Motivation and Problem Statement

- On May 14, the Supreme Court legalized sports betting, paving the way to a new market worth an estimated $150 billion.
- In this project, we attempt to apply Machine Learning algorithms to predict the outcomes of certain betting indicators in the NBA, such as the Money-Line, the Point Spread, or the Over Under.
- More specifically, we focus on estimating the number of points scored by both teams in every NBA game.
- While we were able to closely approximate the number of points scored in a game, our estimates were not precise enough to allow us to "beat the house" on the long run.

Datasets

- The datasets acquired provide two forms of data: betting odds data, which informs us of the bets offered by sportsbooks, and game data, which gives us data summarizing NBA games.
  - Betting odds data: Sports Book Review Online offers betting odds for every NBA game since October 2007.
  - Game data: Basketball Reference provides game-by-game team- and player-level data, which we retrieved using Frank Goitia’s NBA crawler for every season since 2007-2008.

Features

- For every game in our dataset, we extracted the following information:
  - Statistics for both teams’ past three games. This includes simple statistics such as Points Scored or Total Rebounds, but also more complex features like Offensive Rating or Plus/Minus.
  - Season averages for both teams’ respective opponents in the past three game in the same categories.
  - Number of days since the last game for both teams.
  - Distance traveled by both teams.

Models

- **Loss:** MSE

  - **Collaborative Filtering:**
    - Build a sparse matrix $A$ containing past games.
    - We factor $A$ into $UE^T$ using
      $$\min_{u,v} \sum_{ij} (A_{ij} - UE^T_{ij})^2$$
    - Predict $u_iE_j^T$.

  - **Neural Network:**
    - Inputs: 762 features for each team.
    - Architecture: three hidden layers (size 500, 100, 20) with ReLU activations.
    - Output: predicted O/U.

  - **LSTM Network**
    - Process the past three games sequentially.
    - We add a fully connected layer to output the
      points scored.

Results

- **Table 1. Train and Test Errors for Various Models.**

<table>
<thead>
<tr>
<th>Model</th>
<th>Train MSE</th>
<th>Test MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest (baseline)</td>
<td>955.10</td>
<td>910.33</td>
</tr>
<tr>
<td>Collaborative filtering</td>
<td>2.95</td>
<td>459.85</td>
</tr>
<tr>
<td>Neural Network</td>
<td>369.17</td>
<td>369.84</td>
</tr>
<tr>
<td>LSTM Network</td>
<td>398.35</td>
<td>426.56</td>
</tr>
</tbody>
</table>

Future Work

- Augment the dataset:
  - Incorporate the odds lines offered by various sportsbooks as features in our models
    - Perhaps we can learn trends such as that the books tend to overestimate the
      performance of certain teams
  - Incorporate player-level data, not just team-level data (should help account for when
    we know a player is injured before a game starts)
  - Explore novel neural network/LSTM architectures to take the additional features
    mentioned as input and train over longer sequences of games
  - Design architectures more similar to state-of-the-art rating prediction models
    (map user/item relationship to team/team relationship)

Discussion

- **Discussion:**
  - Due to the rapidly changing nature of the NBA it is difficult to acquire sufficient training
    data that reflects the way the game is currently played.
  - Note that the Collaborative Filtering model, which didn’t use any team features,
    outperformed the Random Forest:
    - This shows the high variance in our data as well as the strong seasonal trends that a
      model needs to encompass in order to be accurate on this task.
  - We achieved a test Mean Squared Error of 369.84 for our best model.
    - Encourages future work to be done on feature selection and
      engineering.
    - Current best models can beat the house around 51.5% of the time,
      but successful long-term betting patterns need to be correct at
      least 52-53% of the time.

References

- Fran Goitia, Basketball Reference Scraper, (2017), GitHub repository.
  - [https://github.com/FranGoitia/basketball-reference](https://github.com/FranGoitia/basketball-reference)