Tennis: What separates a win from a loss ?

Data narrative on a tennis dataset

Pranav Joshi
CSE IITGN
Gandhinagar, India
pranav.joshi@iitgn.ac.in

***Abstract*—In this document, I have done analysis of the Tennis Major Tournaments Match Statistics data-set using popular methods like covariance, PCA, Naive Bayes and Linear Classificatier.**

***Index Terms*—Covariance, PCA, Naive Bayes, Linear Classificatier**

I. Introduction

The Tennis Major Tournaments Match Statistics data-set [1] contains 8 c.s.v. files, 4 for men's tournaments, 4 for women's , from which I extracted the data for all the men's tournaments which is the data I am using throughout this data narrative.The data is values for different variables like "Player 1", "Player 2","Result",etc. for each match. I removed the variables which were either string type data, binary data, redundant, or had too many incorrect or missing entries, and then rows with incorrect or missing entries, to get 21 variables, along with the 'Result' variable, which is binary data, where 0 corresponds to Player 2 winning and 1 corresponds to Player 1 winning.

II. Questions and Objectives

- What factors could indicate winning a match?
- Distribution of matches according to most important variables
- Dependence of important variables on other variables
- Better way of predicting the result of a match
- Probability based method for predicting the result of match.
- Explanation for average accuracy of probabilistic method.

III. What factors indicate winning a match ?

By marking every victory for player 1 as 1 and loss as -1, instead of 0 for the 'result' variable, we can generate the covariance values of different parameters (scaled down by their standard deviation) with the 'result' variable.

Variable	covariance	Index
BPW.1	0.646120	8
BPW.2	−0.627855	18
BPC.2	−0.514476	19
BPC.1	0.508826	9
WNR.2	−0.328569	16
ACE.2	−0.270095	14
ACE.1	0.241339	4
WNR.1	0.225747	6
UFE.1	−0.201056	7
UFE.2	0.180922	17
FSW.2	−0.179655	12
DBF.2	0.139628	15
SSW.2	−0.135268	13
FSP.1	0.121506	1
FSP.2	−0.112612	11
FSW.1	0.099938	2
DBF.1	−0.096350	5
SSW.1	0.065182	3
NPA.1	−0.048894	10
Round	0.047018	0
NPA.2	0.019625	20

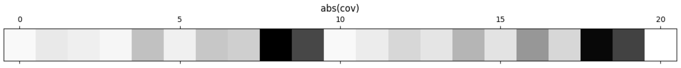

Fig. 1. Covariances of variables with result

As is evident from Fig. 1, the variable with index 8, 9, 18, 19 are the most important, as they have the biggest absolute covariances with the modified 'result' variable. These variables are:

- 8 : BPW.1
- 9 : BPC.1
- 18 : BPW.2
- 19 : BPC.2

Here BPC means Break Points Created and BPW means Break Points Won. The .1 and .2 mean "by player 1" and "by player 2".

Let's look at the distribution of matches using scatter plots with these variables. Here green means that player 1 wins, and blue means player 2 wins.

A. Distribution of matches w.r.t. Break Points Created

Break points are often pivotal in the outcome of a tennis match. Since the server is at and advantage, if one breaks the flow of the opponent's service game, then there's a chance of the match's outcome changing favourably. [2]

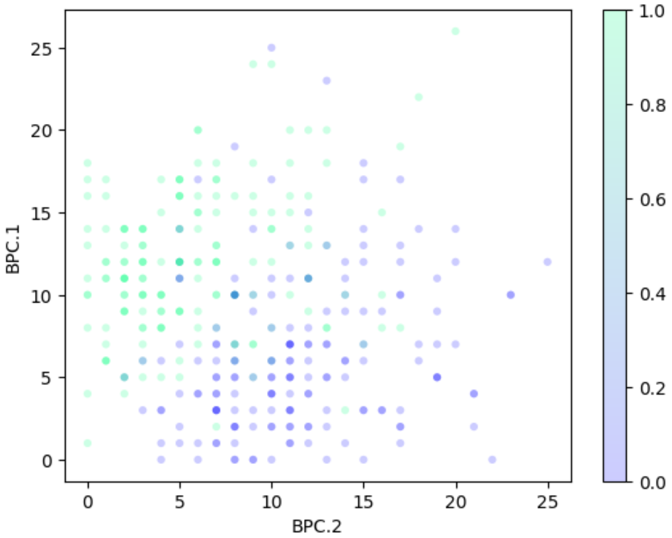

Fig. 2. Distribution of matches w.r.t BPC

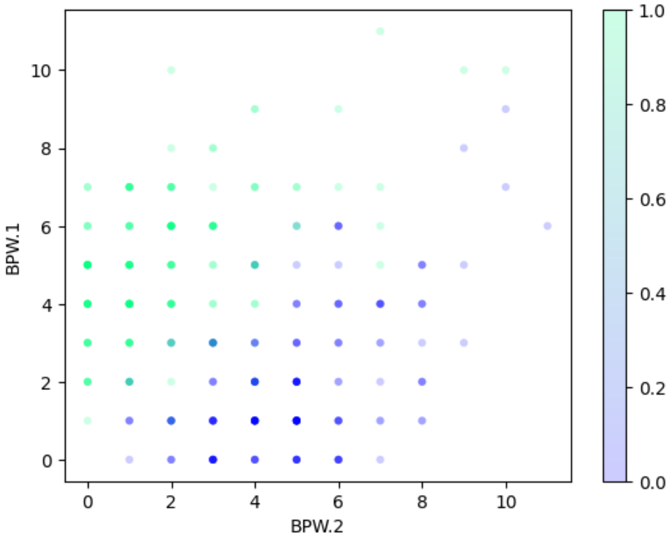

Fig. 3. Distribution of matches w.r.t BPW

B. Distribution of matches w.r.t. Break Points Won

Of course, just creating a break point isn't enough. Winning it is the next necessary step. It's quite common for players just having broken their opponent's serve to be broken back immediately. [2]

C. Dependence of BPC on BPW

Considering that the last 2 graphs were so similar, it's easy to guess that BPC and BPW are related, as is evident from Fig 4 and also the low covariance of BPC.1 and BPW.1, which is shown in the covariance matrix for our data in Fig 5 The cells (8,9) and (9,8) show the covariance of BPC.1 and BPW.1 .

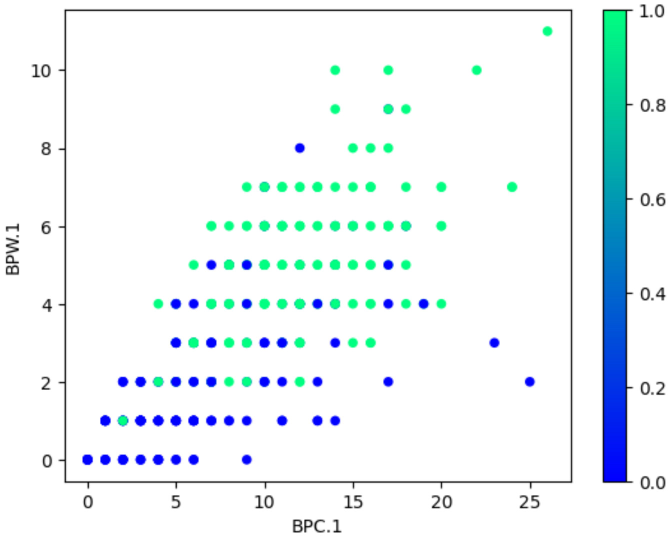

Fig. 4. Dependance of BPW on BPC

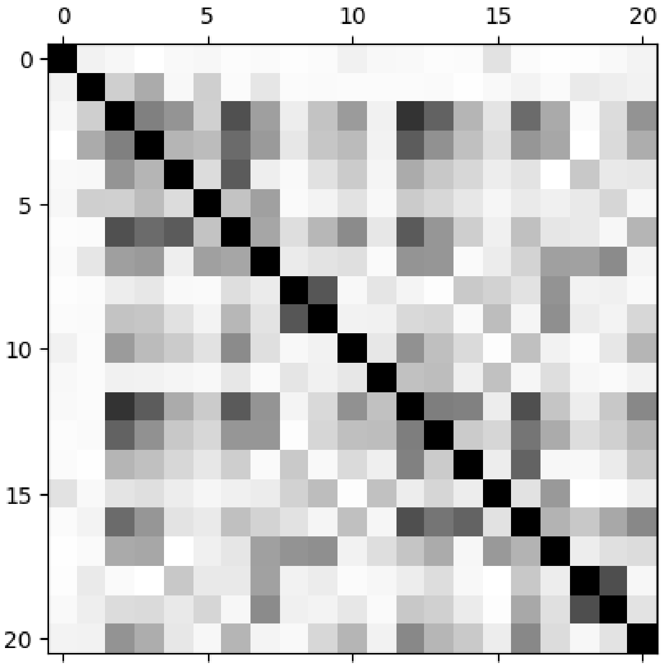

Fig. 5. Covariance Matrix

IV. Can we predict better than this ?

Rather than using two or three variable to predict the result, we can use the value of a linear function of all our variables. Consider x_i to be the i^{th} variable and y to be the result variable . Define $u_i = (x_i - E(x_i))/\sigma(x_i)$ to be the scaled version of the variable. And define $v = 2y - 1$ to be the result variable with -1 in place of 0. We want to saperate the clusters ($y = 0$ and $y = 1$) by a plane $d = c_0 + c_1u_1 + c_2u_2 + \ldots c_nu_n = 0$ where $n = 21$ is number of variables and $\sum c_i^2 = 1$
Here d, as a function of $u_1, u_2 \ldots$ is the signed distance from

this plane. The value dv gives us a measure of how 'typical' a certain point $(U_1, u_2, \dots)$ is, i.e. how much does its features match the typical features of its category. Thus for a good saperation, we want $E(dv)$ to be high. Since

$$E(dv) = E(v(c_0 + c_1 u_1 + \dots))$$
$$= c_0 E(v) + c_1 E(vu_1) + c_2 E(vu_2) + \dots$$

and $E(v) = 0$ in this case because in a match, which player is player 1 and player 2 is randomly decided, thus under the constraint $\sum c_i^2 = 1$, this value is maximized when

$$c_i = \frac{E(vu_i)}{\sqrt{\sum E(vu_i)^2}} \quad \text{for } i = 1, 2, \dots$$

And since c_0 doesn't contribute much, we can set it to be 0. Now, based of the sign of d for any point $(u_1, u_2, \dots)$, we can predict its v value. And thus $y = (v + 1)/2$

```
def CovS(X,y):
'''
X:data of shape (n,f) \n
y:binary (0 or 1) representation
   of results (Win or Loss)
   (for player 1).
   Shape: (n,)
'''
v = (2*y - 1)[:,newaxis]
mu = mean(X,axis=0)
d = sqrt(var(X,axis=0))
U = (X - mu)/d
cov = mean(v*U,axis=0)
def C(x):
    u = (x - mu)/d
    v_pred = sign(sum(u*cov,axis=-1))
    y_pred = (v_pred + 1)/2
    return y_pred
return C
```

The actual code I used has some additional features and different notation, but the main calculations are exactly the same.

This method gives an average accuracy of 92.33% with Leave-One-Out-Cross-Validation.

V. A PROBABILITY BASED METHOD OF PREDICTING

We can try using a Naive Bayes classifier for predicting the result of a match based on all the other parameters. But before that,, since Naive Bayes assumes probabilistically independent variables, the least we can do is make the variables linearly independent, which can be done by P.C.A. The matrix used for transforming old variables to new ones, as obtained from P.C.A. (but rows multiplied by their corresponding eigenvalues), converted to pixels is shown in Fig 6 with values being absolute values of entries and rows arranged from most to least significance (eigenvalues). And Fig 7 is the scatter plot for our data using the 2 most significant new variables : With only these 2 variables, the Gaussian Naive Bayes Classifier

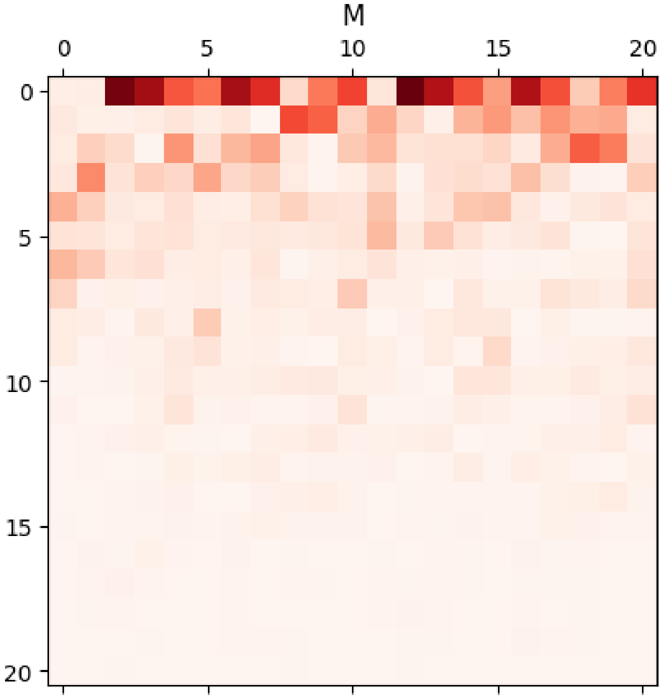

Fig. 6. PCA eigenvector matrix

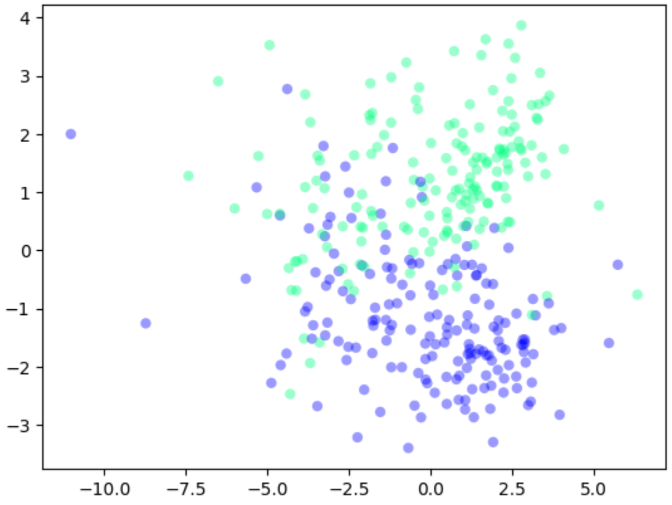

Fig. 7. Distribution in most significant 2 new variables

has 85.84% accuracy (using LOOCV). With 5 most significant new variables, we get accuracy of 91.74%. This is a list of average accuracy of GNB after PCA with different number of variables used:

variables used	Acurracy
1	48.37758112094395
2	50.442477876106196
3	86.72566371681415
4	88.20058997050147
5	88.7905604719764

6	89.08554572271386
7	89.08554572271386
8	89.6755162241888
9	88.49557522123894
10	89.08554572271386
11	88.7905604719764
12	88.20058997050147
13	88.20058997050147
14	87.31563421828909
15	87.61061946902655
16	90.56047197640117
17	90.56047197640117
18	90.56047197640117
19	89.97050147492625
20	89.97050147492625
21	90.56047197640117

Without P.C.A., and using all the variables, we get an accuracy of around 91.4% .

One guess for the reason behind such bad accuracy could be that the new variables are not having a normal distribution. But that's not the case . Fig 8 shows the histograms for most significant 5 of our new variables, with the orange curve being the scaled Gaussian distribution that we assumed when applying Naive Bayes. The real reason is that 0 covariance only implies linear independence, of our variables, not probabilistic independence.

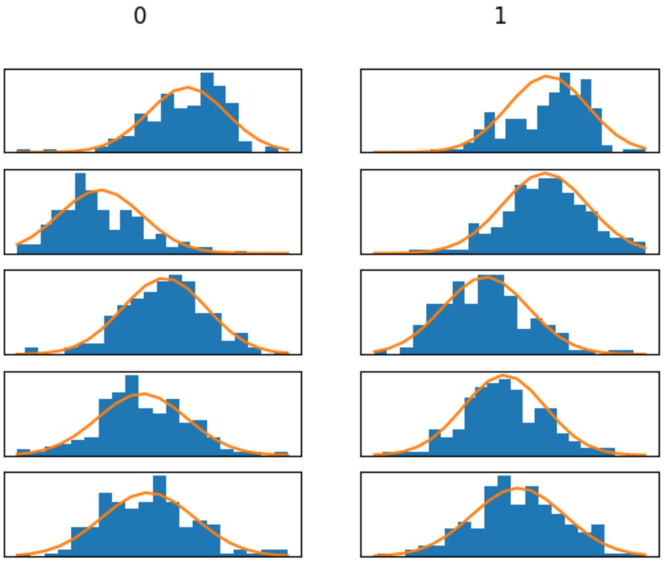

Fig. 8. Distributions of new variables

VI. Summary

- Break Points Created and Break Points Won by the players are the most important factors for predicting the outcome of a match.
- BPC and BPW are co-related.
- The linear classification works well for predicting the result of match
- Gaussian Naive Bayes doesn't work well in this case, because some variables are not probabilistically independent.

VII. Libraries and functions used

- matplotlib Python library
- pandas Python library
- numpy Python library

Acknowledgment

This document was made for an assignment in the ES114 (Probability, Statistics and Data Visualisation) course at IITGN. I want to thank the institute and prof. Shanmuga R. for giving me this opportunity. I also acknowledge the Data Centric Computing course which overlaps a lot with ES114 and provided me prior practice.

References

[1] Tennis Major Tournaments Match Statistics, UC Irvine Machine Learning Repository URL:https://archive-beta.ics.uci.edu/dataset/300/tennis+major+tournament+match+statis

[2] Break Point in Tennis, The Tennis Bros.com URL:https://thetennisbros.com/tennis-tips/tactics/break-point-in-tennis/