Predicting NBA Draft Picks from the NCAA

Heather Monteson, Josh Truong, Alisha Gurule, Steve Naumov

Intro

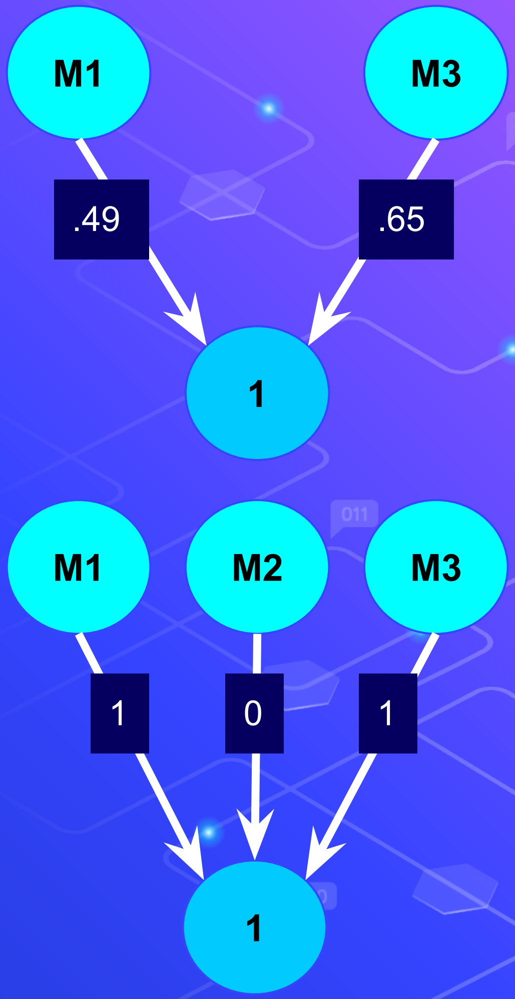
- The National Basketball Association (NBA) draft
  - 60 players drafted each year
  - Majority drafted from National Collegiate Athletic Association (NCAA)
  - Nearly 20,000 players in the NCAA
  - Diverse skills, and performance metrics
  - Challenging to understand competition
  - Difficult to target and recruit

Goal

- Using NCAA player stats and data mining techniques, predict NBA draft picks
- Determine features that could help NCAA players identify how they compare to their peers, and pinpoint areas of opportunity
- Help recruiters identify potential NBA players using data, or understand what stats to look out for

How

- Apply various preprocessing techniques to get our data ready
- Use an ensemble method for making our predictions/classifications
  - Testing 2 similar models with different methods for training
    - K-fold cross validation
    - Semi balancing data



The Data

- Players from the 2009-2021 seasons
- Only 60 players get drafted into the NBA, so the data highly skewed towards undrafted players
- Requires preprocessing and normalization due to the data not being clean and with some incomplete features

Preprocessing: Data Cleaning

- Remove any corrupt/inconsistent/unnecessary data to enhance data quality and improve model performance.
  - Dropping any irrelevant or inconsistent columns
    - Unnamed: 65, ht, num, type, pid, yr,
  - Renaming Columns
    - Unnamed: 64 was renamed to role position.
- Remove columns with >60% missing data (NaN values) to reduce skewness in analysis
  - Identify columns with nan values and calculate the percentage of NaNs in each.
    - Rec Rank NaN % = 69.75%
    - Pick NaN % = 97.65%
  - Drop rec rank column because high percentage of NaN values.
  - Keep pick because this is crucial in identifying which players were drafted.

Preprocessing: High Correlation and Merging

- Remove highly correlated features with correlation above 95%
  - Uses a correlation matrix
  - This will help to reduce data complexity, enhance model accuracy, and improve the efficiency of data analysis.
- Merge each individual player into 1 row/player
  - Identify Unique Player
    - Each player was grouped by their name and the team they played for. This will help to accurately aggregate data for players who share the same name but play for different teams. To ensure accuracy all player names were converted to lowercase
  - Weighted Averages
    - For each player, stats across different seasons were aggregated using a weighted average. This shows a players performance over time and gives more weight to later seasons.
  - Pick
    - This indicates whether a player was drafted. If a player was drafted in any season this was marked as 1

Preprocessing: Improving Runtime and Avoiding Data Leakage

- Remove all players who still had >15% missing data
  - Improve data quality
  - Computational efficiency running KNN to later fill in our values
- Split into training/testing sets
  - Avoid data overlap and leakage
  - Reduce the risk of overfitting in feature selection which can impact the model performance predictions with unseen data (overly optimistic)

Preprocessing: Impute Missing Values with KNN

- To handle missing values within our dataset, we choose K Nearest-Neighbor to impute missing values
- KNN imputation estimates missing values based on similar neighboring data points and multiple features, leveraging data relationships and patterns for accurate imputation.
- Distance metric used is sklearn default: ‘nan_euclidean’
- Since KNN are susceptible to local outliers, we also merged/smoothed our players before imputing.

Preprocessing: Scaling the Data

- Sklearn StandardScaler
    - Standardize features by removing the mean and scaling to unit variance.
- Applicable to only numeric values that are not categorical
- Column units vary with some columns including discrete values like Games Played, to percentages like effective field goal or turnover, etc…

Preprocessing: Feature Selection with Random Forest

- Predictive Power: Random Forests serve as both classifiers and data mining tools, extracting feature importance.
- High Accuracy and Generalization: Recognized for high accuracy, Random Forests demonstrate robust generalization to unseen data.
- Overfitting Mitigation: They help in reduce overfitting, improving generalization by minimizing the number of influential features. In our case, columns were reduced from 50 to 14.
- Interpretability: Random Forests, being decision trees, offer easy interpretation through the assignment of variable importance in decision-making.

Preprocessing: Feature Selection with Random Forest

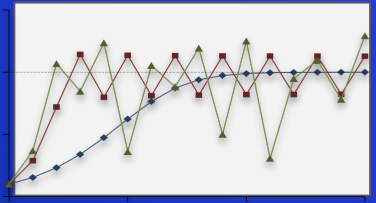
Games Played (GP)	Two-Point Attempts (twoPA)	Offensive Rebound Rate (porpag)	Adjusted Offensive Efficiency (adjoe)
Sum of Mid-Range Shots Made and Mid-Range Shots Missed (midmade+midmiss)	Sum of Dunks Made and Dunks Missed (dunksmiss+dunksmade)	Offensive Rating (adrtg)	Defensive Rebound Rate (dporpag)
Stops	Box Plus-Minus (bpm)	Offensive Box Plus-Minus (obpm)	Game Box Plus-Minus (gbpm)
Offensive Game Box Plus-Minus (ogbpm)	Defensive Game Box Plus-Minus (dgbpm)		

Model 1 with K-fold

- Use the Ensemble Method

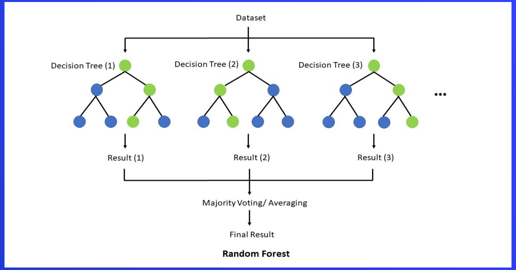
Logistic Regression

- Looks at the players' past performance to make predictions



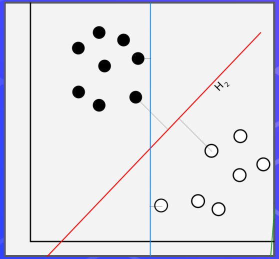
Random Forest

- Trees look at different aspects of a players' performance and combines findings to make predictions



SVM

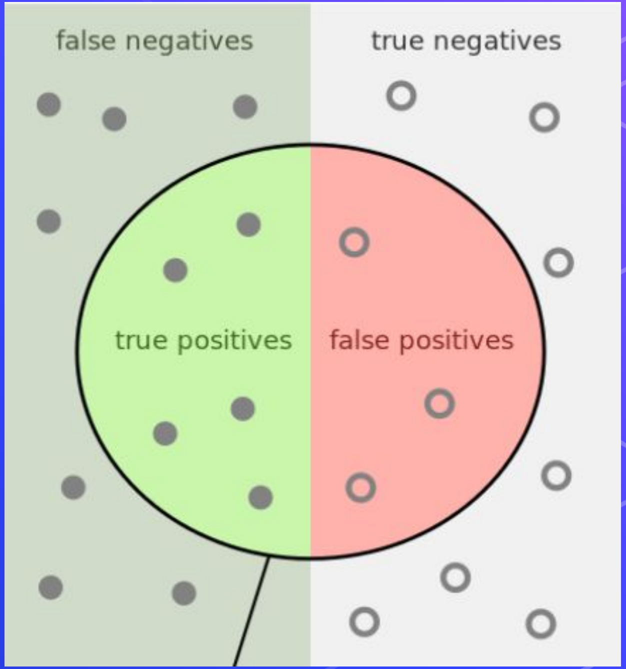
- Finds the best boundary to separate players who were drafted from those who were not drafted.



- Benefits
  - Combining different models leads to improved accuracy and better predictions.
- Cons
  - More time may be required to train multiple models.

Performance Metrics Used

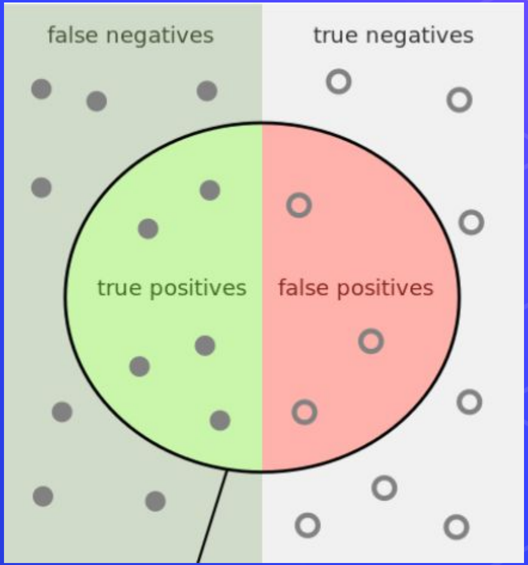
- Unbalanced in favor of players not being drafted (‘pick’==0), thus, our models accuracy was noted for some tests but determined to be a bad representation of overall performance as accuracy is influenced by the class with a larger number of instances
- Instead, used precision, recall and f1-score

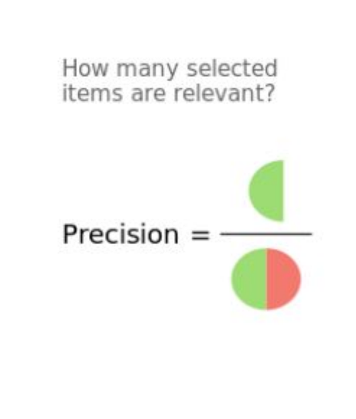


https://towardsdatascience.com/the-case-of-precision-v-recall-1d02fe0ac40f

Precision

- Precision: The accuracy of a models positive predictions
- When model predicts 'pick'==1, how likely is it that that the player was drafted
- **TP/(TP+FP)**

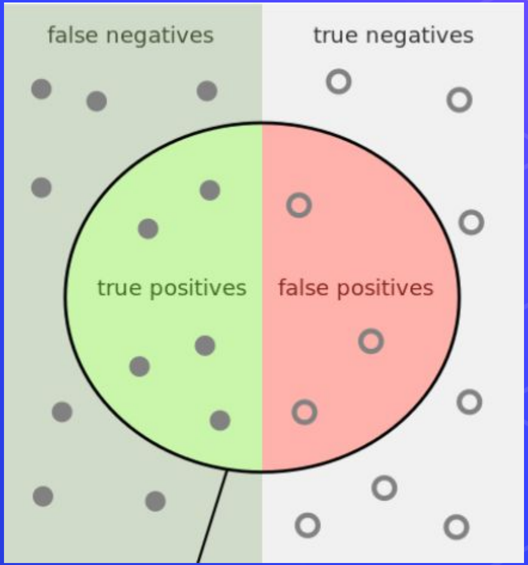


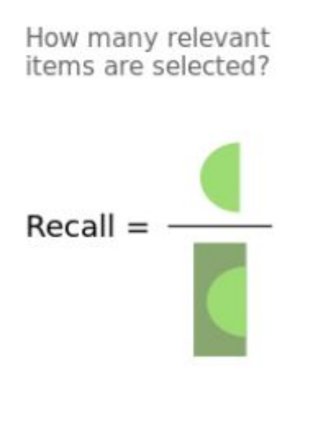


https://towardsdatascience.com/the-case-of-precision-v-recall-1d02fe0ac40f

Recall

- Recall: Models ability to identify the positive instances
- If a player is drafted, how likely is it that the model will classify the player as ‘pick’==1
- **TP/(TP+FN)**





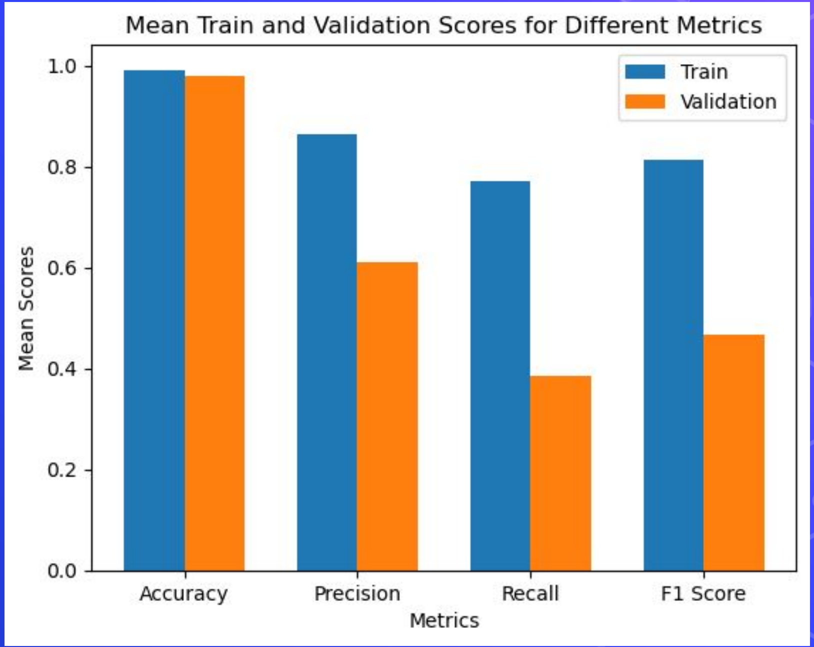
https://towardsdatascience.com/the-case-of-precision-v-recall-1d02fe0ac40f

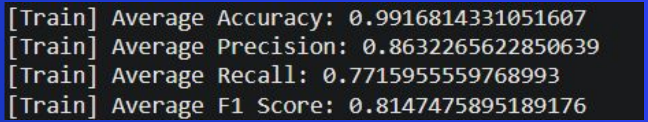
F1-Score

- F-1 Score: The harmonic mean of precision and recall
- Trade off with increasing precision or recall and decrease the other. F1-scores help to represent both values, more representative of actual model performance
- **2 * (Precision * Recall) / (Precision + Recall)**

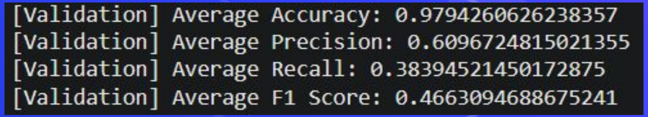
K-Fold Model and Training

- Tested different k values
- Tested various feature combinations
- Estimated performance by taking the mean of each training epoch





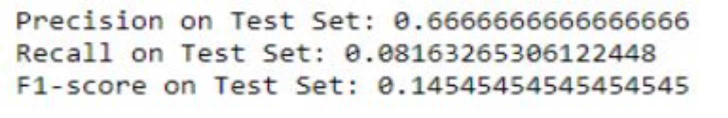
```
[Train] Average Accuracy: 0.9916814331051607
[Train] Average Precision: 0.8632265622850639
[Train] Average Recall: 0.7715955559768993
[Train] Average F1 Score: 0.8147475895189176
```



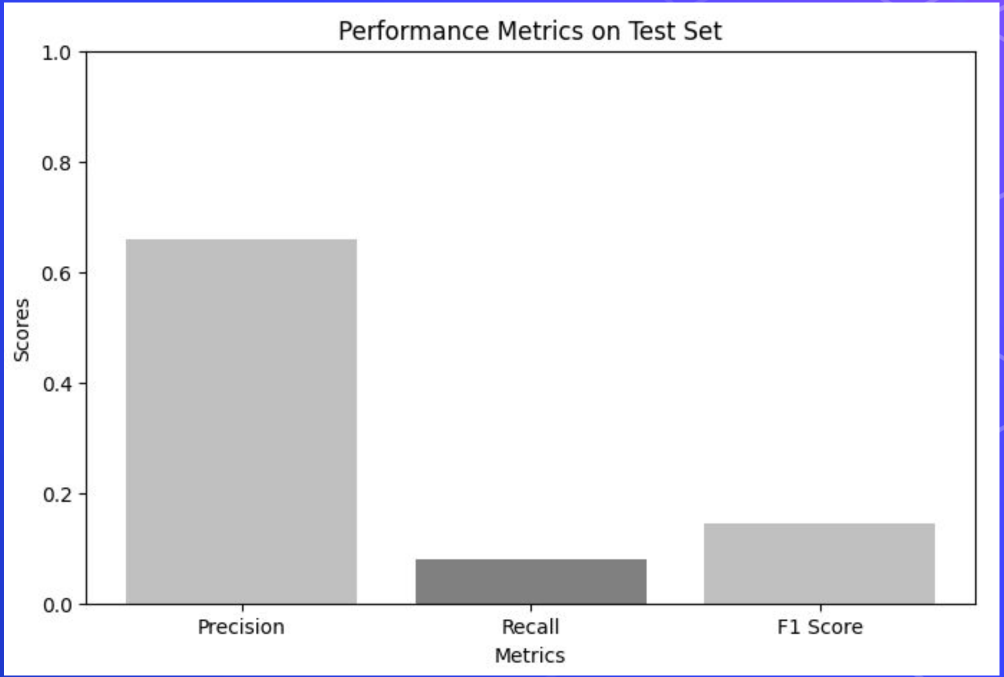
```
[Validation] Average Accuracy: 0.9794260626238357
[Validation] Average Precision: 0.6096724815021355
[Validation] Average Recall: 0.38394521450172875
[Validation] Average F1 Score: 0.4663094688675241
```

5-fold and Model Performance

- Initial performance:
  - Ran with 5-folds
  - Features used from initial feature selection

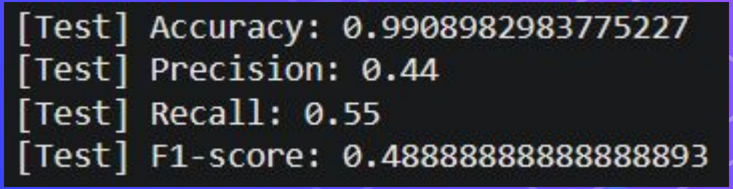


```
Precision on Test Set: 0.6666666666666666
Recall on Test Set: 0.08163265306122448
F1-score on Test Set: 0.14545454545454545
```

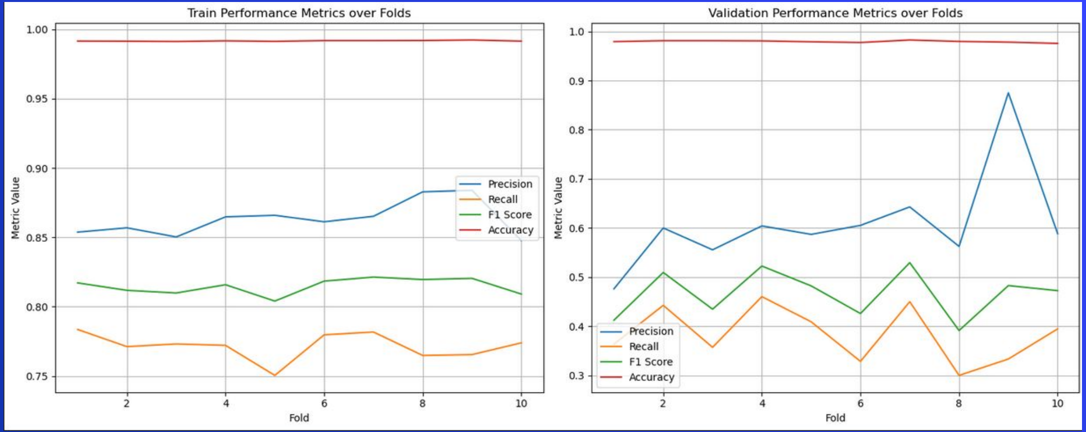


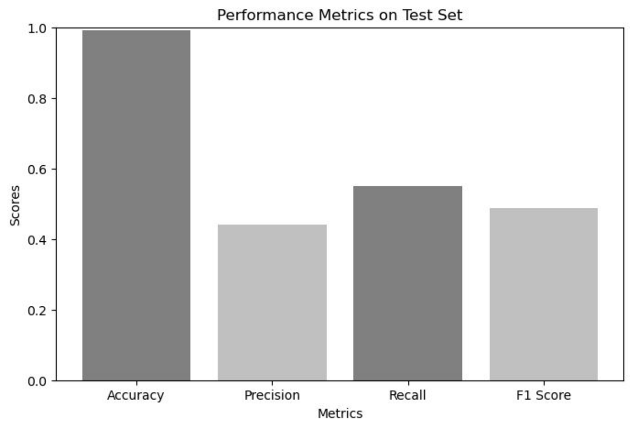
kFolds and Model Performance

- Final Model ran for 10 folds



```
[Test] Accuracy: 0.9908982983775227
[Test] Precision: 0.44
[Test] Recall: 0.55
[Test] F1-score: 0.48888888888888893
```





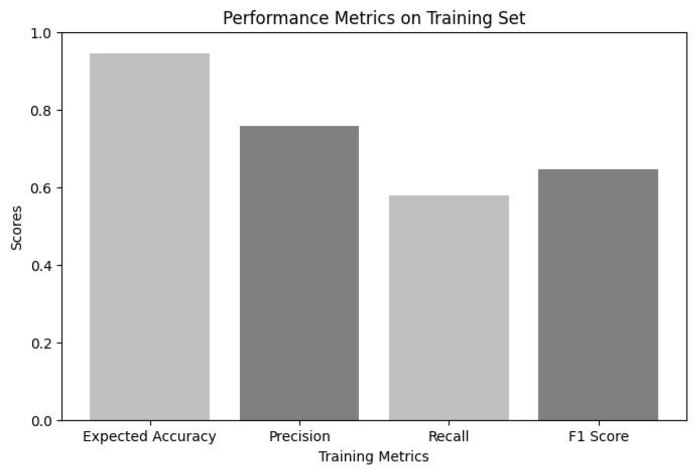
features=['porpag','adjoe','adrtg', 'bpm', 'obpm', 'dbpm', 'ogbpm', 'dgbpm', 'pts']

Model 2 with Balanced Data

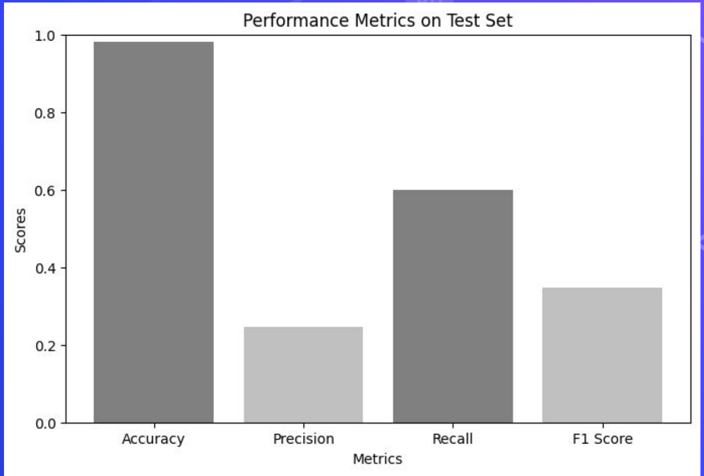
- To undersample the majority class so that the model doesn't only know how to predict the majority class
- Reduce the majority to minority class data from around 35:1 to 10:1 by randomly sampling the training set for both types of data
- Main goal: to increase the F1 score as much as possible
- Try to prevent overfitting by sampling without replacement

Balanced Data and Performance Predictions

- Higher recall on test set
- Higher Recall and F1 score on the training set
- Still a discrepancy between the F1 score on the training and test set, indicating that it could be optimized further



Performance Metrics on Training Set



Performance Metrics on Test Set

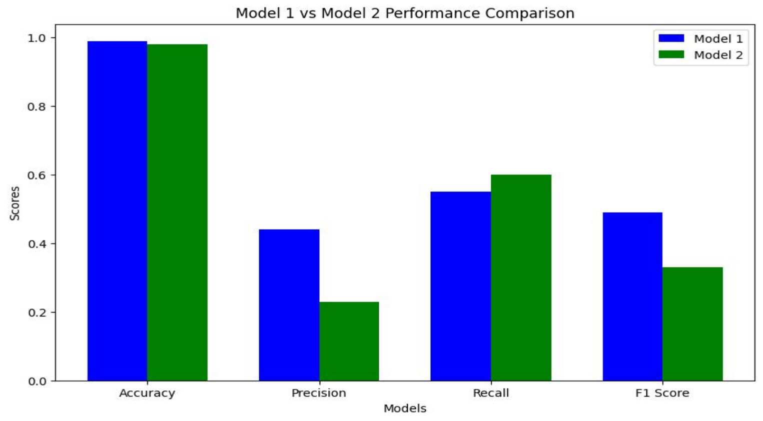
Model 1 vs Model 2 Result Comparisons

Model 1 with K-fold

- Splits data into subsets, in each iteration one subset is used for validation and the rest for training.
- Model 1 outperforms Model 2 in precision.
- Model 1 also has a higher F1 score, which balances precision and recall.
- Overall, Model 1 is more accurate.

Model 2 with Balanced Data

- Focuses on balancing the data by getting the number of drafted players and not drafted players closer equal.
- Model 2 outperforms Model 1 in recall.



Final Conclusion

- Better performance seen with the model and k-fold cross validation
- Made significant improvements from initial performance, however precision, recall and f1 still remained relatively low for both methods
  - Unbalanced data
  - Not all players with good stats apply to the draft
- Best predictions were made with aggregate features. All/many skills are considered and important, may be harder for a player to pinpoint one skill to improve upon aside from their average points made per game.
- Could provide a recruiter with insight into what data to focus on when determining players to potentially pick for the draft

Citations

Han, J., Kamber, M., &amp; Pei, J. (2012). Data Mining: Concepts and Techniques. Elsevier Science.

Alterman, R. (2020, June 13). The case of precision v. recall. Medium. https://towardsdatascience.com/the-case-of-precision-v-recall-1d02fe0ac40f