# Tennis  Grand Slam match statistics Data

- An Internship Report

Submitted by
Sai Teja Gollapinni Venkata
Batch 30
International School of Engineering,
L77, 15th Cross Road, Sector 6, HSR Layout, Bengaluru, Karnataka 560102

# Index

## Section 1
## Introduction

**Data Topic :** Tennis Major Tournament Match Statistics for Year **2013**

**Abstract**: The Data set contains all the statistics of the tournaments happening in a calender year(2013) which are comprised in 8 different **csv** files which wholly contains the Men and women singles Match data.

**Aim**: To Predict the performance of the player and how his point scoring ablility is helping him to score more matches along different tournaments i.e; whether the player is better on which type of tennis court (Hard court(Aus open, US open), Grass court(Wimbledon), clay court(French open) ).

**1.1 Data intuition and Attributes** :

**Result** is 1 :Player 1 won the match

0: Player 2 won the match

so FNL1>FNL2(also no should be 3 out of 5 for men and 2 out of 3 for women )

**FNL**: final no of games won by 1/2 in the match(which determines the winner)

**Round**: It is to determine level of games in the tournament

1.First Round, with 128 players (sixty-four matches)

2.Second Round, with 64 players (thirty-two matches)

3.Third Round, with 32 players (sixteen matches)

4.Fourth Round, with 16 players (eight matches)

5.Quarterfinals, with 8 players (four matches)

6.Semifinals, with 4 players (two matches)

7.Final, with the last two players playing for the title.

**FSP 1**: percentage of serve time player 1 served to player 2 first time

**FSW 1** : No of times player 1 won the points doing a first serve(This can be a primary attribute to win the match).

**FSP2** : % of serve time player 2 served to player 1 first time(this will be positive for any player and negative to opponent player)

**SSP 1** : percentage of time play 1 served to play 2 second time if faulted at first

**SSW 1** : no of times play 1 won the second serve(i.e without double faults)

**SSP2** : % of time play 2 served to play 1 second time(if 1st serve is a fault)

**SSW2** : no of times play 2 won the 2nd serve(without double faults)(This is also important and positive for that player and not to other)

**ACE 1** : ace points won by player 1( if play 1 gets point on the serve(no return shot) itself)**it is a subset of winner points**

**ACE 2**: ace points won by player 2 (if play 2 wins point in the serve itself)**it is a subset of winner points**

**DBF1**: Double faults committed by play 1( This is a negative impact on play 1(good for play 1) and it should be as low as poss)

**DBF2**: Double faults committed by play 2 (also negative impact on play 2 and should be low)

**WNR1** : Winner points earned by play 1( it can be combination of Ace1, NPW1,FSW1,SSW1 (not UFE)).** If more winners most likely match winner.

**WNR2** : winner points earned by play 2 ( Combo of Ace2, NPW2, FSW2, SSW2)

**BPC1** : No of times play 1 creates break points(40-15, 40-30 situation but not winning)

**BPW1** : No of times play 1 won those break points( usually winning through BPC1 or do come back from BPC2(very rare))***this can be Winners(WNR1, ACE1,FSW1,SSW1,NPW1,UFE2(if this point is not a winner for play 1 ))***

**BPC2** : No of times play 2 creates break points(40-15, 40-30 situation but not winning)

**BPW2** : No of times play 2 won those break points( usually winning through BPC2 or do come back from **BPC1**(very rare))***this can be Winners(WNR2, ACE2,FSW2,SSW2,NPW2,UFE2(if this point is not a winner for play 2))***

**TPW1** : The Total points won by Player1 ( usual 4 points for a game or scoring 5 points during a duece situation or scoring 7 points during a tie breaker)

**TPW2** : The Total points won by Player2 ( usual 4 points for a game or scoring 5 points during a duece situation or scoring 7 points during a tie breaker).

**Player 1**:

ST1.1 :Set 1 result for Player 1

ST2.1 :Set 2 Result for Player 1

ST3.1 :Set 3 Result for Player 1

ST4.1 :Set 4 Result for Player 1

ST5.1 :Set 5 Result for Player 1

Like (6-4, 7-5, 6-0 etc)

**Player 2**:

ST1.2 :Set 1 result for Player 2

ST2.2 :Set 2 Result for Player 2

ST3.2 :Set 3 Result for Player 2

ST4.2 :Set 4 Result for Player 2

ST5.2 :Set 5 Result for Player 2

**1.2 Data Understanding**:

- In the given data set, there are 8 data sets of which 4 are Men and the other 4 women respectively.
- Each set has around 76 rows which are for different players data participating in a match.
- The Attributes are around 42 for two players(since it takes only singles data )are actually the different skill set for each player and their points scored during the match.
- According to the rules, Men usually plays 5 sets to win the Match and Women usually play 3 sets to win the match.
- The resultant NAs in the Set 4 and 5 column for women which are not even played by the women players.
- Since I have to get to a single variable understanding on the two players data, I have to Impute those Nas with Central imputation which is most common as it is numerical or Knn if required.
- If NA values are present for both the players which happens when they have not played that shot or scored the particular point then I impute them with Zero(0).
- Finally used KNN imputation and this gives me the best accuracy so far with the data.

**1.3 Models Applied** :

- **Logistic regression** : Used this model with all the attributes(predictor variable) vs the result (response variable)
- **Decision Trees**: Used the C50 method to run a decision tree based on the rules applied on the data.
- **Support Vector Machines**: Used the e1071 method package to run the required SVM model
- **Random Forest** : used the Random forest package to execute the required RF model.

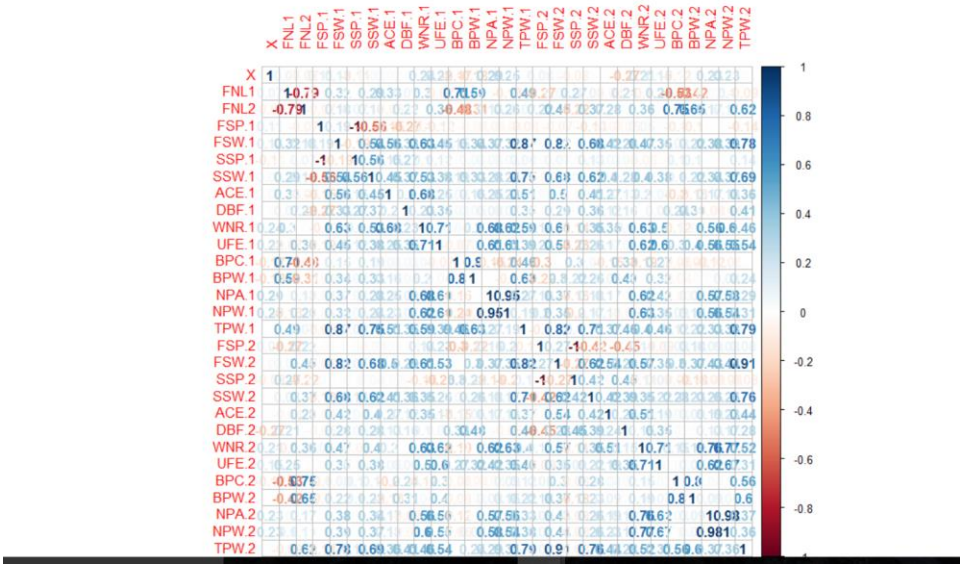
## Section 2: Experiments(Models)

**Section 2.1-Data preprocessing :**

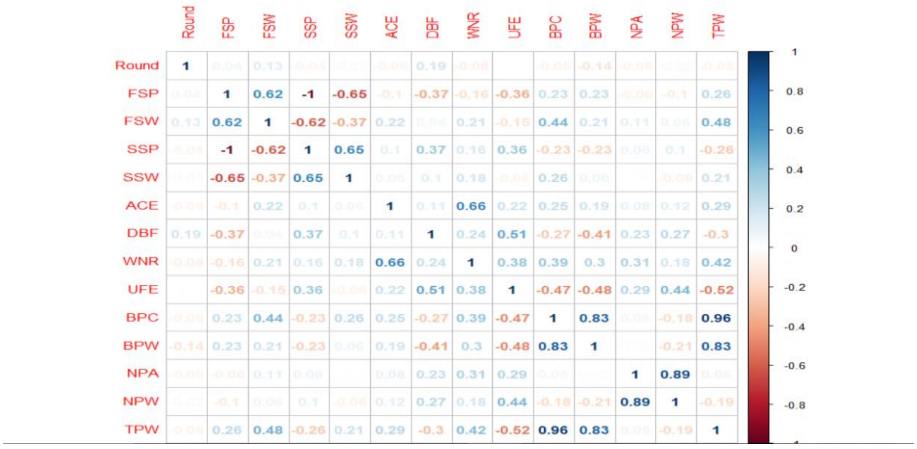
- Performed a Logistic regression on the Dual player data and it is unable to run **glm** on that high correlated data .
- Considered Both players attributes as X variables and result of the Match(where each row is a match statistics with its result).

- So while computing logistic regression with multiple Predictors(Dual) and Result as Response(target) variable.
- Have performed the Correlation plot between all the player 1 and player 2 variables to check relationship between them.
- It is returning all Coefficients and its Z score values as Zero and not even calculating its Deviance and its DoF.
- consolidated the Data set using the Difference of the predictor variables as single variable value to predict its target variable.
- Created a new data set by taking the difference between the two player attributes and taken the resultant set as the final for model building.
- The Resultant data set has eliminated all the NA values and considered the difference of the NA values as Zero.

**Corr plot between all Player 1 and Player 2 variables.**



**Corr Plot of the Difference variables data set(singular)**




## 2.2 Summary

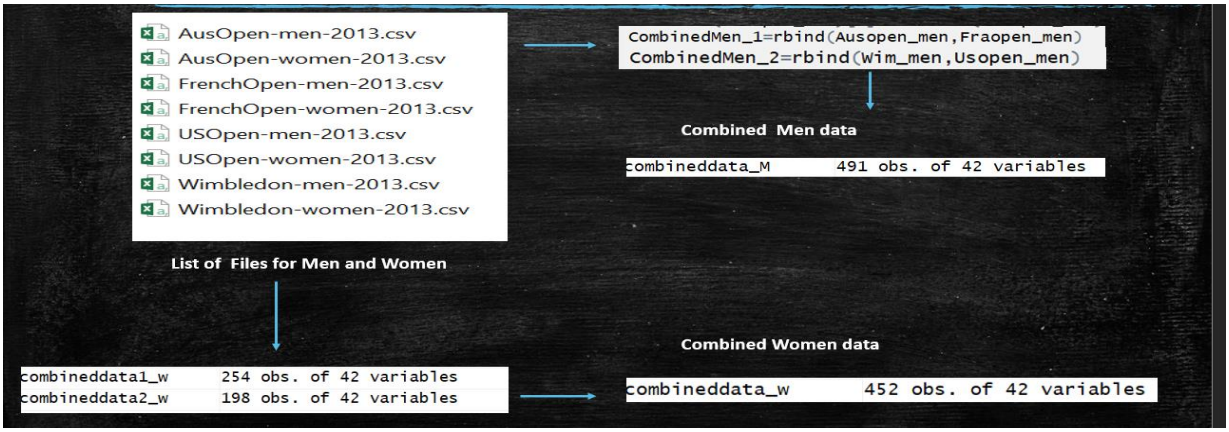
- Performed correlation check for the singular data.
- Encountered some variables(SSP,SSW, FSW, DBF,TPW) with high correlation and does not siginfy on the model accuracy.
- Drop those columns with high correlation to fine tune the model.
- Performed analysis on the all the variables but its not giving a desired result .
- So used one variable to predict and just performed feature selection and added more significant variables.
- Ran the correlation and box plots for all the combinations of variables and the target to remove any outliers present.
- Performed the StepAIC to automatically remove all the highly correlated variables and give out the best fit for the model.
- Also performed the VIF to remove multi collinearity between the variables to see how much variance is between the predictors.
- Performed the train and test split for prediction and performed the predict the train on the test results.
- Obtained Accuracy around 70% and Specificity around 79 % and Sensitivity with 64% range.
- Performed some other Data preprocessing and tried to remove NA values and replace the difference with some default values with some value like 3,4 & 5.
- Not resulted in a great outcomes so performed more Data preprocessing.
- Used Standardization and Normalisation for the data before Splitting the data.
- Used standardization method and performed the logistic regression on the data and got much improved accuracy with increase of 13%.

- Used Range method seperately and performed the glm model and this did not give much different accuracies.
- So Standardization gives us good accuracy of about 83 percent with Sensitivity of 75 and specificity of 94 percent respectively.
- Performed the K fold cross validation on the data for checking on the validation of accuracies.

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 16  5
         1  1 15

               Accuracy : 0.8378
                 95% CI : (0.6799, 0.9381)
    No Information Rate : 0.5405
    P-Value [Acc > NIR] : 0.0001445

                  Kappa : 0.6792
 Mcnemar's Test P-Value : 0.2206714

            Sensitivity : 0.7500
            Specificity : 0.9412
         Pos Pred Value : 0.9375
         Neg Pred Value : 0.7619
```

### 2.3 Data set Combination

- I have aggregated all the data for the Mens data (Ausopen, Frenchopen, Wimbledonopen&USopen)
- Combined the Men data and women data together to optimize the compile multiple models for multiple data sets.
- Since I have the identical columns for both Men and Women data, I have combined using the Rbind function (source “rbind” in Rstudio)
- I have used the combined men and women data for predicting models seperately.
- Ran a scatter plot for all the variables and to detect outliers.
- Based on the businesss decision, the outliers cannot be eliminated since outliers can be exceptional player in a match.



### Section 2.3.1: Imputations

- I have detected mulitiple missing values(NA) of about 490.
- I have performed multiple imputation techniques like Zero imputation, Central imputation& KNN imputation.
- Based on the performance and accuracy, I chose KNN imputed data for further prediction.
- I have used standardization method to further scale the data for better prediction.
- I have formalized the final data set for train test split and run multiple model on the data.

### Section 2.4 Models Applied

---

### 2.4.1: Logistic Regression

- Used the glm method to run on the train data and then predict the same on the test data .
- Obtained good accuracy on train and test data.

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 157  62
         1  19 106

               Accuracy : 0.7645
                 95% CI : (0.7161, 0.8084)
    No Information Rate : 0.5116
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5261
 Mcnemar's Test P-Value : 3.061e-06

            Sensitivity : 0.8920
```

**Train predictions**

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 67 19
         1  8 53

               Accuracy : 0.8163
                 95% CI : (0.7441, 0.8753)
    No Information Rate : 0.5102
    P-Value [Acc > NIR] : 1.116e-14

                  Kappa : 0.6314
 Mcnemar's Test P-Value : 0.05429

            Sensitivity : 0.7361
            Specificity : 0.8933
         Pos Pred Value : 0.8689
         Neg Pred Value : 0.7791
             Prevalence : 0.4898
         Detection Rate : 0.3605
   Detection Prevalence : 0.4150
      Balanced Accuracy : 0.8147

       'Positive' Class : 1
```

**Test predictions**

### 2.4.2 : Decision Tree

- Performed the Decision tree building based on the C50 algorithm and it uses all the rules based on the algorithm.
- Predict the tree split which is based on Information gain from each variable .
- It obtains a better accuracy than the regression model on train and test data .

**Train Predictions**

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 173   5
         1   3 163

               Accuracy : 0.9767
                 95% CI : (0.9547, 0.9899)
    No Information Rate : 0.5116
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9535
 Mcnemar's Test P-Value : 0.7237

            Sensitivity : 0.9830
            Specificity : 0.9702
         Pos Pred Value : 0.9719
         Neg Pred Value : 0.9819
             Prevalence : 0.5116
         Detection Rate : 0.5029
   Detection Prevalence : 0.5174
      Balanced Accuracy : 0.9766

       'Positive' Class : 0
```

**Test predictions**

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 69  9
         1  6 63

               Accuracy : 0.898
                 95% CI : (0.8373, 0.9418)
    No Information Rate : 0.5102
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7957
 Mcnemar's Test P-Value : 0.6056

            Sensitivity : 0.9200
            Specificity : 0.8750
         Pos Pred Value : 0.8846
         Neg Pred Value : 0.9130
             Prevalence : 0.5102
         Detection Rate : 0.4694
   Detection Prevalence : 0.5306
      Balanced Accuracy : 0.8975

       'Positive' Class : 0
```

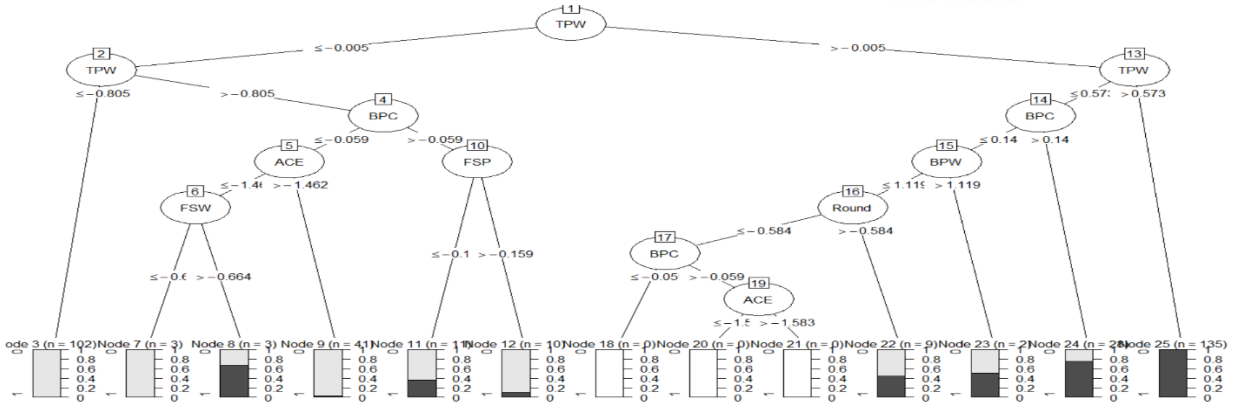



Fig : **Decision tree Structure**

### 2.4.3 Support Vector Machines

- Performed the tuned SVM model using linear slack
- Retrived the best model and predicted both on train and test data
- Retreived the best parameters for gamma as 1e-0.6 and cost as 0.1

**Train Metrics**

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 167   9
         1   7 161

               Accuracy : 0.9535
                 95% CI : (0.9256, 0.9732)
    No Information Rate : 0.5058
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.907
 Mcnemar's Test P-Value : 0.8026

            Sensitivity : 0.9598
            Specificity : 0.9471
         Pos Pred Value : 0.9489
         Neg Pred Value : 0.9583
             Prevalence : 0.5058
         Detection Rate : 0.4855
   Detection Prevalence : 0.5116
      Balanced Accuracy : 0.9534

       'Positive' Class : 0
```

**Test Metrics**

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 70  5
         1  8 64

               Accuracy : 0.9116
                 95% CI : (0.8535, 0.9521)
    No Information Rate : 0.5306
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8229
 Mcnemar's Test P-Value : 0.5791

            Sensitivity : 0.8974
            Specificity : 0.9275
         Pos Pred Value : 0.9333
         Neg Pred Value : 0.8889
             Prevalence : 0.5306
         Detection Rate : 0.4762
   Detection Prevalence : 0.5102
      Balanced Accuracy : 0.9125

       'Positive' Class : 0
```

### 2.4.4 Random Forest

- It employs Randomforest function and chose 10 trees for generating the best random forest tree.
- It gives the best accuracy so far after running a series of models with 94 percent accuracy on the test data.

**Train metrics**

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 176   2
         1   0 166

               Accuracy : 0.9942
                 95% CI : (0.9792, 0.9993)
    No Information Rate : 0.5116
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9884
 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 0.9881
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9888
```

**Test metrics**

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 71  5
         1  4 67

               Accuracy : 0.9388
                 95% CI : (0.887, 0.9716)
    No Information Rate : 0.5102
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8775
 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9306
            Specificity : 0.9467
         Pos Pred Value : 0.9437
```

### 2. 5 Final prediction metrics for my model ecosystem

| Models Applied | Men data | | | | | | Women data | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy (In %) | | Specificity (In%) | | Sensitivity (In %) | | Accuracy (In %) | | Specificity (In %) | | Sensitivity (In %) | |
| | Train | Test | Train | Test | Train | Test | Train | Test | Train | Test | Train | Test |
| **Logistic Regression** | 76 | 82 | 94 | 89 | 89 | 73 | 86 | 84 | 93 | 92 | 79 | 76 |
| **Decision Trees** | 97 | 90 | 97 | 87 | 98 | 92 | 99 | 91 | 1 | 89 | 99 | 92 |
| **Support Vector Machines** | 95 | 91 | 94 | 92 | 95 | 89 | 97 | 95 | 94 | 92 | 98 | 96 |
| **Random forest** | 99 | **93** | 1 | 94 | 99 | 93 | 99 | 95 | 1 | 94 | 99 | 97 |

### 2.6 Future Scope

- Xg boost can be done on the model to minimize the error for misclassification and gives us the best model to predict the result accurately.
- Clustering of all the predictor variables can be done to accurately predict the players performance based on their performance and classify them based on different courts(tournaments).
- The Best predicted model can be used for a product innovation and this thrives to a very lucrative model in sport winner predictions which seems to be an Impossible task in the current world.
- Although the predictions on the result is performed, the further scope could be merging the player names and predict the performance of each player based on their performance in each tournament rather than win or a lose in the match.

### 2.7 Conclusion

**Random Forest** gives me the best prediction with highest accuracy(94%) and other metrics.Accuracy gives me the best intuition based on my models along with specificity(true negatives) and sensitivity(true positives) which helps in better classification.Emphasizing on more number of wins based on the data and used the difference of players attributes for a clear intuition of classification of win and loss.Finally I conclude that by chosing the best Ensemble learning technique, Random Forest the model learns best by itself on the data and predicts the result of a match best on the future match data.I would least consider the computational time for the model and consider its result obtained from the experiment and which results in a scope for higher ensemble learning models for the same data.