Data Analysis of Tennis Matches

Shambhavi Agrawal
Civil Engineering
IIT Gandhinagar
Gandhinagar, India
shambhavi.agrawal@iitgn.ac.in

***Abstract*—This paper analyses eight datasets containing information about tennis matches, the results, the percentage of different types of serves and their winning percentage. It answers scientific questions and proves/disproves hypotheses about the dataset using python codes, plots, graphs etc. A few unanswerable questions are included as well.**

I. Overview Of The Dataset

The datasets include a list of the names of the pairs of players who have played against each other, along with the data of different serves and types of points scored. It contains information about the first-serve and second-serve percentages and the wins related to them. The number of aces won, double faults committed, and winners earned. It tells about the number of winners earned, unforced errors committed, breakpoints created and won. It shows the ability of the sportspersons by giving the net points attempted and won by the particular player. It includes the total points won by each player in the match and the set results for each player. It also has the final number of games won by each player and hence the final result of the match. This dataset is very useful for analyzing the different techniques in tennis.

II. Scientific Questions And Hypotheses

A. *Does the total wins of a player depend on the number of first serves made by the player? How do the wins due to the first serve affect the total win?*

If a player makes a successful first serve, then it can be said that he/she is a good player since they have better accuracy and did not need a second chance to make the serve. Can we deduce from this fact that these players will have a better chance of winning? Also, if these two parameters do not have a direct relation, then can we say that a player who scores more wins on the first serve will lead to a better chance of winning?

B. *How do the total points scored by a player in a match depend on the unforced errors made by the player? What is the correlation between them? How does it relate to the average of the points scored in all the matches?*

Usually, we feel that a player who makes more number of blunders in a match will not be a very good player. Players who can handle the pressure and keep their cool in the match will have fewer chances of making blunders. Losing points due to a player's own silly mistake rather than due to the skill of the opponent might be due to nervousness or less attention. Does this mean that such players will have a lesser number of total points in the match? Can we generalize a correlation between these two parameters for all players? How does this correlation vary with the average of the total number of points? How does the correlation depend on the unforced errors committed?

C. *What part of the total points scored by a player in the match is the net points and the ace points? How does the percentage vary for good players and inexperienced players?*

How much do the net points account for out of the total points scored by a player? What part of the total points were scored by scoring ace points? Do better tennis players have a strategy of scoring more net points or ace points? Can we deduce a pattern from the percentage of different types of points scored for good and bad players? Where do the inexperienced players fail in making their strategy regarding these points? How can they improve upon their strategy by learning from the good players?

D. *Do players committing more faults on serves score fewer points in total? What is the relation between the double faults committed by a player and the total points scored by him/her?*

If a player is not able to make a correct serve in both the chances given, then it shows that they lack accuracy or are not able to manage the force they apply etc. This shows that the player might need more practice or is not a very experienced player as he/she is not able to make good decisions during the match. Hence, can we say that these players will not score very well points in the match?

E. *Can we predict if a player has a chance to win a match with respect to the break points he wins?*

If a player creates more number of breakpoints then he/she has a very good chance of winning the match as he/she is very close to the final score. It shows that both the players playing the match are very good players and are playing competitively. Is there a trend that we can identify between the number of breakpoints won by a player and the final result of the match? To what accuracy can we predict the results of matches based on this parameter?

F. Do players who make more second serves and win those serves have better chances of winning the final games?

Players who make a second serve mean that they have already made a wrong serve on the first chance. Can we say that these players are not very good players since they needed a second chance to make a correct serve? Or do the wins scored on this serve point out to a strategy made by the player? What is the relation between the number of wins scored on the second serve say about the winning chances of the player?

G. Can we estimate which of the two players will win a match depending on data regarding the winners earned by each player?

If a player scores more winners, then it can be generalized that the particular player is a good player. Can we identify a pattern between the number of winners scored by each player in the match and the final result of the match? Following this pattern, can we predict the winner of a match, given the winners scored by two players?

H. Can we identify the winner of a match depending on the most principal aspects of tennis?

In the dataset, there are a number of parameters and scores of a tennis match given. We want to sort out this data and identify the parameters which affect the final result of the match. This helps players in making a good strategy. Can we predict the winner of a match based on these principal parameters?

III. Libraries And Functions

To analyse the given data and answer scientific questions and hypotheses, a few Python libraries need to be installed and then imported to help write the code and plot graphs. A few of the Python libraries used to assess the dataset are Matplotlib, Pandas, NumPy, scikit-learn, and Seaborn. Matplotlib is extremely useful for plotting graphs and curves to visualise the data. It has many features, such as labelling the axes and giving titles, colours, and sizes. Various graphs like bar graphs, double bar graphs, line graphs, scatter, histograms, 3D plots, 2D plots with three variables, pie charts etc., can be plotted using Matplotlib. The Axes3D module helps plot 3D graphs. Pandas is an extremely useful tool for data analysis. It has many functions that can help organise and view the data from different perspectives. It is used to read and change the dataset format into the desired one using functions like DataFrame and Series. The mean value (mean), standard deviation (std), variance (var), maximum (max), and minimum (min) of a dataset can be easily calculated using the respective functions written inside the parenthesis. It can help plot various types of graphs. NumPy helps create arrays and matrices and perform basic mathematical operations. Sci-kit learn is a very good library that helps identify patterns in data and make predictions about it. It is used to identify groups in data and even identify the important features of data. It is very useful in analysing data. Seaborn is a library that is very useful for making good graphs and visual representations to analyse data in different ways. Another library that would be useful in the analysis would be scipy for optimizing data.

IV. Analysing Data and Answering Questions/Hypotheses

A. Relation between the number of first serves, the wins scored by it and the total wins of a player.

1) Procedure

First, we sort the dataset of the Australia Open Men - 2013 to create different series of the first serve, the first serve wins and the final result of both the players. To increase the number of data points in the dataset, we append the second player's data points to the first player's data points. We then plot the relation between the first-serve percentage and the total wins. We also plot the relationship between the first serve percentage and the wins due to the first serve. To check if there is any relation between the data sets, we fit the data points to a linear regression model and plot a line. The following code snippet can be used:

```
aus_men = pd.read_csv("AusOpen-men-
2013.csv")
unique1 = aus_men.Player1.value_counts(
)
idc1 = unique1.index
fs_pct1 = pd.Series(np.zeros(len(idc1))
, index=idc1)
for player in idc1:
  fs_pct1[player] = aus_men[aus_men['Pl
ayer1']==player]['FSP.1'].mean()
df_fs_pct = pd.DataFrame(fs_pct1.append
(fs_pct2, ignore_index=True))
fs_pct = df_fs_pct.iloc[:,0]
model = LinearRegression()
xfit = np.linspace(40, 90, 1000)
plt.plot(fs_pct, fs_win, ".")
model.fit(fs_pct[:, np.newaxis], fs_win
)
yfit2 = model.predict(xfit[:, np.newaxi
s])
plt.plot(xfit, yfit2)
```

2) Observation and Results

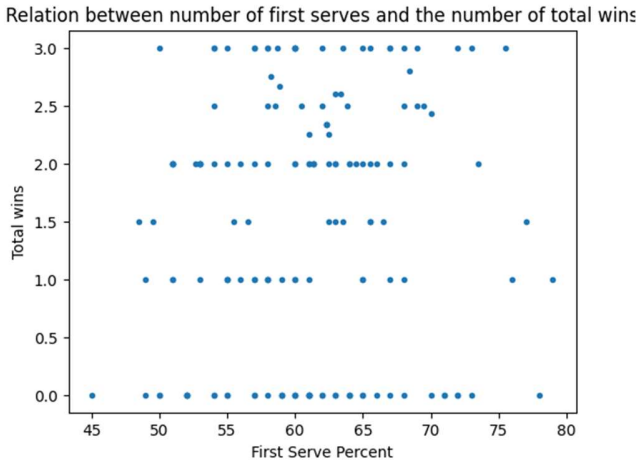

Fig. 1. Scatter plot showing the relation between the number of first serves and the number of total wins

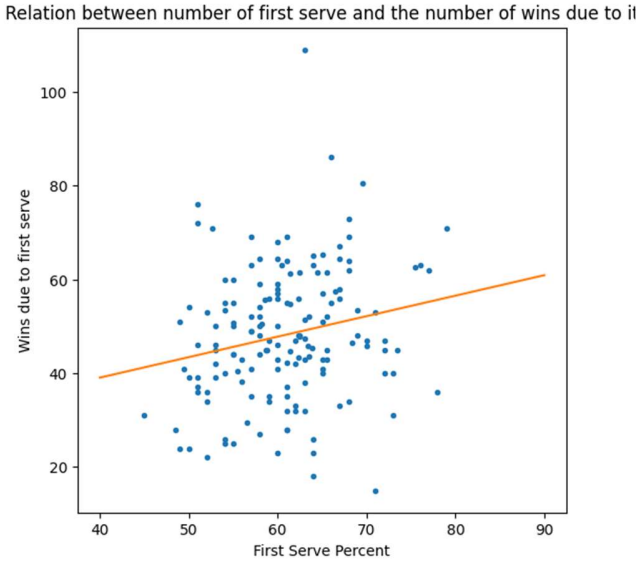

Fig. 2. Scatter plot relation between the number of first serves and the number of wins due to it

In Fig. 1. we can observe that there doesn't seem to be any valid relation between the number of successful first serves made by the player and the total number of wins. We try to check then if we could deduce anything better from the graph between the number of successful first serves made by a player and the number of wins in the game due to this first serve. In Fig. 2. There still doesn't seem to be much valid reasoning we can conclude, although if we do a linear regression of the points plotted, it gives us a faint idea that the number of wins due to the first serve increases as the number of successful first serves made by a player increases. We can say that a player who makes more successful serves is a better tennis player and hence has more chances of scoring wins on the serve. Though one must note that there is much variance in the data points plotted, and hence linear regression is not the best way to point out any significant conclusion. Overall, we can say that our hypothesis is proved to be wrong as there doesn't seem any direct relation between the successful first serves made and the winning chances of a player. This might be due to the fact that each player has a different strategy, some might not always tend to make a successful first serve but rather use the second serve or maybe some don't focus on scoring points on the serve.

B. Correlation between the number of unforced errors committed and the total points scored by a player

1) Procedure

We first divide the dataset containing information about the Australia Open women-2013 into two parts, containing information about the first player and the second player. We create different series containing information about the unforced errors committed and the total points scored. The correlation is then computed between these two parameters, for which the Pearson method is the default setting. We then plot this correlation with the average of the unforced errors, and then the average of the total points scored. We also plot the boxplot of the correlation coefficients to get information about the mean, variance and range etc.

```
aus_women = pd.read_csv("AusOpen-women-2013.csv")
unique1 = aus_women.Player1.value_counts()
idc1 = unique1.index
fp_error = pd.Series(np.zeros(len(idc1)), index=idc1)
fp_points = pd.Series(np.zeros(len(idc1)), index=idc1)
for player in idc1:
  fp_error[player] = aus_women[aus_women['Player1'] == player]['UFE.1'].mean()
plt.figure(figsize=(6, 10))
plt.subplot(311)
plt.plot(df_corr, df_error, '.')
plt.subplot(313)
boxplot = df_corr.boxplot()
```

2) Observation and Results

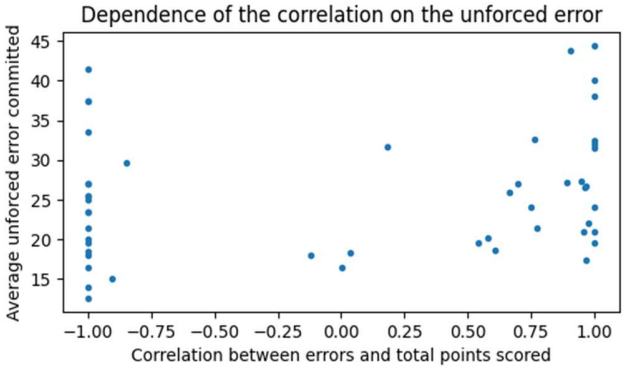

Fig. 3. Scatter plot depicting the correlation between the errors and total points with the average unforced errors committed

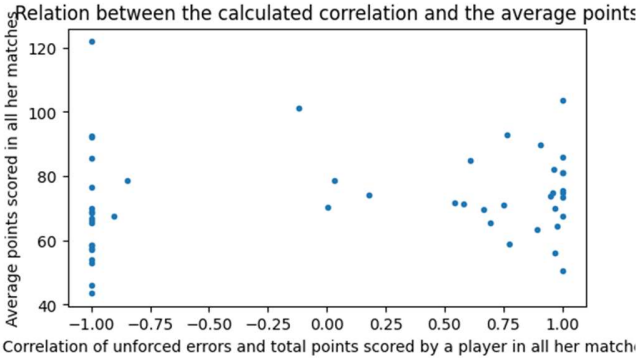

Fig. 4. Scatter plot depicting the correlation between the errors and total points with the average total points scored

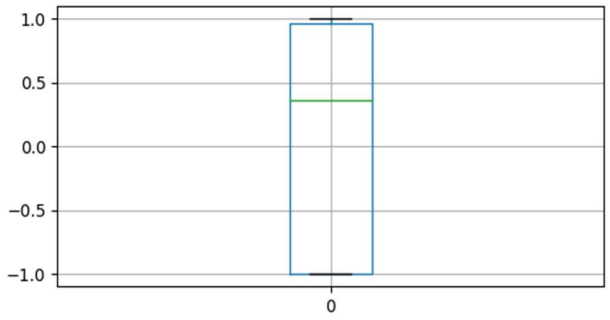

Fig. 5. Boxplot of the correlation matrix

From Fig. 3. we can deduce that the correlation calculated for the unforced errors committed and the total points scored by a player does depend on the unforced errors committed. Although there is some variance, we can see that most of the points are plotted on the extremes of the x-axis. This shows that there is a strong correlation and dependence of it on the unforced errors but for some players it is extremely positive and for some extremely negative. There are a few players who come in between these extremes as well. We can deduce a similar pattern for the relation between the correlation computed and the average total points scored, which is shown in Fig. 4. From Fig. 5. we can see that the median correlation value is approximately 0.35, which is a positive value, and hence quite a lot of data points have a positive correlation which in fact deviates from our hypothesis that more the number of unforced errors committed by a player, less the number of total points scored.

C. Distribution of net points and ace points in the total points scored in a match. Comparison of this distribution for good and inexperienced players

1) Procedure

We divide the dataset of French Open men-2013 into two parts, one containing information about the first player and the other about the second player. We then create individual series containing the sum of all the total points scored by a particular player. To identify the good players, we sort out the players having a sum of total points greater than 500 and, for the inexperienced players, we set this number to be around 50 to 60. For these particular players, we calculate the sum of the net points scored, and the sum of the ace points scored in all the matches played by them. . To increase the number of data points in our dataset, we merge the respective series containing information of both the players. We then stack plot a bar graph to visualize these distributions. The following code snippet can be used:

```
french_men = pd.read_csv("FrenchOpen-
men-2013.csv")
unique1 = french_men.Player1.value_coun
ts()
idc1 = unique1.index
fp_tp = pd.Series(np.zeros(len(idc1)),
index=idc1)
for player in idc1:
  fp_tp[player] = french_men[french_men
['Player1']==player]['TPW.1'].sum()
fp_tp_gt = fp_tp[fp_tp.gt(500)]
fp_tp_lt = fp_tp[fp_tp.lt(50)]
for player in indcs_gt_1:
  fp_ace_gt[player] = french_men[french
_men['Player1']==player]['ACE.1'].sum()
  fp_np_gt[player] = french_men[french_
men['Player1']==player]['NPW.1'].sum()
column_names = ["Ace points", "Net poin
ts", "Other points"]
left_gt = tp_gt-(ace_gt+netp_gt)
fig = plt.figure()
ax_gt = fig.add_subplot(121)
points_gt.plot.bar(stacked=True, ax=ax_
gt)
```

2) Observation and Results

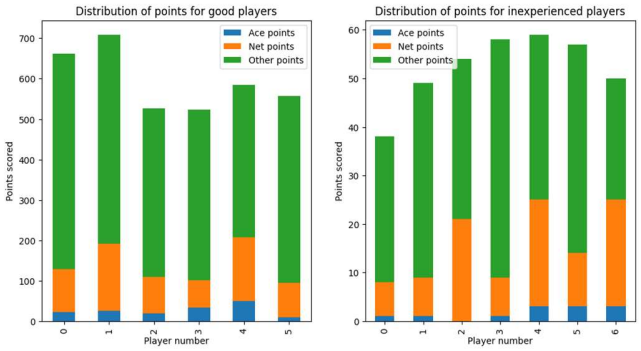

Fig. 6. Side-by-side comparison of the distribution of net points and ace points out of the total points scored both for excellent and inexperienced players

We have assumed that the players who have a high value of the sum of the total points scored in all matches must be excellent and experienced players, while the players who have a lesser value of this sum can be considered inexperienced players. From the stacked bar graph plotted in Fig. 6. we can see the difference in the total number of points for both these types of players has a huge difference. Although from the graph, we can't really point out a significant difference in the distribution of the different ways in which the points are scored. For a few inexperienced players, we can notice that they score more net points and less number of ace points than the experienced players, although we can't generalize this conclusion for both sets of players. One interesting fact to notice is that there is not much difference in the distribution of the points for different players, which shows that there maybe is a similar strategy that all the players follow. It seems that the points scored by giving an ace during the service are significantly few for all the players, maybe because since it is a point scored during the service, the player can score it only when he/she has the chance of making the service, or maybe it's because it is difficult always to serve an ace with accuracy or maybe the players want to change the type of services they want to make every time and as a result don't always give ace serves. Since the net points can be scored during any part of the game, it constitutes a significant portion of the total points. Though we could not see any significant difference in the distribution of the points, this was an interesting observation that aligns with logic.

D. Relation between double faults committed and the total points scored by a player

1) Procedure

We divide the dataset of French open women-2013 into two parts, each containing information about one of the players. Then the mean of the double faults committed by each player in all the matches and the mean of the total points scored by a player in all the matches is calculated. Using seaborn, we then plot a joint plot between these two variables to show the joint distribution and the marginal distributions. The following code snippet can be used:

```
french_women = pd.read_csv("FrenchOpen-women-2013.csv")
unique1 = french_women.Player1.value_counts()
idc1 = unique1.index
fp_tp = pd.Series(np.zeros(len(idc1)), index=idc1)
for player in idc1:
  fp_tp[player] = french_women[french_women['Player1']==player]['TPW.1'].mean()
  fp_dbf[player] = french_women[french_women['Player1']==player]['DBF.1'].mean()
df_tp = pd.DataFrame(fp_tp.append(sp_tp, ignore_index=True))
tp = df_tp.iloc[:,0]
column_names = ["Double Faults", "Total Points"]
faults = pd.concat([dbf, tp], axis=1)
faults.columns = column_names
with sns.axes_style('white'):
  sns.jointplot(x="Double Faults", y="Total Points", data=faults, kind='hex')
```

2) Observation and Results

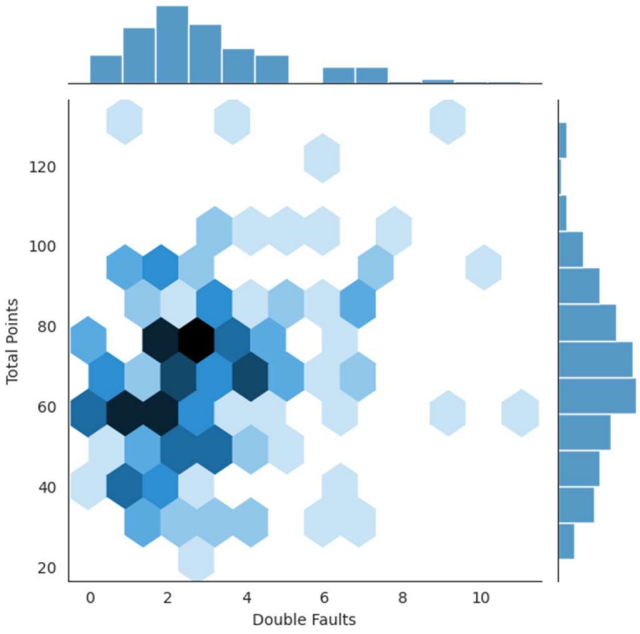

Fig. 7. Joint plot depicting the relation between double faults committed and the total points scored

From the joint plot in Fig. 7. we can make observations based on the joint distributions and the marginal distributions of the double faults committed and the total points scored. From the marginal plots, we can see that most of the players commit around two to three double faults in the game. From the joint plot, it is surprising to note that very few players who have committed a large number of double faults have, in fact, scored pretty well in terms of total points. For the players committing very few double faults, it is difficult to generalize anything from the graph as there is a range of values of the total points that they have scored. Most players have committed around 1-3 double faults and score around 60-80 total points in the matches. Thus our hypothesis that the players committing lesser double faults will be good tennis players and will hence score more total points, is proved to be wrong as the data points have high variance.

E. Analysing the break points won and the final results of players and estimating if a player will win a match

1) Procedure

First, convert the necessary columns of the dataset of US Open men-2013 to NumPy arrays. With the help of scikit-learn, fit the data of the breakpoints won and the result of the match of the first player to the Gaussian naïve Bayes model. Then, calculate the accuracy score of estimating the result of the second player using the model. Scatter plot the training set for this model i.e. the data of the first player. Also, plot the testing set and the prediction of this model i.e. the data of the second player. In both plots, when the result of the match is zero, then it means that the player lost that match, but if the result is one, then it means that the player won this match. The following code snippet can be used:

```
us_men = pd.read_csv("USOpen-men-2013.csv")
bpw1 = us_men['BPW.1'].to_numpy()
result = us_men['Result'].to_numpy()
bpw2 = us_men['BPW.2'].to_numpy()
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(bpw1[:, np.newaxis], result)
ynew = model.predict(bpw2[:, np.newaxis])
res2 = []
for ele in result:
  if ele==0:
    res2.append(1)
  else:
    res2.append(0)
from sklearn.metrics import accuracy_score
print(accuracy_score(res2, ynew))
plt.figure()
plt.subplot(211)
plt.scatter(bpw1, result)
plt.subplot(212)
plt.scatter(bpw2, ynew)
plt.plot(bpw2, res2, "r.")
```

2) Observation and Results

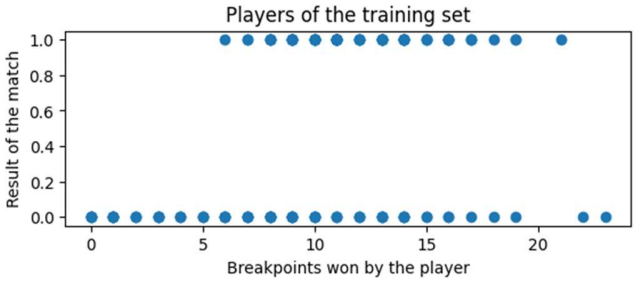

Fig. 8. Plot showing the relation between the number of breakpoints won by a player and the result of the match for the players in the training set.

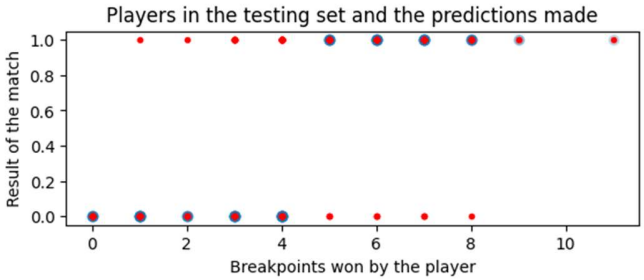

Fig. 9. Plot showing the relation between the number of breakpoints won by a player and the result of the match for the players in the testing set (blue) and the predictions made (red)

In Fig. 8. we can observe that for a very low number of break points won, the player eventually loses the match, but we can't generalise anything for break points won more than 5. After training the Gaussian Model on the data in Fig. 8., we predict the result of other players in Fig. 9. The blue points are the actual results with dark blue showing the density of coinciding points, and the red points are the ones we have predicted. Overall, this model gives us an accuracy of 69.04%, which is not a very good number. The reason is clear from Fig. 8. That there is no concrete relation that can be established between the number of breakpoints won by a player and the result of the match.

F. Relation between the successful second serves made by a player and the wins scored by it and the final result of the match.

1) Procedure

Using the US Open women-2013 dataset, plot a two-dimensional scatter graph of the three variables that are second serve per cent, second serve wins and the final result of the match. To increase the data points in the graph, including the data of both the first player and the second player. The following code snippet can be used:

```
us_women = pd.read_csv("USOpen-women-2013.csv")
plt.scatter(us_women['SSP.1'], us_women['SSW.1'], lw=0.1, c=us_women['FNL.1'])
plt.scatter(us_women['SSP.2'], us_women['SSW.2'], lw=0.1, c=us_women['FNL.2'])
plt.colorbar()
```

2) Observation and Results

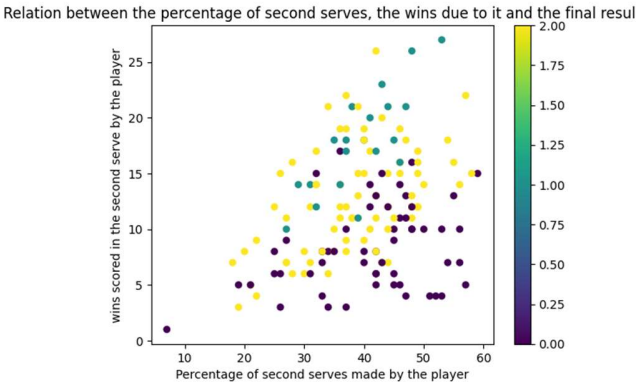

Fig. 10. Two-dimensional scatter plot showing the relation between the percentage of second serves, the wins due to it and the final result

From the graph in Fig. 10. it is clear that there is no relation between the final result of the match and the percentage of second serves made by a player. Although, some patterns can be deduced from the relation between the wins scored in the second serves and the final result of the match. We can see that the yellow points are scattered throughout the plot, but the points close to the x-axis have a majority of purple colour, while the points far from the x-axis have a majority of greenish-blue colour. This shows a faint trend that the number of matches won finally increases as the number of wins scored in the second serve by the player increases. Also, we can see that it is not necessary that players score more wins if they have made more second serves, i.e. maybe some players get nervous and focus only on getting the service right and do not intend on scoring as they already made a fault in the first serve, whereas some players score more wins as the number of serves increases, maybe due to practice and strategy. It is evident that if the number of second serves made by a player is significantly less, than the number of wins scored in that service will also be less. Thus, the graph proves a very small part of our hypothesis.

G. Estimating the winner of a match based on the data of the winners earned by both the players

1) Procedure

First, we convert the dataset of the Wimbledon men-2013 tournament to a data frame. Then, we convert the desired columns to a NumPy array. The desired columns are the winners scored by both the players and the result of the matches. We then apply the Gaussian naïve Bayes model to this data to train and estimate the results. We apply the cross-validation score with the set number equal to five to obtain better accuracy. Compute the score for each set and print it. Plot the scatter plot for the three variables in two dimensions. The following code snippet cam be used:

```
wim_men = pd.read_csv("Wimbledon-men-2013.csv")

win1 = wim_men['WNR.1'].to_numpy()
win2 = wim_men['WNR.2'].to_numpy()
result = wim_men['Result'].to_numpy()

win = np.column_stack((win1, win2))
from sklearn.model_selection import cross_val_score
model = GaussianNB()
print(cross_val_score(model, win, result, cv=5))

plt.scatter(win1, win2, lw=0.1, c=result)
plt.colorbar()
plt.xlabel("Winners earned by the first player")
plt.ylabel("Winners earned by the second player")
plt.title("Relation between the winners earned by both the players and the result of the match")
```

2) Observations and Results

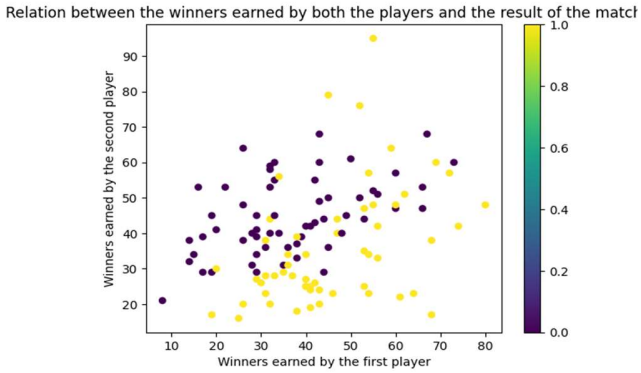

Fig. 11. Scatter plot depicting the relation between the winners earned by both the players and the result of the match

From Fig. 11. we can observe that when the winners earned by both players are approximately same, there is a fair chance of both the players winning the match. When the winners earned by the first player are more than the second player, the points in the graph are yellow, which means that the first player won the match (yellow signifies 1, which means that the first player won the match). Similarly, the purple points signify that the second player won the match. Although there are some anomalies in the graph, the basic idea that a player earning more winners finally wins the match is somewhat proven from this graph. Dividing the dataset into five parts and training and testing a Gaussian Naïve Bayes model on each part of it, we predict the winner of a match based on the number of winners earned by each player. The accuracy score for each test was approximately 82.6%, 86.9%, 78.2%, 78.2% and 63.6%. The mean value of this is 78%, which is not a very good prediction accuracy, and this deviation can be attributed to the anomalies in the plot.

H. Analysing the data to identify the most principal features contributing to the win of a player and predicting winners from data.

1) Procedure

From the Wimbledon-Women 2013 data, we create a dataframe consisting of features like final scores, first and second-serve percentages and wins and aces scored. Now using PCA, we reduce the dimensionality of this data to two. We plot the points according to the reduced dimensions along with the result of the matches. The blue points correspond to the result 0, meaning that the first player lost the match, whereas the orange points represent the first player winning the match. The following code snippet can be used:

```
wim_women = pd.read_csv("Wimbledon-women-2013.csv")
df = wim_women[['Result', 'FNL.1', 'FNL.2', 'FSP.1', 'FSW.1', 'SSP.1', 'SSW.1', 'ACE.1', 'FSP.2', 'FSW.2', 'SSP.2', 'SSW.2', 'ACE.2']].dropna(axis=0)
mat = df.to_numpy()
print(mat.shape)
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(mat)
X = pca.transform(mat)
df['PCA1'] = X[:, 0]
df['PCA2'] = X[:, 1]

sns.lmplot(x="PCA1", y="PCA2", hue='Result', data=df, fit_reg=False)
```

2) Observation and Results

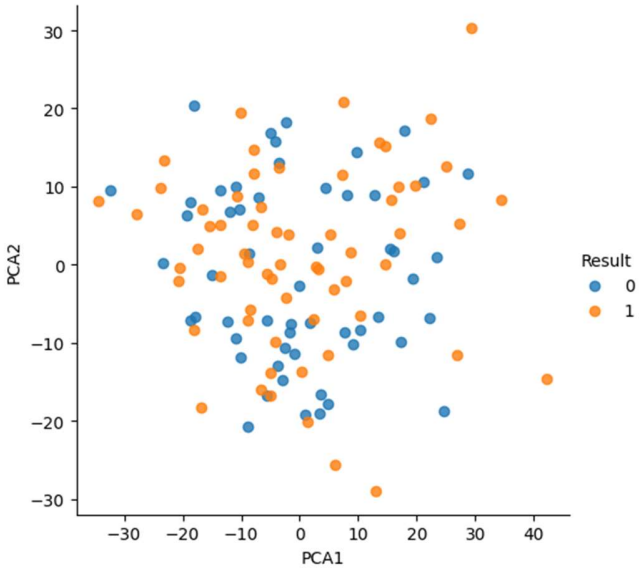

Fig. 12. Graph showing the result of the match based on the dataset reduced to two dimensions.

From Fig. 12. we cannot deduce any pattern for the result of the match from the reduced dimensionality dataset. It seems like there is much variance in the data points as the points are too scattered in the graph. Hence our hypothesis stating that there should have been (at least) a vague demarcation for the results, but we see that this condition is not satisfied here. Hence, this shows that maybe reducing the data to two dimensions is not a good practice as it eliminates the possibility of any relation between the result of the match and the features in the dataset

V. Summary Of The Observations

Using the data about the different tennis matches, we analysed how different scores, points, errors etc., contribute to a player's

win in a match. We analysed the number of successful first serves and second serves, the wins scored due to them and the effect they have on the total points scored and the final result of the match. Also, we observed that double faults and unforced errors committed do not have much effect on the final result of the match as the data points have high variance. We attempted to predict the winner of a match using different machine learning models based on information like the number of breakpoints scored, the number of winners earned etc. The distribution of different types of points scored by good and bad players did not show any significant variation. The data set of the tennis matches is very useful to improve strategies for game playing and predicting the result of a match.

VI. Unanswered Questions

1) Can we group players based on their playing style? Can we see any relation between these groups with the final number of wins? In other words, to a group of players having a specific style of playing have more probability of winning a match?

2) Can we predict the outcome of the final match of a player based on the information from the previous matches?

3) Is there any difference between the overall performance of male and female players?

4) As the rounds progress, does it affect the player's performance? Does the level of competition increase as we move on to the next round?

5) Is there a relationship between the number of sets played and the final outcome of the match?

Acknowledgement

The author would like to thank Prof. Shanmuganathan Raman and his entire team for their support and guidance in this project. The author would also like to extend her gratitude to Mrigankashekhar Shandilya for reviewing the report.

References

[1] "Pandas Documentation#." pandas documentation - pandas 1.5.3 documentation. Accessed April 15, 2023. https://pandas.pydata.org/docs/.

[2] "Matplotlib 3.7.1 Documentation#." Matplotlib documentation - Matplotlib 3.7.1 documentation. Accessed April 15, 2023. https://matplotlib.org/stable/index.html.

[3] NumPy documentation. Accessed April 15, 2023. https://numpy.org/doc/.

[4] "Seaborn: Statistical Data Visualization — Seaborn 0.12.2 Documentation". Accessed April 15 2023. https://seaborn.pydata.org/.

[5] "Scikit-Learn: Machine Learning in Python — Scikit-Learn 1.2.2 Documentation," Accessed April 15, 2023. https://scikit-learn.org/stable/.