EXPERIMENT REPORT

Student Name: Yasaman Mohammadi
Project Name: advmla-2023-spring

Date:
01/09/2023
Deliverables:
notebook name: Mohammadi_Yasaman-24612626-week3_XGBOOST-2

1.EXPERIMENT BACKGROUND

1.a. Business Objective

This project aims to create a model that predicts whether a college basketball player will be drafted into the NBA based on their current season's statistics. As a result, NBA teams can make better draft selections and allocate resources more efficiently.

By making accurate predictions, draft decisions can be improved, players can be developed, and teams can perform better. Missed opportunities, wasted resources, and potential damage to the model's credibility can all be attributed to incorrect predictions. Draft choices, player development strategies, and fan engagement are all impacted by the model's success.

1.b. Hypothesis

Do college basketball player statistics provide a sufficient basis for predicting their likelihood of being drafted into the NBA?

> Accurate predictions can lead to improved draft selections, resulting in a better roster for the team and an overall improvement in performance.
>
> Data-driven predictions can enhance sports analysis, media coverage, and fan discussions. As a result, it can generate excitement and stimulate informed discussion about the potential outcomes of a draft.

1.c. Experiment Objective

The experiment is expected to result in a predictive model with a high AUROC score that accurately predicts whether a college basketball player will be drafted into the NBA based on their current season's statistics. AUROC scores should be significantly higher than random chance, indicating that the model can reliably distinguish between players who are likely to be drafted and those who are not.

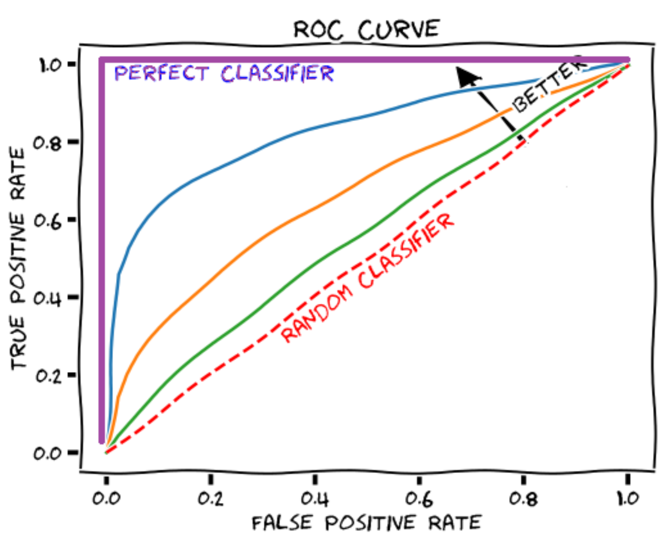

Figure 1- ROC curve

Different scenarios might happen based on AUROC curve performance. The area under the receiver operating characteristic (AUROC) is a performance metric for evaluating classification models. The larger the area under this curve, the better the performance.

2.EXPERIMENT DETAILS

2.a. Data Preparation

	the steps taken for preparing the data for previous experiment.	**Explanation**
1	Data collection	Building a machine learning model begins with the collection of data from a variety of sources. In this case, the data was collected from a Kaggle competition.
2	Explore Dataset	A general understanding is derived from the data in this step.
3	Data cleaning	Data should be preprocessed to remove any irrelevant, missing, or corrupted data points and any discrepancies in the data.
4	Data visualization	Visual understanding of data distribution is gained by plotting different charts in both Python and Tableau.
5	Feature engineering	In feature engineering, new features are created, or existing features are transformed to improve the model's performance.
6	Splitting the dataset	The dataset was randomly split into training, validation, and testing. The training set is used to train the model, the validation set is used to tune the model's hyperparameters and prevent overfitting, and the test set is used to evaluate the final performance of the model. The test size parameter specifies the percentage of data allocated to the test set, in this case, 20%. The random state parameter is used to ensure the reproducibility of the split. For many datasets, the 80/20 split is a good rule of thumb since it provides enough data to train the model while still leaving a considerable amount for testing. The ratio may vary depending on the size and complexity of the dataset, as well as the specific problem being addressed. As part of this project, there was a test set without drafted sections and the objective was to predict a probability for the drafted variables at the end.
7	Scaling the dataset	The data is crucial to prevent sensitivity to some features, so all features have the same scale. Since some machine learning algorithms are sensitive to the scale of input features, this is important.

Table 1- the steps taken for preparation of the data for previous experiment.

Data Preparation Steps for the Experiment:

1. **Data Cleaning:**

"The type" and "num" columns have been removed because the type has only one unique value, while the num column serves as an identifier.

2. **Feature Engineering:**

   a. *Categorical features:*

Having understood from the tableau dashboard that teams and conference columns play a vital role, we have used one hot encoding to turn these categorical features into numerical ones. Since there are 358 unique values in the team column, using one-hot-encoding created too many columns, resulting in a more time-consuming machine learning model at the end, but with better results than removing these columns as in the previous experiment.
the "year" column underwent label encoding due to its relevance.
reconsidered the "ht" column. In this instance, columns containing month information were transformed into numerical representations of month numbers. Additionally, a conversion to height in feet format was performed, supplementing the dataset with more comprehensive information.

   b. *Numerical features:*
   In the data set, there are two critical columns with a large number of missing values.

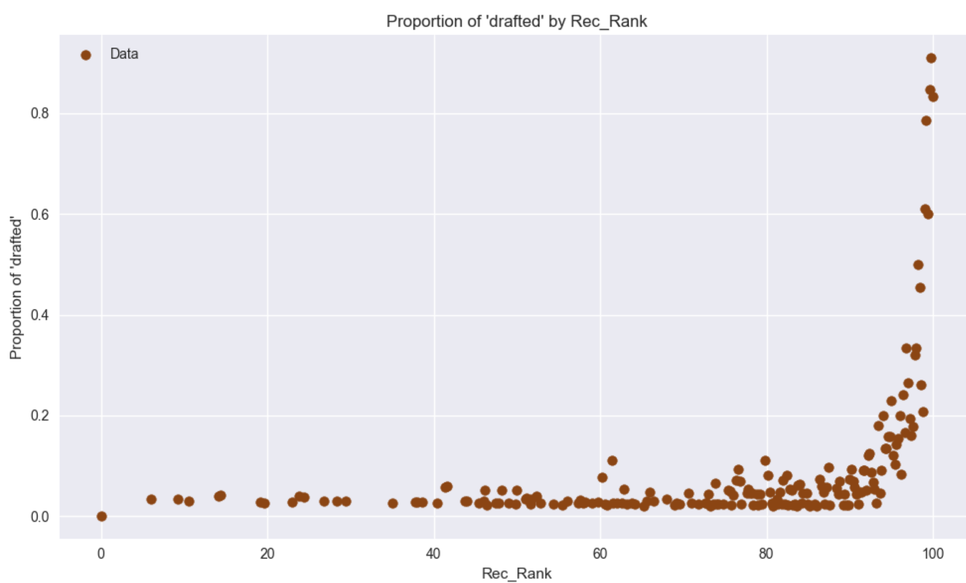

Figure 2-proportion of y vs. Reck_Rank

We fill the missing values of this column with 0.

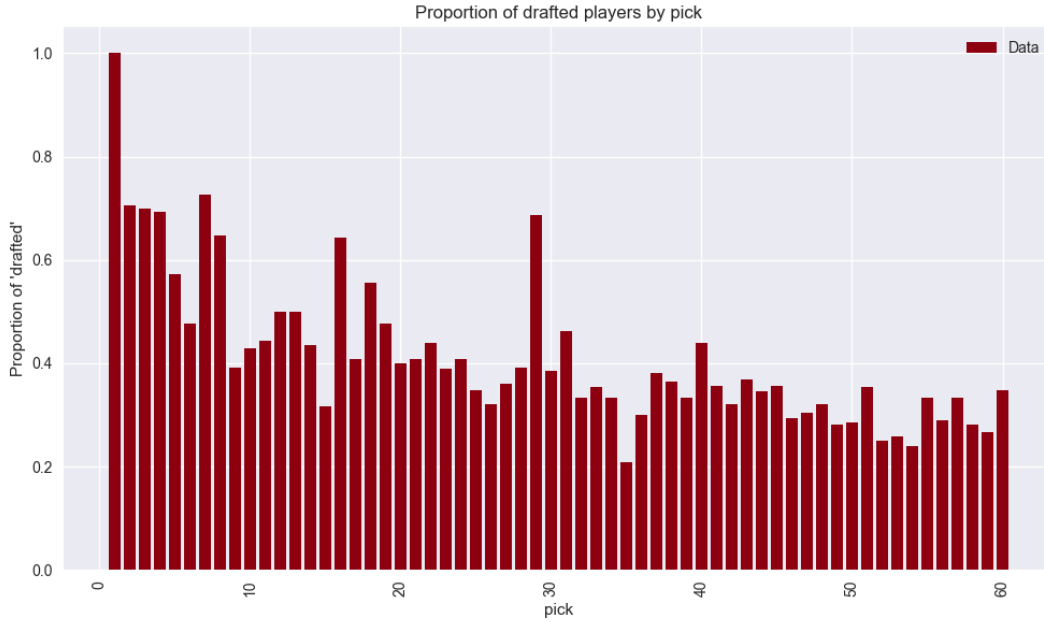

Figure 3-Proportion of drafted players vs pick

Since as the number of pick increases, proportion of drafted decrease we fill missing value in this column with high number (100) to result in 0 at the end.

Imputation technique:

The missing value is estimated as the mean of the non-missing variables for the remainder of the numerical values.

Figure4-The mnso bar chart for fiding missing values

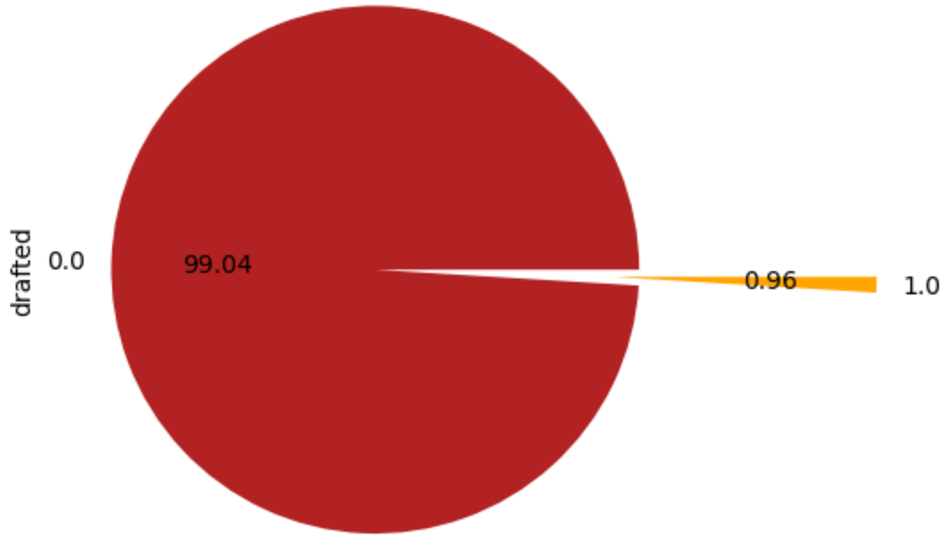

Figure 5-Pie chart of drafted distribution

2.b. Feature Engineering

As discussed in the data preparation step, one-hot-encoding, label-encoding, and feature imputation were performed on the data to prepare it for machine learning.
It is important to note that the feature engineering in this step differs from the last experiment, resulting in a higher prediction accuracy at the end.

2.c. Modelling

In the previous experiment, we used XGBOOST, as it is a powerful model and showed good performance; however, for this experiment, we tested XGBOOST a second time with entirely different feature engineering and achieved higher prediction accuracy.

	Hyperparameter	Explanation
1	N estimator	indicate how many trees are present in the forest. Generally, the larger the number of trees, the better the ability to learn the data. To find the optimal value, a grid search was used to find the optimal number of trees. However, adding too many trees can significantly slow down the training process. With grid search, a n estimator of 250 to 450 was selected for this experiment, and 250 was determined to be the best value.
2	Max depth	In XGBOOST, the maximum depth is calculated as the longest path between the root node and the leaf node. With grid search, a maximum depth of 15 to 20 was selected for this experiment, and 17 was determined to be the best value. To prevent overfitting, higher values are not given.
3	Gamma (Minimum loss reduction)	Gamma specifies the minimum loss reduction required for a split. As a result, the algorithm becomes more conservative. Depending on the loss function, the values may vary. With grid search, eta values between 0.05 to 0.06 were selected for this experiment, and 0.05 was determined to be the best value. A larger gamma will result in a more conservative algorithm, so higher values were not used.
4	eta (learning rate)	To prevent overfitting, step size shrinkage is used in the update process. At each step of the boosting process, the weights of new features are achieved, and eta shrinks the weights of new features to make the boosting process more conservative. With grid search, eta values between 0.1 and 0.3 were selected for this experiment, and 0.1 was determined to be the best value.
5	subsamples	The calculation accuracy decreased when we used subsamples between 0 and 1, so we eliminated them.

Table 2-Hyperparameters tuning explanation

It was decided to use stratified k-fold cross-validation due to the imbalanced distribution of the target feature in the primary data. Therefore, training and test data in each fold will reflect the imbalanced distribution of the target feature in the primary data.

3. EXPERIMENT RESULTS

3.a. Technical Performance

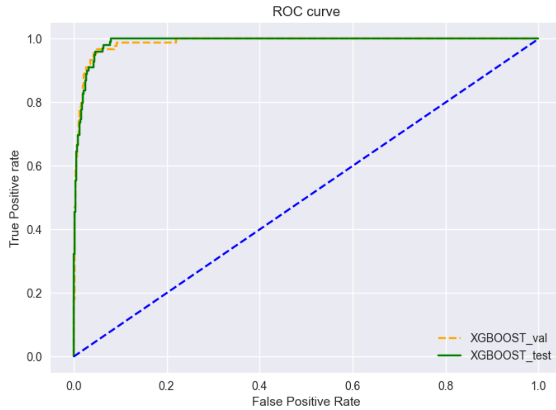

Figure 6-AUROC curve of previous XGBOOST model

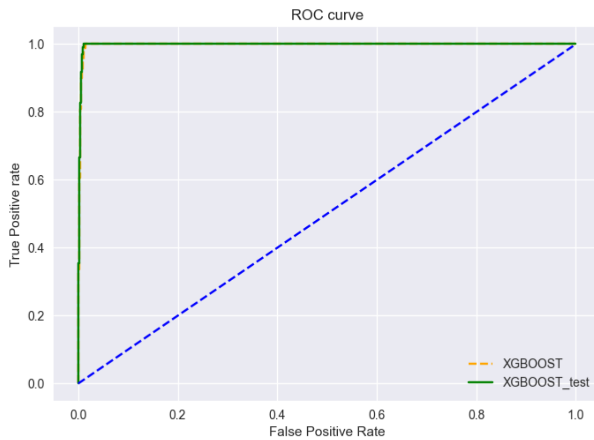

Figure7- AUROC curve of current XGBOOST model

XGBOOST has greatly improved with more feature engineering and a final score of 0.99933 was achieved on Kaggle, which represents a significant improvement over the previous XGBOOST model which had a score of 0.9612.
Furthermore, this new model appears to perform better based on AUROC curves.

3.b. Business Impact

The success of the model affects several aspects of the NBA teams' operations, including draft choices, player development, resource allocation, fan engagement, and team performance. Errors in results can have a wide range of negative impacts, including wasted resources, missed opportunities, credibility damage, and long-term competitive disadvantages. For the model to provide meaningful value to NBA teams and stakeholders, accurate predictions are critical.

3.c. Encountered Issues

As we mentioned in the previous experiment, imputing missing values correctly resulted in a better score.

4.FUTURE EXPERIMENT

4.a. Key Learning

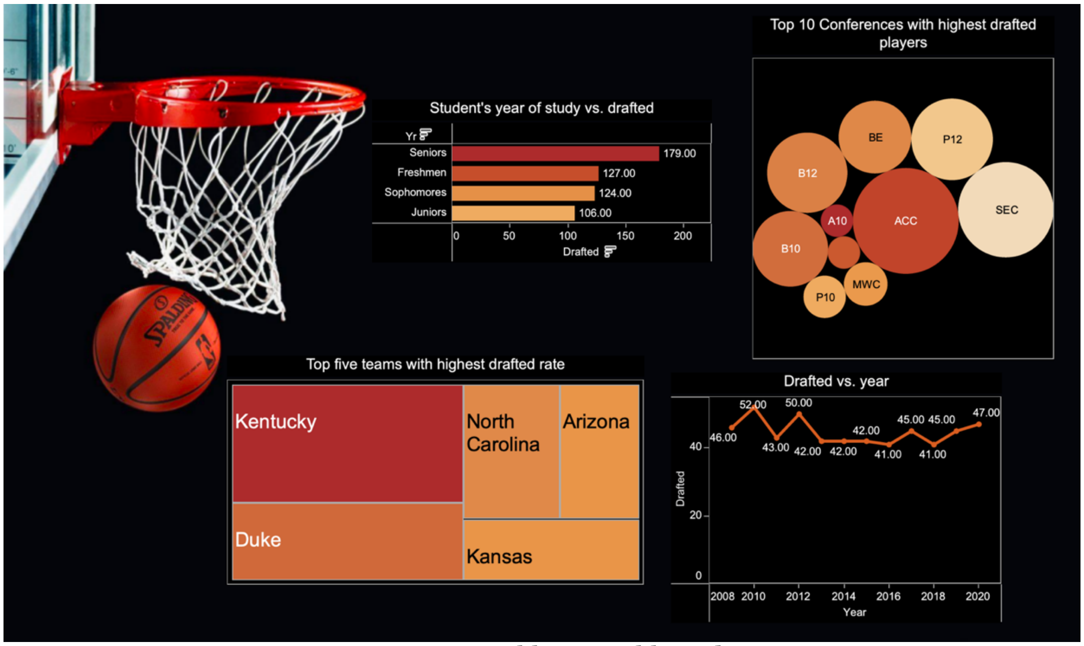

Figure 8- Tableau Dashboard

An interactive dashboard in Tableau has been created to visualize the correlation between drafted players' rates and teams, seasons, and conferences.

As demonstrated in this experiment, applying one-hot-encoding to these features improved performance when compared with removing them in the previous experiment.

Performance was improved by using XGBOOST in conjunction with additional feature engineering. However, it's worth noting that even better machine learning models, like ADABoost, could lead to achieving even higher scores.

4.b. Suggestions / Recommendations

In future experiments, different machine learning models with hyperparameter tuning will be used. In order to deploy the final solution into production, the user interface must be refined so that predictions are easy to access, ethical biases must be addressed proactively, and ongoing model adjustments should be monitored in real-time.

References:

1. Anderson, A., & Semmelroth, D. (2015). *Statistics for big data for dummies*. John Wiley & Sons.

2. Prashant Shekhar. A Guide on XGBoost Hyperparameters Tuning. Kaggle. https://www.kaggle.com/code/prashant111/a-guide-on-xgboost-hyperparameters-tuning

Private repo link:

https://github.com/JYasimo/Kaggle_competition_NBA_league/tree/main