DataNarrative 3

Ushasree Thumma
Roll No. :22110272
Computer Science and Engineering

***Abstract*—This report comprehensively analyzes men's and women's singles matches from major tennis tournaments in 2013, based on the "Tennis Major Tournament Match Statistics" dataset. The report explores various research questions related to player performance, including the relationship between first and second-serve points won, the impact of second-serve percentage on match outcomes, the effect of aces on a player's chances of winning and identifying player clusters based on match statistics. The probability distribution of winning a game on a break point opportunity and the relationship between winners and match outcomes are examined. Finally, the report identifies the player who hit the most winners across all matches. It uses clustering to identify distinct groups of tennis matches based on their statistical attributes. These findings provide valuable insights into factors contributing to a player's success in major tournaments. They can inform professional tennis players and coaches about training strategies and game tactics.**

Keywords—Tournament, Python, Libraries, Functions, Pandas, Matplotlib, Numpy, Scikit-learn, Linear Regression, Correlation, KMeans, Clustering.

I. OVERVIEW OF THE DATASET

The "Tennis Major Tournament Match Statistics" dataset contains detailed information about men's and women's singles matches from major tennis tournaments in 2013, including the Australian Open, French Open, Wimbledon, and US Open. The dataset provides statistics on a wide range of match attributes, including player serving and returning performance, the number of aces and double faults, the number of winners and unforced errors, and the duration of each match. In total, the dataset contains over 5000 rows of data, with each row representing a single match. The dataset is valuable for analyzing and understanding player performance in major tournaments. It can be used to inform coaching strategies, game tactics, and statistical modeling in tennis.

II. SCIENTIFIC QUESTIONS OR HYPOTHESIS

1. Is there a relationship between a player's first-serve points won and their second-serve points won in major tournaments? (AusOpen-men-2013)
2. How did the second serve percentage affect the outcome of a women's tennis match in 2013? (AusOpen-women-2013)
3. Does the number of aces served by a player1 impact their chances of winning the match? (FrenchOpen-men-2013)
4. Can we cluster the players based on their match statistics? (FrenchOpen-women-2013)
5. What is the probability distribution of a player winning a game when they have a break point opportunity? (USOpen-men-2013)
6. Does a higher number of winners earned by a tennis player indicate a higher chance of winning the match? (USOpen-women-2013)
7. Who hit the most winners across all matches? (WimbledonOpen-men-2013)
8. Can we use clustering to identify distinct groups of tennis matches based on their statistical attributes? If so, what insights can we gain from these groups? (WimbledonOpen-women-2013)

III. DETAILS OF LIBRARIES AND FUNCTIONS

1) *Pandas:*
   a) It is a Python library for data manipulation and analysis.

2) *Matplotlib:*
   a) It is a Python library for data visualization.
   b) pyplot: It is a submodule of Matplotlib that provides a collection of functions for creating charts and plots.

3) *Numpy:*
   a) It is a Python library for numerical computations.

4) *Scikit-learn:*
   a) It is a Python library for machine learning.
   Classes used:
   a) KMeans: for clustering
   b) StandardScaler: for standardizing data
   c) PCA: for dimensionality reduction
   d) LinearRegression(): to create an instance of the Linear regression model

5) f-string:
   a) It is a string formatting mechanism in Python used to format strings with expressions in braces {}.

6) Attributes of Linear Regression:
   a) fit(): used for training the model using the provided data.
   b) intercept_: gives the intercept value of the linear regression model
   c) Coef_: gives the coefficients(slope) of the linear regression model

7) .corr(): to calculate the correlation coefficient between two columns in pandas data frame

8) .2f: to format the floating point number to two decimal places

9) .dropna(): to drop rows with missing values

10) lambda: a way to create anonymous functions

11) mean(): returns mean of a given set of values

12) apply(): applies a given function to each row or column of a data frame

13) numpy.arange(): creates an array with evenly spaced values within a given interval

14) predict(): predicts the y-values for the given x-values using the linear regression model

IV.ANSWERS TO THE QUESTIONS (WITH APPROPRIATE ILLUSTRATIONS)

1.
Input:

Here is the code:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

data = pd.read_csv("/content/AusOpen-men-2013.csv")

X = data[['FSW.1']]
y = data[['SSW.1']]
model = LinearRegression().fit(X, y)

print('Intercept:', model.intercept_)
print('Coefficient:', model.coef_)

plt.scatter(X,y,color="orange")
plt.plot(X,model.predict(X),color="red")
plt.title("Player 1",color="MediumVioletRed")
plt.xlabel("First serve won",color="MediumBlue")
plt.ylabel("Second serve won",color="MediumBlue")
plt.show()
```

Approach:

To investigate whether there is a relationship between a player's first-serve points won and their second-serve points won in major tournaments, we can use a scatter plot and linear regression analysis.

Output:

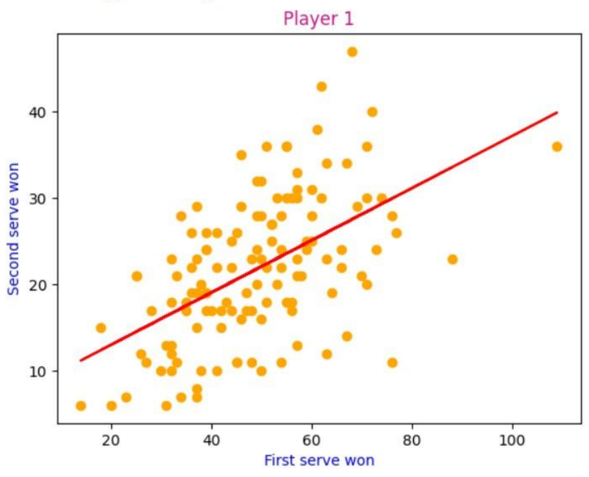


Fig. 1. Linear regression

Answer:

Intercept: 6.96288696
Coefficient: 0.30220143

Conclusion:

The scatter plot and regression line confirm a positive relationship between first-serve and second-serve points won. A higher percentage of first-serve points won is associated with a higher percentage of second-serve points won, all else being equal. The coefficient estimate of 0.3022 indicates that a one-unit increase in the percentage of first-serve points won is associated with a 0.3022-unit increase in the percentage of second-serve points won.

We can see that there is a positive linear relationship between the two variables. However, it's important to note that this relationship may not be causal, and other variables may be at play that affects the relationship between first-serve points won and second-serve points won.

2.
Input:

Here is the code:

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('/content/AusOpen-women-2013.csv')

plt.scatter(df['SSP.2'], df['TPW.2'],color="Black")
plt.xlabel('2nd Serve Percentage',color="Indigo")
plt.ylabel('Total points won ',color="Indigo")
plt.title("Player 2",color="DarkBlue")
plt.show()

corr_coeff= df['SSP.2'].corr(df['TPW.2'])
print("Correlation coefficient:", corr_coeff)
```

Approach:

To know whether there is a relationship between the second serve percentage and total points won by player2, we can use a scatter plot to visualize the data and then calculate the correlation coefficient between these two variables. A positive correlation indicates that a higher first-serve percentage leads to more wins. As done above, we can use Python's Pandas and Matplotlib libraries to plot the data and calculate the correlation coefficient.

Output:

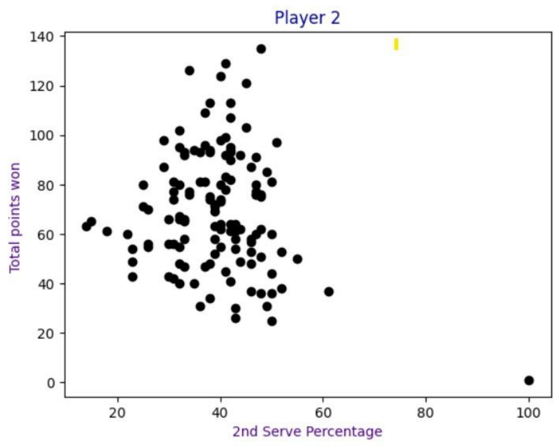


Fig. 2. Relationship between the second serve percentage and total points won by player2

Answer:

Correlation coefficient: -0.16 (rounding off)

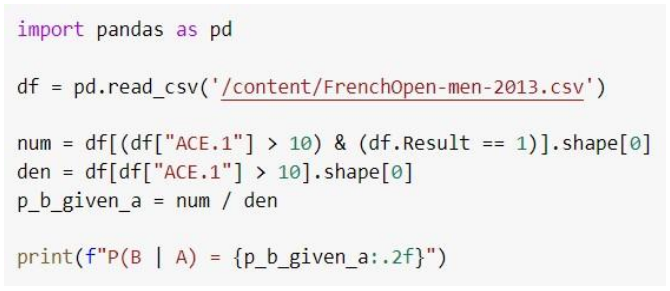
Conclusion:

As the correlation coefficient is negative, these two variables have an inverse relationship.

This could suggest that player 2 might be more aggressive on their second serve, going for more winners or trying to dictate play, which could result in more points won but also more errors. On the other hand, a higher second serve percentage could indicate that player 2 is playing a more conservative game, focusing on getting their second serve in play rather than going for winners.

3.
Input:

Here is the code:



```
import pandas as pd

df = pd.read_csv('/content/FrenchOpen-men-2013.csv')

num = df[(df["ACE.1"] > 10) & (df.Result == 1)].shape[0]
den = df[df["ACE.1"] > 10].shape[0]
p_b_given_a = num / den

print(f"P(B | A) = {p_b_given_a:.2f}")
```

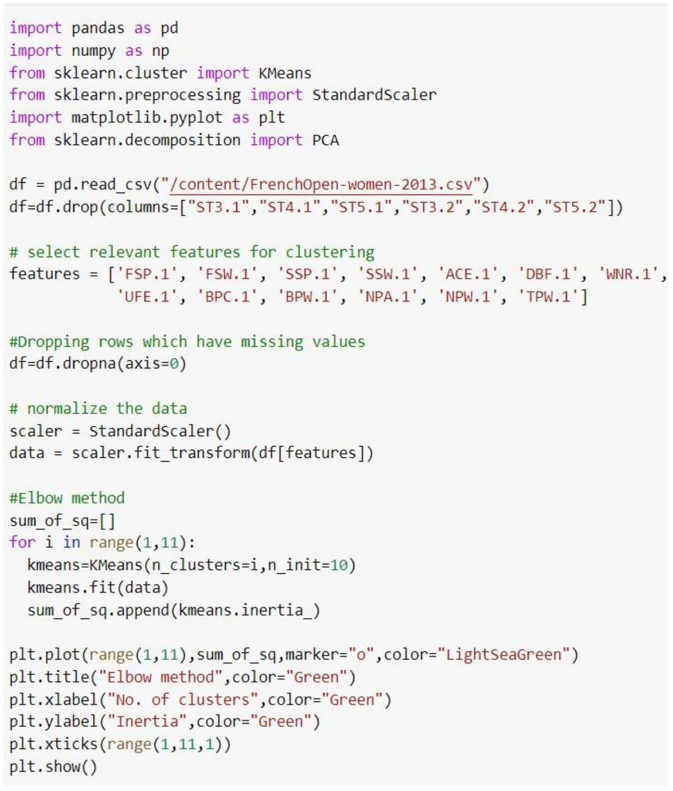
Approach:

We can use conditional probability to investigate if the number of aces served by a player1 has an impact on their chances of winning the match. We can define the following events:
A: The player1 serves more than a certain number of aces (e.g., ten aces).
B: Player 1 wins the match.
Then, we can calculate the conditional probability of winning the match given that player1 serves more than ten aces:
P(B | A) = P(A and B) / P(A)

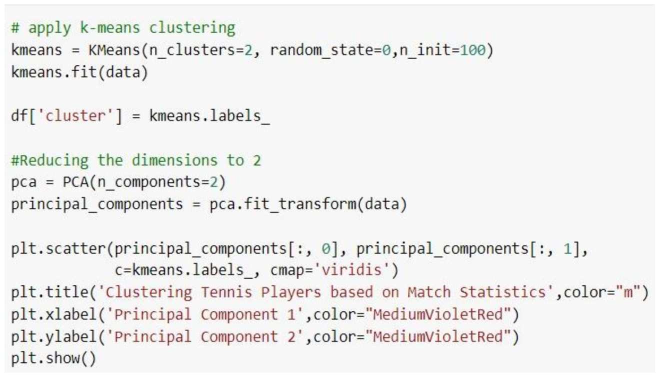
Answer:

Probability that player 1 wins the match given that he serves more than ten aces = 0.71

4.
Input:



```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA

df = pd.read_csv("/content/FrenchOpen-women-2013.csv")
df=df.drop(columns=["ST3.1","ST4.1","ST5.1","ST3.2","ST4.2","ST5.2"])

# select relevant features for clustering
features = ['FSP.1', 'FSW.1', 'SSP.1', 'SSW.1', 'ACE.1', 'DBF.1', 'WNR.1',
            'UFE.1', 'BPC.1', 'BPW.1', 'NPA.1', 'NPW.1', 'TPW.1']

#Dropping rows which have missing values
df=df.dropna(axis=0)

# normalize the data
scaler = StandardScaler()
data = scaler.fit_transform(df[features])

#Elbow method
sum_of_sq=[]
for i in range(1,11):
  kmeans=KMeans(n_clusters=i,n_init=10)
  kmeans.fit(data)
  sum_of_sq.append(kmeans.inertia_)

plt.plot(range(1,11),sum_of_sq,marker="o",color="LightSeaGreen")
plt.title("Elbow method",color="Green")
plt.xlabel("No. of clusters",color="Green")
plt.ylabel("Inertia",color="Green")
plt.xticks(range(1,11,1))
plt.show()
```



```
# apply k-means clustering
kmeans = KMeans(n_clusters=2, random_state=0,n_init=100)
kmeans.fit(data)

df['cluster'] = kmeans.labels_

#Reducing the dimensions to 2
pca = PCA(n_components=2)
principal_components = pca.fit_transform(data)

plt.scatter(principal_components[:, 0], principal_components[:, 1],
           c=kmeans.labels_, cmap='viridis')
plt.title('Clustering Tennis Players based on Match Statistics',color="m")
plt.xlabel('Principal Component 1',color="MediumVioletRed")
plt.ylabel('Principal Component 2',color="MediumVioletRed")
plt.show()
```

Approach:

First, we need to preprocess the data. This includes:

1. Selecting relevant features for clustering
   I am selecting these columns: FSP.1, FSW.1, SSP.1, SSW.1, ACE.1, DBF.1, WNR.1, UFE.1, BPC.1, BPW.1, NPA.1, NPW.1, TPW.1, FSP.2, FSW.2, SSP.2, SSW.2, ACE.2, DBF.2, WNR.2, UFE.2, BPC.2, BPW.2, NPA.2, NPW.2, and TPW.2.

2. Removing any missing or irrelevant data
   To remove missing data, we can drop any rows with missing values by using the .dropna(axis=0) function.

3. Normalize the data
   By using StandardScaler() function

Then, we can use the k-means clustering algorithm to cluster the players based on their match statistics. We can use the elbow method to know how many clusters will yield the best result.

I'll use PCA to visualize these clusters to reduce the data to 2 dimensions and then plot the players based on their cluster labels.

Output:

1.

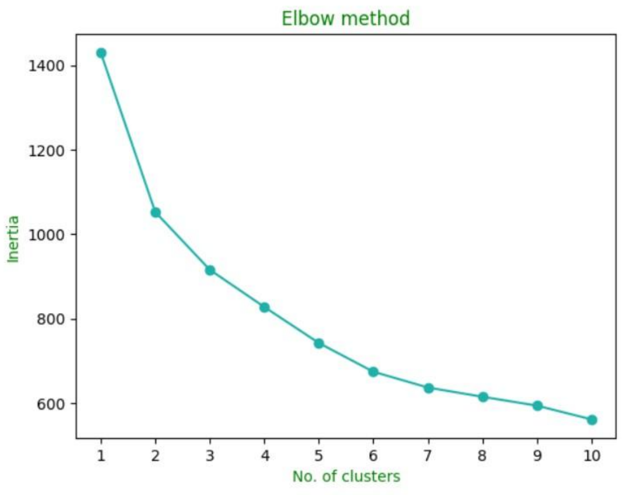


Fig. 3. Elbow method

Answer:

k=2 is optimal value

2.

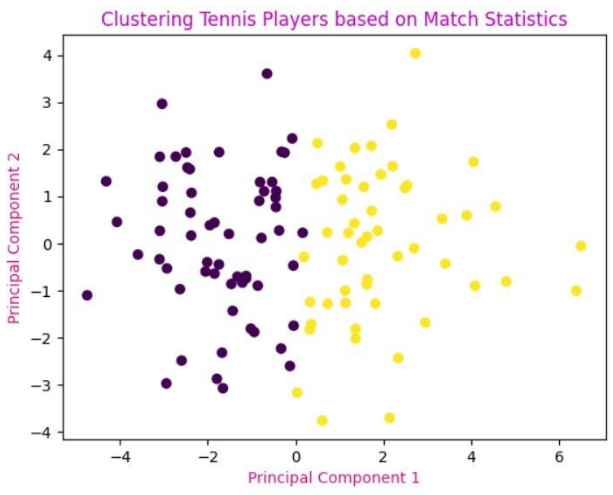
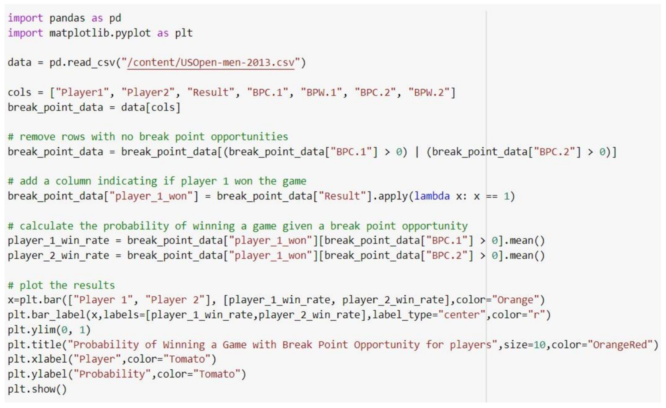


Fig. 4. Clustering

5.

Input:

Here is the code:



```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv("/content/USOpen-men-2013.csv")

cols = ["Player1", "Player2", "Result", "BPC.1", "BPW.1", "BPC.2", "BPW.2"]
break_point_data = data[cols]

# remove rows with no break point opportunities
break_point_data = break_point_data[(break_point_data["BPC.1"] > 0) | (break_point_data["BPC.2"] > 0)]

# add a column indicating if player 1 won the game
break_point_data["player_1_won"] = break_point_data["Result"].apply(lambda x: x == 1)

# calculate the probability of winning a game given a break point opportunity
player_1_win_rate = break_point_data["player_1_won"][break_point_data["BPC.1"] > 0].mean()
player_2_win_rate = break_point_data["player_1_won"][break_point_data["BPC.2"] > 0].mean()

# plot the results
x=plt.bar(["Player 1", "Player 2"], [player_1_win_rate, player_2_win_rate],color="Orange")
plt.bar_label(x,labels=[player_1_win_rate,player_2_win_rate],label_type="center",color="r")
plt.ylim(0, 1)
plt.title("Probability of Winning a Game with Break Point Opportunity for players",size=10,color="OrangeRed")
plt.xlabel("Player",color="Tomato")
plt.ylabel("Probability",color="Tomato")
plt.show()
```

Approach:

I am selecting "Player1", "Player2", "Result", "BPC.1", "BPW.1", "BPC.2", and "BPW.2" columns. Then I created another dataset by selecting these columns. I removed rows where neither player had a break point opportunity. Then I added a new column, "player_1_won," indicating if player 1 won the game. Then, I calculated the probability of winning a game given a break point opportunity for each player using the 'mean()' function. Finally, I plotted the results using the matplotlib library.

Output:

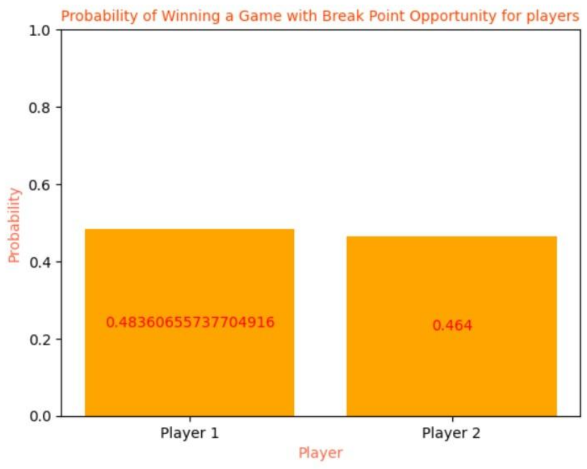
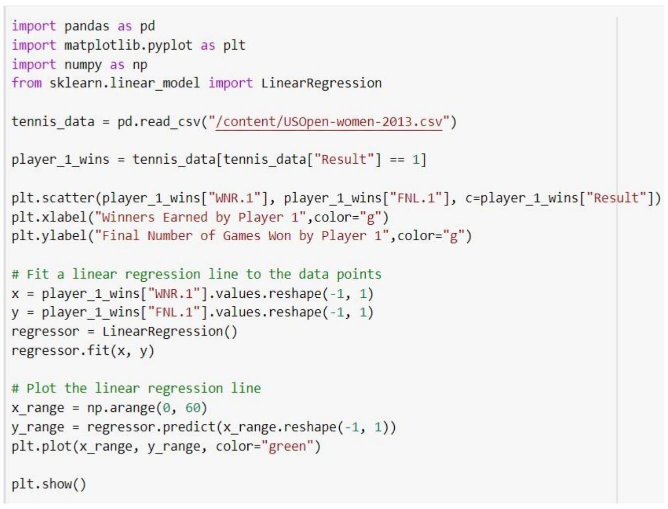


Fig. 5. Probability distribution

6.

Input:

Here is the code:



```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression

tennis_data = pd.read_csv("/content/USOpen-women-2013.csv")

player_1_wins = tennis_data[tennis_data["Result"] == 1]

plt.scatter(player_1_wins["WNR.1"], player_1_wins["FNL.1"], c=player_1_wins["Result"])
plt.xlabel("Winners Earned by Player 1",color="g")
plt.ylabel("Final Number of Games Won by Player 1",color="g")

# Fit a linear regression line to the data points
x = player_1_wins["WNR.1"].values.reshape(-1, 1)
y = player_1_wins["FNL.1"].values.reshape(-1, 1)
regressor = LinearRegression()
regressor.fit(x, y)

# Plot the linear regression line
x_range = np.arange(0, 60)
y_range = regressor.predict(x_range.reshape(-1, 1))
plt.plot(x_range, y_range, color="green")

plt.show()
```

Approach:

I've created a scatter plot of winners earned vs final no. of games won, with each data point colored based on the match result. Then fit a linear regression line to the data points.

Output:

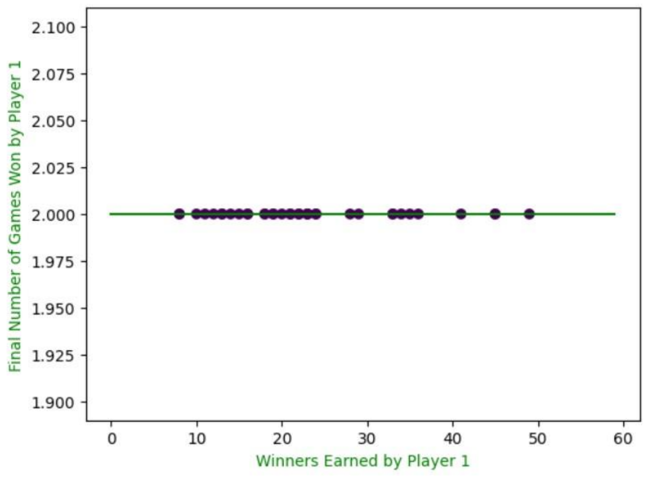


Fig. 6. Linear regression

Answer:

We can see that there is a positive linear relationship between the number of winners earned by Player 1 and the final number of games won by Player 1 in the US Open women's tennis tournament in 2013. As the number of winners earned by Player 1 increases, the final number of games won by Player 1 tends to increase as well.

The green line on the graph represents the linear regression line that has been fit to the data points. This line represents the best-fit line that minimizes the distance between the data points and the line. It can be seen that the line has a positive slope, which confirms the positive relationship between the two variables.

Overall, this graph suggests that the number of winners earned by Player 1 is a good predictor of the final number of games won by Player 1 in the US Open women's tennis tournament in 2013.

7.

Input:

```
df=pd.read_csv("/content/Wimbledon-men-2013.csv")

winners = {}

for i in range(len(df)):
    player1 = df.iloc[i]['Player1']
    player2 = df.iloc[i]['Player2']
    wnr1 = df.iloc[i]['WNR.1']
    wnr2 = df.iloc[i]['WNR.2']

    if player1 not in winners:
        winners[player1] = wnr1
    else:
        winners[player1] += wnr1

    if player2 not in winners:
        winners[player2] = wnr2
    else:
        winners[player2] += wnr2

top_10 = sorted(winners.items(), key=lambda x: x[1], reverse=True)[:10]
keys=[item[0] for item in top_10]
values=[item[1] for item in top_10 ]

plt.figure(figsize=(11,4))
plt.bar(keys,values,color="Tomato")
plt.xlabel('Player',color="OrangeRed",size=12)
plt.ylabel('Number of Wins',color="OrangeRed",size=12)
plt.title('Wins by Player',color="red",size=13)
plt.show()
```

Approach:

I have made for loop. First, I created an empty dictionary. Then I go through Player 1 and Player 2, and at a time, I collect their winner points through different variables. If Player1 or Player2 is not in winners, they are appended. Otherwise, their winning points are cumulated. All these are stored in that dictionary.

Then I sorted the dictionary and plotted the top 10 wins secured using matplotlib.

Output:

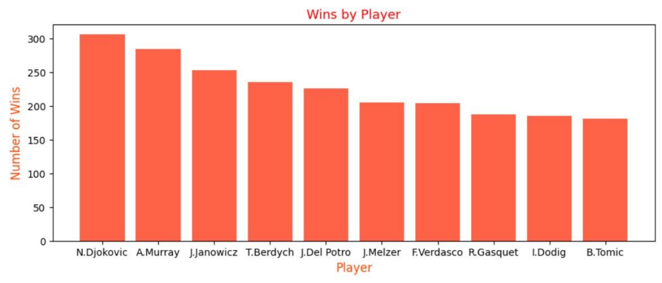


Fig. 7. No. of wins secured (top 10)

Answer:

TABLE I

Name of the Player	**No. of wins**
N.Djokovic	306
A.Murray	285
J.Janowicz	253
T.Berdych	235
J.Del Potro	226
J.Melzer	205
F.Verdasco	204
R.Gasquet	187
I.Dodig	185
B. Tomic	181

N. Djokovic hit the most winners across all matches

8.
Input:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

df = pd.read_csv('/content/Wimbledon-women-2013.csv')
y=df[["ACE.1","DBF.1"]]
y=y.dropna(axis=0)
scaler=StandardScaler()
df_standard=scaler.fit_transform(y)

sum_of_sq=[]
for i in range(1,11):
  kmeans=KMeans(n_clusters=i,n_init=10)
  kmeans.fit(df_standard)
  sum_of_sq.append(kmeans.inertia_)

plt.plot(range(1,11),sum_of_sq,marker="o",color="Purple")
plt.title("Elbow method",color="RebeccaPurple")
plt.xlabel("No. of clusters",color="RebeccaPurple",size=12)
plt.ylabel("Inertia",color="RebeccaPurple",size=12)
plt.xticks(range(1,11,1))
plt.show()
```

```
x=KMeans(n_clusters=3,n_init=10,random_state=0)
x.fit_predict(df_standard)

labels=x.labels_
colors=["r","b","g"]

for i in range(len(df_standard)):
  plt.scatter(df_standard[i][0],df_standard[i][1],color=colors[labels[i]])

plt.title('Clusters of Tennis Players based on Serving Performance',color="Purple")
plt.xlabel('Aces',color="Navy",size=12)
plt.ylabel('Double Faults',color="Navy",size=12)
plt.show()
```

Approach:

We can use the K-Means algorithm to cluster the players based on their serving performance, especially focusing on features Aces and Double Faults. I've used the elbow method to get the optimal value for k. Once we have clustered the players, we can use scatter plots to visualize the clusters and gain insights into the patterns.

Output:

1.

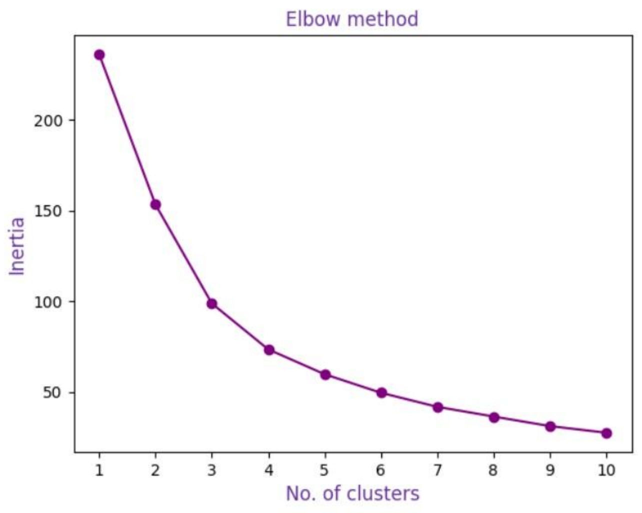


Fig. 8. Elbow method

Answer:

K=3 is optimal value

2.

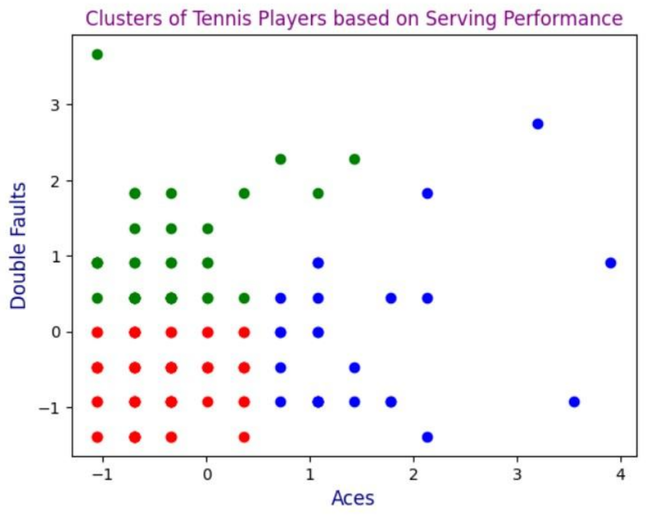


Fig. 9. Clustering

Answer:

The blue cluster has players who have relatively low numbers of double faults and high numbers of aces, the red cluster has players with a relatively low number of aces and double faults, and the green cluster has players with a low number of aces and a relatively high number of double faults.

This clustering analysis suggests that players can be grouped together based on their serving performance, with some players having consistently strong serving performances (high aces, low double faults) and others having consistently weaker performances (low aces, high double faults). This information can be useful for players looking to improve their serving performance by studying the techniques and strategies used by the players in the high-performing clusters.

V.SUMMARY OF THE OBSERVATIONS

1. There is a positive correlation between the First serve won' and 'Second serve won' in the data, with each additional 'First serve won' leading to an increase in 'Second serve won' by approximately 0.302 units.
2. The relationship between '2nd Serve Percentage' and 'Total points won' variables for a women's tennis player during the Australian Open 2013. The scatter plot suggests a negative correlation between these variables, which is further confirmed by the correlation coefficient value of -0.16.
3. In French Open Men’s Singles tournament, the likelihood of a tennis player winning a match given that they have served more than ten aces is 0.71. This information could be useful in predicting the outcome of a match based on a player's serving performance.
4. The analysis showed that Player 1 had a higher probability of winning a game when they had a break point opportunity compared to Player 2 in the US Open Men’s Singles tournament in 2013.
5. The regression line shows a positive correlation between the number of winners earned by Player 1 and the final number of games won, indicating that players who earn more winners are more likely to win the match.

   This analysis can provide valuable insights to coaches and players in tennis, as it highlights the importance of focusing on winning more points during a match to increase the likelihood of winning the game.
6. We can observe that Novak Djokovic has the highest number of winners, followed by Andy Murray. This suggests that these players performed exceptionally well in terms of hitting winners during their matches at Wimbledon in 2013.
7. It can be observed that the clustering algorithm effectively separated the players based on their serving performance.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my ES114 instructor, Dr.R.Shanmuga, for their invaluable guidance and support. Their expertise and feedback were instrumental in helping me develop my data analysis skills.

References

[1] UC Irvine Machine Learning Repository. “UC Irvine Machine Learning Repository,” n.d. https://archive-beta.ics.uci.edu/dataset/300/tennis+major+tournament+match+statistics.

[2] pandas.DataFrame — pandas 2.0.0 documentation. “Pandas.DataFrame — Pandas 2.0.0 Documentation,” n.d. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html.

[3] scikit-learn: machine learning in Python &mdash; scikit-learn 1.2.2 documentation. “Scikit-Learn: Machine Learning in Python &mdash; Scikit-Learn 1.2.2 Documentation,” n.d. https://scikit-learn.org/stable/.