FIFA PLAYER CLUSTERING

datalab

This is a FIFA 23 Clustering Project - The goal of this project is to:

- take hundreds of player data (unlabeled) and to cluster it as best as we can regarding the following attributes: Player Overall, Player Age, Player Potential

```
# Uploading the dataset and getting insights as to what it looks like

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
df = pd.read_csv('Fifa 23 Players Data.csv')

df
```

	Known As	Full Name		P.	Value(in ...	Positions Pl...
0	L. Messi	Lionel Messi	91	91	54000000	RW
1	K. Benzema	Karim Benzema	91	91	64000000	CF,ST
2	R. Lewandowski	Robert Lewandowski	91	91	84000000	ST
3	K. De Bruyne	Kevin De Bruyne	91	91	107500000	CM,CAM
4	K. Mbappé	Kylian Mbappé	91	95	190500000	ST,LW
5	M. Salah	Mohamed Salah	90	90	115500000	RW
6	T. Courtois	Thibaut Courtois	90	91	90000000	GK
7	M. Neuer	Manuel Neuer	90	90	13500000	GK
8	Cristiano Ronaldo	C. Ronaldo dos Santos Aveiro	90	90	41000000	ST
9	V. van Dijk	Virgil van Dijk	90	90	98000000	CB
10	H. Kane	Harry Kane	89	89	105500000	ST
11	Neymar Jr	Neymar da Silva Santos Jr.	89	89	99500000	LW
12	H. Son	Heung Min Son	89	89	101000000	LW,LM
13	Casemiro	Carlos Henrique Venancio Casimiro	89	89	86000000	CDM
14	J. Oblak	Jan Oblak	89	91	85500000	GK
15	S. Mané	Sadio Mané	89	89	99500000	LM,CF

Rows: 1,123 ⚠ truncated from 18,539 rows Expand Table

In order to give an accurate representation of clusters, we want to get a random sample of players. We're choosing the number 500 because it is large enough to show where the different clusters are, but not too big where there is diluted cluster quality

```
df = df.sample(n=500, random_state=28)

df
```

	Known As	Full Name		P.	Value(in ...	Positions Pl...	Best
15	S. Mané	Sadio Mané	89	89	99500000	LM,CF	LM
33	H. Lloris	Hugo Lloris	87	87	9000000	GK	GK
30	A. Rüdiger	Antonio Rüdiger	87	88	73500000	CB	CB
141	A. Davies	Alphonso Davies	84	89	60500000	LB,LM	LM
103	J. Grealish	Jack Grealish	84	84	46000000	LW,LM,CAM	LW
151	M. Ødegaard	Martin Ødegaard	84	89	63500000	CAM,CM	CAM
210	Anderson Talisca	Anderson Souza Conceição	82	82	31000000	CF,ST,CAM	CF
266	Isco	Francisco Román Alarcón Suárez	82	82	29500000	CAM,LW,CM	CAM
229	B. Saka	Bukayo Saka	82	89	60500000	RM,LM	LM
245	J. Pickford	Jordan Pickford	82	84	26000000	GK	GK
336	J. Vertonghen	Jan Vertonghen	81	81	7000000	CB	CB
288	N. Zaniolo	Nicolò Zaniolo	81	88	53000000	CF,RW,RM	CAM
499	A. Romagnoli	Alessio Romagnoli	80	80	19500000	CB	CB
471	A. Rebić	Ante Rebić	80	80	21000000	LW,LM,ST	ST
376	F. Neuhaus	Florian Neuhaus	80	84	29000000	CM	CM
487	W. Benítez	Walter Benítez	80	82	17500000	GK	GK

Rows: 500 Expand Table

```
# Selecting the three attributes that we want to see - Overall, Potential and Age

df = df[['Overall', 'Potential', 'Age']]

df
```

index	Overall	Potential	Age
15801	59	59	39
2291	74	74	38
4976	70	70	37
10776	64	64	37
17414	55	55	37
16705	57	57	36
10364	65	65	36
11475	64	64	35
1386	76	76	35
33	87	87	35
9399	66	66	35
4473	70	70	35
1725	75	75	35
336	81	81	35
2601	73	73	35
11590	64	64	35

Rows: 500 Expand Table

This code standardizes the features of the df dataset using StandardScaler, transforming them into a new dataframe fifa_preprocessed with scaled values

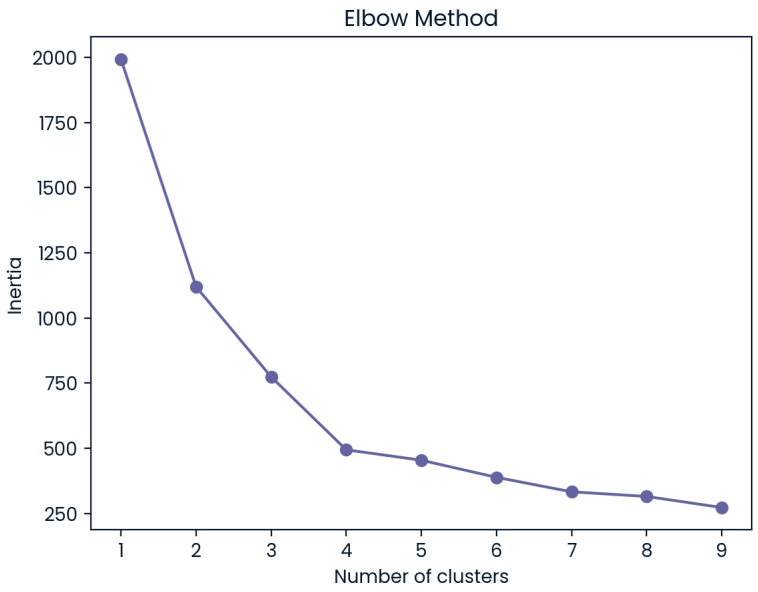
```
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df)
fifa_preprocessed = pd.DataFrame(data=scaled_data,columns=df.columns)
fifa_preprocessed
```

index	Overall	Potential	Age
0	2.0153578138	1.3444881416	0.3475044216
1	-0.5810092524	0.2926981336	-1.3332908462
2	0.2844464364	-0.0078132972	-0.0726943953
3	-1.8791927855	-0.1580690126	-1.5433902547
4	-0.2925240228	-0.9093475897	0.5576038301
5	-0.4367666376	-1.2098590206	1.1879020555
6	-0.8694944819	-0.308324728	-0.4928932123
7	1.7268725842	1.494743857	0.1374050131
8	-0.7252518672	-0.1580690126	-0.7029926208
9	-0.2925240228	-0.4585804435	-0.0726943953
10	-0.2925240228	-1.0596033051	2.0282996895
11	-0.2925240228	-0.9093475897	0.7677032386
12	-0.7252518672	-0.7590918743	-0.7029926208
13	0.8614168955	0.1424424182	0.7677032386
14	-0.0040387932	1.3444881416	-1.5433902547
15	0.8614168955	0.5932095645	0.1374050131

Rows: 500 Expand Table

Here we are using the Elbow Method - We can see that the ideal number of clusters in this scenario is 4.

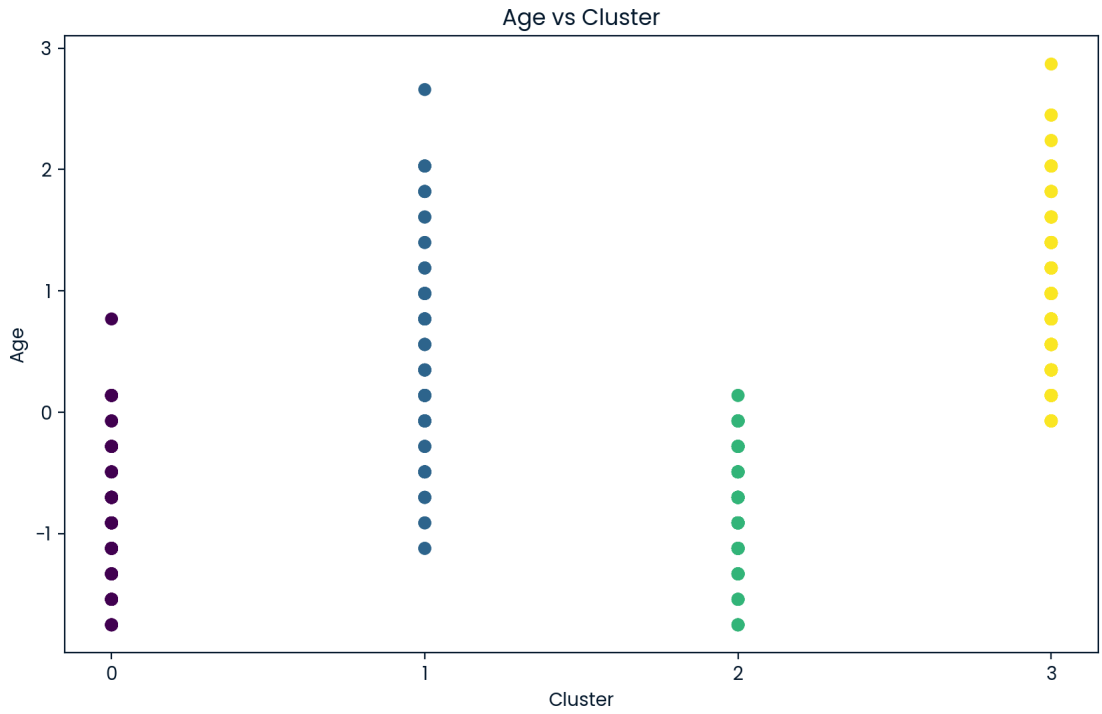
```
 ## Using the Elbow Plot to see the inertia for different values of k, helping to determine the optimal number of clusters for K-means clustering by identifying the "elbow" point in the graph

inertia = []
for k in range(1, 10):
    kmeans = KMeans(n_clusters=k, random_state=42).fit(fifa_preprocessed)
    inertia.append(kmeans.inertia_)
plt.plot(range(1, 10), inertia, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method')
plt.show()
```



```
# We assign a cluster to every data point and create a new column showing this in our dataframe

kmeans = KMeans(n_clusters=4, random_state=42)
kmeans.fit(fifa_preprocessed)
fifa_preprocessed['Cluster'] = kmeans.labels_
```

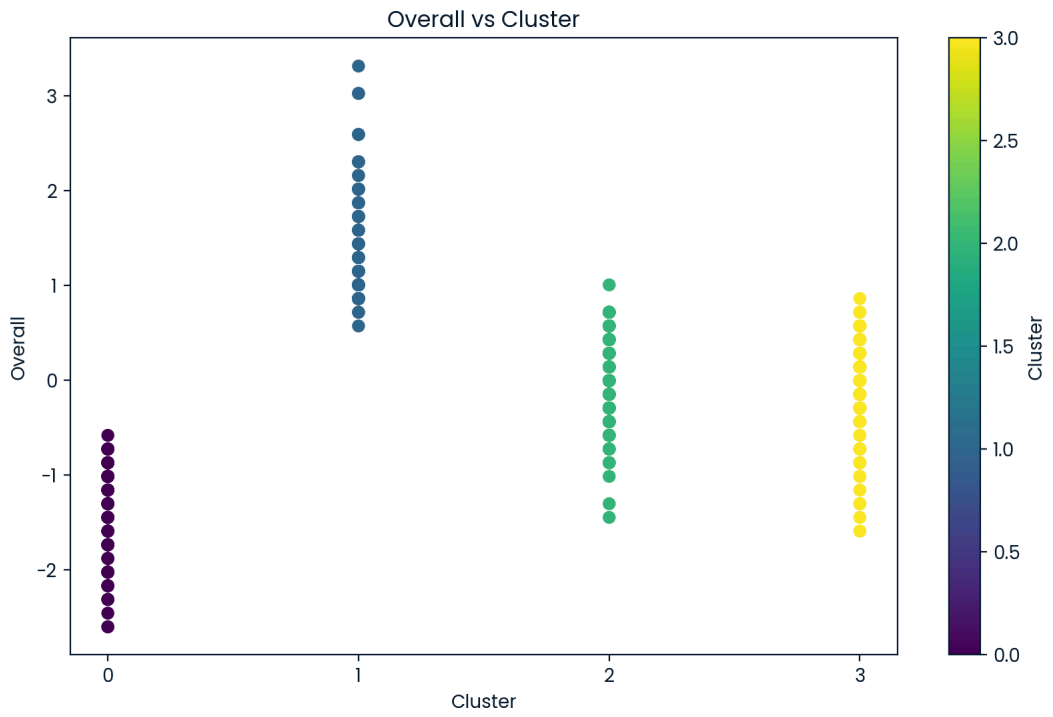
Graph showing our 4 clusters and how they are distrubuted along age. Negative values for age are younger and positive values for age are older.

```
plt.figure(figsize=(10, 6))
plt.scatter(fifa_preprocessed['Cluster'], fifa_preprocessed['Age'], c=fifa_preprocessed['Cluster'], cmap='viridis')


plt.xlabel('Cluster')
plt.ylabel('Age')
plt.xticks(range(fifa_preprocessed['Cluster'].min(), fifa_preprocessed['Cluster'].max() + 1))  # Set x-ticks to cluster numbers
plt.title('Age vs Cluster')

plt.show()
```



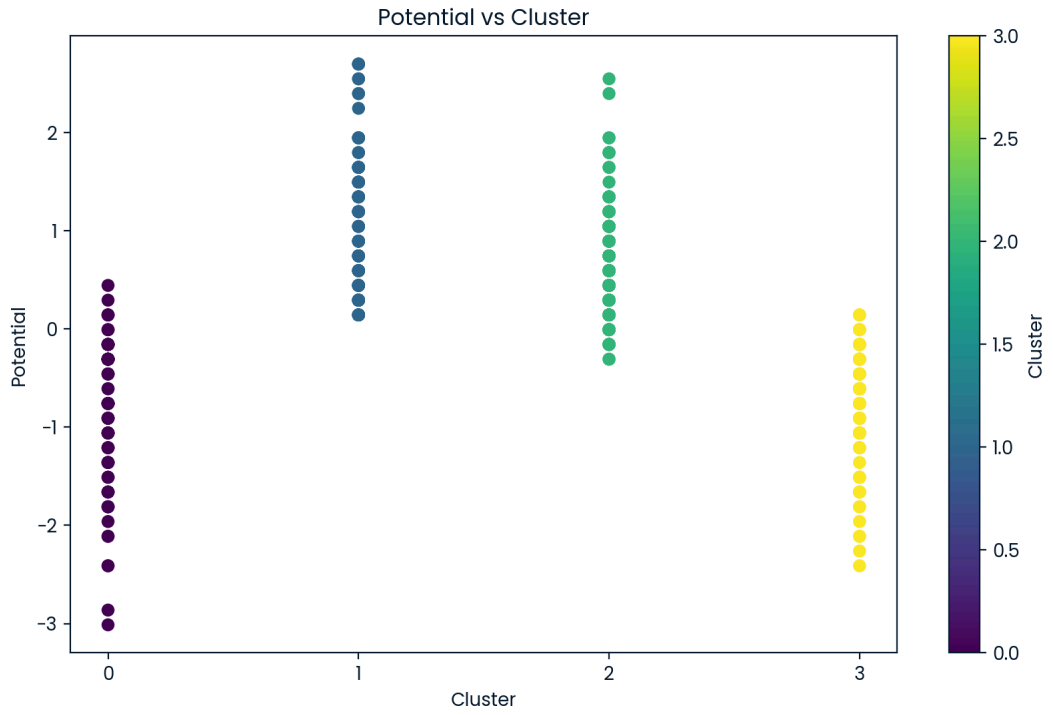
Graph showing our 4 clusters and how they are distrubuted along overall rating . Negative values for overall have a lower rating and positive values have a higher rating

```
plt.figure(figsize=(10, 6))
plt.scatter(fifa_preprocessed['Cluster'], fifa_preprocessed['Overall'], c=fifa_preprocessed['Cluster'], cmap='viridis')

# Add labels and title
plt.xlabel('Cluster')
plt.ylabel('Overall')
plt.xticks(range(fifa_preprocessed['Cluster'].min(), fifa_preprocessed['Cluster'].max() + 1))  # Set x-ticks to cluster numbers
plt.title('Overall vs Cluster')

# Show the color bar to indicate the cluster color mapping
plt.colorbar(label='Cluster')

# Show the plot
plt.show()
```



 # Graph showing our 4 clusters and how they are distrubuted along potential. Negative values for potential show a lower future potential rating and positive values show a higher future potetial rating

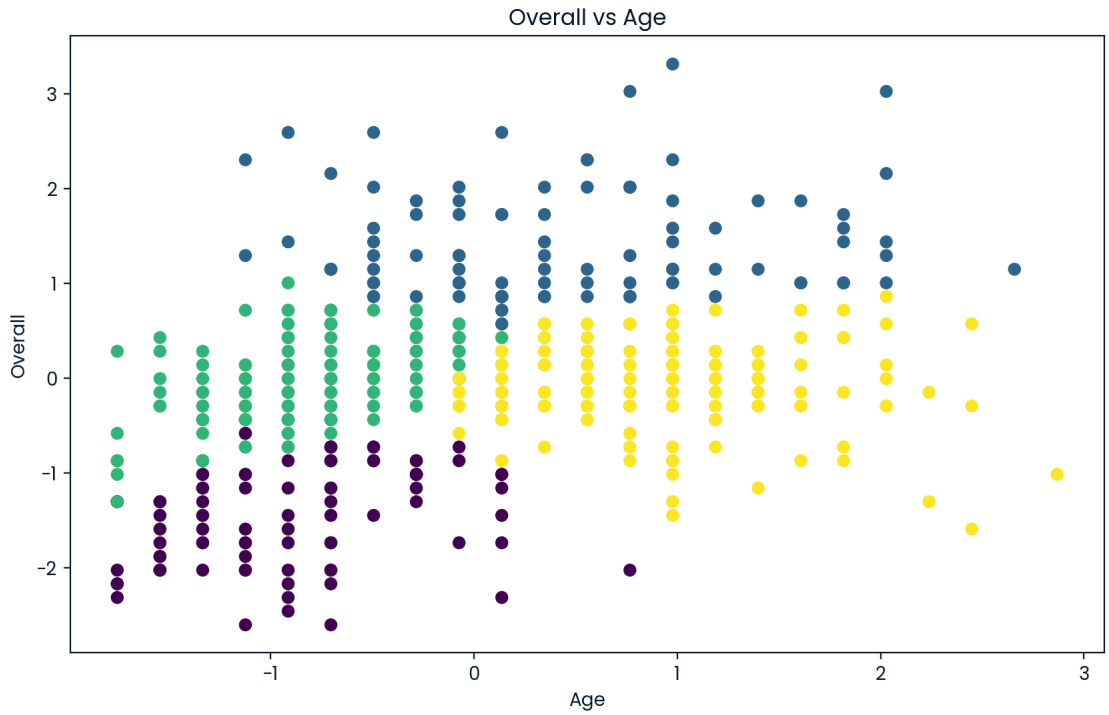
```
plt.figure(figsize=(10, 6))
plt.scatter(fifa_preprocessed['Cluster'], fifa_preprocessed['Potential'], c=fifa_preprocessed['Cluster'], cmap='viridis')

# Add labels and title
plt.xlabel('Cluster')
plt.ylabel('Potential')
plt.xticks(range(fifa_preprocessed['Cluster'].min(), fifa_preprocessed['Cluster'].max() + 1))  # Set x-ticks to cluster numbers
plt.title('Potential vs Cluster')

# Show the color bar to indicate the cluster color mapping
plt.colorbar(label='Cluster')

# Show the plot
plt.show()
```



Graph showing Overall vs Age and where the clusters lie. We can see 4 well defined clusters:

1. The dark blue represents younger players that have lower ratings / starting out in career
2. The green represents younger players that have an overall rating at around the mean/ starting out in career and good current level
3. The lighter blue represents that are the most highly rated in the game / current stars (mostly average age and slightly older guys)
4. The yellow represents older players that have an overall rating at around the mean/ guys who still havent reached stardom or are past their peak

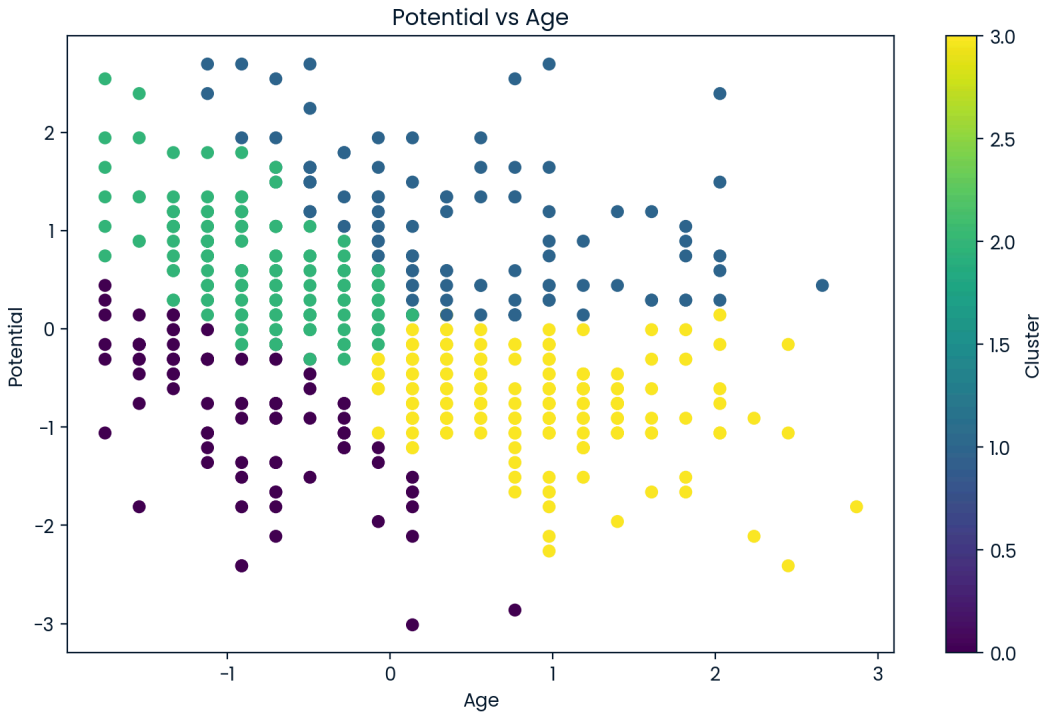
```
plt.figure(figsize=(10, 6))
plt.scatter(fifa_preprocessed['Age'], fifa_preprocessed['Overall'], c=fifa_preprocessed['Cluster'], cmap='viridis')


plt.xlabel('Age')
plt.ylabel('Overall')
plt.title('Overall vs Age')

plt.show()
```



Graph showing Potential vs Age and where the clusters lie. We can see 4 well defined clusters:

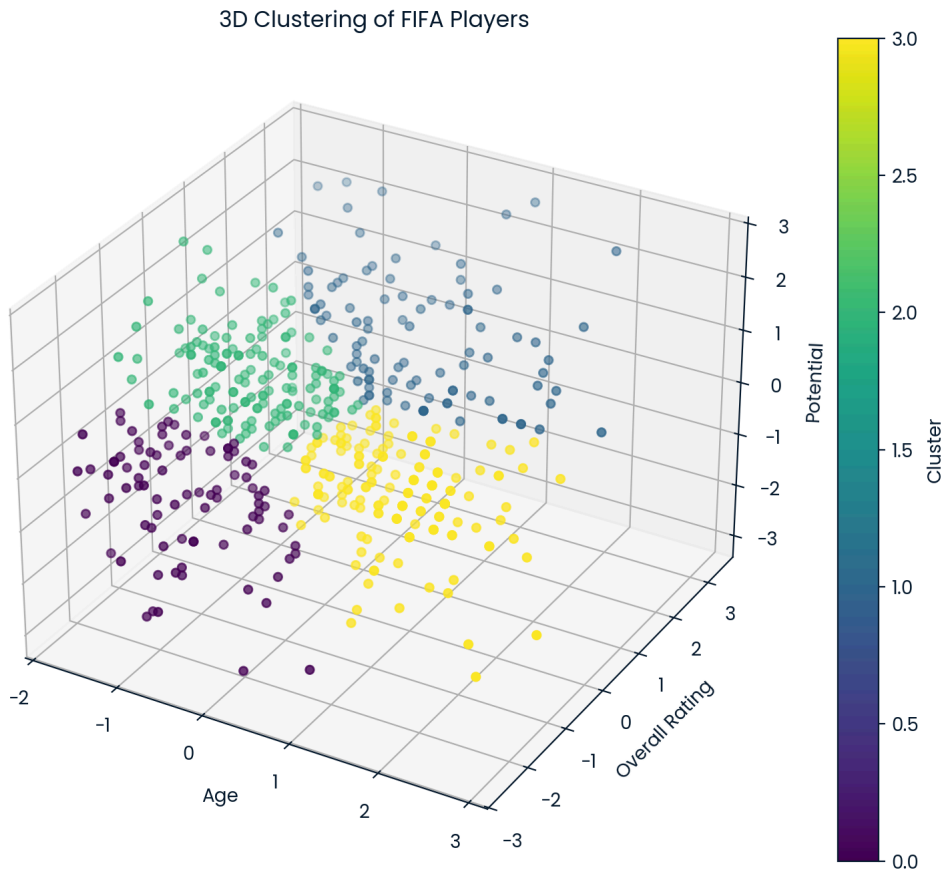
1. The dark blue represents younger players that have lower potential/ showing these guys will be average to below average players later on
2. The green represents younger players that have high potential/ these guys will be the games future stars
3. The lighter blue represents players that are the most highly rated in the game / current stars (mostly average age and slightly older guys)
4. The yellow represents older players that have an overall rating at around the mean/ guys who still havent reached stardom or are past their peak

```
plt.figure(figsize=(10, 6))
plt.scatter(fifa_preprocessed['Age'], fifa_preprocessed['Potential'], c=fifa_preprocessed['Cluster'], cmap='viridis')

# Add labels and title
plt.xlabel('Age')
plt.ylabel('Potential')
plt.title('Potential vs Age')

# Show the color bar to indicate the cluster color mapping
plt.colorbar(label='Cluster')

# Show the plot
plt.show()
```



Here we can see a 3-D Plot of our graph

```
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

scatter = ax.scatter(fifa_preprocessed['Age'],
                     fifa_preprocessed['Overall'],
                     fifa_preprocessed['Potential'],
                     c=fifa_preprocessed['Cluster'], cmap='viridis', marker='o')

ax.set_xlabel('Age')
ax.set_ylabel('Overall Rating')
ax.set_zlabel('Potential')
ax.set_title('3D Clustering of FIFA Players')

fig.colorbar(scatter, ax=ax, label='Cluster')

plt.show()
```



Here, we can see the 4 clusters and their averages in regards to the metrics we have been looking at

```
cluster_averages = fifa_preprocessed.groupby('Cluster')[['Overall', 'Potential', 'Age']].mean()

print(cluster_averages)
```

```
          Overall  Potential       Age
Cluster
0       -1.461560  -0.842956 -0.881333
1        1.410422   1.062375  0.446123
2       -0.012870   0.618763 -0.793035
3       -0.062931  -0.725307  0.879590
```

The averages of the clusters are as follows:

- Cluster 1: Overall - 56, Potential - 65, Age - 21 / These are the young players with lower potential
- Cluster 2: Overall - 76, Potential - 78, Age - 27 / These are the players in the prime / stars
- Cluster 3: Overall - 66, Potential - 75, Age - 22 / These are the young players with high potential/ future stars
- Cluster 4: Overall - 66, Potemtial - 66, Age - 30 / These are the veterans, likely on decline

```
original_cluster_averages = scaler.inverse_transform(cluster_averages)

original_cluster_averages
```

```
array([[55.89534884, 65.44186047, 21.15116279],
       [75.80612245, 78.12244898, 27.46938776],
       [65.93877551, 75.17006803, 21.57142857],
       [65.59171598, 66.22485207, 29.53254438]])
```