This project proposes various machine learning models, which predict probabilities of winning a match for both players. It includes: logistic regression, SVM with RBF kernel, random forest, XGBoost, 1-layer ANN.

For this project I use files:
atp_matches_qual_chall_2022.csv
atp_matches_qual_chall_2023.csv
atp_matches_qual_chall_2024.csv
atp_matches_2022.csv
atp_matches_2023.csv
atp_matches_2024.csv

All datasets have been taken from https://github.com/JeffSackmann/tennis_atp repository. These datasets have data for 2022, 2023, 2024 matches from ATP, challengers, qualifications (individuals). File
https://github.com/JeffSackmann/tennis_atp/blob/master/matches_data_dictionary.txt has an explanation of features of datasets. Also I use file odds.xlsx, it has the same structure, but 2 additional columns in the end which contain bookmaker's odds (bet365) on the opening line. I use this file for evaluating models.

File features_create.ipynb contains different approaches for building the feature space. As a result I create following features for player using information of all his previous matches:

1. Average number of aces per match
2. Average number of double faults per match
3. Average number of break points faced per match
4. Average number of break points saved per match
5. Average number of break points faced per match for rivals
6. Average number of break points saved per match for rivals
7. Winning % on 1st serve = [1st serve win] / [total 1st serve]
8. Winning % on 2nd serve = [2st serve win] / [total 2sr serve]
9. Winning % Return serve = [Return Serve Win] / [Total Return Serve]
10. Winning % Match Played = [Match Played Win] / [Total match Played]
11. Average of Winning Point per match: [Point Win] / [Number of matches]

Also, I calculate the same features but for previous 3 player's matches. Additionally, I append ATP rank and ATP rank points of players. So that, we already have 24 features
After that I add 3 features:

1. Hard Court: this feature represents the match played on hard court.
2. Clay Court, this feature represents the match played on clay court.
3. Grass Court, this feature represents the match played on grass court.

After that I add one more feature:

1. Same handedness

Eventually, I calculate the first 24 features for the winner and loser and subtract the second set from the first set and add other 4 features(surface of court and same handedness). It has corresponding prediction = 1. I also subtract the first set from the second set and add other 4 features, it has corresponding prediction = 0. I add these 2 rows into training dataset, and I do it for every existing match.
This feature space building is contained in ‘player_statistics2’ and ‘player_statistics_last’ functions and the last cell of features_create.ipynb file.

Training of machine learning models is contained in tennis_ANN.ipynb file. For training I use odds.xlsx file. For every model I pick parameters using GridSearchCV. Best result is shown by xgboost model, accuracy = 0.68. Bookmakers accuracy for this data is 0.592.
After that, using predicted odds, we simulate a situation where we bet 1$ for every match where some predicted odd is lower than the corresponding odd offered by bet365.
ROI of xgboost model = 0.02.