### Objectives

Our project had two main components:

- Use historical data from tennis matches to develop a model to predict match outcomes.
- Use this model and historical betting data to develop a betting strategy.

### Data

- Tennis match data was retrieved from an open source data set available on GitHub. It includes all match results from the Open Era (1968) to September of this year. Only the more recent matches have statistics associated with them.
- Betting data was retrieved from [http://tennis-data.co.uk/](http://tennis-data.co.uk/), which has odds from various betting services from 2001 on.
- These two datasets had to be merged. We were able to merge about 93% of the data. The merged dataset has 46,114 matches.
- This was split into a training set of size 41,324 and a test set of 4,790. (Roughly a 90-10 split.)

### Features

- We constructed features from match statistics that we thought would be relevant for match prediction.
- Careful not to have any “look-ahead”... features one could not know in advance of the match being played.
- In total we had 85 features. Each feature was of the form, for symmetry:

  $$F_{\text{FEATURE}_i} = \text{STAT}_{\text{player}_1} - \text{STAT}_{\text{player}_2}$$

Some of the player statistics used were:

- Rank, ranking points
- Head-to-head wins
- Head-to-head wins on the given surface
- Match wins in the last 5, 10, 20 matches
- Match statistic averages for last 1, 2, 3, 4, 5, 10, 20 matches (such as aces, double faults, etc.)

### Prediction Models

We tried a variety of models and evaluated them using 5-fold cross validation.

<table>
<thead>
<tr>
<th>Model</th>
<th>5-Fold CV Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>69.7</td>
</tr>
<tr>
<td>Neural Network</td>
<td>65.2</td>
</tr>
<tr>
<td>SVM</td>
<td>54.0*</td>
</tr>
<tr>
<td>(1 HL, 300 nodes, logit)</td>
<td></td>
</tr>
<tr>
<td>SVM - Linear Kernel</td>
<td>69.9</td>
</tr>
<tr>
<td>SVM - RBF Kernel</td>
<td>51.0</td>
</tr>
<tr>
<td>SVM - Polynomial Degree 3</td>
<td>54.0*</td>
</tr>
<tr>
<td>Logistic Regression w/L1 Reg</td>
<td>69.9</td>
</tr>
<tr>
<td>Logistic Regression w/L2 Reg</td>
<td>69.7</td>
</tr>
</tbody>
</table>

* Training the polynomial kernel SVM was too slow to cross validate.

### Betting Strategy

The betting strategy is modeled as a single-shot decision problem, where we choose the bet that maximizes the expected returns. Possible bets include betting on player 1 ($b = 1$), betting on player 2 ($b = -1$), or not betting ($b = 0$).

Model for Choosing the Bet

$$b^* = \arg \max_{b \in \{0,\pm1\}} E[U(\text{win, odds, } b)]$$

### Error Analysis

Two of the important features were the ranking and the odds.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Total #</th>
<th>Correct</th>
<th>Pct.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower Ranked Player</td>
<td>814</td>
<td>486</td>
<td>59.7</td>
</tr>
<tr>
<td>Higher Ranked Player</td>
<td>3976</td>
<td>2850</td>
<td>71.7</td>
</tr>
</tbody>
</table>

### Further Work

- More sophisticated betting strategies
- Trying additional models
  - Some of our ideas required more computational resources than we had
  - Deeper neural networks, tune hyper-parameters
  - Including polynomial order features
- Richer data such as on injuries, expert predictions, weather conditions, anything else that could affect match outcome.
- Further research into SVM models
- Speeding up training on non-linear kernels
- Best strategies to yield probabilities from SVM model predictions
- Understanding why our betting strategy is so streaky... we have longs periods of winning and losing