# Predicting How Points End in Tennis

Jonathan Williams, Zhongyi Zhang, Stevan Zlojutro

# Business Problem & Objective

- Data competition hosted by CrowdANALYTIX and sponsored by Tennis Australia
- Seeking an automatic solution for identifying point-ending shots in tennis through the use of tracking data
- Current system uses manual coding and subjective decision-making which could include inconsistencies from one coder to the next
- Potential to provide more accurate point-ending classifications

# Our Data

Focus on classifying point-ending shots in tennis: **Winners, Forced Errors, and Unforced Errors**
**Data Dictionary defined 25 features**

## Overview

- Men’s and women’s data sets
- Player and ball tracking data for the penultimate (second-to-last) shot and final shot of a rally

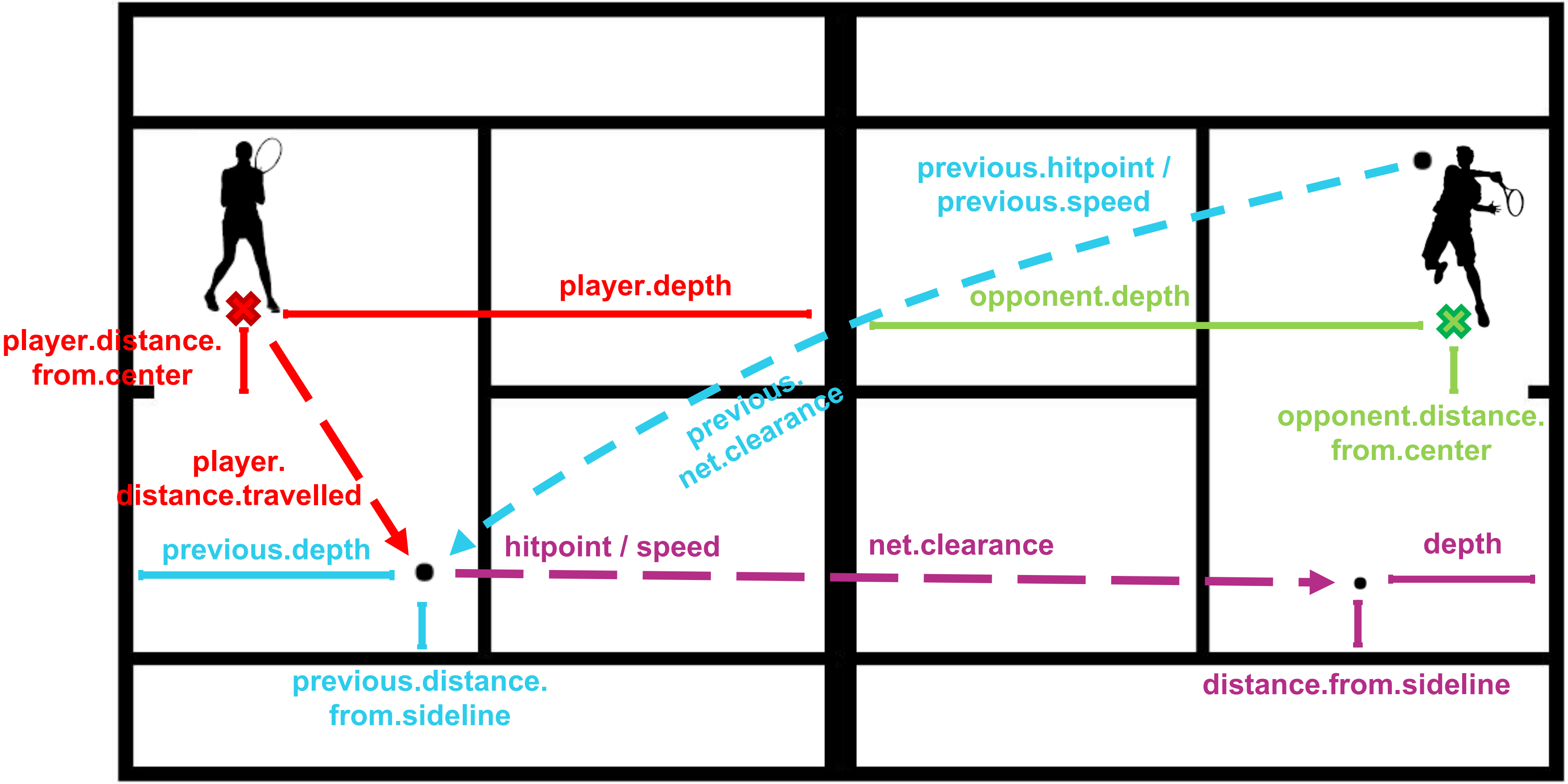
## Player Tracking Data

- Speed
- On-court coordinate position at time of shots
  - 2D coordinates of player position
- Distance travelled to hit final shot

## Ball Tracking Data

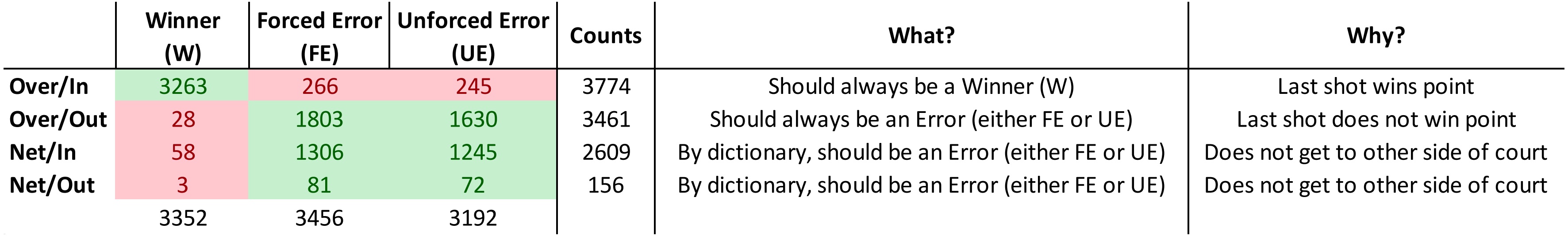
- Type of shot hit
- Speed
- Net clearance
- On-court coordinate location
  - 2D coordinates of ball position

# Our Data


previous.hitpoint /
previous.speed
player.depth
opponent.depth
player.distance.
from.center
previous.
net.clearance
opponent.distance.
from.center
player.
distance.travelled
previous.depth
hitpoint / speed
net.clearance
depth
previous.distance.
from.sideline
distance.from.sideline

# Data Assumptions

- Almost equal distribution of outcomes
- Misclassifications in the data due to technological tracking errors and missed calls (6% of entire data set)

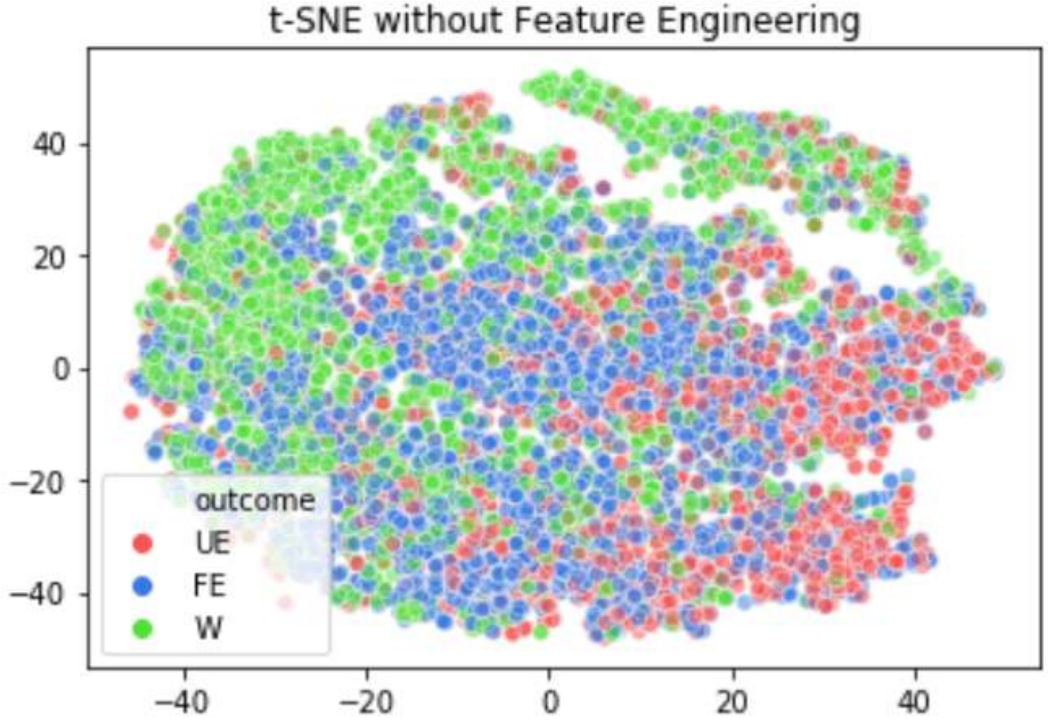


| | Winner | Forced Error | Unforced Error | Counts | What? | Why? |
|---|---|---|---|---|---|---|
| | (W) | (FE) | (UE) | | | |
| **Over/In** | 3263 | 266 | 245 | 3774 | Should always be a Winner (W) | Last shot wins point |
| **Over/Out** | 28 | 1803 | 1630 | 3461 | Should always be an Error (either FE or UE) | Last shot does not win point |
| **Net/In** | 58 | 1306 | 1245 | 2609 | By dictionary, should be an Error (either FE or UE) | Does not get to other side of court |
| **Net/Out** | 3 | 81 | 72 | 156 | By dictionary, should be an Error (either FE or UE) | Does not get to other side of court |
| | 3352 | 3456 | 3192 | | | |

- Treated men’s and women’s data as one data set
    - Originally provided separately

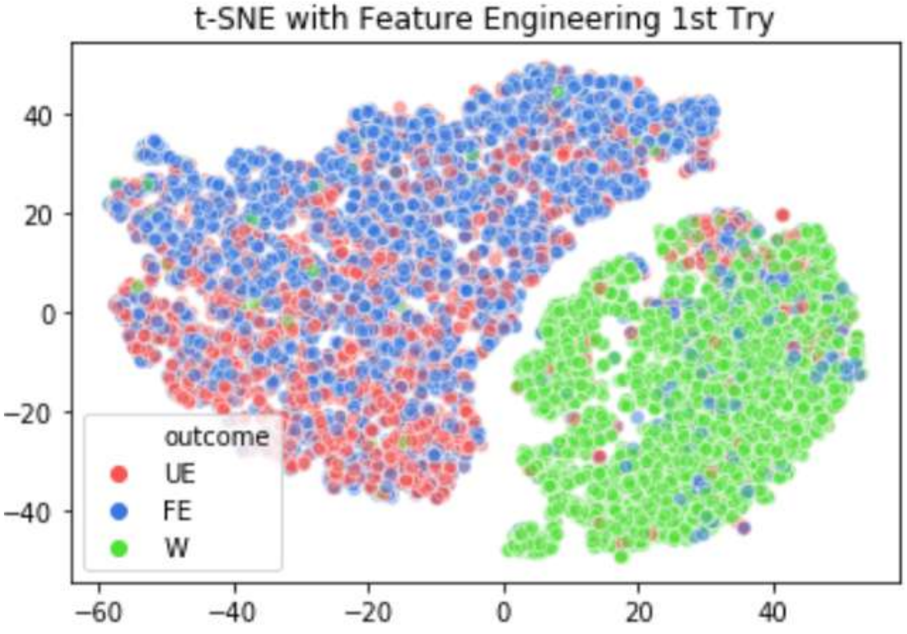
# Data Properties & Initial Visualization

- Categorical data (total of 8 – e.g. shot type, in/out of court, serving player)
- Continuous data (total of 17 – e.g. on-court player position, ball position)

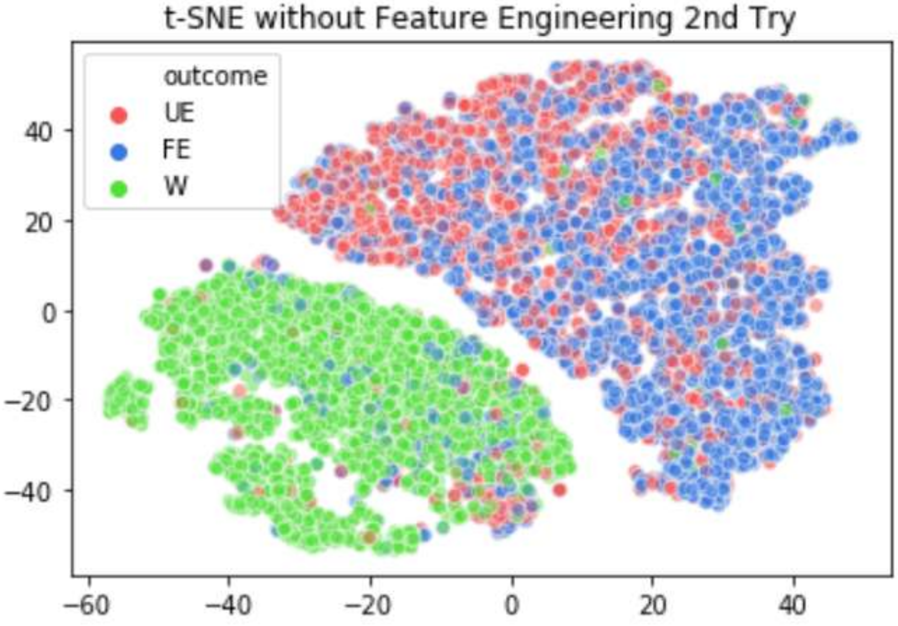


# Feature Engineering

- Learned that this was a feature engineering problem



t-SNE with Feature Engineering 1st Try



t-SNE without Feature Engineering 2nd Try

# Feature Engineering

### Data Inaccuracies Features (1st try)

- Over In
- Over Out
- Net In
- Net Out

### Euclidean Distance Features (1st try)

- Player Distance Start
- Player Distance End

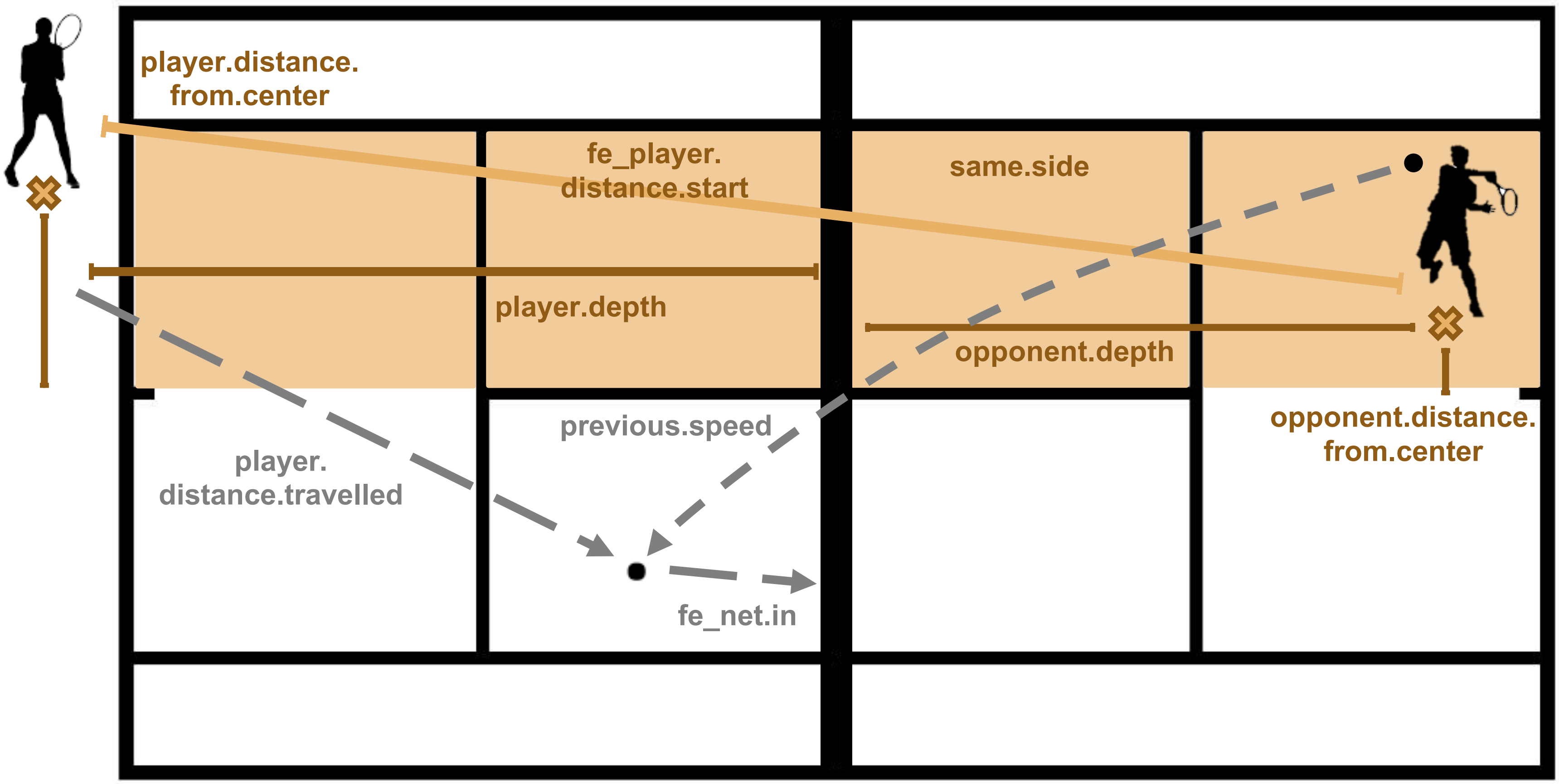
### Error Shot Differentiation Features (1st try)

- Rushed Shot
- Flopped Shot

### Gender Error Shot Differentiation Features (2nd try)

- Men Speed
- Women Speed
- Men Previous Distance from Sideline
- Women Previous Distance from Sideline
- Men Opponent Depth
- Women Opponent Depth
- Men Player Distance Travelled
- Women Player Distance Travelled

# Feature Engineering



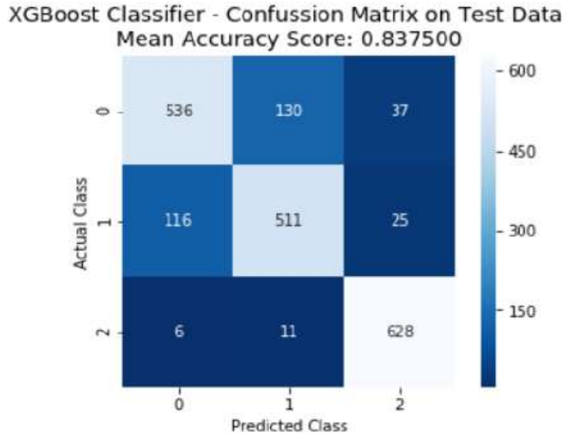
# Data Transformations

- Dummy Variables for Categorical Variables
- Feature Variables
- Scaled Data
- Explored Second-Order Polynomial Interactions

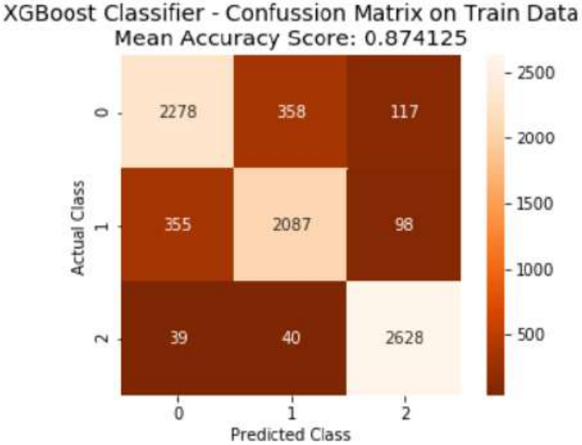
# Model Performance

| | | Base Model | | Final Model | |
|---|---|---|---|---|---|
| | | Original Dataset | | with FE | |
| | Models | Test | Train | Test | Train |
| **1** | Random Forest Classifier | 0.7665 | 0.7749 | 0.8325 | 0.8339 |
| **2** | Ada Boost Classifier | 0.7625 | 0.7626 | 0.8070 | 0.8156 |
| **3** | Gradient Boost Classifier | 0.8490 | 0.9923 | 0.8545 | 0.9903 |
| **4** | XGBoost Classifier | 0.8585 | 0.8834 | 0.8375 | 0.8741 |
| **5** | SVC + Poly Kernel | 0.8215 | 0.8620 | 0.8590 | 0.9060 |
| **6** | SVC + RBF Kernel | 0.8370 | 0.8720 | 0.8660 | 0.9053 |
| **7** | K-Nearest Neighbors | 0.7660 | 1.0000 | 0.8340 | 1.0000 |
| **8** | Soft Voting Ensemble | 0.8580 | 0.8884 | 0.8605 | 0.9545 |

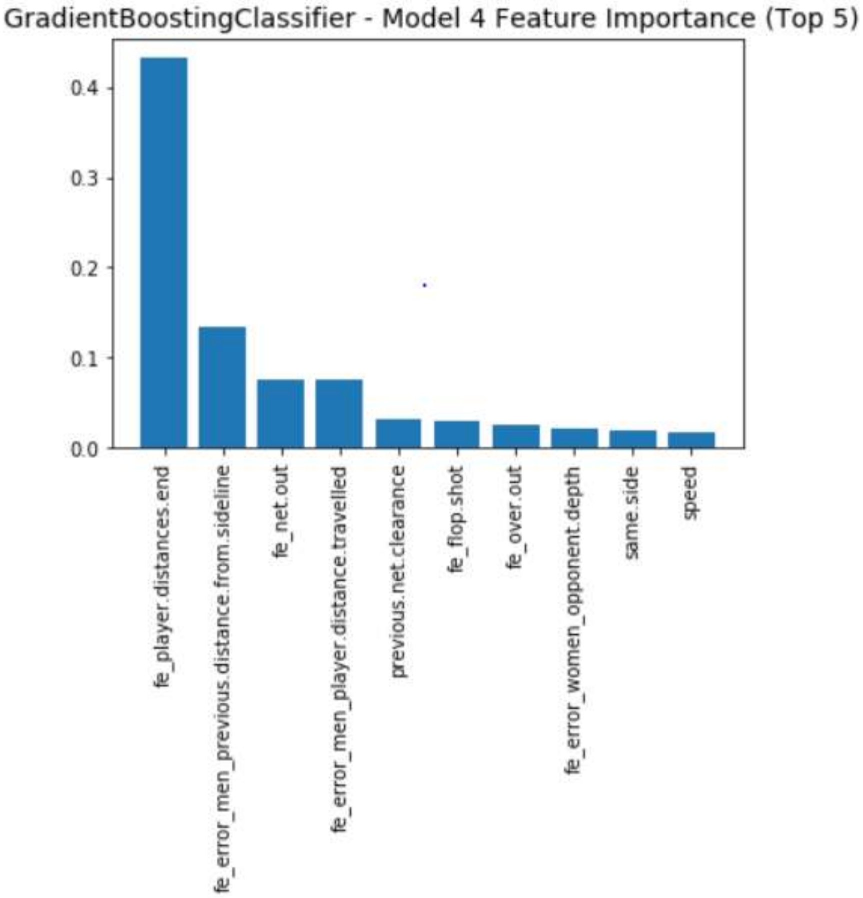
# Model Feature Importance



XGBoost Classifier - Confussion Matrix on Test Data
Mean Accuracy Score: 0.837500



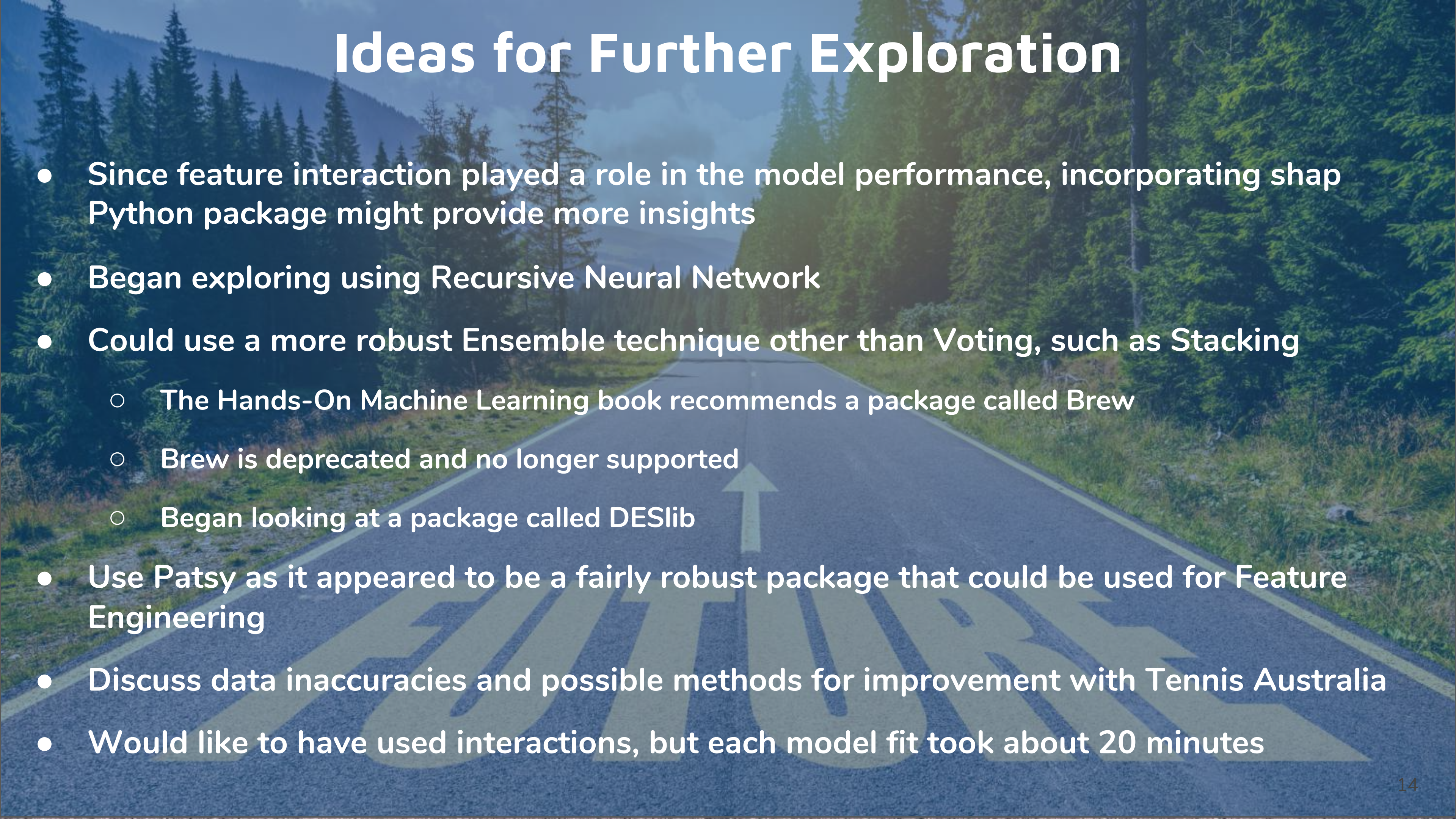
XGBoost Classifier - Confussion Matrix on Train Data
Mean Accuracy Score: 0.874125



GradientBoostingClassifier - Model 4 Feature Importance (Top 5)

# Selected Model:
# Soft Voting Ensemble

# Ideas for Further Exploration



- Since feature interaction played a role in the model performance, incorporating shap Python package might provide more insights
- Began exploring using Recursive Neural Network
- Could use a more robust Ensemble technique other than Voting, such as Stacking
  - The Hands-On Machine Learning book recommends a package called Brew
  - Brew is deprecated and no longer supported
  - Began looking at a package called DESlib
- Use Patsy as it appeared to be a fairly robust package that could be used for Feature Engineering
- Discuss data inaccuracies and possible methods for improvement with Tennis Australia
- Would like to have used interactions, but each model fit took about 20 minutes

# Questions