Predicting NBA Draft Picks from NCAA Players

Predicting NBA Draft Picks Using Two Ensemble Voting Models Training One with K-Fold Cross Validation and One with Balanced Data

Heather Monteson
University of Colorado
Boulder, Colorado USA
hemo7757@colorado.edu

Joshua Truong
University of Colorado
Boulder, Colorado USA
jotr4800@colorado.edu

Alisha Gurule
University of Colorado
Boulder, Colorado USA
algu5256@colorado.edu

Steve Naumov
University of Colorado
Boulder, Colorado USA
stna7972@colorado.edu

ABSTRACT

In the context of the highly competitive National Basketball Association (NBA) draft, our project employs advanced data mining techniques to predict which National Collegiate Athletic Association (NCAA) players will be drafted. We aim to assist players in evaluating their draft potential and identifying areas for improvement, while aiding NBA teams in their decision-making process.

Using ensemble models that were constructed using Logistic Regression Models (LRM), Random Forest Classifiers (RFC), and Support Vector Machines (SVM), we analyzed NCAA player data from 2009 to 2021. Our goal was to determine the key features impacting a player's draft selection. In an attempt to enhance model performance, we explored two different training methods: k-fold cross-validation, and splitting the training data into semi-balanced sets. We then employed data preprocessing steps, including the use of a correlation matrix to remove redundant data, k-nearest neighbors (KNN) for finding missing values, and an RFC for feature selection, to prepare our data for the models.

After preprocessing, training, and testing, our k-fold cross-validation approach yielded the best predictive performance over the two methods. The model achieved a precision of 44%, recall of 55%, and an f1-score of 49% with 10 folds, incorporating eight aggregate features, and one non-aggregate feature (average points per game). Although a semi-balanced training approach showed a recall of 60%, the lower precision (23%) resulted in a lower f1-score (33%). Thus, the k-fold cross-validation method was deemed superior.

Despite efforts to increase the f1-score, challenges persisted due, we speculate from data exploration, to some well-performing players not applying for the draft, and the data's inherent skewness in favor of not drafted players. Additionally, the model excelled with aggregate features, which may make it challenging for players to pinpoint specific areas for improvement as the features incorporated a vast range of various metrics. However, these findings could aid recruiters in identifying key metrics when evaluating potential players.

1 INTRODUCTION

Each year, the NBA carefully selects 60 eligible players from a pool of hundreds of applicants to join the league. This selection process unfolds through two distinct draft events where each of the 30 NBA teams selects two players. Among those who enter the draft is a cohort of NCAA players aspiring to make their mark in the competitive world of professional basketball. With such a substantial pool of talent, identifying college standouts becomes a multifaceted challenge that prompts exploration into advanced methodologies for understanding player performance, and provides an opportunity to apply data mining techniques to enhance the precision of player evaluations and draft decisions.

College and professional sports alike have witnessed a surge in the use of sports data analysis and data mining. This surge has significantly influenced how team owners, managers, and coaches approach improving their team’s performance. It also provides players with an opportunity to understand their overall performance, what their strengths

are, and what their areas of opportunity might be in comparison to their peers.

Our project aims to combine various data mining techniques to predict prospective NBA draft picks from NCAA players. Making such classifications has the potential to support NBA teams in their decision-making process for draft picks as they can easily narrow down potential players based on their past season performance, or even see what players may be outside of the typical draft selection, and thus give the team better odds of being the first to select that player in the draft.

Beyond supporting draft decisions in the NBA, our project could also potentially provide valuable insights for players aspiring to enter the draft as it can give them the ability to weigh their own performance metrics against others who have been drafted in the past, and thus help to identify areas in that players performance that, if improved upon, may increase their chances of being drafted.

Thus, by combining various data mining techniques for this project, we aim to assist teams in identifying promising talents, potentially reshaping conventional draft selections, and offering valuable insights for aspiring players. Through this endeavor, we strive to contribute to the evolving landscape of talent evaluation through data mining in basketball.

2 **LITERATURE REVIEW**

2.1 A Review of Data Mining Techniques for Result Prediction in Sports [5]

This paper was published in the November 2013 issue of the Advances in Computer Science: an International Journal (ACSIJ) and reviews various data mining systems for sports result predictions. It was authored by Maral Haghighat, Hamid Rastegari, and Nasim Nourafza.

Advancements in technology have led to an increase in data about players, teams, games, and entire sporting seasons. Traditionally, this data was only understood and utilized by sporting experts like coaches and team managers. However, now data mining techniques are enabling more sporting organizations to assist them in making predictions about game outcomes, possible injuries, potential draft picks, and in planning game strategies. This particular study examines previous research on how data mining is used to predict sports game results.

The section on feature selection from the article is relevant to our project because it highlights the importance of feature selection in creating accurate predictive models. Feature selection is a good section from this paper to focus on because it's an important step in identifying which player statistics are most predictive for our purposes.

The article discusses how different studies in data mining and sports have chosen specific features to improve model accuracy and speed. It mentions different methods researchers have used, such as manual selections made by expert opinions and algorithm-based selections. The discussion on the importance of feature selection emphasizes how critical it is to carefully select the right features for predictive models. .

The article also discusses different classification techniques similar to those used in our project. Logistic regression is ideal for predicting binary outcomes, such as whether an NCAA player will be drafted into the NBA or not. The article’s explanation of logistic regression’s use in sports result predictions mirrors our use of it in predicting draft picks.

SVM was also confirmed to be an effective classification tool for handling linear and non-linear data, praised for its ability to create complex decision boundaries. Our project also has complex player performance data that needs to be evaluated to classify players.

Decision trees were discussed as another powerful classification tool. This article also covers techniques not implemented in our project, like the use of Artificial Neural Networks or (ANN)’s.

Whereas, this article might rely on one specific model, our project uses a combination of classification techniques, in an ensemble approach to obtain the best results.

Focusing on how the article describes these methods is important because these all affect the results that are rendered by the predictive model. Some tools might obtain a higher accuracy than others. Learning how they work and how they are interrelated could help us get more accurate results from our own model. It helps to provide a better understanding for how these methods can be applied to sports data, particularly in predicting outcomes.

2.2 Jumping in the Pool: What Determines Which Players the NBA Considers in the Draft? [7]

The paper delves into the intricate process of NBA player selection from the vast pool of college talents, aiming to decipher the criteria NBA decision-makers employ to identify potential draft picks. The study confirms that, consistent with prior research, points scored remain a dominant factor in evaluating player talent. Additionally, the research reveals challenges in distinguishing individual player performance from the success of their college teams. The analysis emphasizes the significance of understanding the dynamics that lead to a player's inclusion in the draft, crucial for predicting career trajectories and optimizing player selection strategies.

The paper begins by highlighting the complexity faced by NBA decision-makers in selecting players, exemplified by instances where talented players like Tyler Johnson may be overlooked. The authors stress the specific rules governing the NBA draft, including eligibility criteria for non-international players and the two-round selection process. The study departs from previous research focusing on factors impacting a player's draft position and instead investigates the criteria determining a player's inclusion in the draft pool.

The paper concludes by summarizing the key findings, reiterating the dominance of point-scoring in player evaluation, and underscoring the importance of team success in predicting draft inclusion. The research provides valuable insights for NBA decision-makers, emphasizing the need for a nuanced understanding of player evaluation beyond traditional metrics. It sets the stage for future studies on refining AI models to predict NBA draft selections based on a holistic understanding of player performance and team dynamics.

This highlights some of the metrics to take into consideration, while also considering that there are certain key factors that may be difficult to capture. Such as a player’s individual performance may not be correlated with the player as part of a particular team. Some additional things to consider is a particular team’s coaching staff performance not being captured by the data, but still influencing the team and player performance.

2.3 The Determinants of Draft Position for NBA Prospects [6]

The paper explores the determinants of NBA draft positions, primarily focusing on understanding the factors influencing a player's draft position.

Scoring proficiency, particularly points scoring per 40 minutes, emerges as a consistently influential factor in determining draft selection. This finding emphasizes the pivotal role of offense in shaping draft outcomes.

Shooting efficiency plays a crucial role, with two-point shooting percentage (2PT%) standing out as a significant determinant. Surprisingly, three-point shooting percentage (3PT%) does not correlate significantly with draft position.

Team performance is a key factor, as players from winning teams tend to secure better draft outcomes. A ten percentage point increase in a college team's winning percentage is associated with a two-slot improvement in the draft, underscoring the impact of collective success on individual player evaluations.

Coaching experience influences draft selection. Each additional year of head-coaching experience is associated with a small but significant improvement in draft position, highlighting the influence of a coach's tenure on a player's perceived value.

Player attributes, specifically height, significantly influence draft position, even when standardized by position. Taller players tend to be drafted earlier, emphasizing the importance of height in player selection. However, body mass index (BMI) is not found to be a significant factor in draft outcomes.

The research extends its analysis by categorizing players into early entrants and four-year players, revealing significant differences in determinants between these two groups. This categorization emphasizes the evolving dynamics based on the number of collegiate years.

Position-specific regressions for point guards, shooting guards, small forwards, power forwards, and centers highlight a surprising degree of disparity in the factors influencing draft positions. While scoring proficiency remains universally important, the study reveals nuanced differences based on player positions.

In conclusion, the study contributes valuable insights into the complex dynamics of the NBA draft, emphasizing the importance of scoring proficiency, team success, coaching experience, and player attributes.

3 **METHOD**

In order to prepare our data for a model to make NBA draft predictions, we first completed a series of preprocessing steps. The preprocessing included removing inconsistent data and redundancies, combining player data to make the data more clear and simplified, KNN to replace null values, splitting our data into testing and training, scaling the data to better fit our models and finally determining what features may be the best for predicting if a player was drafted using a random forest classifier (RFC).

Following preprocessing, we then created two similar ensemble voting machine learning (ML) models as we wished to test two different methods for training to determine the best means for predicting NBA draft picks. The ensemble models combine three different ML models: LRM, SVM, and a RFC to then make the predictions. The first model was trained and evaluated using k-fold cross-validation with 10 folds, and the second used semi-balanced training data sets.

3.1 Preprocessing

The original non-cleaned data set came from Kaggle dataset containing several different files, but we focused on 2009-2021 NCAA player stats as they would be the most relevant for our goals. The players who did not play during 2021 were selected as the training set, and players who did play during 2021 were selected as the test set [1].

Initially, the dataset was not clean, and required multiple preprocessing steps to make it usable. There were several columns in the data that were irrelevant to gauging how a player might perform and were inconsistent. An unnamed column was removed as the data was inconsistent. The player height was removed due to the way in which the height was recorded being inconsistent. Some columns only had 1 unique value, so they were removed due to being not interesting, and some categorical data was removed such as the player's school year. Columns with over 60% of missing values were then removed due to there being too much missing data to eventually use KNN imputation to determine the most likely values with reasonably good levels of accuracy.

Upon refining our dataset to a relatively cleaner state, we prioritized investigating the presence of highly correlated features. This step holds significant importance in our preprocessing workflow, as the identification and removal of such features can enhance model performance by reducing the computational burden associated with redundant inputs. The impact of highly correlated features is particularly noteworthy for our chosen supervised learning models—logistic regression and random forest. In linear models like logistic regression, employed within our voting classifier, the existence of highly correlated features can lead to multicollinearity, introducing variability in solutions and causing numeric instability. Similarly, within a random forest model, highly correlated features have the potential to obscure interactions between different features.

Furthermore, removing highly correlated data can be viewed as an opportunity to simplify our model by reducing the number of inputted features. In our preprocessing step, we determine whether or not a feature is highly correlated with a correlation matrix, for visual purposes the matrix was also projected on a heatmap to understand why a feature might be highly correlated. Then when the values for each feature is determined a threshold of 0.95 was used to remove any features above it. However, we've exercised caution to avoid eliminating all pairs of highly correlated features. Instead, we selectively removed only one feature from each correlated pair to mitigate potential issues.

Following the removal of redundant data, we then sought to simplify our data. The initial dataset contained multiple rows for players who played for more than one year. Because of this, we faced potential problems with increased dimensionality and redundancy between rows. Allowing this structure in the data would also make running our model with the dataset computationally expensive, and more prone to overfitting. Because of this, any rows with duplicate player names were combined into one row of data.

In order to achieve this, data was grouped by player_name and then by team before applying a weighted average. The string player_name was updated to be all lowercase in order to avoid any mistakes in comparisons, ensuring that all records of the same player were correctly identified and aggregated.

Next, the data was grouped by the player names and their respective teams. This helped to distinguish between players who might share the same name but played for different teams. A weighted average of all the numeric statistics was calculated to reflect a player's overall performance, and put more emphasis on their most recent seasons as players, on average improve over their NCAA career.

In the weighted average function, the ‘pick’ column was used to identify any players who got drafted in the NBA. In cases where a player was drafted in any season, their final record was marked with a 1 to reflect true. By performing these weighted averages, each player was represented by a single row of data that summarized their entire NCAA basketball careers for a particular team. This was essential in helping to provide a more accurate, and concise representation of player performance.

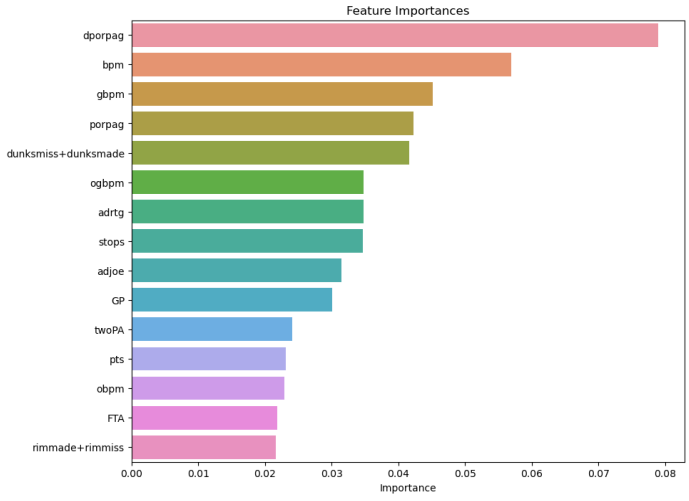
Since machine learning models typically require numerical input, it was crucial that, following the merging of players, we encoded properly to reflect the categorical data. To achieve this, Scikit-Learn’s LabelEncoder was used, and we manually inspected the data to ensure the same number of unique values remained for each category in the dataset.

Once our categorical data was encoded, we split the data into training and testing sets. This was done during preprocessing to avoid data leakage, and overlap while filling in missing values, and performing feature selection. It was important as well to ensure that we were using completely unseen data for our testing set so we could accurately evaluate the performance of our model. To split the data we simply selected all players who were part of the NCAA during the 2021 season and removed them from the data to create our testing set. The remaining players who did not play during the 2021 season were then used as our training data.

Since our dataset spans from 2009 to 2021 and has over 50 features, there are some missing values. To handle these gaps, we use a KNN algorithm to fill in the missing data. We opt for KNN imputation because it has the advantage of supporting multivariate imputations. This means the algorithm can take into account multiple variables to better capture relationships and improve imputation accuracy. After tweaking the parameter for k, we found that using the average value of 11 neighbors enhances the accuracy of our predictive model.

Data normalization is a critical step in our preprocessing workflow, addressing the impact of varying feature scales on model performance. Differences in feature scales can erroneously prioritize features with larger scales, affecting models like LRM and SVM. LRMs are sensitive to input feature values, and experience convergence challenges with large scale differences. SVM, particularly with an RBF kernel, sees its distance metric influenced by feature scales. To mitigate these issues as we use both SVM and LRM in our model, we employed Scikit-learn's StandardScaler, which is a Z-score normalization, to standardize features, centering data around 0 and scaling standard deviation to 1.

In our data preprocessing and exploration, managing over 50 features posed challenges in terms of computation and code redundancy. To address this, we leveraged the Random Forest algorithm for feature selection. By utilizing RandomForestClassifier from scikit-learn, we calculated feature importance scores. We then retrieved the feature support and retained only those above a certain threshold. As a result the number of features used in our predictive models was significantly reduced from 50 to 14 features. This systematic approach streamlined our dataset, improved computational efficiency, and ensured we retain the most influential variables for subsequent analyses. While we did not use all of the selected features that were deemed of significant importance by the random forest, it did help narrow down the features, and gave the team an informed decision on what features to include in our model. In our basketball dataset, the key features identified for predicting player performance included 'GP', 'twoPA', 'porpag', 'adjoe', 'dunksmiss+dunksmade', 'adrtg', 'dporpag', 'stops', 'bpm', 'obpm', 'gbpm', 'ogbpm', 'dgbpm', and 'pts'. These metrics are aggregate values with the exception of ‘pts’, or points per game.



3.2 Classification Models

To make our draft prediction, we chose to use an ensemble voting method which allowed us to combine different ML techniques. The ensemble method was chosen because it has improved performance and robustness over using a

single model, and can allow for one model to overcome the weaknesses of another [4].

Expected performance, and evaluation for the models was initially captured by recording the accuracy, precision, recall and f1-score during testing and training respectively. However, it was later determined that accuracy was a poor representation for our model's performance given that the data was unbalanced, and accuracy is most influenced by the class with a larger number of instances. As such, precision, recall and f1-score were able to provide a better representation of our models performance. Precision worked to capture the accuracy of our model's positive predictions. That is, when the model predicted a player was drafted, how likely was it that that player was drafted:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}.$$

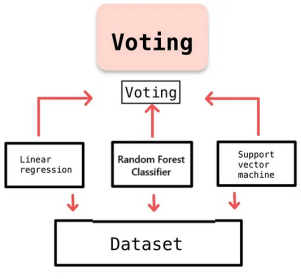
Recall then showed the models ability to identify the positive instances, or if a player was drafted, how likely was it that the model classified the player as such:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}.$$

Then, the f1-score represents the harmonic mean of precision and recall. This metric is useful as there is often a tradeoff when increasing either precision or recall such that raising one will decrease the other. The f1-score can then be found such that

$$f1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)}.$$

3.3 Ensemble Model



The first ensemble included three different ML models: LRM, SVM, and RFC. For each model, we took advantage of the scikit-learn library for Python which offers a simple, compact framework for creating ML models. Once built, the ensemble model used results from the three different models, and leveraged each of the predictions such that the winning classification for a player was that which was determined by at least two of the models in the ensemble.

The first model selected for use in the ensemble was a LRM, and, along with SVM and RFC, models a binary outcome such that the player is predicted to be picked for the draft ('pick'=1) or not picked ('pick'=0). LRM was selected to be part of the ensemble as it is typically used in binary predictions, and its simplicity and interpretability made it well-suited for the ensemble. The equation used for logistic regression in our model is given as

$$P(Pick = 1) = \frac{1}{1+e^{-(b_0+b_1X_1+b_2x_22+\ldots+b_nX_n)}}$$

where $P(pick = 1)$ is the probability of the dependent variable pick being 1, which in this case was the probability of the given player being drafted. The values $b_0, b_1, \ldots, b_n$ are coefficients which represent relationships between independent variables $X_1, X_2, \ldots, X_n$. The resulting value from the equation is then between 0 and 1 and is interpreted as the probability that the given player is drafted or not.

The LRM was trained with the other two models under supervised learning using labeled data where the expected binary outcome for each player was provided. The coefficients were then learned by the model during the training phase in the ML process as it seeks to minimize the difference between the predicted probabilities and actual outcomes.

The second model used was a RFC, which is, in itself, an ensemble learning method. RFC builds a number of decision trees, and determines a classification based on the majority prediction. In the scikit-learn library, RFC uses Gini impurity as the default criterion for splitting nodes in the decision trees. The Gini impurity is given as

$$Gini(D) = 1 - \sum_{i=1}^{m} {p_i}^2,$$

where D is a data partition or set of training tuples, m denotes the number of classes, and p_i is the probability that a tuple D belongs to class i [2, p.341]. The Gini impurity value determines the minimizing value, or the value with the

lowest likelihood of misclassifying a player as drafted or not, and that feature is then chosen as the feature to split the node on.

Once constructed, each tree in our RFC then classified a player as drafted ('pick'=1) or not ('pick'=0), and the RFC determined the final result based on the majority vote between all the decision trees. In our model, we found that the best performance occurred in both testing and training when using 53 decision trees to make the final classification.

The third model selected for the ensemble was SVM. SVM was chosen to be part of the model because of its effectiveness in handling linear, and non-linear data. It also performs well at identifying more complex patterns, modeling non-linear decision boundaries, and works well with classification tasks.

For binary classification tasks, SVM tries to maximize the margin between the closest points of two classes, in our case the classes being a player being drafted ('pick'=1) or not ('pick'=0). The model works by taking the data, or players, and separates them into the two classes using a decision boundary. The players closest to the decision boundary are the support vectors and help determine where the boundary will go. When a new player data point is added, the model can easily determine what class it belongs to based on where it is in terms of the boundary line.

The decision boundary for SVM is determined using a decision function which is built during training and helps to split the two classes across features. The decision function is then given as

$$f(X) = W \cdot X + b$$

where the vector W is the weight vector such that $W = \{w_1, w_2, ..., w_n\}$, and n is the number of features selected. The value of b is a scalar, or the bias, and X is the feature vector, or individual players feature data. Each players feature data instance (X) is then plugged into the decision function and the sign of the resulting value defines the predicting class such that if $f(x)$ is negative, the player is predicted to belong to the drafted class and if it is positive then they are predicted to be part of the non drafted class [2, p. 409-413].

In other words, it separates the classes in a way where there are the biggest margins on either side of the decision boundary. The points closest to the decision boundary are the support vectors and help determine where the boundary will go. When a new data point is added, the model can easily determine what class it belongs to based on where it is in terms of the boundary line.

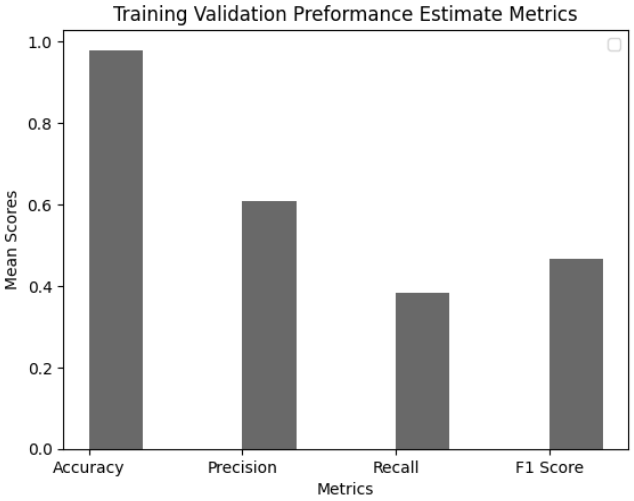
3.3.2 Training with K-Fold

Training for the first ensemble training was done using k-fold cross-validation. We opted to test this method of training for our first model, as our data was highly unbalanced in favor of undrafted players, and k-fold is well suited for unbalanced data.

For the k-fold cross-validation, we ran a series of training epochs testing the variability in performance based on the value k, as well as making some changes to the features being used. Ultimately, we found that the best performance occurred with 10 folds along with 9 features selected.

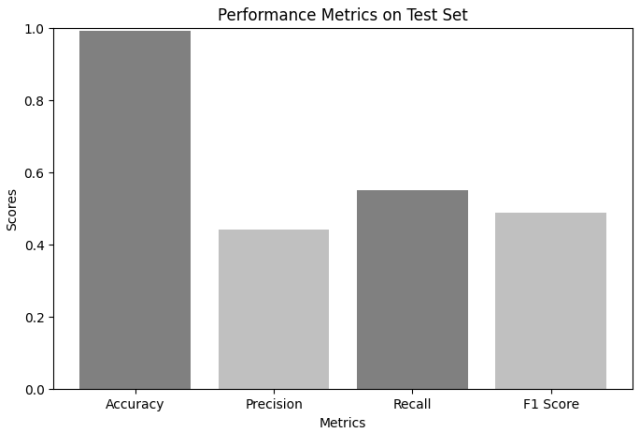
For the 10-fold cross-validation, we split our data into 10, nearly equal sized, non-overlapping subsets. Training was run 10 times, and each time a subset was used once and only once as the validation set. The 9 features selected were made up primarily of aggregate data already provided in the dataset, along with a player's average points per game. All of the features utilized were suggested by our feature selection from preprocess, but we removed some features that we felt were adding noise to the mode. The aggregate features used were 'porpag', which gives a value for the pointers over replacement per adjusted game; 'adjoe', a players adjusted offensive efficiency, 'adrtg' a players adjusted defensive rating; 'bpm', box plus/minus which represents a players quality and contribution to the team from play-by-play regression [3]; 'obpm' and 'dbpm, which are a players offensive and defensive box plus/minus rating; and 'ogbpm', 'dgbpm' which are two additional offensive and defensive aggregate data points that were not directly defined in the Kaggle dataset.

Expected performance, and evaluation of the model was then captured by recording the accuracy, precision, recall and f1-score for each pass through the training data, and then taking the mean for each. The final expected results from the training data were found to be approximately 97% for accuracy, 60% for precision, 38% for recall, and 47% for the f1-score:



3.3.3 K-Fold Testing Results

The training data used included only players who had played in the year 2021, and had been separated from all the players in the test data prior to being seen by the model in the preprocessing phase. In testing the model, our best performance evaluation found that accuracy was 99%, precision was 44%, recall was 55% and the f1-score was 48%:



As mentioned previously, while measured, the accuracy did not provide an accurate representation of our model due to our data being biased in favor of drafted players. However, the moderate precision suggests that, among the players predicted as drafted, a notable proportion were still labeled as false positives. The somewhat higher recall then suggests that the model captured a relatively moderate portion of actual positive instances. Because of the lower score for precision, the f1-score is also somewhat low indicating that the model has room for improvement, particularly in reducing false positives while capturing more actual positive instances.

3.3.4 Training with Semi-Balanced Data

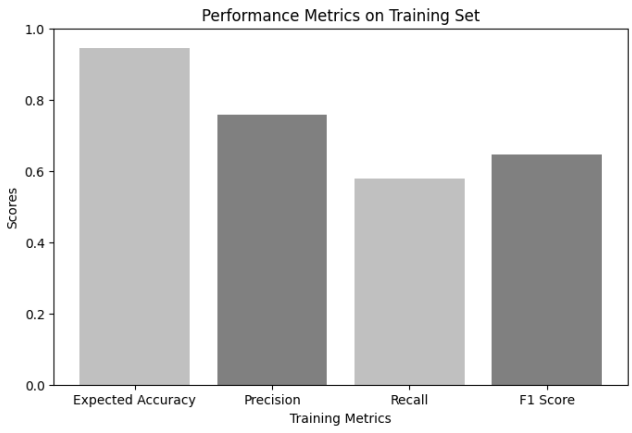
In the NBA, only 60 people are selected each draft, therefore there is a much larger class with undrafted players. To try to prevent the model from being extremely biased towards one class the majority class of undrafted players will be undersampled to try to provide a way to more precisely predict the player drafts instead of it just predicting the most common outcome with 99% accuracy.

To evaluate the model, the F1 score was used as it balances both precision and recall as it shows both correctly identifying positives and not missing the positives in the data. Accuracy was not considered as accuracy will be high for any imbalance data.

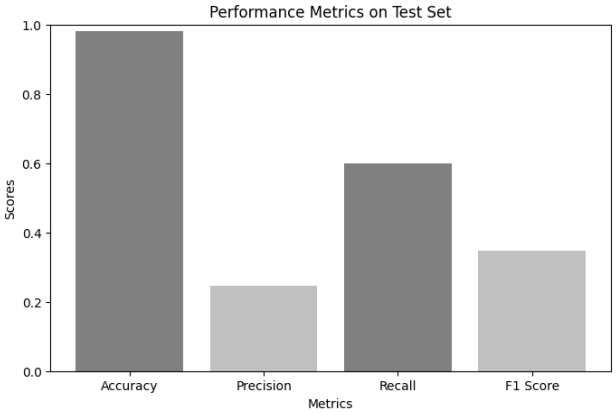
The original data had a ratio of about 30:1 of undrafted to drafted players, and with balancing that ratio was reduced to 10:1. The players randomly were sampled without replacement from years 2009-2021 and put into 10 different data frames that were all used to train the model.

3.3.5 Testing Semi-Balanced Results

The Accuracy, Precision, Recall, and F1 Score metrics for the training data results were 0.9437, 0.7571, 0.5794, 0.6458 respectively, which at first indicated that the model was going to perform better.



The Accuracy, Precision, Recall, and F1 Score metrics for the test results were 0.9813, 0.2448, 0.6, 0.3478 respectively, which indicates that the model is not performing well for new never seen data. The model could be optimized further by using varying techniques for balancing, although the discrepancy in the F1 Score may indicate that the model is overfit. The accuracy is also still high which indicates that there is still some imbalance. Some further improvements may include sampling the test data with the same ratio at which the training data was sampled.



3.3.6 Comparison of Methods

Comparative analysis of method 1 which trained the model using k-fold cross validation, and method 2, which focuses on training with semi-balanced data, offered valuable insights into the application and efficiency of different data mining approaches when applied to predicting NBA draft picks from NCAA players. We can see from the model's f1-score that the model trained with k-fold cross-validation performed better overall. Using the k-fold cross-validation approach was effective in managing the unbalanced nature of our dataset. It accomplished this by ensuring that each fold was a good representation of the data as a whole.

Each model used for both testing methods combined the strengths of different models, which led to improved performance metrics and more reliable predictions. Using multiple models can also increase the time and cost to run the predictive model. It does help eliminate potential weaknesses from one individual model, but might take more time and resources to run on larger datasets.

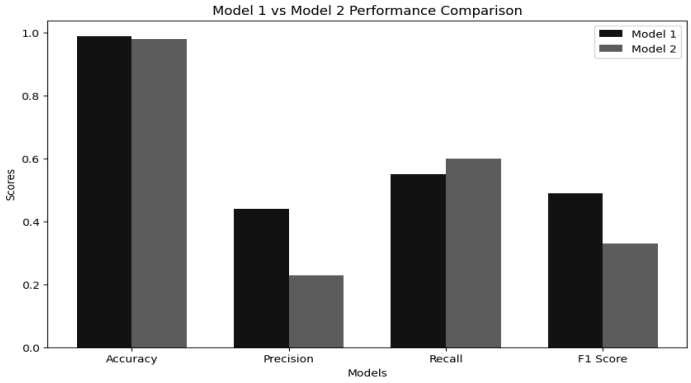
This model trained with k-fold scored an accuracy of99%, precision was 44%, recall was 55% and the f1-score was 48%. This suggests that the model is accurate in making positive predictions of a player being drafted and is relatively precise in its predictions. This has a lower risk of possible false positives, but at the risk of overlooking some potential draft picks. A higher F1 score suggests that there is more balance between precision and recall.

Overall, method 1's approach is designed for datasets with a class imbalance, which is present in cases where the number of positive predictions of drafted players is much lower than the negative predictions of undrafted players. This shows the importance of choosing models and training techniques that match your dataset and task you are trying to achieve.

Method 2 focused on trying to get the data more semi-balanced. It was designed to try to more directly address the imbalance between the number of undrafted and drafted players in the dataset by undersampling the undrafted class.

This method did achieve a higher recall score than method 1, achieving an accuracy of 98%, a precision of 23%, a recall of 60%, and an F1 score of 33%. This shows that enhancing the model's sensitivity to class-drafted players lost some precision but grew in recall score. This could indicate that balancing the data could have led to underperformance in terms of accuracy, precision, and the balance between precision and recall.

Method 2's balanced approach would be better suited at making sure no potential drafted player is overlooked. Method 1's performance overall was better than method 2 in terms of precision and the quality of predictions, and would therefore be the preferred model.



4 **CONCLUSION**

In conclusion, we found that, due to the higher f1-score from training with k-fold, the second method for training was better suited for our data. However, while we made some significant improvements from our initial models performance (ex k-fold: 66% precision, 8% recall and 14% f1-score), precision, recall and f1 still remained relatively low for the model trains with each method. This is likely due to the degree to which the data is unbalanced, and the fact that players with significant performance metrics with respect to our features, may not even apply to the draft. Because of this, it may be difficult for our model to discern “draftable” players from those which are not.

With respect to the features we found to be the most effective with our models, we felt their complexity might hinder players' understanding of specific areas for improvement. This was the case as each feature required a complex equation with multiple player stats, as such, it might make it difficult for them to understand exactly what needs to be improved upon. However, if for instance they do have a low players adjusted defensive rating, while it may not express that they need to box out more, or get more rebounds, it might encourage them to focus more on their overall defensive game.

We did feel, however, that the features used can help teams and recruiters find and target players. If a player has good performance for a number of the features used, it could encourage a recruiter to consider them further during the draft.

Future related work might benefit from focusing more closely on just the features that a player can better focus on to improve their game. Features such as dunks made, rim shots made, defensive rebounds and so on. This targeted approach may enhance the practicality and effectiveness of player development strategies, while also giving recruiters more insight into player performance.

REFERENCES

[1] College Basketball 2009-2021 + NBA Advanced Stats. (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/adityak2003/college-basketball-players-20092021

[2] Han, J., Kamber, M., & Computer, P. (2012). Data mining : concepts and techniques. Third Edition, Elsevier/Morgan Kaufmann.

[3] Box Plus-Minus (BPM) Explained. (2017, May 8). Www.nbastuffer.com. https://www.nbastuffer.com/analytics101/box-plus-minus/#:~:text=Box%20plus%2Dminus%20is%20based

[4] Hosni, M., Abnane, I., Idri, A., Carrillo de Gea, J. M., & Fernández Alemán, J. L. (2019). Reviewing ensemble classification methods in breast cancer. Computer Methods and Programs in Biomedicine, 177, 89–112. https://doi.org/10.1016/j.cmpb.2019.05.019

[5] Haghighat, Maral, et al. “A Review of Data Mining Techniques for Result Prediction in Sports.” *ACSIJ*, vol. 2, no. 5, 2013, pp. 7-12. *https://citeseerx.ist.psu.edu/*, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=37767ab1f559dda7a8d239aec0af1c9485d3d426.

[6] Evans, B. A., & Pitts, J. D. (2022). "The Determinants of Draft Position for NBA Prospects." New York Economic Review, Summer 2022, Issue 52, pp. 22-37.

[7] Greer, Tiffany; Price, Joshua A.; Berri, David J. "Jumping in the Pool: What Determines Which Players the NBA Considers in the Draft?" The Free Library 01 February 2019. 13 December 2023 <https://www.thefreelibrary.com/Jumping in the Pool: What Determines Which Players the NBA Considers...-a0583252510>.