CS 105 Final Project

Presented by
Isiah Montalvo
Amir Shirazi
Regan Jia
Xirong Xie
Samantha Nino

Project description

We are going to analyse a dataset about Valorant's leaderboards. We will use the data to predict a player's rank based on their headshot statistic with the K-means clustering method.

Techniques we use

- K-means clustering
    - Elbow Method
- Data cleaning

Dataset 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

- Valorant Leaderboard Statistics
- We will need the stats of players, including their rank, headshots, headshot percentage, etc
- We can find these in the data set
- When we do the data cleaning, we replace “,” with “ ”, and we also have different numbers represent for different gun names. In this case, we could just use numbers, which will be easier to make graphs.

Dataset we use

```
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

- Cleaned Data Set

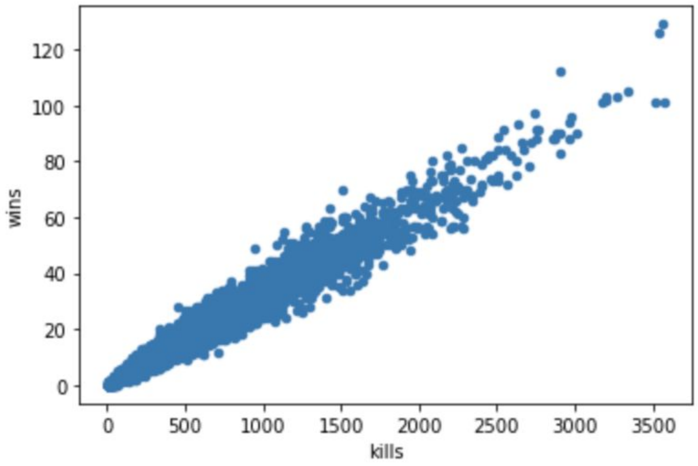
Data Cleaning

```
df['gun1_name'].replace(['Vandal', 'Operator', 'Phantom', 'Judge', 'Odin',␣
↪'Guardian', 'Spectre', 'Classic', 'Ghost', 'Sheriff', 'Marshal', 'Shorty',␣
↪'Bulldog', 'Frenzy', 'Bucky', 'Ares', 'Stinger'],␣
↪[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16], inplace=True)
df['gun2_name'].replace(['Vandal', 'Operator', 'Phantom', 'Judge', 'Odin',␣
↪'Guardian', 'Spectre', 'Classic', 'Ghost', 'Sheriff', 'Marshal', 'Shorty',␣
↪'Bulldog', 'Frenzy', 'Bucky', 'Ares', 'Stinger'],␣
↪[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16], inplace=True)
df['gun3_name'].replace(['Vandal', 'Operator', 'Phantom', 'Judge', 'Odin',␣
↪'Guardian', 'Spectre', 'Classic', 'Ghost', 'Sheriff', 'Marshal', 'Shorty',␣
↪'Bulldog', 'Frenzy', 'Bucky', 'Ares', 'Stinger'],␣
↪[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16], inplace=True)

df['headshots'] = df['headshots'].str.replace(",","",regex = False)
df['headshots'] = df['headshots'].astype(int)
df['kills'] = df['kills'].str.replace(",","",regex = False)
df['kills'] = df['kills'].astype(int)
df['gun1_kills'] = df['gun1_kills'].str.replace(",","",regex = False)
df['gun1_kills'] = df['gun1_kills'].astype(int)
df['gun2_kills'] = df['gun2_kills'].str.replace(",","",regex = False)
df['gun2_kills'] = df['gun2_kills'].astype(int)
df.dtypes
```

EDA

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

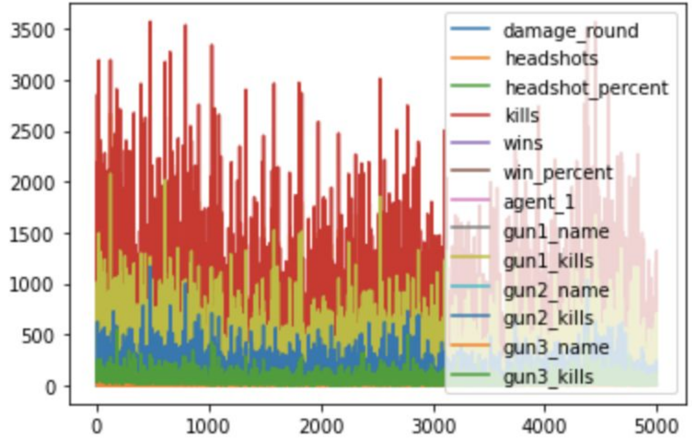
```
[6]: <AxesSubplot:xlabel='kills', ylabel='wins'>
```



- Scatter plot that shows the correlation between the amount of kills and wins a player has
- With this graph, we can see that the more kills a players has, the more wins they receive

EDA

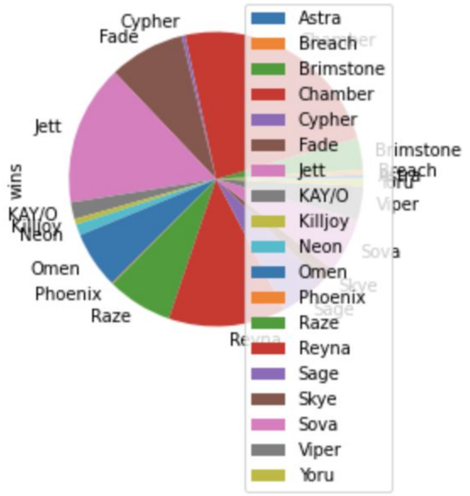
```
[6]: df.plot()

[6]: <AxesSubplot:>
```



EDA

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

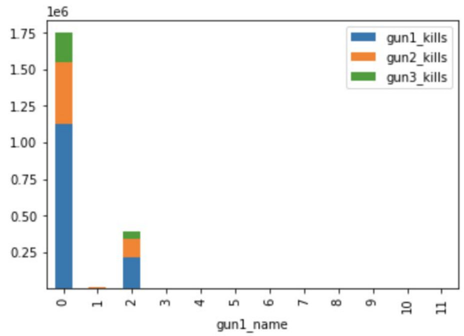
```
[13]: <AxesSubplot:ylabel='wins'>
```



- Pie graph that shows the total amount of wins each agent has
- Chamber, Jett, and Reyna has the most wins, Jett, and Reyna has the most wins

EDA

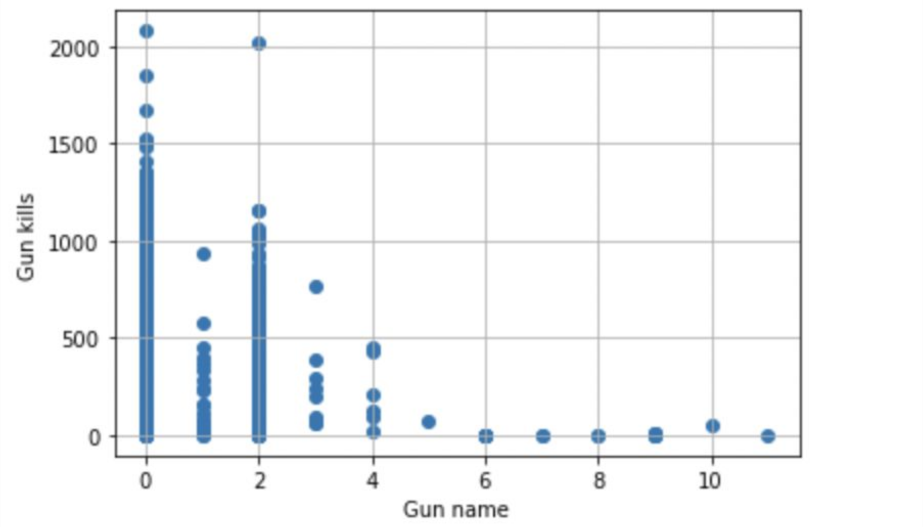
```
[32]: df.groupby(["gun1_name"]).sum().plot(kind='bar',stacked=True,y=["gun1_kills","gun2_kills","gun3_kills"])
[33]: <AxesSubplot:xlabel='gun1_name'>
```



0: Vandal
1: Operator
2: Phantom
3: Judge
4: Odin
5: Guardian
6: Spectre
7: Classic
8: Ghost
9: Sheriff
10: Marshal
11: Shorty

- Stacked bar graph to show which guns resulted in the most kills
- We can see that Vandal(0) and Phantom(2) was the most used guns with the most kills

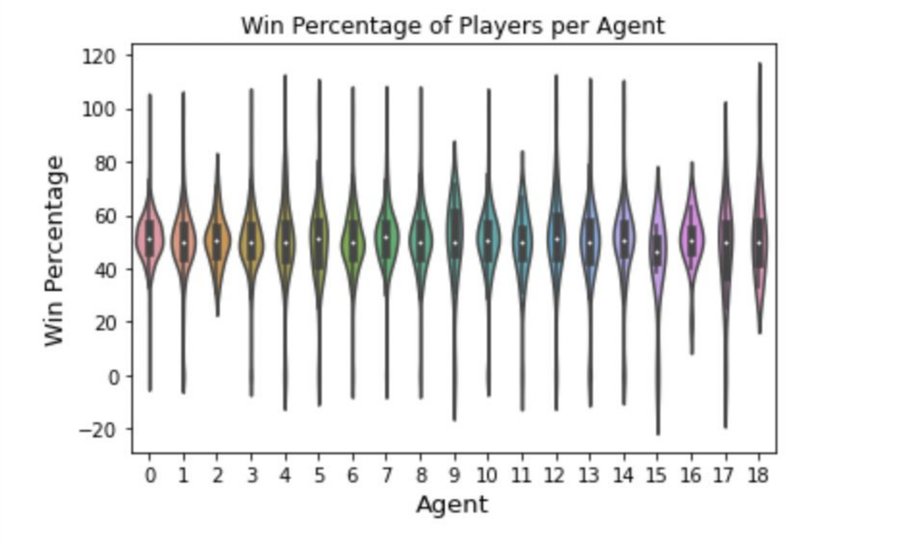
EDA




```
df['gun1_name'].replace(['Vandal', 'Operator', 'Phantom', 'Judge', 'Odin', 'Guardian', 'Spectre', 'Classic', 'Ghost', 'Sheriff', 'Marshal', 'Shorty'], [0,1,2,3,4,5,6,7,8,9,10,11], inplace=True)
```

- Scatter plot that shows the gun kills perspective to the gun name
- Each point is an individual using that gun name (x-axis) and the amount of kills with that gun (y-axis)
- Attached below is the respective numerical gun name to the actual gun name

EDA



- Violin plot displaying the agents that players play with the respective win rates
- White circle indicates median
- Attached below is the respective numerical agent name to the actual agent name

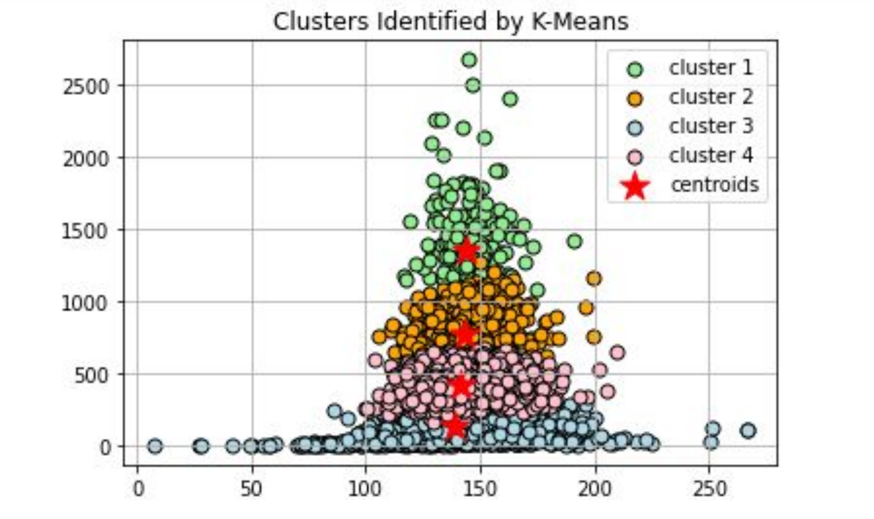
```
df['agent_1'].replace(['Fade', 'Chamber', 'Yoru', 'Jett', 'Sage', 'KAY/O', 'Sova', 'Raze', 'Omen', 'Breach', 'Reyna', 'Neon', 'Skye', 'Viper', 'Brimstone', 'Phoenix', 'Astra', 'Killjoy', 'Cypher'], [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18], inplace=True)
```

K-Means Clustering

- Performed K-Means clustering to group players of similar Rank.
- Set K=4, since there are four ratings Radiant, Immortal 3, Immortal 2, Immortal 1

```
In [11]: kmean_df=df
         ##kmean_df.head()

In [12]: 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]
```

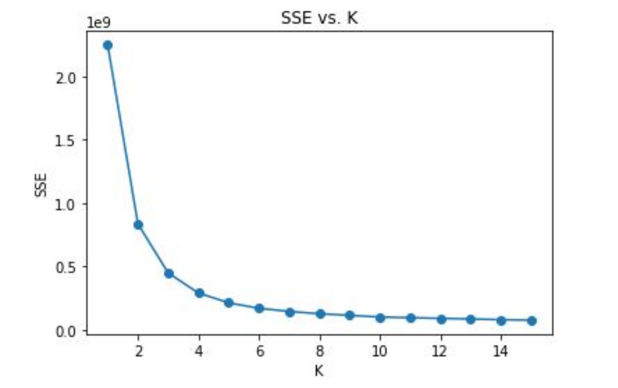
K-Means Clustering



- 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

- Visualization method to estimate the optimal number of K Clusters.
- Using within-clusters Sum of Square Error to measure the variability.
- We pick the optimal K by finding the elbow in the plot.
- We can notice that the elbow is at K=4.



Questions for you

Q1: What do the clusters represent and what pattern did the data form as the headshot rating increased

Q2: What agent had the highest contribution to a players wins?

Q3: How could clustering be used to improve matchmaking in competitive games?

Thank you for watching this video!