Background Information: One of the most common basketball discussions is similarity between players. People constantly compare different players' ability and playstyle against each other, especially when debating who the better player is and whether a new talent will be successful in the league. However, it is extremely difficult to quantitatively compare players, since there are so many factors that define a player for who they are and what makes them great. Additionally, today's NBA players have evolved so much from the past as players have such a wide range of skills. As a result, I've always been curious about how we can quantitatively compare NBA players.

As I thought about this question, one player stood out to me among all NBA players: my glorious king LeBron James. As arguably the best NBA player of all time, he doesn't have the flashy dribbles of Kyrie, the marksman three-point shooting like Curry, the alien-like build of Wemby. Yet, with his athleticism, high basketball IQ, and simple playstyle, he is able to dominate the League for 21 years.

Problem Statement: Group NBA players together based on similarity in the 23-24 regular season, and find which players are the most similar to LeBron James.

Hypothesis: Some players that are the most similar to LeBron are Jimmy Butler, Paul George, and Kawhi Leonard.

Method: To determine similarity between players, I will look at 9 factors that define a player for who they are: Physical Attributes, Usage & Creation, Shooting Tendencies, Shooting Ability, Portability, Passing Ability, Defense Ability, and Dominance. In order to evaluate players based on these factors, I wanted to find the most aggregate and advanced NBA statistics that precisely summarizes these factors. Using player data from CraftedNBA and NBA Stats, I was able to quantify these factors using the followings statistics (Glossary of each statistic at the end of the writeup) :

1. Physical attributes: Height, Weight, Wingspan
2. Usage & Playmaking: BC (Box Creation), USG % (Usage Percentage), LOAD (Offensive Load)
3. Shooting tendencies: Less than 5ft FGA%, 5-9 ft FGA%, 10-14 ft FGA%, 15-19 ft FGA%, 20-25 ft FGA%, 25-29ft FGA%
4. Shooting ability: SQ (Shooting Quality), TS% (True Shooting %)
5. Portability: PORT(Portability)
6. Passing ability: PR (Passer rating)
7. Defense: VR (Defensive Versatility Rating), CDPM (Defensive Crafted Plus Minus)
8. Dominance: CPM (Crafted Plus Minus)
9. Playstyle: Handoff%, Cut%, Iso%, Prbh%, Post%, Pnrr%, OffScreen%, SpotUp%

To add the data into a pandas dataframe, I copied these data into two Excel sheets and put them each into a dataframe, one for the data collected from CraftedNBA, and one for NBA Stats. I only included the players with the 250 most minutes played last season, since players with low playtime have skewed data. To combine the two dataframe, I first filtered both dataframes to only include the necessary columns, calculated new columns when needed, then merged the two dataframes by matching the player names.

To prepare the data for the K-Means model, I needed to deal with the null values and normalize the data. Since there were a lot of null values in the playstyle data, with multiple statistics with half or more null values, I decided to only include the 4 statistics where LeBron is the highest in, which were Prbh%, Iso%, Post%, and SpotUp%. For the null values in these, I used a flag and fill method, where I created a new column of binary data that indicates whether that value was filled, and filled the null values with the median of the dataset. The same flag and fill method was used for the Shooting tendencies data. Now that the dataframe has no null values, I used sklearn's standard scaler to normalize the dataframe and added them to a new dataframe.

Before running the clustering algorithm, I determined how many clusters to create using the elbow chart with sklearn's inertia attribute. Based on the elbow plot, it seems like 6 clusters is where the elbow is. However, six clusters is too little for 250 NBA players and won't separate them enough based on all the features we are including. My reasoning is that if you simply cluster players by their position (PG, SG, SF, PF, C), that is already five clusters. Therefore, we need a lot more than 5 to properly separate the players using the 9 factors we are considering. I decided to use 10 as the number of clusters, since there is another slight elbow after 10.

With the data fully processed for clustering, I fitted the data in a K-Means algorithm with 10 clusters, and random state = 1 to ensure the same result is produced every time. Then, I used the predict function to determine the cluster that each player belongs to, appended player name and the cluster they are in into a new dataframe, and printed the output to see which player is in which cluster.

To improve the model, I used Principal Component Analysis to reduce the dimensionality of the model into two variables so it can be plotted on a scatter plot to visualize the clusters. It also accounts for redundancy in the data. I used PCA on my normalized dataframe, and graphed the data points on a scatter plot using plotly so I can hover over the data points and see its information.

To evaluate the model, I used silhouette score and calinski scores, since I don't have the true values of the model. I calculated the silhouette and calinski scores for the PCA model using a range of number of clusters to further evaluate the best number of clusters, and graphed the results. The result showed that 9 clusters is the best, since it has the highest silhouette score and calinski score for more than 3 clusters, with a silhouette score of 0.378 and calinski score of 242.200. The silhouette score indicates a fair clustering. Although it doesn't indicate a strong clustering (score > 0.5), it must be noted that models with high dimensionality are much harder to get a high value because of the curse of dimensionality.

After evaluating the model, I changed the number of clusters to 9, ran the PCA algorithm again, and got my final model. Using that model, I created a function where you can input a player's name and it will return the 10 most similar players based on their euclidean distance to the inputted player.

Result and Discussion: With my final model, I ran the similar_players function on LeBron James, and found that the 10 most similar players are: 'Paolo Banchero', 'Jimmy Butler', 'Kevin Durant', 'Kawhi Leonard', 'Julius Randle', 'DeMar DeRozan', 'Jayson Tatum', 'Nikola Jokic', 'Jalen Williams', and 'Brandon Ingram.

This result seems decently accurate, as many of these players, especially Paolo Banchero, Jimmy Butler, and Kawhi Leonard are consistently compared to LeBron James by sports media and experts. This also partially supports my hypothesis, as two out of the three players in my hypothesis are among the top 10 most similar players based on this model.

Outside Resources:
Sources of dataset:

- https://craftednba.com/
- https://www.nba.com/stats/players/shooting?PerMode=Totals&Season=2023-24&SeasonType=Regular%20Season

Sources used to create project:

- https://blog.dailydoseofds.com/p/how-to-evaluate-clustering-results
- https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html
- https://builtin.com/data-science/step-step-explanation-principal-component-analysis
- https://scikit-learn.org/1.5/modules/clustering.html
- https://www.youtube.com/watch?v=iNlZ3IU5Ffw&t=10s
- https://www.youtube.com/watch?v=-jfHDMb7Ioc
- https://www.youtube.com/watch?v=EItlUEPCIzM