[y == 0, 0], kmean_df[y == 0, 1],s=50, c='lightgreen',marker='o', edgecolor='black',label='cluster 1')
plt.scatter(kmean_df[y == 1, 0], kmean_df[y == 1, 1],s=50, c='orange',marker='o', edgecolor='black',label='cluster 2')
plt.scatter(kmean_df[y == 2, 0], kmean_df[y == 2, 1],s=50, c='lightblue',marker='o', edgecolor='black',label='cluster 3')
plt.scatter(kmean_df[y == 3, 0], kmean_df[y == 3, 1],s=50, c='pink',marker='o', edgecolor='black',label='cluster 4')
plt.scatter(km.cluster_centers_[:, 0], km.cluster_centers_[:, 1],s=250, marker='*',c='red', edgecolor='red',label='centroids')
plt.legend(scatterpoints=1)
plt.grid()
plt.show()
```



We can see that through the use of k-means clustering, we were able to successfully cluster the headshot data into 4 distinct player rankings that confirm our intuition. With cluster 1 being the best players with the highest headshot ratings and cluster 4 being the lower ranked players with lower headshot ratings. We can see that the graph and clusters tighten up as headshot rating increases implying that there are a lower amount of players able to achieve that level of play. We see that cluster 1 has a few standout outliers that go far beyond their predecessors, clusters 2 and 3 are fairly tight knit with most players being very similar in skill and cluster 4 has the most variance with both low and high outliers at the lower ranks.

```
##elbow method
sse = []
##runnning for K 1:15
for i in range(1, 16):
    km = KMeans(
        n_clusters=i, init='random',
        n_init=10, max_iter=300,
        tol=1e-04, random_state=0)
    km.fit(kmean_df)
    sse.append(km.inertia_)

#plot of SSE vs. K
plt.plot(range(1, 16), sse, marker='o')
plt.title("SSE vs. K")
plt.xlabel('K')
plt.ylabel('SSE')
plt.show()
```



Here we confirm out choice of K using the elbow method. We can see that 4 was a solid choice for K compared to most of the other available choices.

Analyze and Conclusion

We were able to successfully run EDA on multiple aspects of the data as well as implement k-means clustering to group up players within their skill level based on their headshot rating. This shows that it is possible to further improve skill-based matchmaking to ensure that players are matched up with other players around their skill level to avoid the problems plaguing current competitive gaming such as smurfs and matchmaking on insufficient conditions. We can conclude that based on the clusters, as a player improves their headshot rating, in other words, their skill, they can be successfully grouped among other individuals who are around their skill level to ensure the playing field is even and tough!

Contribution:

- Isiah:
  - Data cleaning
  - Data analyze
  - K-means clustering
- Amir:
  - Data analyze
  - K-means clustering
  - Edit slides/ report
- Regan:
  - EDA
  - Data cleaning
  - K-means clustering
- Samantha:
  - K-means clustering
  - EDA
  - Data analyze

- Xirong:
  - Data analyze
  - Edit slides/report
  - EDA