Jesse Gan (jzg219), Nisarg Patel (nrp341), Jason Huang (jjh596), Alan Ma (am7216)

Data Mining for Business Analytics Final Paper

05/22/19

# Data Mining for NBA Analytics

## Our Goals

The goals, operations, and overall success of NBA franchises revolve entirely around the players that make up each team. For both small and large market teams, roughly 60% of a franchise's revenues go towards player salaries (6% going towards the luxury tax for large market teams), with the salary floor and cap set at $91 million and $102 million, respectively. Because so much of their money goes towards paying their players, it's critical for NBA teams to recruit the optimal players such that their money is worth it and their team performs well. By extension, being able to accurately evaluate and value rookies is an extremely important task is an important part of assembling the best possible roster for any team.

The first goal of our project is to enable teams to better identify which rookies are most likely to reach a star-level in their career. If teams can predict which rookies have the most future potential, then these teams will be able to sign them to long-term contracts before these players reach stardom and the cost premium attached to it, in addition to potential loyalty by these players to stay with the team. The decision of these franchises can alter the team's performance for the next decade. The problem is, this decision is extremely difficult. Since 1980, 76% of #1 picks end up being all-stars while none of the other top 8 picks (lottery picks) have above a 50% chance of being all-stars. In addition, of all 59 active NBA players who have been selected as all-stars at least once in their careers, nine have been second-round draft pick (Draymond Green,

Goran Dragic, DeAndre Jordan, Marc Gasol, Isaiah Thomas, Paul Millsap, Kyle Korver, Manu Ginobili). This is a not-insignificant amount, and these figures together illustrate how teams are often unsuccessful in identifying the talents that will best develop into the best players, and thus best investments, for the team ("rookie contracts are guaranteed for the first two years of a player's career, with teams then having the option to extend the contracts in the third and fourth years as the player's salary increases exponentially each year")(Huddleston).

The other component of the business problem is being able to identify whether a team's current rookies have enough future potential. Doing so would help teams decide on a number of things: whether or not to include them in trades, extend their rookies contracts, and how much to pay them. There are many examples where teams trade away rookies in their first or second year and those players end up becoming all-stars at their new team. Thankfully, there are vast amounts of player-specific information that is publicly available through the NBA. We will use historical information on players to determine a feature set and create a model(s) to predict which rookies will become NBA all-stars at some point in their careers, a strong indicator of a successful player.

In short, we want to help NBA franchises in finding players early in their careers that are likely to develop into successful players, which will benefit the organizations by streamlining scouting, avoiding the need to pay high salaries for players who do not deliver, and reducing the opportunity cost of missing out on the best talent. A data mining solution will help address this issue by utilizing the extensive amounts of NBA data and historical information on players and their performance to predict future all-stars.

**Data Understanding and Preparation**

**Overview**

To perform this data mining task, we will be using per-game NBA statistics from each player's rookie season. This data is pulled directly from the NBA website where they have collected statistics for all players each year starting from 1997. This data includes mostly numerical data with the only categorical feature being the team played for. Along with the data collected, we perform some data cleaning techniques to add and adjust a few of the features. We will discuss those changes later in this section. For each of the players we choose to include in our final dataset, we determine if they have been an All-Star and use that binary data point as our target variable.

The final features are as follows (with abbreviations and description): Age of player (AGE), games played (GP), win percentage (W%), loss percentage (L%), minutes playerd (MIN), points (PTS), field goal made (FGM), field goals attempted (FGA), field goal percentage (FG%), 3-pointers made (3PM), 3-poiners attempted (3PA), 3-pointer percentage (3P%), free throws made (FTM), free throws attempted (FTA), free throw percentage (FT%), offensive rebounds (OREB), defensive rebounds (DREB), rebounds (REB), assists (AST), turnovers (TOV), stealing (STL), blocks (BLK), personal fouls (PF), fantasy points (FP), double doubles (DD2), triple doubles (TD3), plus-minus (+/-), features for the rank of each player in all of the previous features - except AGE and GP - relative to the players in their rookies season and all-star roster (ALL STAR). You can view an example of the final dataset in *final_dataset_1.csv*.

**Selection of Data**

The data for NBA statistics are separated by seasons (we refer to by the year the season ends; 2018-2019 season is referred to as 2019 season) and the data for each player is also

separated by each season they play in. This makes choosing the correct seasons to extract our data from important. There were a few key decision we made when selecting the exact data to extract: 1) which season of a player's career do we select, 2) which years of players to use to select from, 3) which years of all-stars to refer to, and 4) which years of rookies to include in final dataset.

Our final dataset includes only the features from each selected player's rookie season. This was an important thing to clarify in order to avoid any data leakage. Because we aim to use the model to help NBA management determine the future success of their rookies after their first season, we can only include data from the rookie season of the player statistics we train on. This decision could be modified to include more years of data or the average of a certain set of years of players if the business purpose of the model was changed.

We did not have clear answers to the other three decisions so we created four final datasets that are a combination of a few possible answers:

**Dataset 1:** 2015-2019 Players, 2013-2019 All-stars, 1997-2019 rookies

**Dataset 2:** 2015-2019 Players, 2015-2019 All-stars, 1997-2014 rookies

**Dataset 3:** 1997-2019 players, 1997-2019 All-stars, 1997-2019 rookies

**Dataset 4:** 1997-2019 players, 1997-2019 All-stars, 1997-2014 rookies

We did recognize that the seasons we choose for players to include in our final dataset are linked to the seasons we choose as a list of all-stars. This is because we want our data set to include players within the same season that were competing for the same spots on the all-star list. There is no reason to include a player playing in 2000 while only including a list of all-stars from 2019. Thus the answer to those two questions is the same. We, however, were unable to choose

whether to include all the seasons of data we could collect or only just the most recent 5 years. The reasoning for only including a small subset of years is that we recognize that the nature of players who are the best in the league changes from season to season. Including only the most recent 5 years may allow us to isolate the features that help predict modern NBA all-stars. This assumption may be false.

The other decision we need to test is whether it is important to exclude rookies from the most recent 5 seasons because they may include instances that have would be predicted to be an all-star but just have not played in the NBA long enough for that to be true. These instances may affect the training of the model and lead to more false positives. We want to test if the exclusion of those rookies season statistics affects performance.

**Cleaning the Data**

The original dataset extracted from the NBA website included only the nominal values for per-game statistics. We added a key set of features that represents the ranking of each player for each statistic category. The rankings are made relative to other rookies in their rookie season. The goal here is to represent their statistical performance relative to players playing in the same playing conditions (same season) in order to eliminate any bias from changing conditions of the league. As shown in our testing, we use these ranking features as an alternative to the nominal features.

Other minor data cleaning steps involved removing names as a feature, removing team as a feature, and changing total wins and losses to win and loss percentages.

**Steps in Data Preparation**

1. Scrape NBA player statistics from nba.com for each season from 1997-2019 for all players in the season and all rookies.
    a. Using Selenium and BeautifulSoup, to automate clicks to select a few key options on the website to scrape all files in one run. Files are stored in separate cv files named by whether all players or rookie and by the season.
    b. For each season, add features for the ranking of each statistical category relative to other players in that same season.
    c. Refer to *scraper_nba.com.py*
2. Compile a list of all All-Star players for the last 25 years and label each player with the most recent year they were an All-Star.
    a. The result is a 2-column table with names and season years.
    b. Refer to *All-Stars.csv*
3. Create a list of player names that played during the seasons selected.
    a. Done by reading each *player_stats_*[season].*cv* files for the corresponding seasons selected to be included
    b. Refer to *Data_file_maker.py*
4. Create a list of all-star players in the selected seasons to be included
    a. Done by adding player names in *All-Stars.csv* that have a matching season number to the list of seasons selected.
    b. Refer to *Data_file_maker.py*
5. Collect rookie season statistics for every player in the list of players selected that have played a rookie season in the selected rookie season years.

    a. Done by reading through each player instance in each rookie season statistics file and comparing the name to the list of players.
    b. If matches, add the rookie season statistics to the final Dataframe
6. Add All-Star label.
    a. Go through each instance in the final Dataframe. If the name matches one on the all-stars list, then set 'ALL STAR" label to 1, else set it to 0.

## Running Our Classification Models

### Note on Measuring Performance

We chose recall as the preferred performance measure because it makes the most sense given the business problem. For an NBA franchise, missing a potential all-star is far most costly than investing in a player who will not reach all-star status. That means the model's performance should be minimizing false negatives because that could mean trading away a player with a lot of potential. We want to be sure that when a NBA team uses this model, they can be certain that the model will be able to provide insights to how the team's rookies should be valued.

We think this is better than using area under the curve because we want the model's performance to indicate its success in predicting all of the rookies that become all-stars. AUC does not punish the score of a model for false negatives because the class probability is so small. It is easy for the model to predict the instances that are most likely to become all-stars which is why the AUC scores always look good.

### Note on Feature Selection

Our goal was for our models to achieve the best score possible which is why we decided to employ feature selection for both our ranking datasets and our raw score datasets. Instead of

limiting our dataset to an arbitrary number of features, we modified the feature selection method to return as many features as it could that improved the recall of the models until there was no more improvement.

For the Decision Tree model we initially attempted to do feature selection, but only the Age of the players came up as the best feature using the feature selection method. However, we realized that the feature selection for the Decision Tree was flawed as we would already be doing such a thing by deciding the ideal depth level for the tree, so we decided to forego the feature selection for the Decision Tree model. From then on, we began to create our models to determine whether or not All-Stars could be predicted from their rookie player stats, beginning with Logistic Regression.

**Logistic Regression:**

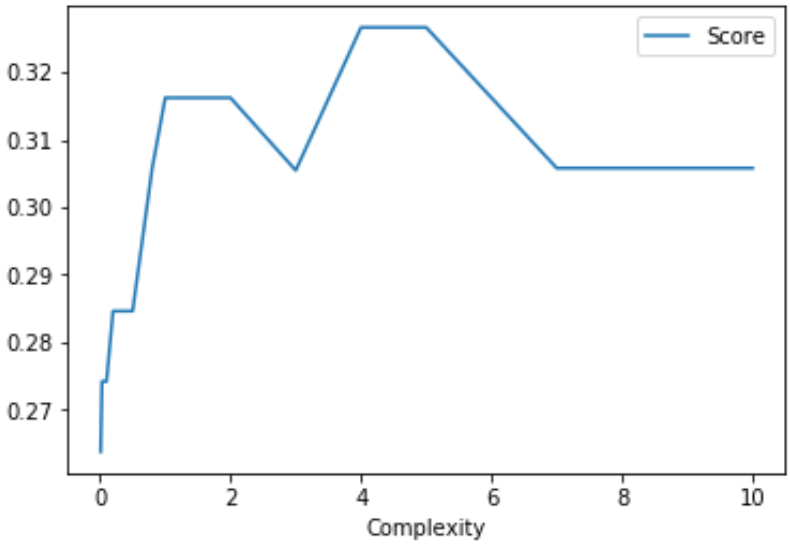
**Feature Selection**

We performed feature selection technique as discuss on each of the 4 datasets and with 3 combination of features (raw features, ranked features, and both). The result of running these 12 feature selection algorithms were the following: Dataset 4 using both raw and ranked features produced a recall score a 31.62% with the best features being fantasy points (FP), age (AGE), 3-point percentage rank (3P% Rank), win percentage (W%), field-goals made (FGM), turnovers (TOV), and plus-minus (+/-).

For the next steps, we chose to only focus on this dataset and feature set collection that produced the highest score.

**Regularization**

The Logistic Regression model is the typical classification model for us to use, and to do so we defined a `log_model` method which incorporates a 3-fold cross-validation score to determine at what level of complexity does the model operate the best.

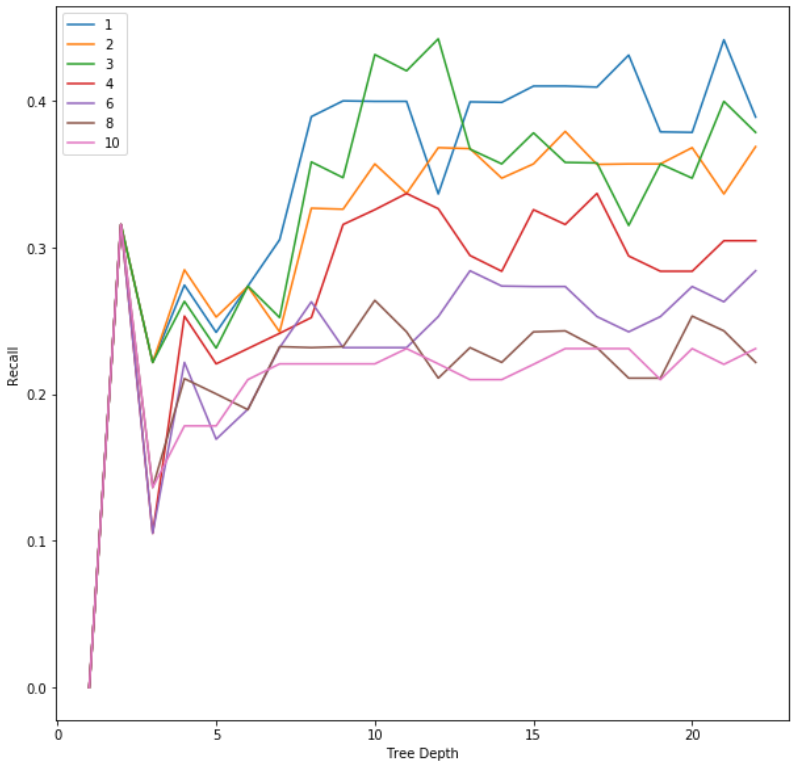



The results of this show that the show that the model performs best with a regularization of 5. Results of other tests can be viewed at the end of the report.

**Decision Tree:**

We decided to conduct our project using the decision tree model as well given the business context we are working with as it would be easier to explain to people in a client setting. In a similar process, we attempted to determine which complexity parameters (tree depth and minimum leaf size) were the best for our project so we could create the best recall possible.

The best model based on recall score was again dataset 4 with both raw and ranked features. This model yield a recall score of 44.25% with a tree depth of 12 and a minimum leaf size of 3. A visualization of complexity parameter testing is shown below with the green bar (min leaf = 3) yielding the highest score at depth of 12.




This model implicitly selects the following features to split the instances: AGE, GP, W%, MIN, FGM, FT%, 3PM, REB, OREB, DREB, FP, PTS RANK, FGA RANK, FG% RANK, FTA RANK, 3PA RANK, 3PM RANK, 3P% RANK, AST RANK, STL RANK, BLK RANK, PF RANK, FP RANK, and +/- RANK. In total, this model selects 24 of the 49 features. You can use the *tree.dot* file to create the visualization shown below:

The recall scores from the other variations of datasets we used can be found in our iPython notebook, for your reference.

Results of other tests can be viewed at the end of the report.

## K-Nearest Neighbor Classifier

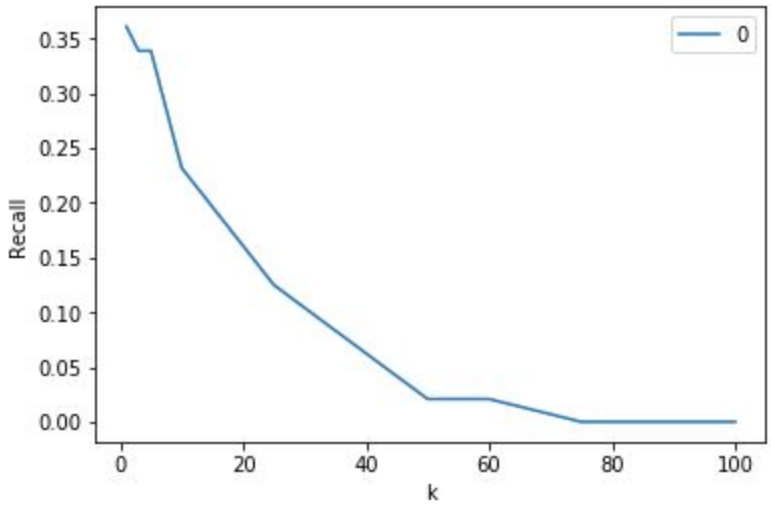
### Feature Selection

We decided to use the k-NN model as well to predict which rookies will become All-Stars because the k-NN is one of the core classification models we worked with in the course, and has the key benefit of being versatile to use in real-time scenarios because it conducts instance based learning. We performed the same feature selection as for the logistic regression model to determine the best features set and baseline performance for each dataset with each feature set. The best results were with Dataset 1 using both raw and ranked features.

The best feature set was triple doubles rank (TD3 RANK), points(PTS), 3-pointers attempted (3PA), and free throws attempted (FTA).

The following regularization was subsequently performed on this dataset and feature set.

**Choosing K**




The model determines that the best value for k is 1 from dataset 1 with a recall score of 36.11%. The iPython notebook is setup in such a way that, for your reference, you can simply change the file name in the read_csv method and return all of the graphs shown for the different datasets.

Results of other tests can be viewed at the end of the report.

## Evaluation

**Selecting the best model**

From our work, we were able to determine that the best model is the Decision Tree, which yielded a recall score of 44.25% with a tree depth of 12 and a minimum leaf size of 3 for its regularization parameters. Even besides the lower scores of the other models when compared to their best performing datasets, on average, the decision tree yielded better recall scores, making

it more of a versatile model in a the basketball setting where the game itself is changing year over year depending on player styles.

**Understanding Value of the Model**

The low recall score of the model may indicate the failure of the model to predict successful cases, however the results must be compared to the base rate of rookies that become all-stars. In 2019, there were 103 rookies (60 whom were drafted). In the past 40 years of the NBA, a rookie class has produced at most 11 all-stars which occured in 1996. This makes a conservative base-rate of 16%. So if we think that if our model is able to provide some additional insight beyond this base rate, we will be able to help NBA teams decide the future value of their current rookies, especially ones that are not obviously future all-stars.

**Business Case and ROI**

We think it will be very difficult to create an business case to measure ROI because of the nature of the problem. First off, a NBA team will only have a few instances (rookies) to build a business case around, this gives a team limited cases to test on. This could potentially be mitigated by evaluating the result of the model for other teams' rookies but that still limits the instances to around 80-100 instances each year. Second, the impact or ROI of these decisions are both hard to evaluate and long-term. The impact is hard to evaluate because success for an NBA franchise depends on so many different factors and having an all-star franchise player is only one of those factors that can impact metrics such as fan attendance, total wins, merchandise sales, among other metrics for success for an NBA franchise. This makes the impact of successfully predicting a rookie to be an all star extremely difficult to quantify. Also, these metrics may take 5-10 years to materialize because players often take a few years to develop before becoming

all-stars. This is especially true for players who are not obvious all-star players. Because of the undefined timeline for returns, ROI is even more difficult to quantify and evaluate for this model.

The evaluation of this model will be more based on the qualitative assessment of the deployment of the results. For example, has a player we predicted to be a future all-star continually improved year after year until they become an all-star or has a player we traded away that we didn't predict to be an all-star perform poorly in subsequent seasons.

## Deployment

There are various deployment techniques that we have identified. Some require additional data mining techniques but all of them use this model to drive decisions. We will be discussing two techniques here:

### 1. Salary and Contract Decisions

The model can be deployed as a way to guide contract decisions. A team can continually train the model each year using data from each new season. Once the model is created, it can be used to rank current rookies in the league on their likeliness to become an all-star. Each team can then use this as guidance to determining whether to extend a rookie contract and the size of the contract.

To give context on the business problem, an NBA team has the option to extend a rookie's 2-year contract into their third year but this must be done from the end of the rookies season until the following October 31. If not done, a rookie will have the option to leave the team and sign with a contract offer from another team after their second season. If the extension is signed, the team also needs to decide how much to pay. Rookie contracts are locked to either

80% to 120% of a predetermined value assigned to each draft pick number. Thus, at the end of a player's rookie season, teams can use that players likeliness to becoming an all-star as a guide to these two decisions.

The threshold for each team in determining what percentage is a a value they want to act on depends on numerous external factors. For example, a team like the Golden State Warriors does not need to invest in finding the next rookie all-star because they already have an franchise player and are already successful. This means the threshold will be higher and they will only care if the model predicts a relatively higher likeliness to become an all-star. On the other hand, a team like the Memphis Grizzlies are one of the worst team in the NBA and have no big-name all-star players that will help the franchise grow. This means they will be more willing to take a risk on some rookies that have a reasonable chance at becoming an all-star. Their threshold would be lower.

**2. Trade Value**

Similar to the previous technique, the model can be used to rank all rookies in the NBA and thus be used to compare one rookie to another. Especially in today's NBA, rookies and second-year players have become a valuable asset to any team that is looking to trade for more established players. Even if not for trading up for better players, young players are still often swapped around the league. This model can be used as another metric for determining the trade value of players in these deals. For example, perhaps a team is looking to trade away one of their rookies after their first season. They can use the model to predict the likelihood of other players on the market of becoming all-stars and use that information to make sure they are not making a mistake trading away a young player with all-star potential.

**Risks**

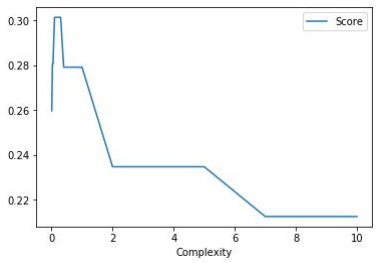
The biggest risk of using this model is that ultimately it is only predicting about 40% of the future all-stars from their rookies stats. Although this is better than the base case that we discussed earlier, it will still result in many false negatives. This means the firm needs to ensure that the results are coupled with many other metrics gained from having a business understanding of the problem. This model is just another evaluation method that NBA franchises can use to find an edge in identifying the best players in an extremely competitive league.

## Model Regularization Visualizations:

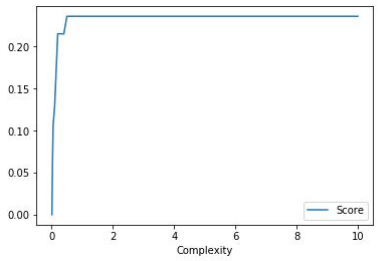
For your reference, the following graphs show the regularization results of all of the datasets we used with their respective models and the three variants of ranked data, raw data, and both raw and ranked data. The underlined dataset is the best performing one in that model.

## Logistic Regression

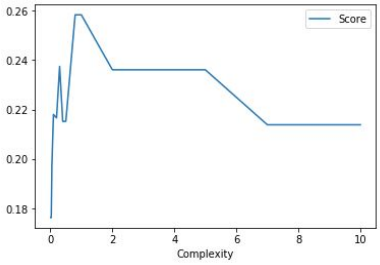
**Dataset 1: Ranked Data**



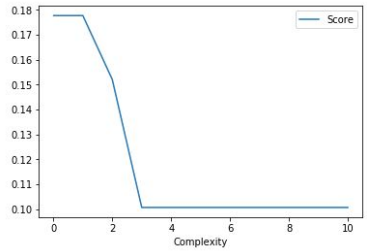

**Dataset 1: Raw Data**



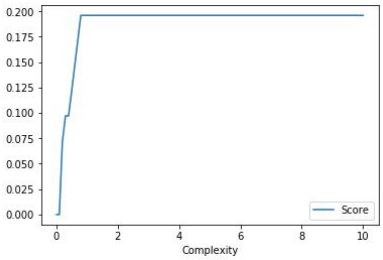

**Dataset 1: Combined Data**



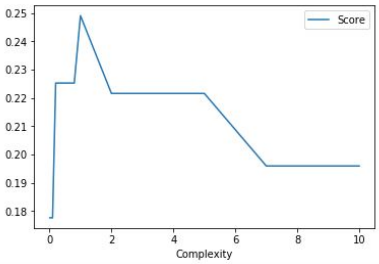

**Dataset 2: Ranked Data**



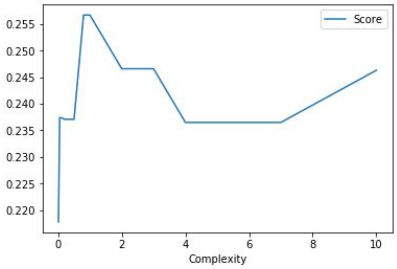

**Dataset 2: Raw Data**



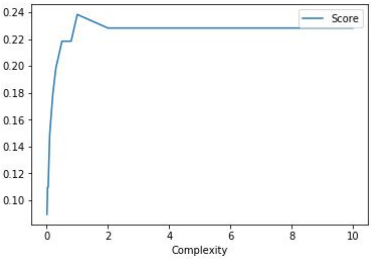

**Dataset 2: Combined Data**



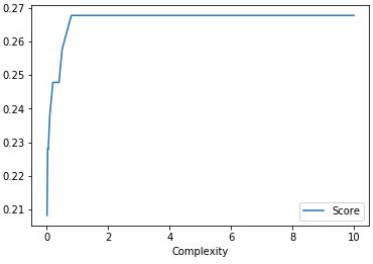

**Dataset 3: Ranked Data**



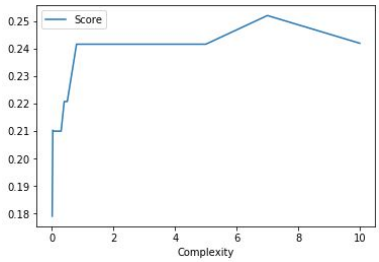

**Dataset 3: Raw Data**



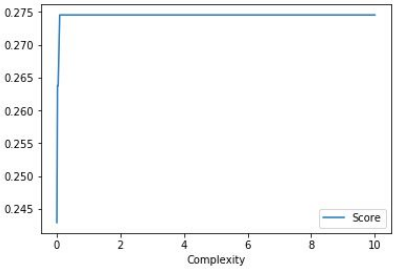

**Dataset 3: Combined Data**



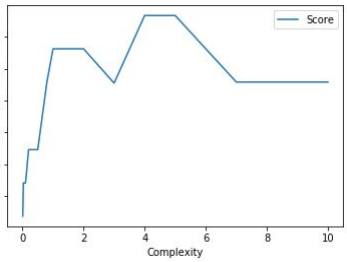
**Dataset 4: Ranked Data**




**Dataset 4: Raw Data**

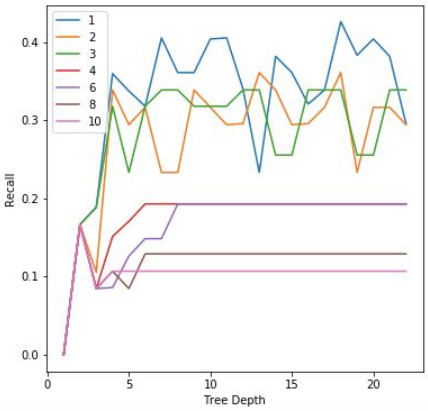



**Dataset 4: Combined Data**

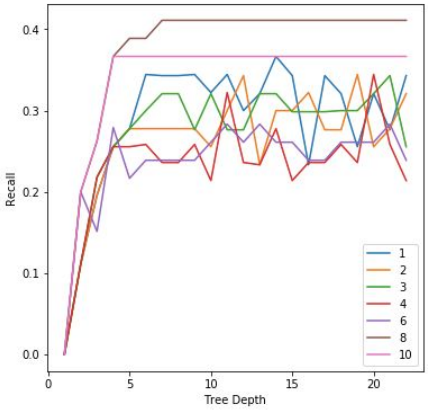



## Decision Tree

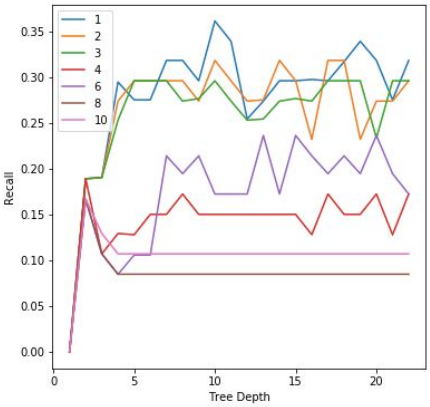
**Dataset 1: Ranked Data**



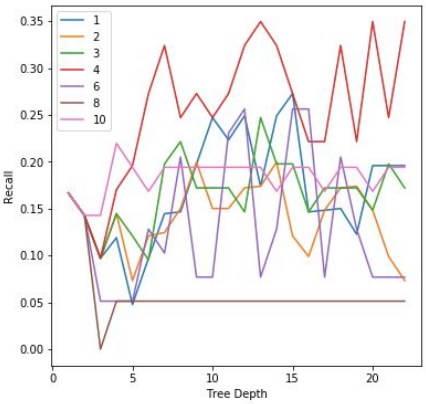

**Dataset 1: Raw Data**



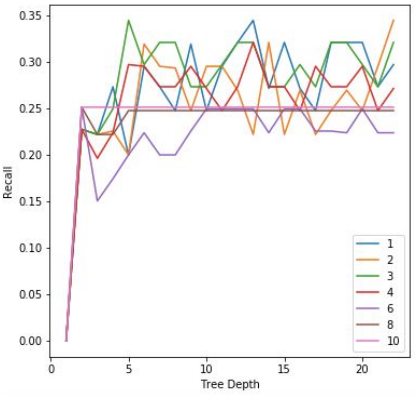

**Dataset 1: Combined Data**



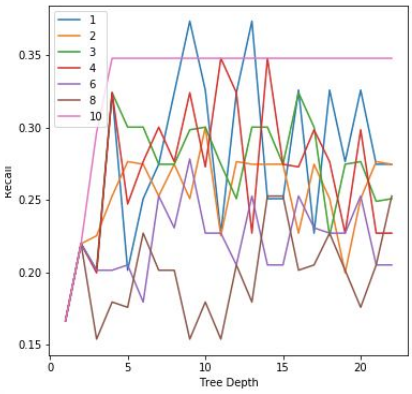

**Dataset 2: Ranked Data**



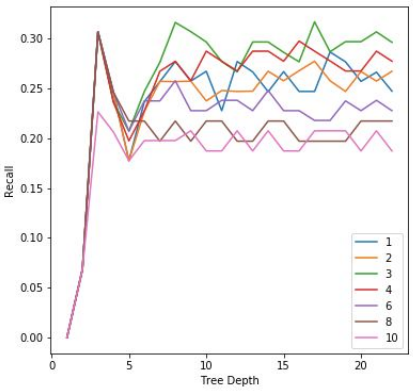

**Dataset 2: Raw Data**



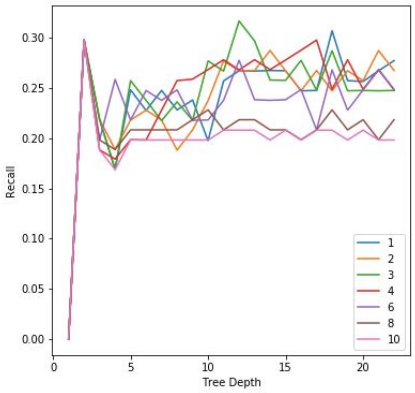

**Dataset 2: Combined Data**



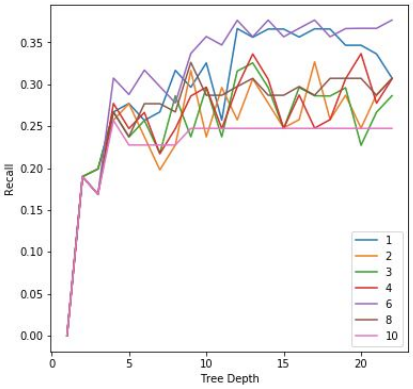
**Dataset 3: Ranked Data**



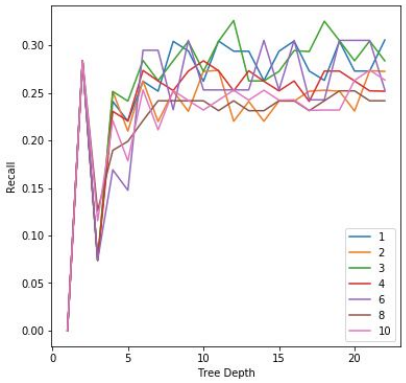

**Dataset 3: Raw Data**



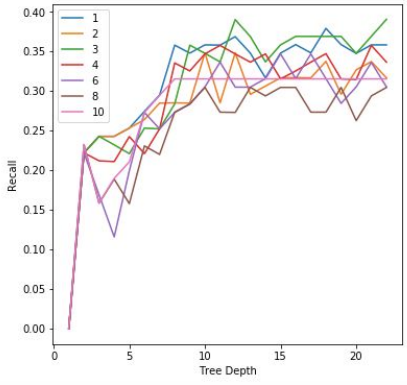

**Dataset 3: Combined Data**



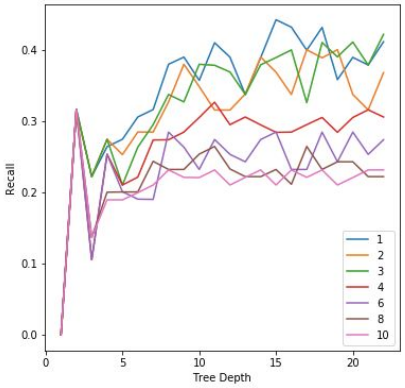

**Dataset 4: Ranked Data**




**Dataset 4: Raw Data**

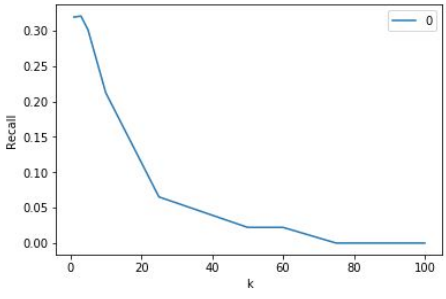



**Dataset 4: Combined Data**

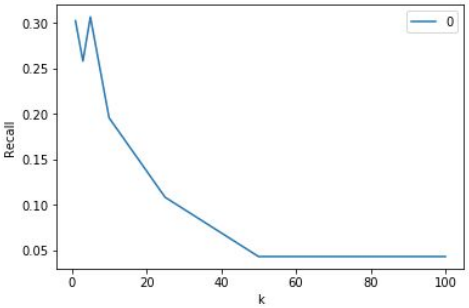



## k-NN

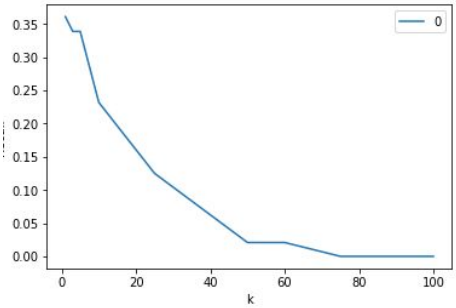
**Dataset 1: Ranked Data**



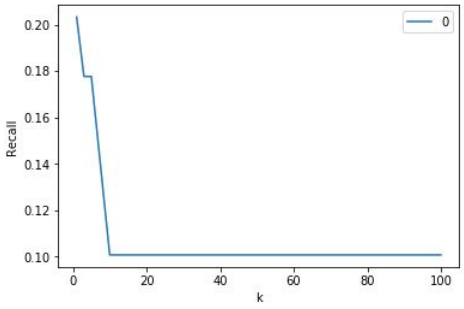

**Dataset 1: Raw Data**



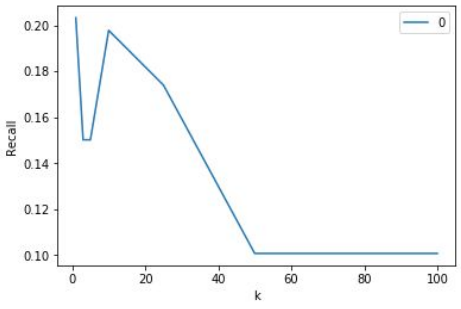

**Dataset 1: Combined Data**



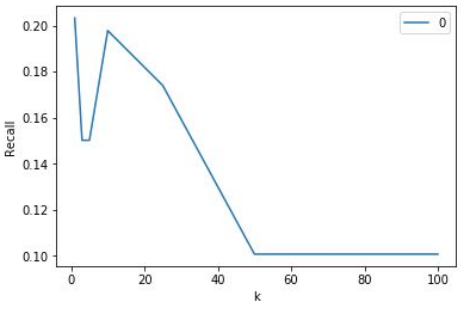
**Dataset 2: Ranked Data**



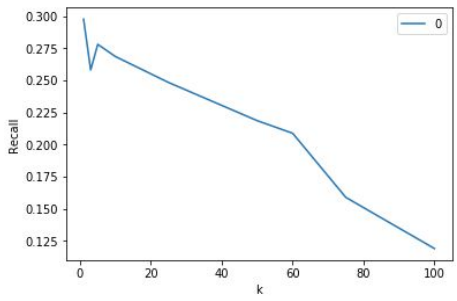

**Dataset 2: Raw Data**



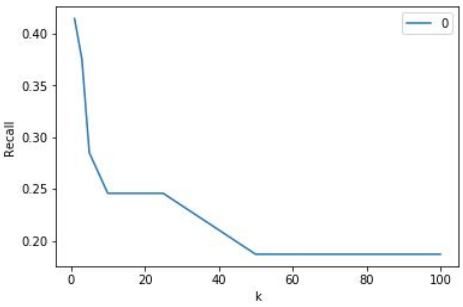

**Dataset 2: Combined Data**



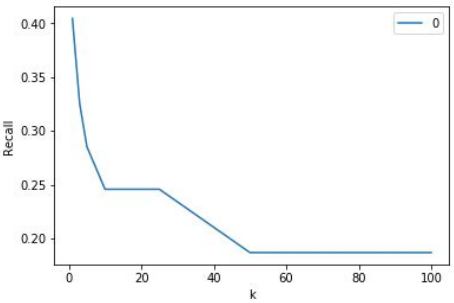

**Dataset 3: Ranked Data**



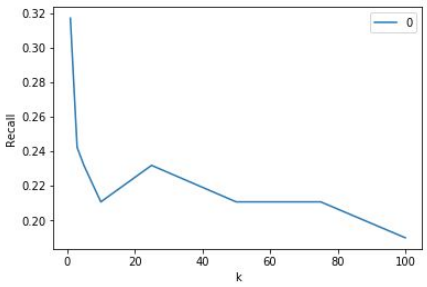

**Dataset 3: Raw Data**



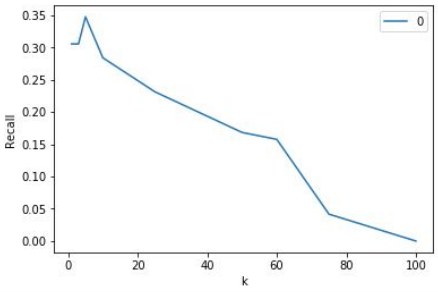

**Dataset 3: Combined Data**



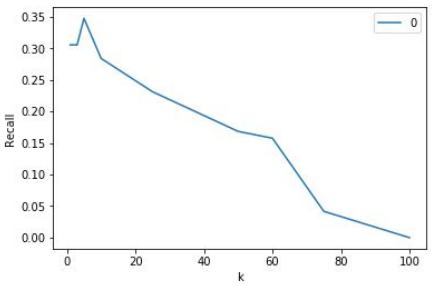

**Dataset 4: Ranked Data**




**Dataset 4: Raw Data**




**Dataset 4: Combined Data**




For more specific recall scores and features for each dataset variant, please see our attached iPython Notebook.

**Contribution by Each Teammate**

Jesse Gan: Collecting & Cleaning Data, k-NN Model, Deployment & Risks

Nisarg Patel: Decision Tree, Logistic Regression, Model Visualizations

Jason Huang: Business Understanding

Alan Ma: Model Evaluation

**Final Submission Folder Notes**

- The python files and jupyter notebooks contain pathnames that will need to change based on where the correct files are stored on the local machine.
- There are 2 python files: *Data_file_make.py* and *scraper_nba.com.py* used in the making of the final datasets. These 2 files create and use various csv files stored on the *csv_files* folder to create the 4 final data sets
- The *tree.dot* file can be used to visualize the results of the decision tree model.