DATA NARRATIVE

Kishan Ved
Computer Science Engineering
Indian Institute of Technology Gandhinagar
Roll Number - 22110122

***Abstract*—This data narrative aims to provide a comprehensive insight into the Tennis Major Match Statistics and analyze the data it contains.**

I. OVERVIEW OF THE DATASET

The Tennis Major Statistics dataset is a collection of 8 files containing the match statistics for women and men at the four major tennis tournaments of 2013. Each file has 42 columns and a minimum of 76 rows. Each row has attributes like first serve percentage, number of double faults, the result of the match etc.

II. SCIENTIFIC QUESTIONS AND HYPOTHESES

All 8 datasets have been combined to create a complete dataset for data on Tennis matches in 2013 from famous tournaments. The dataset has been analyzed and researched to answer eight scientific questions which include 2 hypotheses. A machine learning model to predict the outcome of a match is also asked for in the end.

Scientific Questions and Hypotheses:

1) What is the probability that a player wins the game even if the number of unforced errors exceeds 20?
2) Which were the most interesting games of 2013, which gripped the audience to their seats until the final set?
3) The successive tournaments of the year 2013 were – Australian Open (Jan), US Open (May), French Open (Jun) and Wimbledon (Aug). Was there any player(s) who was out of the Australian Open in the first round itself but worked hard and made it to the semi-finals of any other 2013 tournament?
4) What is the probability that a player wins the game even on losing the first set?
5) Among the winners of the 8 major tennis tournaments in 2013, who committed the least errors in all 2013 tournament games combined?
6) What is the probability that a player commits more errors than his opponent but wins the game?
7) ***Hypothesis*** – A player skilled at scoring net points is also good at scoring break points.
8) ***Hypothesis*** – As the player's experience level increases, the number of Ace points won per match increases.
9) ***Machine Learning Model -*** Design a simple machine learning model to predict the outcome of a tennis match by taking in only the first and second serve percentage and the number of unforced errors, net points, breakpoints and forced errors statistics.

III. PYTHON LIBRARIES AND FUNCTIONS USED

The following Python libraries and their mentioned functions were used to analyze the dataset:

- NumPy:
  - np.array() – To generate a NumPy array to operate upon.
  - np.arange() –To get a NumPy array having consecutive values.
- Pandas:
  - pd.read_csv() – To read data from a csv file.
  - pd.DataFrame() – To convert data into a Pandas DataFrame.
  - describe() – To get statistical information about the data.
- Matplotlib:
  - plt.plot() – To plot data.
  - plt.pie() – To display a pie chart of the data.
  - plt.bar() – To plot a bar graph.
- Sklearn module:
  - KMeans – To cluster the data.
  - LinearRegression – To perform Linear Regression.
  - PolynomailFeatures, make_pipeline – To make a ML pipeline and perform polynomial regression.

IV. ANSWERS TO THE QUESTIONS

Question 1)

What is the probability that a player wins the game even if the number of unforced errors exceeds 20?

A) Approach:

Iterate through the "UFE.1" column and if the value is >20, then check the value in the "Result" column to know if player 1 won the game. Repeat the same for the "UFE.2" column.

Answer –

```
# Probability of a player winning even if the number of unforced errors > 20.

tot = 0
ctr = 0
for i in range(len(df)):
    if df.loc[i,"UFE.1"]!=None and df.loc[i,"UFE.1"]>20:
        tot+=1
        if df.loc[i,"Result"]==1:
            ctr+=1

for i in range(len(df)):
    if df.loc[i,"UFE.2"]!=None and df.loc[i,"UFE.2"]>20:
        tot+=1
        if df.loc[i,"Result"]==0:
            ctr+=1

prob = ctr/tot
```

The probability that a player wins the game even if the number of unforced errors exceeds 20 is: `0.42488038277511964.`

Question 2)

Which were the most interesting games of 2013, which gripped the audience to their seats until the final set?

A) Approach:

A match is interesting if the game is unpredictable till the very end, so we need to find the games that lasted for 5 sets and had maximum number of points. To do so, iterate over the "ST1.5" column and see if it contains a numeric value (not a Nan value). Among all such games, take the games, top 5 games are the ones with the maximum number of points scored.

Answer –

```
# Which were the most interesting games of the year 2023?

ndf = pd.DataFrame(columns=df.columns)
for i,row in df.iterrows():
    if not np.isnan(row["ST5.2"]):
        ndf = ndf.append(row)
        # print(row)

ndf

ndf["Total Points"] = ndf.loc[:,"TPW.1"] + ndf.loc[:,"TPW.2"]
ndf = ndf.drop(['Result', 'FNL1', 'FNL2', 'FSP.1',
       'FSW.1', 'SSP.1', 'SSW.1', 'ACE.1', 'DBF.1', 'WNR.1', 'UFE.1', 'BPC.1',
       'BPW.1', 'NPA.1', 'NPW.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',  'NPPT.1', 'NPP.1', 'NPPT.2',
       'NPP.2', 'BPPT.1', 'BPP.1', 'BPPT.2', 'BPP.2'],axis=1)
ndf =ndf.dropna()
ndf = ndf.sort_values("Total Points",ascending=False)
ndf = ndf.drop([ 'TPW.1', 'ST1.1', 'ST2.1', 'ST3.1',
       'ST4.1', 'ST5.1', 'TPW.2', 'ST1.2', 'ST2.2', 'ST3.2', 'ST4.2', 'ST5.2',
       ],axis=1)
```

The top 5 most interesting games are:

Most interesting matches of the year 2013:

	Player1	Player2	Round	Tournament	Total Points
36	Daniel Brands	Gilles Simon	1	Australian Open 2013	461.0
562	Tommy Haas	John Isner	3	French Open 2013	437.0
365	Richard Gasquet	Milos Raonic	1	US Open 2013	402.0
542	Fernando Verdasco	Janko Tipsarevic	2	French Open 2013	381.0
299	Benoit Paire	Alex Bogomolov Jr.	1	US Open 2013	372.0

Thus, we observe that the game Daniel Brands vs Gilles Simon in Round 1 of the Australian Open 2013 lasted for all 5 sets and had a total of 461 points scored, making it the most interesting game of the year 2013.

Question 3)

The successive tournaments of the year 2013 were – Australian Open (Jan), US Open (May), French Open (Jun) and Wimbledon (Aug). Was there any player(s) who was out of the Australian Open in the first round itself but worked hard and made it to the semi-finals of any other 2013 tournament?

Approach:

To find such a player, store the players in rounds 1 and 2 (say set1 and set2) of the Australian Open 2013 in 2 different sets. Store the players in round 7 (final round) of other 2013 tennis tournaments in their respective sets. Now check if there is a player in set1 but not in set2, who is in a set of any other tournaments too.

Answer – Create sets as shown:

```
aus_r1 = set()
aus_r2 = set()
for i in range(len(df)):
    if df.loc[i,"Round"]==1 and df.loc[i,"Tournament"]=="Australian Open 2013":
        aus_r1.add(df.loc[i,"Player1"])
        aus_r1.add(df.loc[i,"Player2"])
    if df.loc[i,"Round"]==2 and df.loc[i,"Tournament"]=="Australian Open 2013":
        aus_r2.add(df.loc[i,"Player1"])
        aus_r2.add(df.loc[i,"Player2"])

us_r6 = set()
for i in range(len(df)):
    if df.loc[i,"Round"]==6 and df.loc[i,"Tournament"]=="US Open 2013":
        us_r6.add(df.loc[i,"Player1"])
        us_r6.add(df.loc[i,"Player2"])
```

Apply the condition:

```
s = set()
for name in aus_r1 :
    if name in aus_r2:
        continue
    if name in fr_r6 or name in w_r6 or name in us_r6:
        s.add(name)
```

There was 1 player, Sara Errani, who was out of the Australian Open 2013 tournament in the first round itself but worked hard to make it to the finals of another major tournament in the summer.

Question 4)

What is the probability that a player wins the game even after losing the first set?

Approach:

Iterate over the “ST1.1” and “ST1.2” columns and see which has a lesser value. From the result column, check which player won the game.

Answer –

```
# What is the probability that the player who loses the first set wins the match?

tot = len(df)
ctr = 0
for i in range(tot):
    if df.loc[i,"ST1.1"]>df.loc[i,"ST1.2"] and df.loc[i,"Result"]==1:
        ctr+=1
    elif df.loc[i,"ST1.1"]<df.loc[i,"ST1.2"] and df.loc[i,"Result"]==0:
        ctr+=1
```

The probability that a player who loses the first set wins the match is: `0.17709437963944852.`

Question 5)

Among the winners of the 8 major tennis tournaments in 2013, who committed the least errors in all 2013 tournament games combined?

Approach -

Store the winners of all 8 tournaments in a dictionary as keys and the total errors committed by them as the values. Total errors are calculated by 2*double faults + unforced errors found by iterating over the dataset.

Answer –

Create the dictionary as shown:

```
f = set()
for i in range(len(df)):
    if df.loc[i,"Round"]==7 and df.loc[i,"Tournament"]=="Australian Open 2013":
        f.add(df.loc[i,"Player1"]) if df.loc[i,"Round"]==1 else f.add(df.loc[i,"Player2"])

for i in range(len(df)):
    if df.loc[i,"Round"]==7 and df.loc[i,"Tournament"]=="US Open 2013":
        f.add(df.loc[i,"Player1"]) if df.loc[i,"Round"]==1 else f.add(df.loc[i,"Player2"])

for i in range(len(df)):
    if df.loc[i,"Round"]==7 and df.loc[i,"Tournament"]=="French Open 2013":
        f.add(df.loc[i,"Player1"]) if df.loc[i,"Round"]==1 else f.add(df.loc[i,"Player2"])

for i in range(len(df)):
    if df.loc[i,"Round"]==7 and df.loc[i,"Tournament"]=="Wimbledon 2013":
        f.add(df.loc[i,"Player1"]) if df.loc[i,"Round"]==1 else f.add(df.loc[i,"Player2"])

dict = {name:0 for name in f}
for i in range(len(df)):
    for name in f:
        if df.loc[i,"Player1"] == name:
            dict[name]+=(df.loc[i,"UFE.1"]+2*df.loc[i,"DBF.1"])
        elif df.loc[i,"Player2"] == name:
            dict[name]+=(df.loc[i,"UFE.2"]+2*df.loc[i,"DBF.2"])
```

The least errors were committed by M. Bartoli, who made only 169 errors (including double faults and unforced errors).

Question 6)

What is the probability that a player commits more errors than his opponent but wins the game?

Approach –

Create a new column to store total errors as defined in the previous question’s answer. Now, iterate over the “Result” column and see which player won the game, and check if that player has lesser total errors.

Answer –

```
for i in range(len(df)):
    df['TE.1'] = 2*df.loc[i, "DBF.1"] + df.loc[i,"UFE.1"]
    df['TE.2'] = 2*df.loc[i, "DBF.2"] + df.loc[i,"UFE.2"]
tot =len(df)
ctr=0
for i in range(len(df)):
    if df.loc[i,"Result"]==1 and df.loc[i,"TE.1"]>df.loc[i,"TE.2"]:
        ctr+=1
    if df.loc[i,"Result"]==0 and df.loc[i,"TE.1"]<df.loc[i,"TE.2"]:
        ctr+=1
```

The required probability is: 0.4931071049840933

Question 7)

Hypothesis – A player skilled at scoring net points is also good at scoring break points.

Approach-

Correct the data and swap values in NPW,NPA and BPW,BPA when needed.

Make a ML pipeline using sklearn.pipeline's make_pipeline. Into this, append sklearn's PolynomialFeatures. Use this to perform polynomial regression. Use sklearn's LinearRegression to perform linear regression.

Perform polynomial regression to fit a line and a polynomial of degree 3, then plot the scatterplot and the curves. Find the correlation coefficient.

Answer –

Code snippet for polynomial regression-

```
# Performing Linear and Polynomial Regression
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import seaborn; seaborn.set()  # plot formatting
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import make_pipeline
# increases the dimension by making points like x1*x2 etc.

def PolynomialRegression(degree=2, **kwargs):
# kwargs - keyword arguments, position does not matter, these are passed
    return make_pipeline(PolynomialFeatures(degree),
                         LinearRegression(**kwargs))
X_test = np.linspace(-0.1, 1.1, 500)[:, None]
```

The plot-

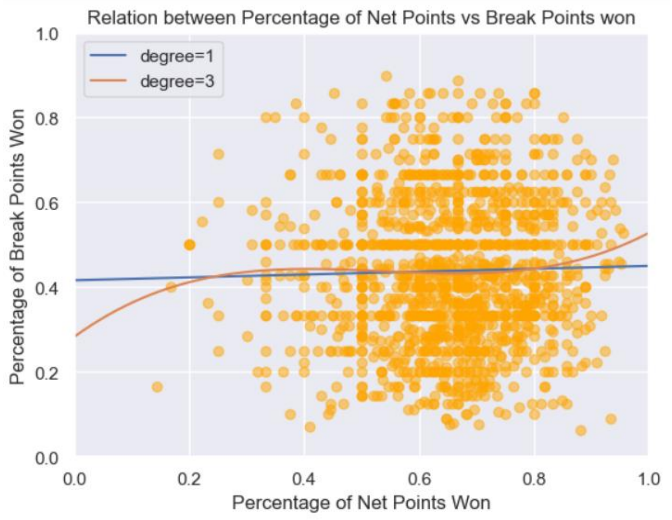

Figure 1: Relation between percentage of Net Points vs Break Points Won

The correlation coefficient between % of Net points won and % of Break points won is `0.02582502873.`

As the graph and the correlation coefficient suggest, there is no strong correlation between the percentage of Net points won, and percentage of Break points won.

Conclusion –

The hypothesis is not valid. A player skilled at scoring Net points may not be so at scoring break points. They are nearly independent variables.

Question 8)

Hypothesis – As the player's experience level increases, the number of Ace points won per match increases.

Approach –

A player's experience is decided by the number of winners, so we consider the relationship between the number of winners and the number of Aces won per match.

Make a ML pipeline using sklearn.pipeline's make_pipeline. Into this, append sklearn's PolynomialFeatures. Use this to perform polynomial regression. Use sklearn's LinearRegression to perform linear regression.

Perform polynomial regression to fit a line and a polynomial of degree 3, then plot the scatterplot and the curves. Find the correlation coefficient.

Answer –

The code for polynomial regression remains the same as in the previous question's answer

The correlation coefficient between the player's experience and the number of Aces won is `0.4213.`

As the graph and the correlation coefficient suggest, a strong correlation exists between the player's experience and the number of Aces won per match.

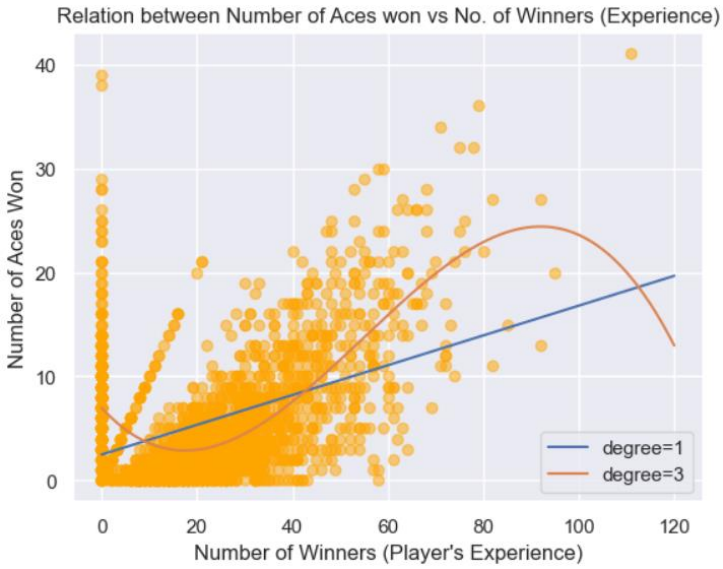

Figure 2: Relation between the Number of Aces Won per match and the player's experience.

Conclusion – The hypothesis – "*As the player's experience level increases, the number of Ace points won per match increases*" holds true.

Question 9)

Machine Learning Model **-** Design a simple machine learning model to predict the outcome of a tennis match by taking in only the first and second serve percentage and the number of unforced errors, net points, breakpoints and forced errors statistics.

Approach –

Perform clustering using sklearn.cluster's KMeans.

Test the model's accuracy using sklearn.model_selection's train_test_split and accuracy_score

Answer – The code -

```
from sklearn.model_selection import train_test_split
df = df.fillna(0)
train = df.filter(items=['FSP.1',
       'FSW.1', 'SSP.1', 'SSW.1', 'ACE.1', 'DBF.1', 'UFE.1', 'BPC.1',
       'BPW.1', 'NPA.1', 'NPW.1', 'FSP.2', 'FSW.2', 'SSP.2', 'SSW.2',
       'UFE.2', 'BPC.2', 'BPW.2', 'NPA.2', 'NPW.2'])
target = df.loc[:,"Result"]
X,XX,y,yy = train_test_split(train,target)

from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score

model = KMeans(n_clusters=2)
model.fit(X,y)
y_model = model.predict(XX)
print("Accuracy of the model is nearly: ",end="")
print(accuracy_score(yy,y_model)*100,"%")
```

Explanation – The model clusters the data into 2 clusters (1 if player 1 won and 0 if player 2 won). We check the accuracy by splitting the data available and training our model on only one part. The other part is used to test the model. This is hence supervised Machine Learning.

Accuracy – The model gave an accuracy of nearly 60%.

V. SUMMARY OF THE OBSERVATIONS

The Tennis Major Statistics datasets were mined to answer 8 questions, including 2 hypotheses and a Machine Learning model. The overview is –

- The probability that a player wins the game even if the number of unforced errors exceeds 20 is: `0.42488038277511964.`
- The game Daniel Brands vs Gilles Simon in Round 1 of the Australian Open 2013 lasted for all 5 sets and had a total of 461 points scored, making it the most interesting game of the year 2013.
- Sara Errani was out of the Australian Open 2013 tournament in the first round itself but worked hard to make it to the finals of another major tournament in the summer.
- The probability that a player who loses the first set wins the match is: `0.17709437.`
- The least errors among winners were committed by M. Bartoli, who made only 169 errors (including double faults and unforced errors).
- The probability that a player commits more errors than his opponent but wins the game is 0.493107.
- A player skilled at scoring Net points may not be so at scoring break points. They are nearly independent variables.
- The hypothesis – "*As the player's experience level increases, the number of Ace points won per match increases*" holds true.