Tennis Match Prediction

Machine Learning Project

Anastasios Sarafidis (mtn2028)

Supervisor: Theodoros Giannakopoulos

Data Gathering

- Tennis ATP singles matches data gathered
  - https://github.com/JeffSackmann/tennis_atp
  - For the time period 2010-2019

- Tournament information
  - Tournament id, surface, draw size etc
- Players' information
  - Name, height, playing hand etc
- Matches statistics
  - Score, minutes, aces etc

Data Cleaning and Feature Engineering

- Data Cleaning
  - Remove stats that are unknown before the match
  - Remove unnecessary columns
  - Surface & Rank Points are important
    - Remove entries with no info about those

- Feature Engineering
  - Transform data so that we have *First Player* & *Second Player*
  - Create *"label"* column
    - "0" if First Player wins
    - "1" if Second Player wins

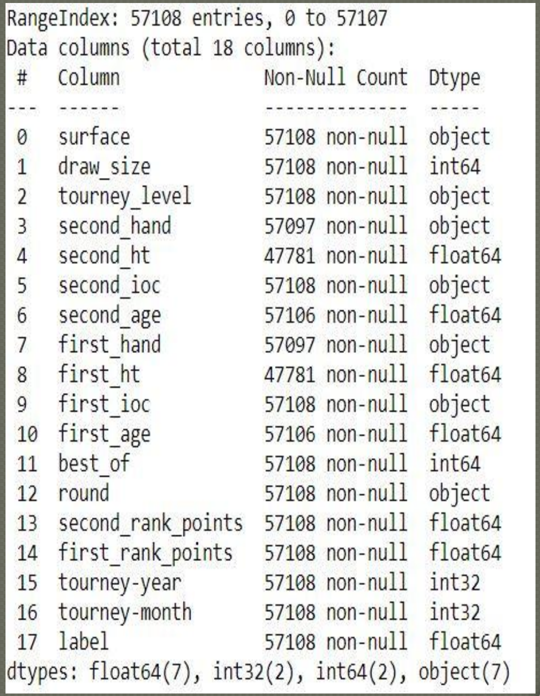
Data Cleaning and Feature Engineering

- Final shape of Dataset

- Multiple null values in “Height” columns
  - Use columns’ means to fill these values

- Categorical values → Numerical

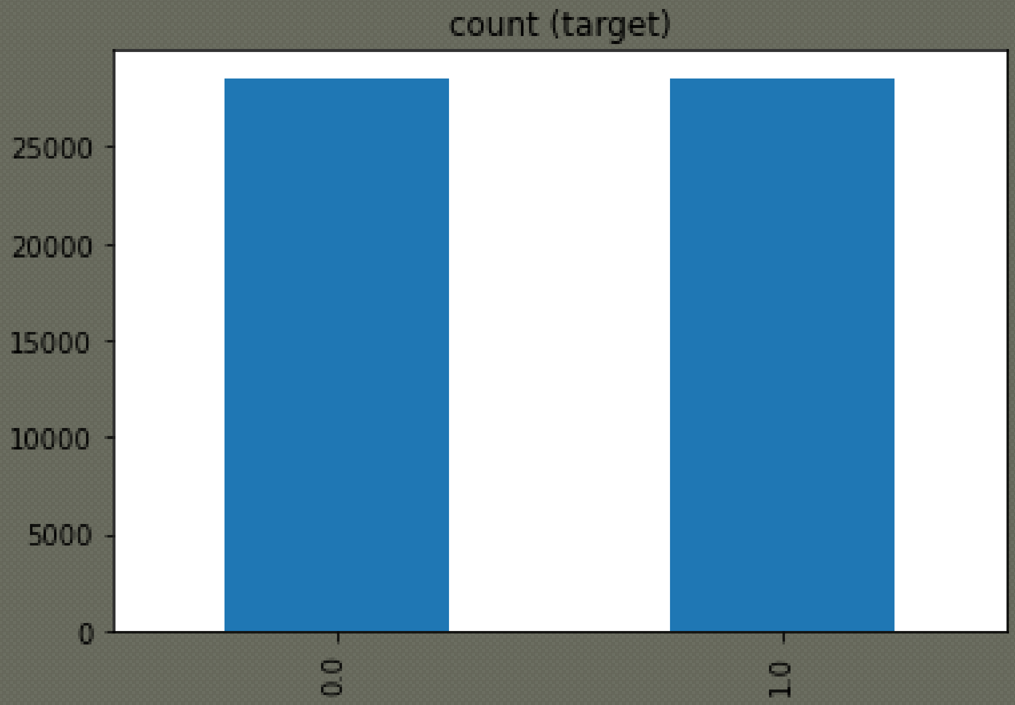
- Remove missing values



```
RangeIndex: 57108 entries, 0 to 57107
Data columns (total 18 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   surface             57108 non-null  object 
 1   draw_size           57108 non-null  int64  
 2   tourney_level       57108 non-null  object 
 3   second_hand         57097 non-null  object 
 4   second_ht           47781 non-null  float64
 5   second_ioc          57108 non-null  object 
 6   second_age          57106 non-null  float64
 7   first_hand          57097 non-null  object 
 8   first_ht            47781 non-null  float64
 9   first_ioc           57108 non-null  object 
 10  first_age           57106 non-null  float64
 11  best_of             57108 non-null  int64  
 12  round               57108 non-null  object 
 13  second_rank_points  57108 non-null  float64
 14  first_rank_points   57108 non-null  float64
 15  tourney-year        57108 non-null  int32  
 16  tourney-month       57108 non-null  int32  
 17  label               57108 non-null  float64
dtypes: float64(7), int32(2), int64(2), object(7)
```

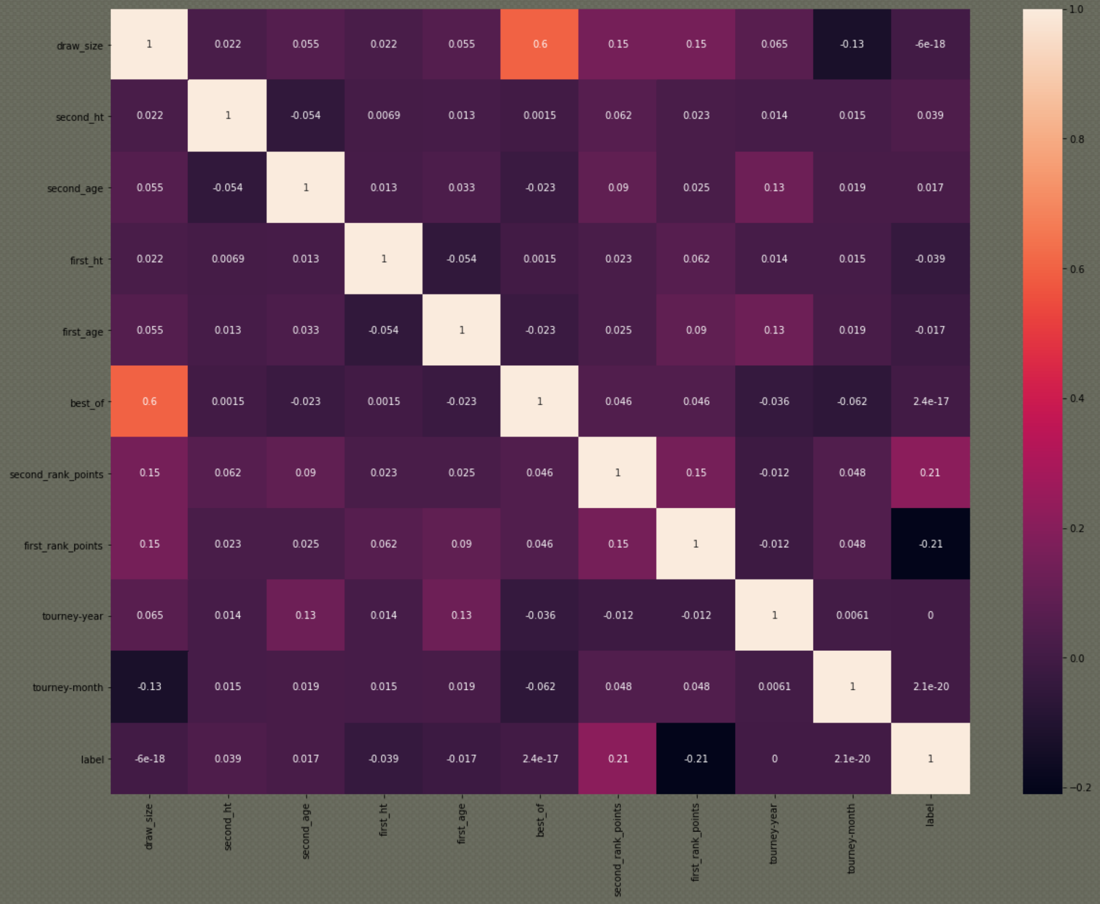
Data Exploration

- Balance of Dataset



Data Exploration

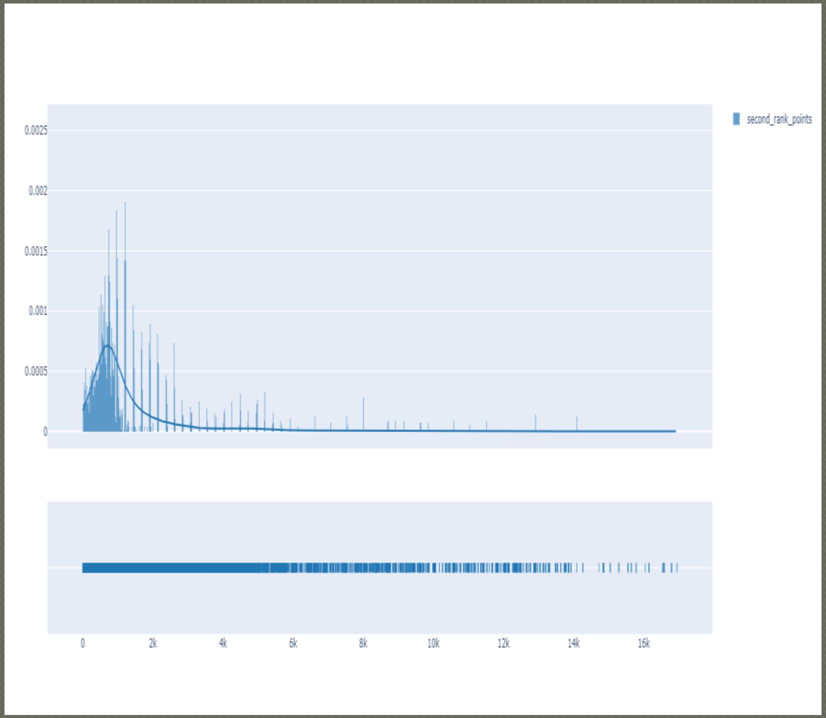
- Correlations

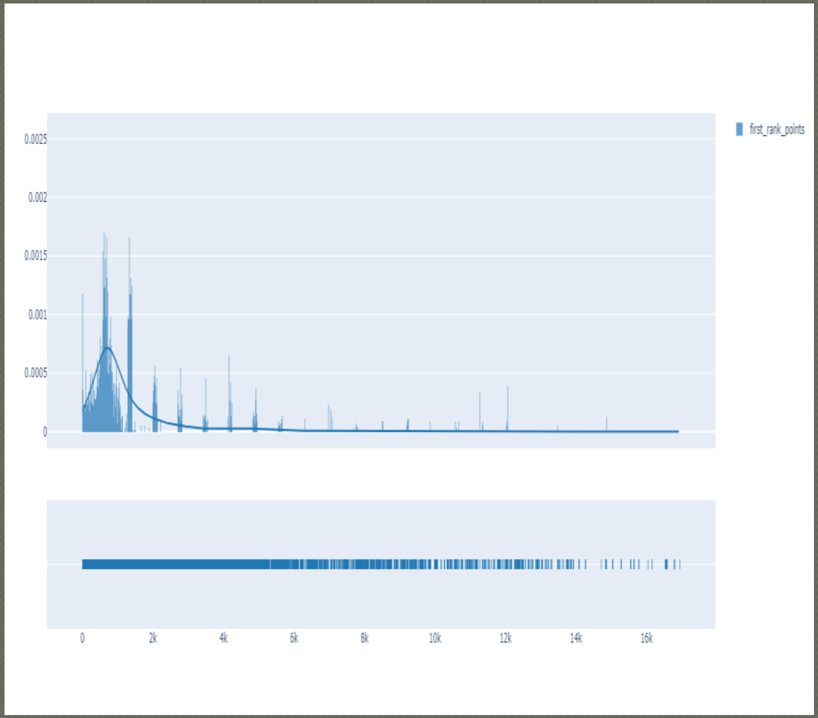


	draw_size	second_ht	second_age	first_ht	first_age	best_of	second_rank_points	first_rank_points	tourney-year	tourney-month	label
draw_size	1	0.022	0.055	0.022	0.055	0.6	0.15	0.15	0.065	-0.13	-6e-18
second_ht	0.022	1	-0.054	0.0069	0.013	0.0015	0.062	0.023	0.014	0.015	0.039
second_age	0.055	-0.054	1	0.013	0.033	-0.023	0.09	0.025	0.13	0.019	0.017
first_ht	0.022	0.0069	0.013	1	-0.054	0.0015	0.023	0.062	0.014	0.015	-0.039
first_age	0.055	0.013	0.033	-0.054	1	-0.023	0.025	0.09	0.13	0.019	-0.017
best_of	0.6	0.0015	-0.023	0.0015	-0.023	1	0.046	0.046	-0.036	-0.062	2.4e-17
second_rank_points	0.15	0.062	0.09	0.023	0.025	0.046	1	0.15	-0.012	0.048	0.21
first_rank_points	0.15	0.023	0.025	0.062	0.09	0.046	0.15	1	-0.012	0.048	-0.21
tourney-year	0.065	0.014	0.13	0.014	0.13	-0.036	-0.012	-0.012	1	0.0061	0
tourney-month	-0.13	0.015	0.019	0.015	0.019	-0.062	0.048	0.048	0.0061	1	2.1e-20
label	-6e-18	0.039	0.017	-0.039	-0.017	2.4e-17	0.21	-0.21	0	2.1e-20	1

Data Exploration

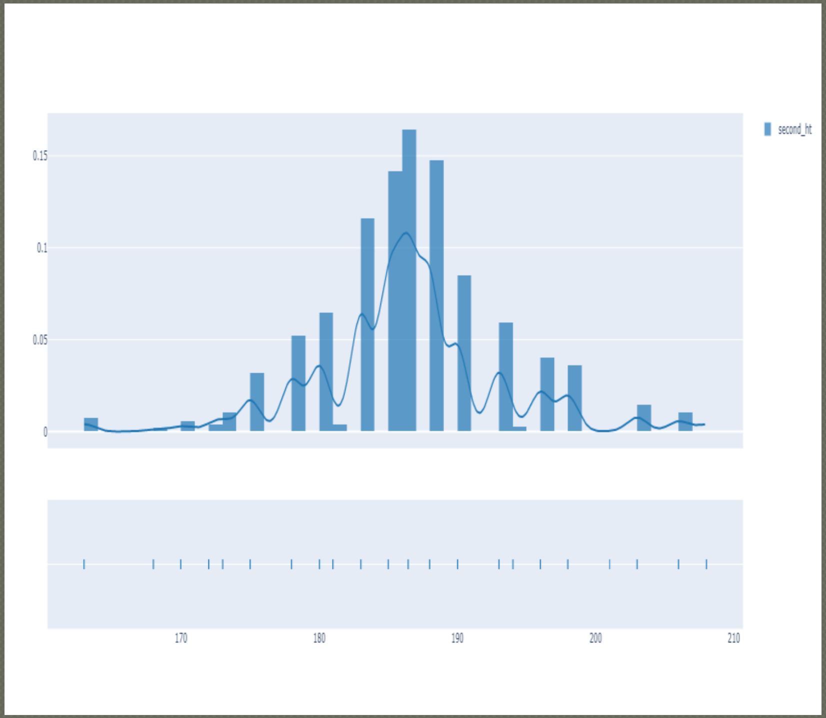
- Players’ rank points Histogram

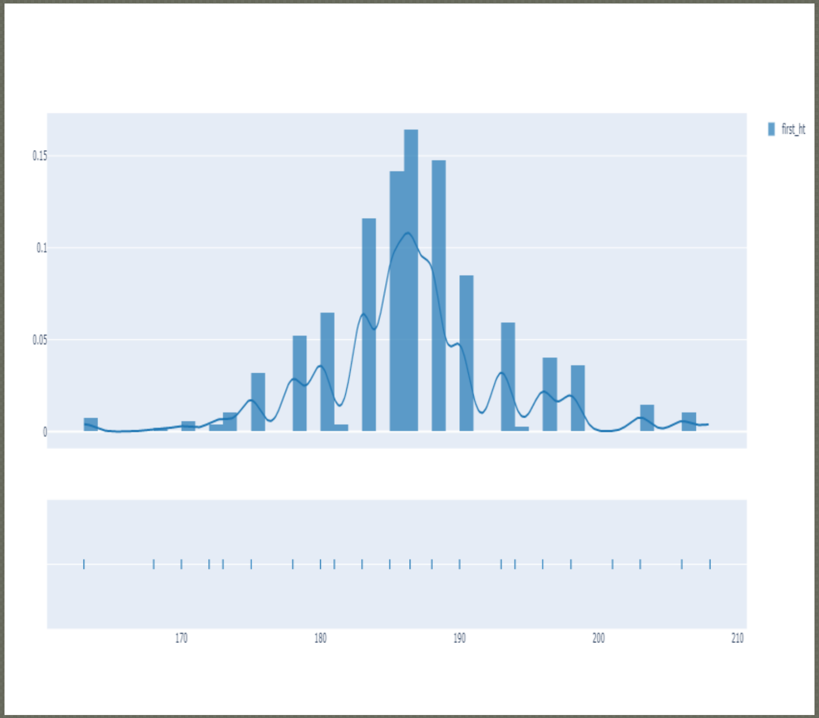




Data Exploration

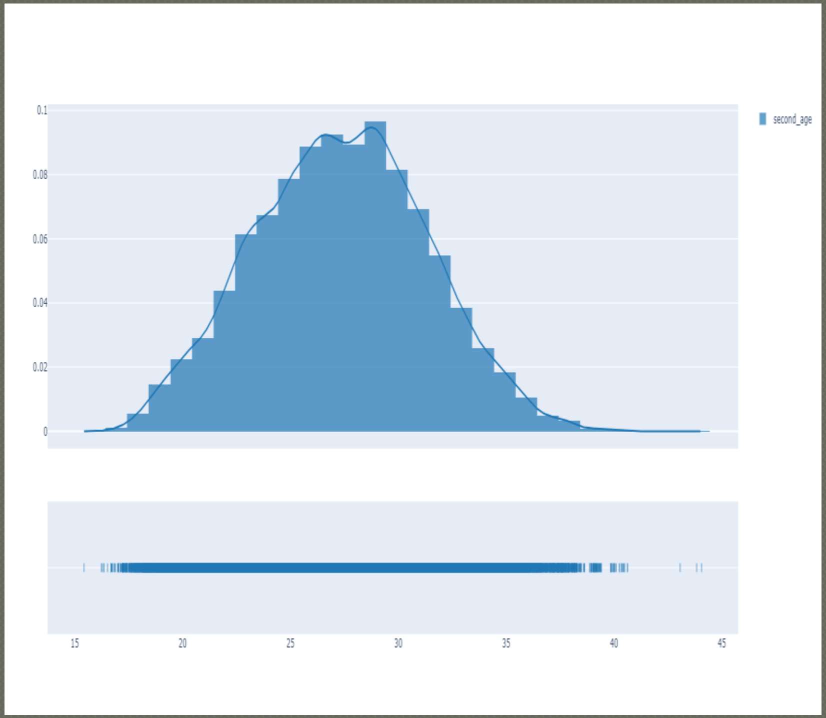
- Players' height Histogram

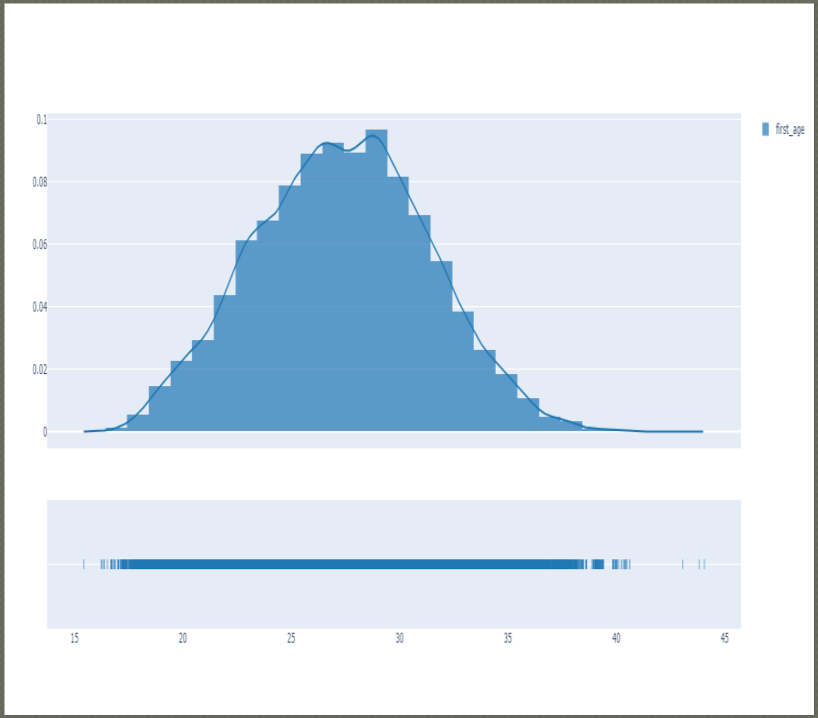




Data Exploration

- Players’ age Histogram



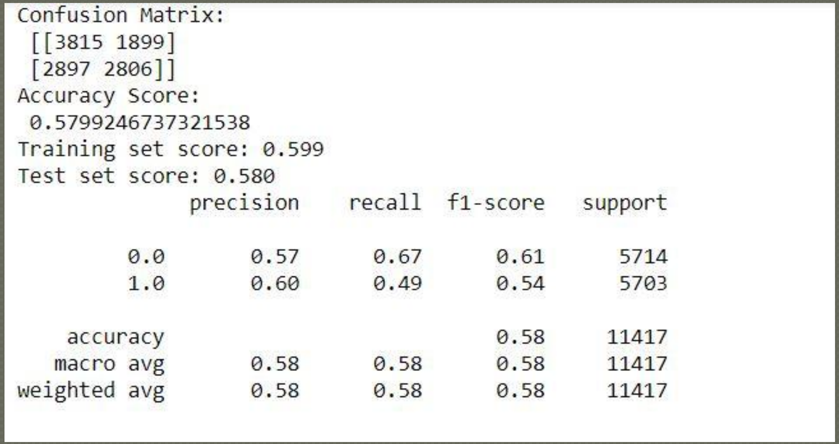


Model Training & Evaluation

- Train – Test split for fitting and evaluation
  - 80% train size
  - 20% test size
- Standardize Features

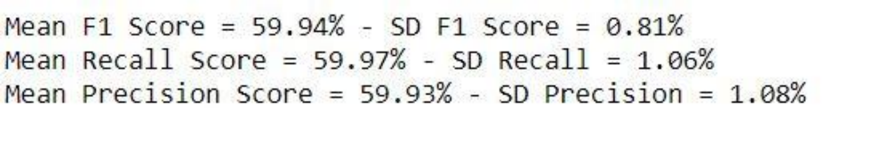
ML Models Evaluation

◎ Naïve Bayes



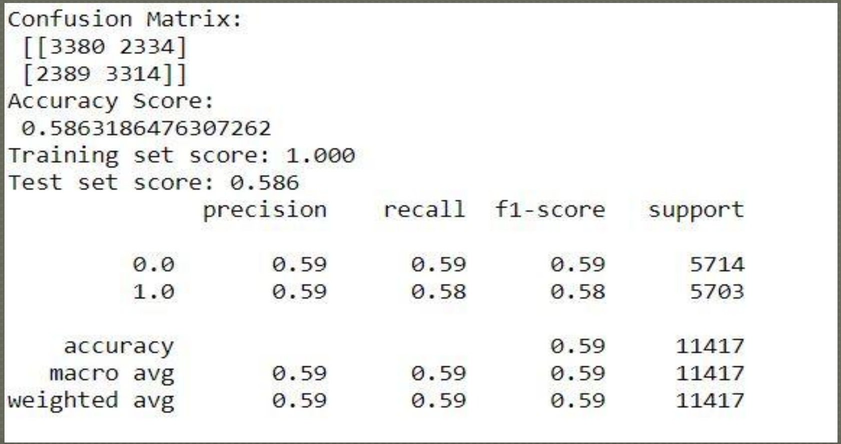
```
Confusion Matrix:
 [[3815 1899]
 [2897 2806]]
Accuracy Score:
 0.5799246737321538
Training set score: 0.599
Test set score: 0.580
              precision    recall  f1-score   support

         0.0       0.57      0.67      0.61      5714
         1.0       0.60      0.49      0.54      5703

    accuracy                           0.58     11417
   macro avg       0.58      0.58      0.58     11417
weighted avg       0.58      0.58      0.58     11417
```

KFold



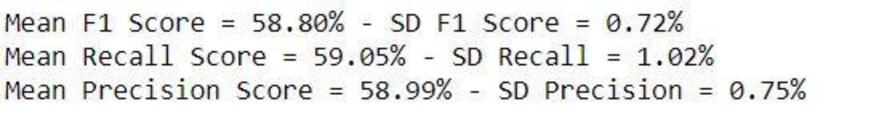
```
Mean F1 Score = 59.94% - SD F1 Score = 0.81%
Mean Recall Score = 59.97% - SD Recall = 1.06%
Mean Precision Score = 59.93% - SD Precision = 1.08%
```

◎ Decision Tree



```
Confusion Matrix:
 [[3380 2334]
 [2389 3314]]
Accuracy Score:
 0.5863186476307262
Training set score: 1.000
Test set score: 0.586
              precision    recall  f1-score   support

         0.0       0.59      0.59      0.59      5714
         1.0       0.59      0.58      0.58      5703

    accuracy                           0.59     11417
   macro avg       0.59      0.59      0.59     11417
weighted avg       0.59      0.59      0.59     11417
```

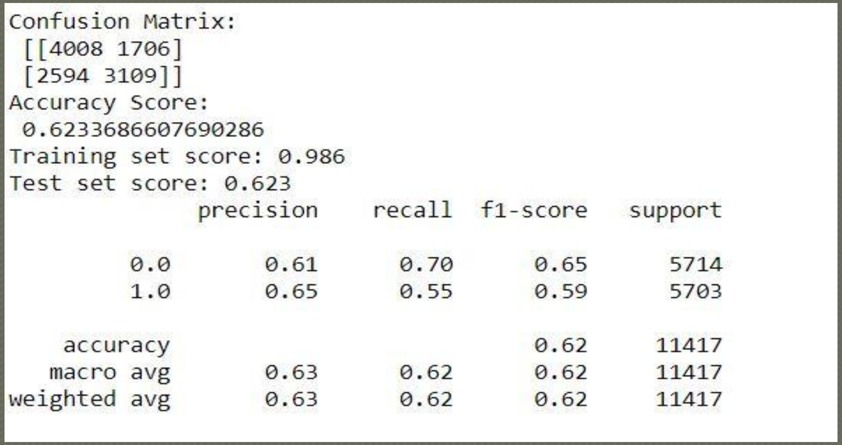
KFold



```
Mean F1 Score = 58.80% - SD F1 Score = 0.72%
Mean Recall Score = 59.05% - SD Recall = 1.02%
Mean Precision Score = 58.99% - SD Precision = 0.75%
```

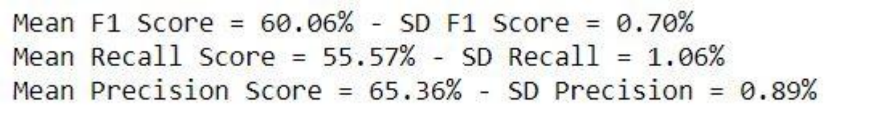
ML Models Evaluation

- Random Forest



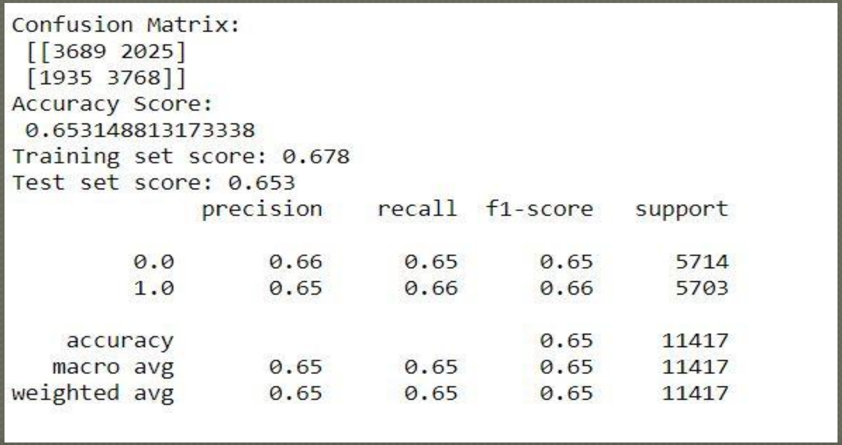
```
Confusion Matrix:
 [[4008 1706]
 [2594 3109]]
Accuracy Score:
 0.6233686607690286
Training set score: 0.986
Test set score: 0.623
              precision    recall  f1-score   support

         0.0       0.61      0.70      0.65      5714
         1.0       0.65      0.55      0.59      5703

    accuracy                           0.62     11417
   macro avg       0.63      0.62      0.62     11417
weighted avg       0.63      0.62      0.62     11417
```

KFold



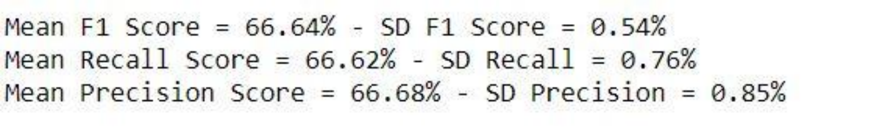
```
Mean F1 Score = 60.06% - SD F1 Score = 0.70%
Mean Recall Score = 55.57% - SD Recall = 1.06%
Mean Precision Score = 65.36% - SD Precision = 0.89%
```

- XGBoost



```
Confusion Matrix:
 [[3689 2025]
 [1935 3768]]
Accuracy Score:
 0.653148813173338
Training set score: 0.678
Test set score: 0.653
              precision    recall  f1-score   support

         0.0       0.66      0.65      0.65      5714
         1.0       0.65      0.66      0.66      5703

    accuracy                           0.65     11417
   macro avg       0.65      0.65      0.65     11417
weighted avg       0.65      0.65      0.65     11417
```

KFold

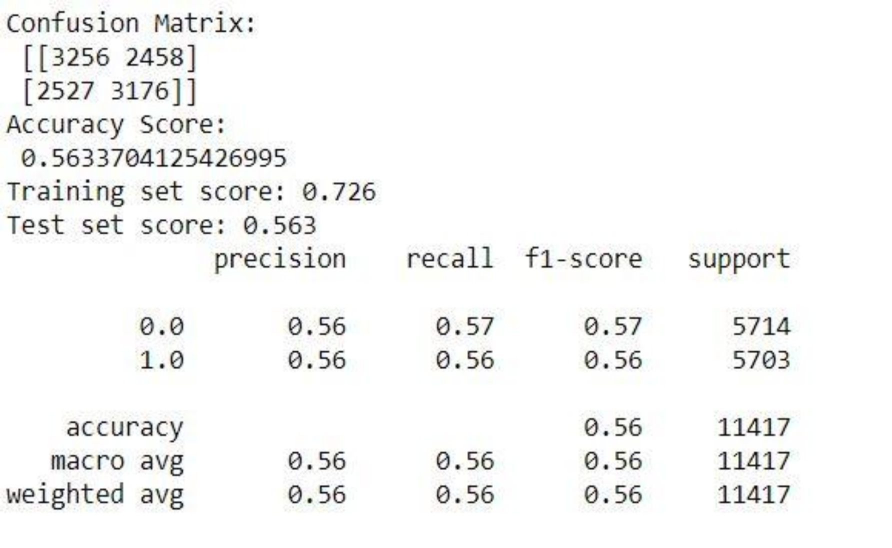


```
Mean F1 Score = 66.64% - SD F1 Score = 0.54%
Mean Recall Score = 66.62% - SD Recall = 0.76%
Mean Precision Score = 66.68% - SD Precision = 0.85%
```

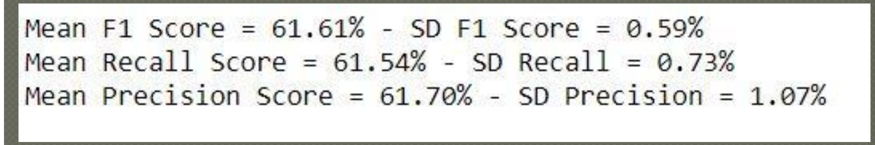
ML Models Evaluation

- K-Nearest Neighbors

KFold



```
Confusion Matrix:
 [[3256 2458]
 [2527 3176]]
Accuracy Score:
 0.5633704125426995
Training set score: 0.726
Test set score: 0.563
              precision    recall  f1-score   support

         0.0       0.56      0.57      0.57      5714
         1.0       0.56      0.56      0.56      5703

    accuracy                           0.56     11417
   macro avg       0.56      0.56      0.56     11417
weighted avg       0.56      0.56      0.56     11417
```



```
Mean F1 Score = 61.61% - SD F1 Score = 0.59%
Mean Recall Score = 61.54% - SD Recall = 0.73%
Mean Precision Score = 61.70% - SD Precision = 1.07%
```

Best Model Decision

- XGBoost is clearly the best choice for our Dataset
  - Based on Mean F1 Scores
  - Every other model has a lower Mean F1 Score
    - 5% or more

Thank you for your time!