Predicting Playstyle from Physical Attributes and Match Statistics in Tennis

By:
Spencer Fradkin

Problem Description:

Tennis is one of the most popular sports worldwide. In spite of this, there are very few tools that help the increasing number of amateur and recreational players understand how they should be playing the game strategically. This project aims to create a tool to help amateur tennis players understand how they should be playing based on their physical characteristics and compare this to how they are currently playing based on some of their match statistics. The customer of a product like this is an amateur player that is just beginning to play the sport of tennis competitively and lacks information about the various strengths and limitations physical attributes impose on players of all levels.

Data Curation:

In order to solve this problem, we are in need of data about the physical characteristics of players, various match statistics about players, and a classification of each player based on how they play in matches. Since amateur data is not collected reliably, all models will be fit using information about professional tennis players. Typically, professional tennis match data is not available publicly at a level that is detailed enough for this project. However, there are organizations and third parties that collect match data (usually by hand) and make it publicly available. The Match Charting Project (http://www.tennisabstract.com/charting/meta.html) does this and makes the data available on Github.

The Match Charting Project collects data on a match by match basis and uses a custom Excel file for each type of shot (serves, returns, rally balls, etc..). Additionally, they collect data associated with matches, not players. Thus, some wrangling was necessary to get the data into a usable format for this project. In short, I needed to change each unique Excel file into a consistent CSV format that could be parsed without any errors, join the new datasets containing match statistics on a particular match ID number, create a new dataframe that contains separate statistics for each player in the match, and then group the datasets by player and average all of the statistics for each player. This created a dataset with rows containing player names and the average match statistics associated with each name. Following this, I did some basic calculations to create columns that I thought would be the most useful for this analysis.

The remaining dataset has roughly 250 professional players from the 1980s to present and 14 features which contain information about each player such as their first serve percentage, the percentage of times they come to the net, their ace percentage, and how long their points are, averaged over all of the matches they played. Additionally, a second dataset was collected from the ATP which contains the height and weight of each player in the first dataset. Since much of the match statistics were collected by hand, there are some inconsistencies in how the data was recorded. Specifically, about 25% of players were missing statistics on how often they

immediately come to the net after serving (serve and volley). This data is crucial in determining how aggressive a player is in a match. Since this data is crucial, I was able to impute the missing column using LASSO Regression. This will be discussed in the Data Exploration/Unsupervised Learning section of the paper.

Lastly, we need labels for each player we have physical attributes and match statistics for so we can train supervised learning models. This was done by surveying 5 USPTA certified tennis professionals and having them rate players from 0 to 3 with 0 being the most aggressive to 3 being the most defensive, and averaging their results.

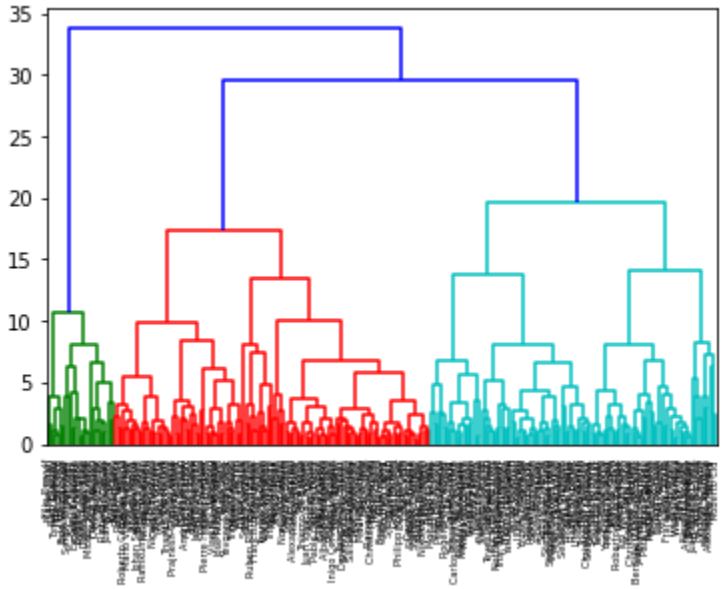
Data Exploration/Unsupervised Learning:

Data Imputation Using LASSO:
As mentioned earlier, the custom dataset created for this project is missing 25% of a column. Since the dataset is not very large to begin with, it is necessary to impute this data. Since the dataset contains many features, I assumed that there was a subset of features that would be good predictors for the missing column. First, I looked at plots of each feature and the feature I wanted to impute the data for (serve and volley percentage) along with a pairwise correlation matrix for the entire dataset not including the rows with missing values. I noticed that the percentage of net points played was positively correlated with the column that had missing values (.91).

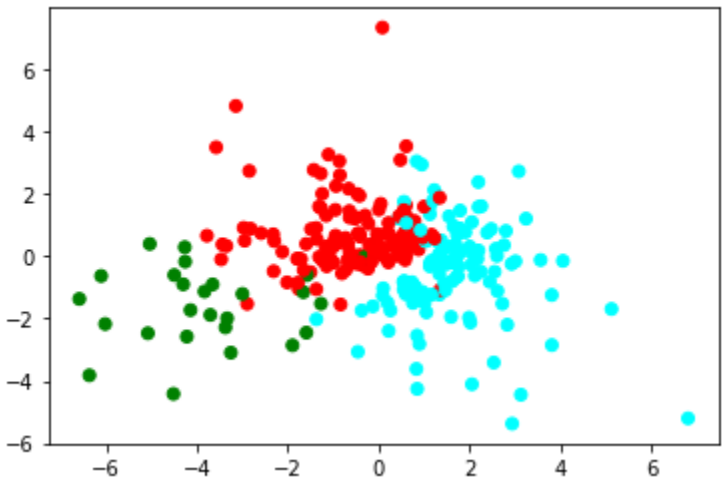
I decided to use LASSO for feature selection. The column I was looking to predict is a value between 0 and 1, and LASSO does not guarantee that the predictions are between 0 and 1. Therefore, I used a logit function to transform the targets from [0,1] to (-inf,inf) and then used these targets for the regression. Next, I used cross validation to find the lambda that gave the best test results and fit the model. The R^2 associated with the fit was .85, showing that a good amount of the variability in the target is explained by the model. Next, I transformed the predictions back to probabilities using a sigmoid function and calculated the test MAE using the original untransformed targets. The resulting MAE was .04 on the test set, which means that on average, our prediction is roughly 4% away from the target. This is an acceptable result as we are just looking for a prediction that is consistent with the rest of the data, which we definitely achieved.

Unsupervised Learning:
From an unsupervised standpoint, the techniques I used were principal component analysis (PCA) for dimensionality reduction and hierarchical clustering for visualization of potential clusters in the data corresponding to players of different play styles. Since the dataset containing match statistics has a relatively large number of features (some of which are certainly correlated), I used PCA to reduce the dataset from 14 features to 6 features while explaining 82% of the variance in the original dataset.

The chosen method of clustering for this dataset was hierarchical clustering as it is the most appealing visually and the most flexible because the dendrogram it creates can be cut at various heights depending on results. Hierarchical clustering was tried with multiple linkage parameters, namely, complete linkage, single linkage, average linkage, and ward linkage. The dendrogram from the hierarchical clustering algorithm using ward linkage can be found below.



From the dendrogram, we can see that the most logical cut point is somewhere between 20 and 30 so that 3 distinct groups are created. Upon further investigation into the players in each group, the green group contains mostly players that are very aggressive and the red and blue groups contain a relatively similar bunch of players. It does seem that the red group contains more aggressive players than the blue group, but as a whole they are difficult to distinguish by their contents. The plot below shows how the groups look when projected onto the first two principal components.



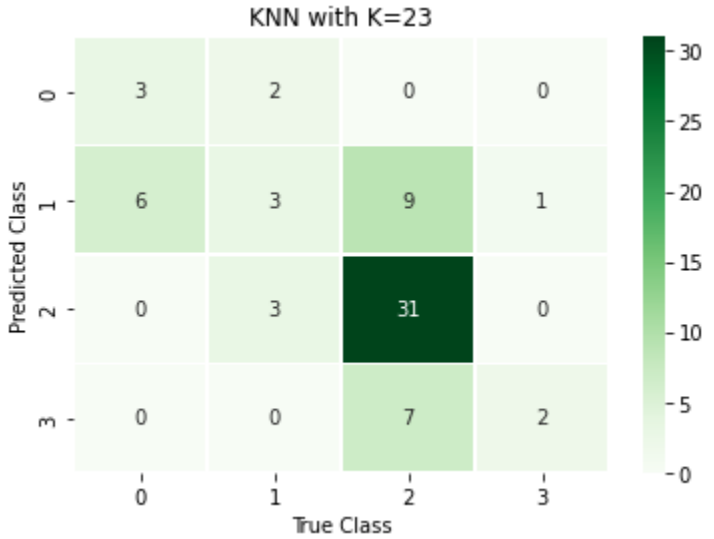
Since the additional dataset containing physical attributes of players (height and weight) is relatively simple, there were no interesting unsupervised learning results from this dataset.

Supervised Analysis Results:

The goal of this project is to create two models. The first model should classify players into 4 groups based on their match statistics and the second model should classify players into 4 groups based on their physical characteristics. Therefore, a variety of supervised classification models were tried on each dataset and the best method for each problem is shown below. In both cases, the models compared were K Nearest Neighbors (KNN), Random Forest, Support Vector Machines (SVM), and an Artificial Neural Network (ANN).

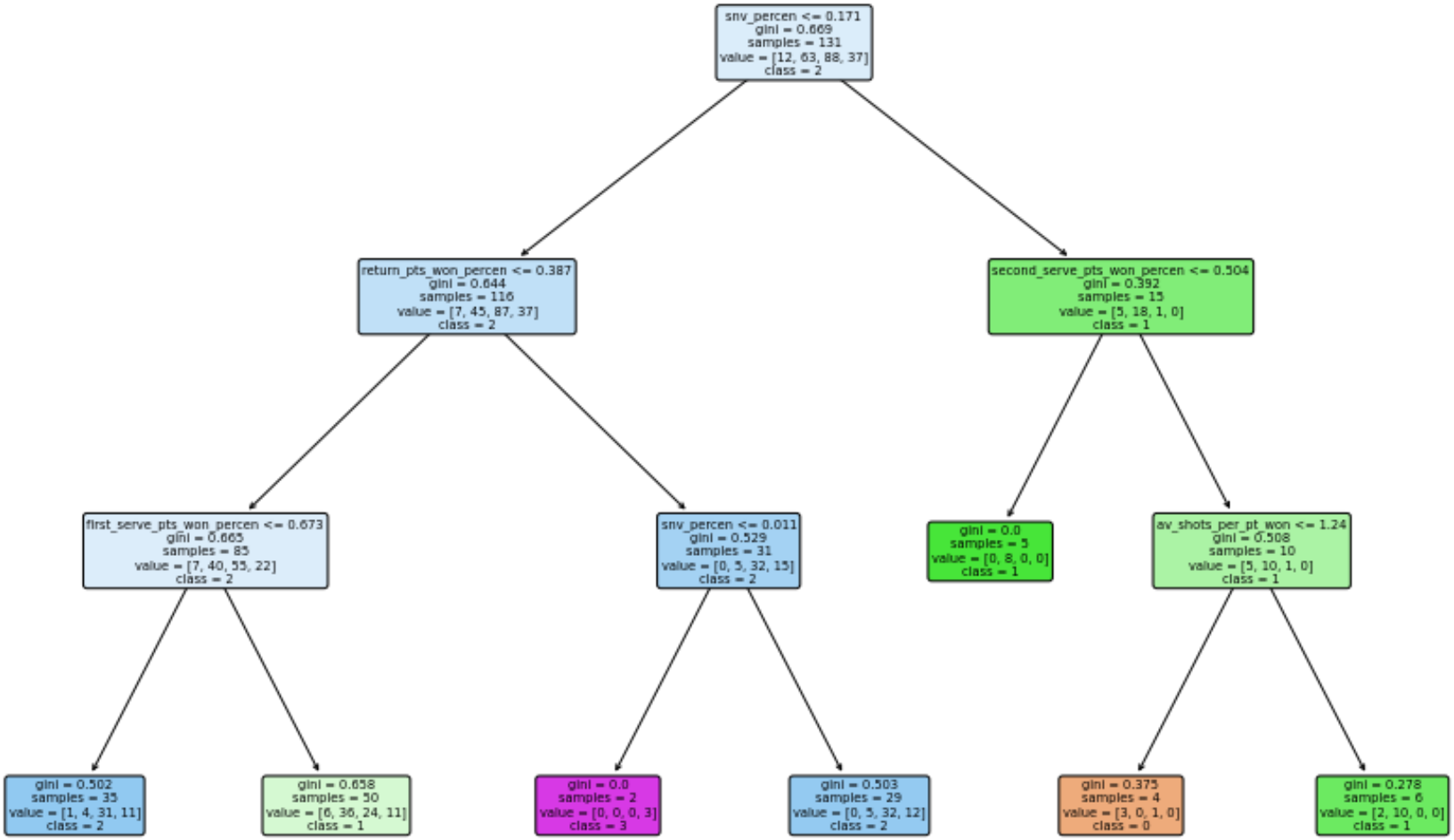
Predicting Class Based on Match Statistics:

Each model was fit separately with the 14 feature original dataset and the 5 feature dataset that was reduced using PCA. Additionally, each model was tuned using cross validation over a grid of various hyperparameters. The best performing model after fitting and tuning the models mentioned above with both datasets was the KNN on the 5 feature dataset that was created with PCA with roughly 58% accuracy. What's interesting about this result is that this model significantly outperformed the KNN model on the 14 feature dataset. Intuitively, this makes sense as KNN typically does not perform well on high dimensional data due to problems with calculating distance in high dimensional spaces. PCA reduced the data by nearly 10 features while also keeping much of the variance in the data which increased the KNN accuracy. The confusion table along with the optimal hyperparameters for this model are shown below.



From the confusion matrix above, we can see that although this model is not perfect, it rarely misses by more than one class. This suggests there is a pattern in the labeled dataset. The

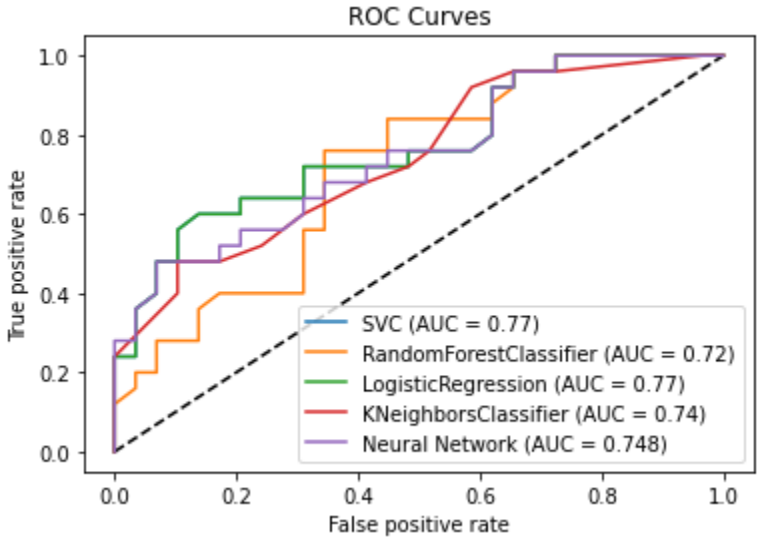
Random Forest on the full 14 feature dataset didn't perform nearly as well with an accuracy of 54%, but the branches of some of the trees in the forest can give us a good idea of which features may be the most important for predicting playstyle. A sample of one of the decision trees in the Random Forest is shown below.



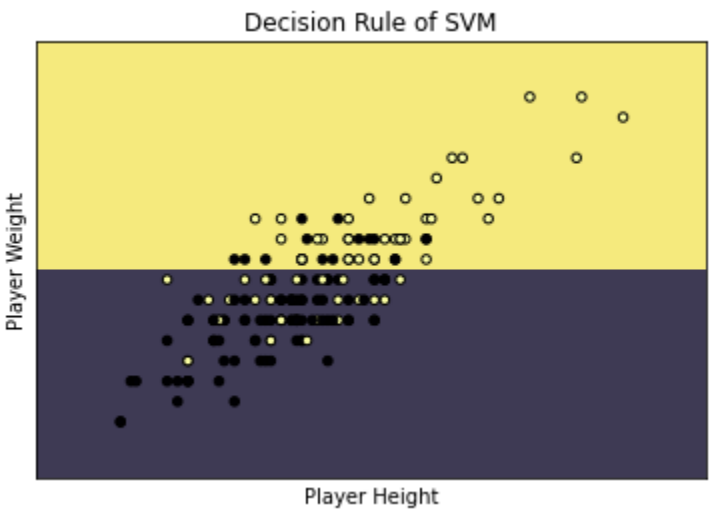
Here, we can see that this particular tree of the random forest initially splits based on serve and volley percentage (the column we would've lost had we not imputed missing rows) and seems to classify players as a 0 (orange node) or 3 (pink node) based on how short the points are and how many points they win on their return games, which is intuitively how a human that knows about tennis would probably classify them as well. Typically, players that are more aggressive tend to get more free points on their serves and come to the net more which shortens points, which in a sense is recognized by the model. The performance of every other model mentioned above can

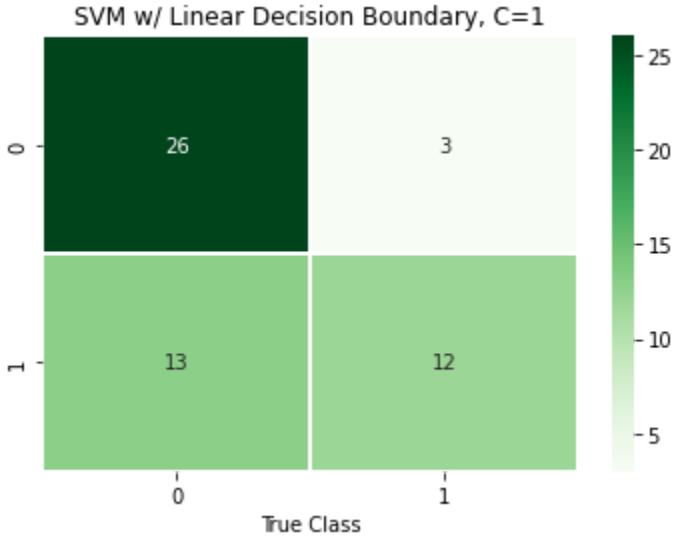
be found in the appendix. Next, we examine which model will be the best using physical characteristics to predict player style.

Predicting Class Based on Physical Characteristics:
The same set of models was used although only on a single dataset containing the height (inches) and weight (lbs) of players. In this case, models performed poorly with the best model barely reaching above 50% accuracy. Since the models were having trouble predicting classes accurately in the 4 class setting, I simplified the problem from multiclass classification to binary classification. Any target which was previously 0 or 1 was classified as a 1 and any target which was 3 or 4 was classified as a 0. Therefore, aggressive players were classified as 1s and defensive players were classified as 0s. Next, I trained the same set of models using the physical characteristics dataset along with the new labels. As expected, the results were significantly better. The ROC curves for the models used are plotted below.



All of the models performed relatively well, but the SVM has one of the highest AUCs and an accuracy on the test set of 71%. Below is a visualization of the decision boundary and a confusion matrix to visualize results.





What's interesting to note here is that both the SVM with a linear kernel and the logistic regression model had exactly the same ROC curves and the same confusion matrix. Thus, their decision boundaries are likely identical.

Conclusion:

The objective of this project was to create a tool that allows players to compare how they should be playing based on their physical characteristics to how they are actually playing using their match statistics. To achieve this, we needed models to accurately predict one of four classes based on the two datasets containing physical characteristics and match statistics. Unfortunately, the accuracy we achieved is not quite high enough in the multiclass setting. If we decided to limit the problem to binary classification, the models above would probably do a good enough job to create the tool, but this is too constrained and not helpful enough for amateur players.

The reason why the models did not perform accurately enough could be because of any of the following problems:

(1) Detailed data was only available for 250 professionals which is just not enough data
(2) The majority of players fall in group 2 so there was possibly not enough data on classes 0 and 3 for the models to train correctly
(3) Only 5 USPTA certified coaches were surveyed to create the labels for the dataset so some of the labels for the players could be incorrect or suboptimal.
(4) Playstyle is slightly influenced by personality which is impossible to model accurately

Although the models for the multiclass classification problem did not give optimal results, the binary classification models performed well. This indicates that there is a strong relationship between playstyle and physical characteristics, but we may need to know more about the player or have more data in order to go deeper than binary classification.

Next Steps:

This project could be improved by using more detailed physical data (top speed, muscle mass, etc.) and including more professional players. Additionally, surveying professional tennis sports analysts to create consistent playstyle labels for the dataset would probably create more accurate results as well. A logical next step for this project would be to talk to some of the organizations that handle the data for the ATP and see what the requirements are for getting access to some of their larger and more detailed datasets. Training models with more than 200 observations (roughly 50 observations were reserved for a test set) and more physical characteristics would probably generate very different results in the multiclass setting.

Appendix:

Below is a table of the models that were fit during the project. The code for each supervised model and all unsupervised learning techniques can be found in the code section.

Model:	Dataset:	Test Accuracy:	Tuned:	Optimal Params:	PCA:	Problem
Random Forest 1	Match Stats	54%	Yes	n_trees=31, max_depth=3 ,min_samples_leaf=4 min_samples_split=2	No	Multiclass
Random Forest 2	Match Stats	53%	Yes	n_trees=41, max_depth=1 ,min_samples_split=2	Yes	Multiclass
SVM 1	Match Stats	55%	Yes	C=1,gamma=.1,kernel=radial	Yes	Multiclass
SVM 2	Match Stats	52%	Yes	C=.1, kernel=linear	No	Multiclass
KNN 1	Match Stats	58%	Yes	n_neighbors=23	Yes	Multiclass
KNN 2	Match Stats	52%	Yes	n_neighbors=23	No	Multiclass
ANN 1	Match Stats	54%	No	2 hidden layers, dropout=.4, activation function=relu	No	Multiclass
Random Forest 3	Physical Chars	52%	No	n_trees=20	No	Multiclass
SVM 3	Physical Chars	46%	Yes	C=10,kernel=linear	No	Multiclass
KNN 3	Physical Chars	48%	No	n_neighbors=50	No	Multiclass
ANN 2	Physical Chars	48%	No	3 hidden layers, dropout=.4 activation=relu	No	Multiclass

Random Forest 4	Physical Chars	63%	Yes	n_trees=10, max_depth=3	No	Binary
Logistic Regression	Physical Chars	70%	NA	NA	No	Binary
SVM 4	Physical Chars	70%	Yes	C=.1, kernel=linear	No	Binary
ANN 3	Physical Chars	68%	No	3 hidden layers, dropout=.4, relu activation, sigmoid activation for final layer	No	Binary
KNN 4	Physical Chars	66%	Yes	n_neighbors=23	No	Binary