# Assignment 1

# Kaggle Competition

Monali Patil
Student ID: 14370946

08/09/2023

36120 - Advanced Machine Learning Application
Master of Data Science and Innovation
University of Technology of Sydney

# Table of Contents

# 1. Executive Summary

Project Overview:

The project involves working with the statistical data of college students playing basketball. This learning task aims to predict whether a college basketball player will be drafted to join the NBA league based on their statistics. The significance of this project lies in its potential to provide insights into the factors that influence NBA draft selections, allowing teams to make more informed decisions about which players to select. Additionally, it can help sports analysts and fans anticipate which college players are likely to transition to the professional NBA league.

Problem Statement and Context:

The problem addressed in this project is predicting whether a college basketball player will be drafted into the NBA. This is a highly relevant and impactful challenge in the world of sports, as it can shape the careers of young athletes and have significant financial implications for both players and NBA teams.

The context of this project involves analyzing a dataset containing various statistics related to college basketball players. These statistics include information about the player's performance, efficiency, shooting percentages, rebounding, assists, turnovers, and much more.

By training a machine learning model on historical data, the goal is to create a predictive tool that can evaluate a college player's likelihood of being drafted based on their performance metrics. This model can be used by NBA teams, scouts, and analysts to identify promising talents and make more informed draft choices.

Achieved Outcomes and Results:

The achieved outcomes of this project include the development of a predictive model with the ability to classify whether a college basketball player will be drafted into the NBA. The results of the model, based on historical data, reveal insights into the factors that influence NBA draft selections. These insights can assist in talent scouting and player evaluation. Additionally, this model will offer valuable insights to sports commentators, fans, and scouts, aiding them in predicting individual players 'potential NBA draft prospects.

In summary, this project addresses a significant problem in the world of sports by leveraging machine learning to predict NBA draft outcomes for college basketball players. The achieved model provides a data-driven approach to talent evaluation and draft decision-making, benefiting both players and NBA teams.

## 2. Business Understanding

The project's primary business use case is centered around predicting whether a college basketball player will be drafted to the NBA league based on their performance statistics. This prediction has several practical applications:

- Talent Scouting: Helps NBA teams identify promising college players.
- Player Career Guidance: Assists players in making informed career decisions.
- Fan Engagement: Generates excitement among basketball fans.
- Sports Commentators: Enhances sports journalism and analysis.

Challenges and Opportunities:

- Predictive Accuracy: Ensuring accurate draft predictions.
- Data Complexity: Handling diverse player and team data.
- Real-time Updates: Providing real-time draft predictions.
- Fan Expectations: Managing fan expectations responsibly.
- Ethical Considerations: Addressing data privacy and fairness.

Key Objectives:

- Accurate Predictions: Build a highly accurate draft prediction model.
- Scalability: Ensure the model can handle real-time data during the draft.
- Privacy and Fairness: Handle player data ethically and avoid bias.
- User-Friendly Interface: Create an easy-to-use interface for stakeholders.
- Stakeholder Collaboration: Work closely with NBA teams, players, and media.
- Ethical Guidelines: Establish guidelines for responsible prediction use.
- Continuous Improvement: Incorporate new data to improve predictions.

# 3. Data Understanding

**Dataset:**

The dataset provided contains a wide range of features that illuminate players' performance during their college basketball season. The dataset comprises 64 players' performance attributes, including Games Played (GP), Minutes Played (Min_per), Offensive Rating (ORtg), Defensive Rating (DRtg), Field Goals Made (twoPM), Free Throws Made (FTM), and many others offer insights into various facets of a player's playing style and contribution to their team.

**Data Source and Collection Methods:**

The dataset, available in CSV format, was accessible via the canvas portal, with distinct files designated for training and testing. As it was acquired through the University portal as part of student resources, there were no concerns related to copyright or privacy issues.

- metadata.csv: Metadata of the Basketball Players
- train.csv: Basketball Players Training dataset
- test.csv: Basketball Players Testing dataset

**Data Limitations:**

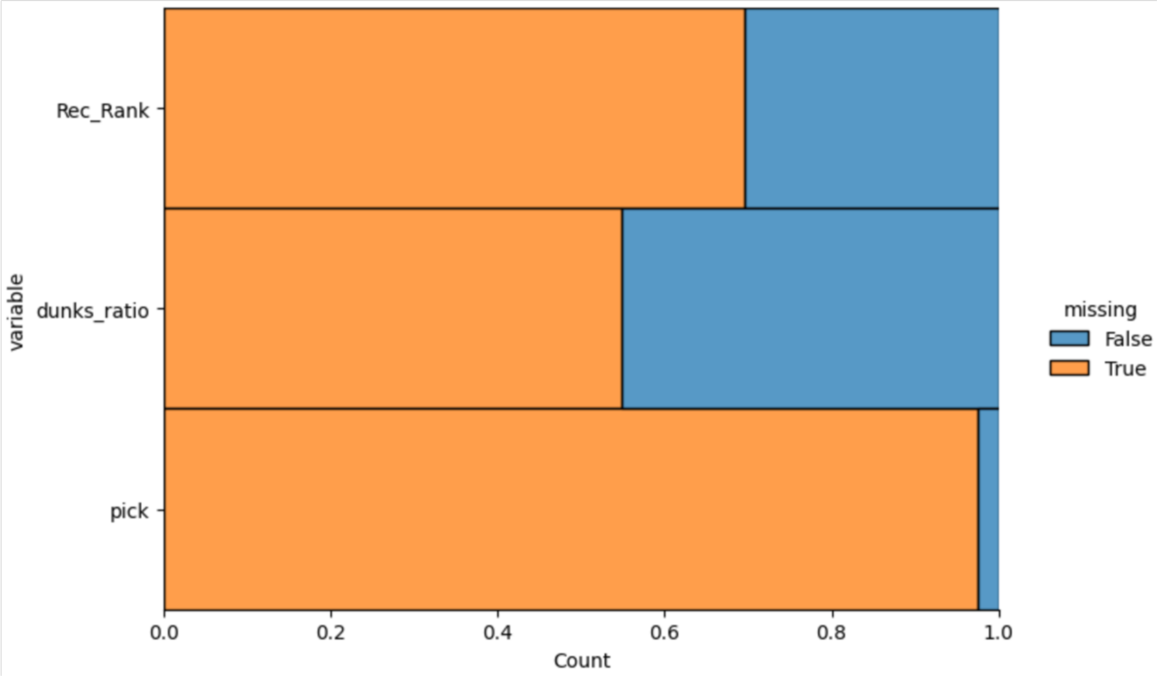
- The absence of metadata for two features, 'ftr' and 'pfr,' presents a challenge in comprehending their definitions.
- The 'ht' column indicates the player's height, with the feature values presented in a data format. Consequently, some processing is necessary to extract meaningful values.
- Additionally, the dataset includes unique identifiers such as player IDs, player numbers, and conference names, which could potentially lead to overfitting of the model.

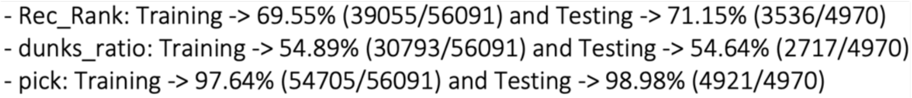
The exploratory data analysis techniques, checking the dimension of the database, feature names, accessing the initial datapoints etc. using different pandas functions were carried out to examine and study players' information to comprehend and uncover patterns, aiming to identify prospective players for NBA draft selection.

# 4. Data Preparation

To ensure the quality of the data to be utilized by the model analysis, conducted the below activities.

- **Handing missing/null values.**





```
- Rec_Rank: Training -> 69.55% (39055/56091) and Testing -> 71.15% (3536/4970)
- dunks_ratio: Training -> 54.89% (30793/56091) and Testing -> 54.64% (2717/4970)
- pick: Training -> 97.64% (54705/56091) and Testing -> 98.98% (4921/4970)
```

Among the features with missing data, the following three attributes exhibit notably high levels of missing values, exceeding 50%. Therefore, it is practical to exclude these features in order to prevent potential biases in the model arising from imputation.

Given that the remaining features have missing values comprising less than 2%, it is a reasonable approach to fill these missing values using the mean for numerical attributes and the mode for categorical attributes.

- **Eliminating identifiers.**

Removed unique identifiers namely 'player_id', 'num', 'team' and 'conf' as its inclusion in the analysis can lead to overfitting, where the model fits to these specific values rather than the underlying generalized patterns in the sportsman's records.

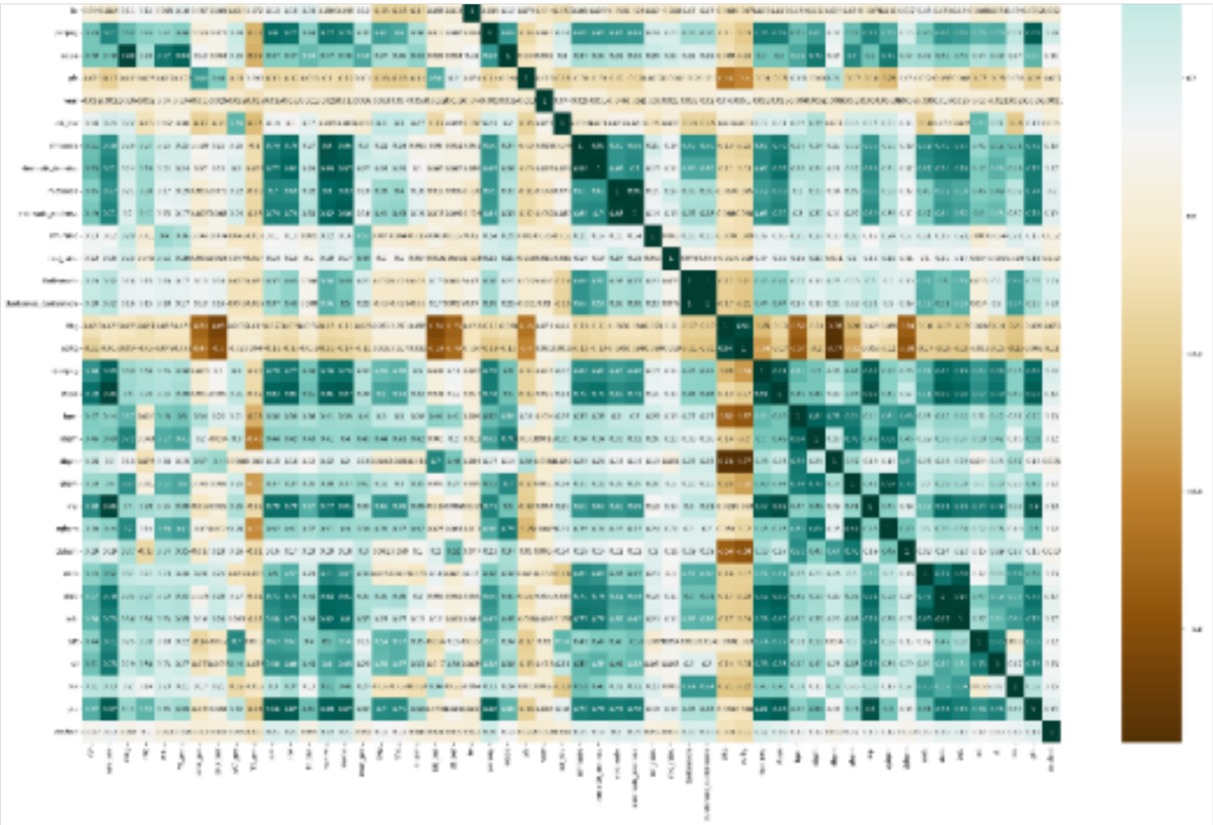
- **Duplicate records.**

Checked duplicate records is a crucial step in data preprocessing and quality assurance, ensuring that the analysis is based on reliable and unbiased data.  There were no duplicate data points in the dataset representing a unique and distinct observation, which is essential for various analytical tasks.

- **Feature Engineering - Processing ‘ht’ feature to derive suitable information.**

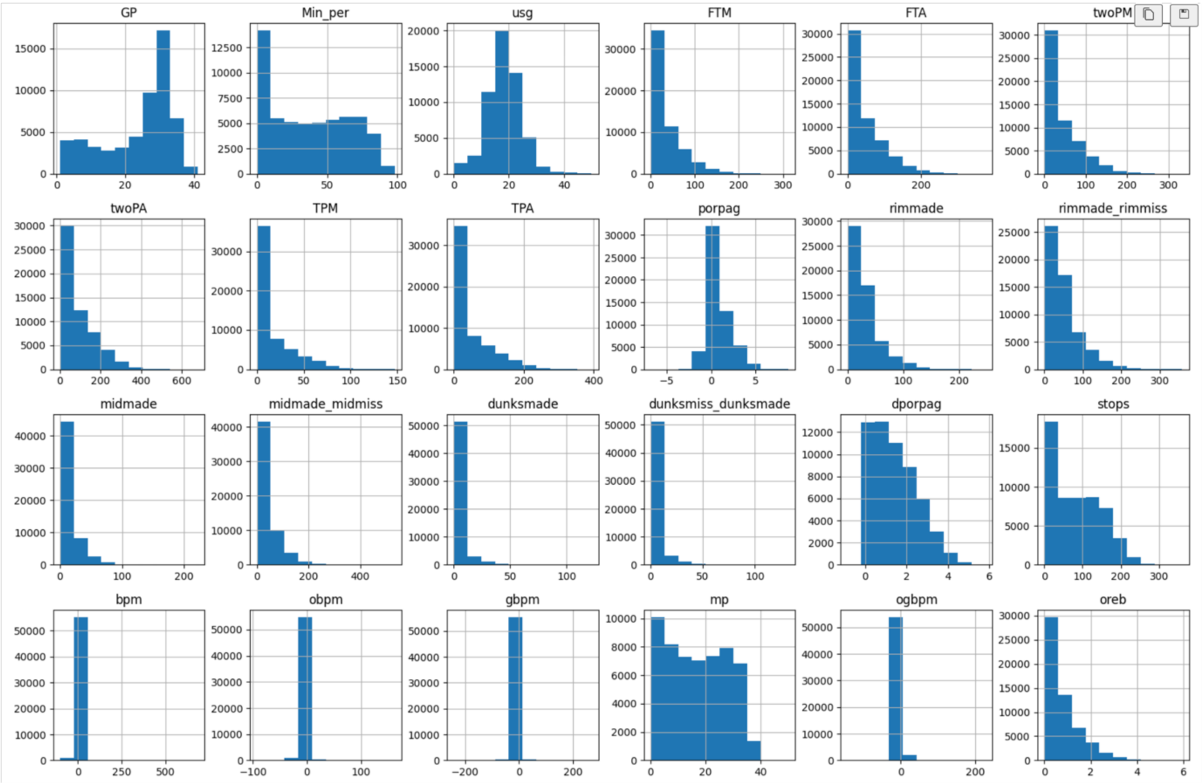
The 'ht' attribute, initially in a date format, represents player height, vital in basketball. Unique value analysis revealed that 'Jun,' 'Jul,' and 'Aug' don't signify months but likely correspond to heights like 6 feet, 7 feet, etc. Consequently, these values were converted into numerical 'ht_cm' for centimetres as numerical input for machine learning.

- **Selecting appropriate features based on the correlation coefficient.**



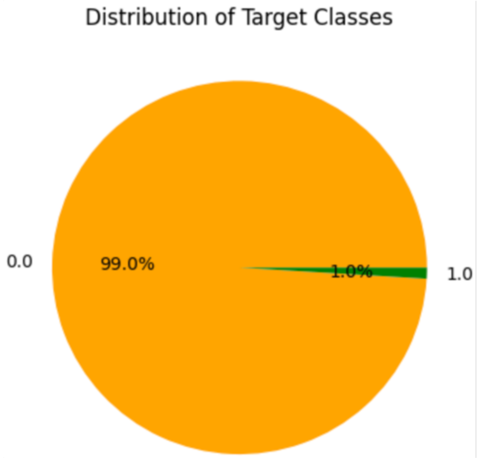
Performed correlation analysis to determine which features would be most valuable for building a predictive model for the NBA draft prediction task.

- Data distribution of various features.



It allows to detect outliers or anomalies in the data and explains characteristics of your data, such as central tendency, spread, and shape. Assessing the above chart informs decisions about scaling data preprocessing steps, as the feature data is present on various scales.

- Accessing if imbalance targets classes.



The above pie chart illustrates a significant class imbalance within the dataset. The majority of observations are attributed to a single target class, representing players who have not been drafted denoted by the value 0.

Therefore, performed Oversampling with the SMOTE (Synthetic Minority Over-sampling Technique) method to address class imbalance by generating synthetic observations for the minority class representing players who have been drafted denoted by the value 1.

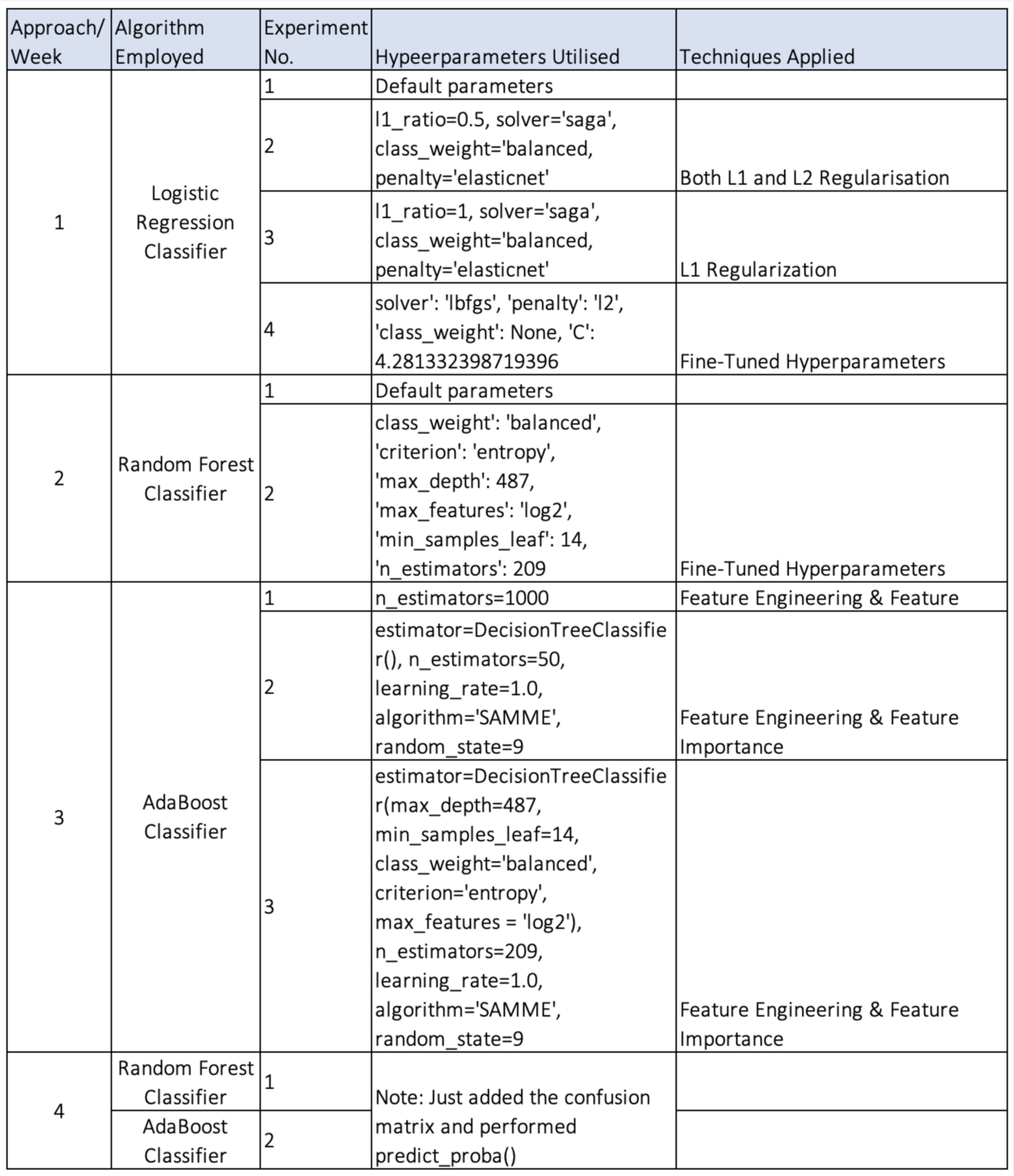
- Features Scaling.

Using feature scaling prevents the algorithm from prioritizing high-value features over other more informative ones. It ensures uniformity in feature values, enabling the algorithm to learn generalized patterns from all features for accurate player identification and predictions.

Employed StandardScaler method because it maintains the features data distribution's shape and retains outliers by scaling data using the mean of 0 and standard deviation of 1 across the entire dataset, rather than for individual data points.

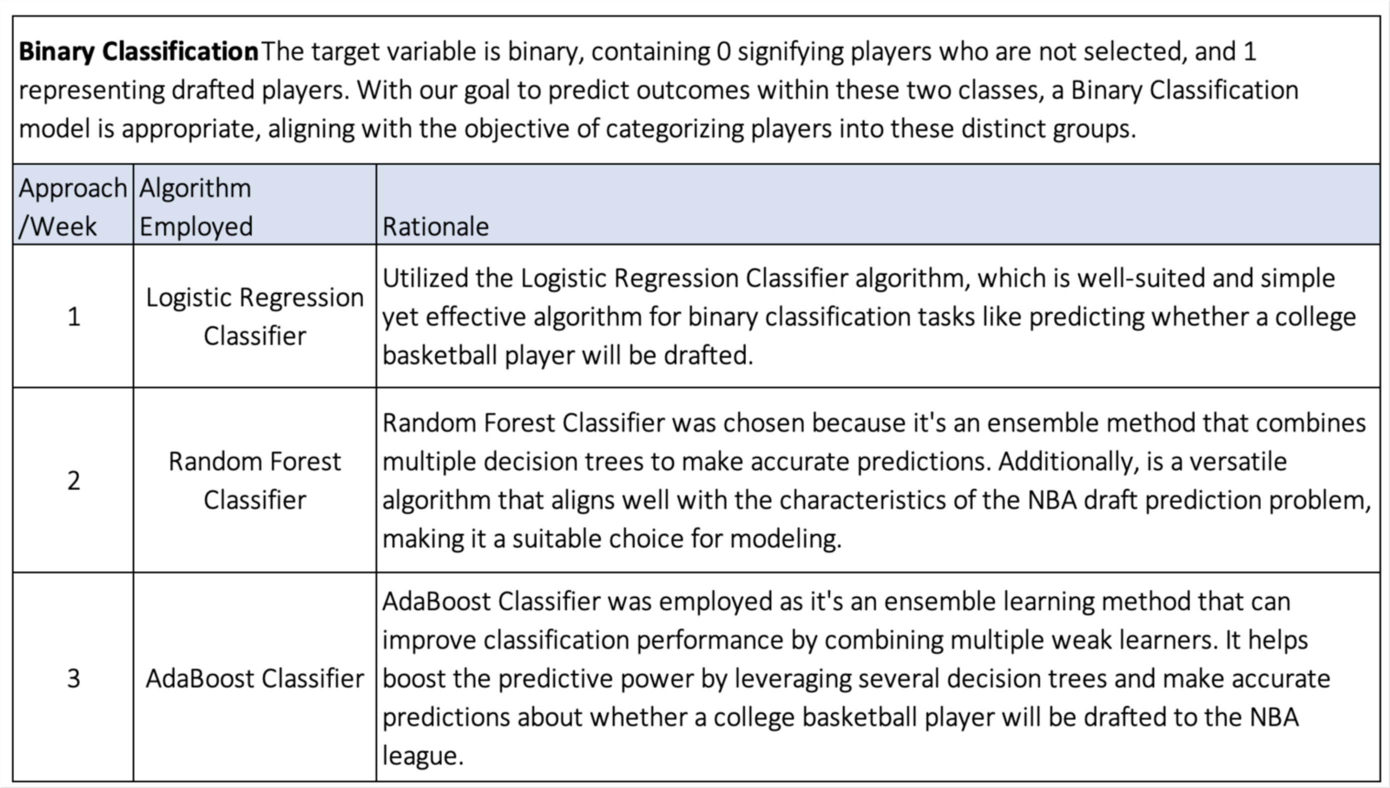
# 5. Modeling

As part of the learning process, below are the Classifier models built, trained, and tested for this binary classification problem.



| Approach/ Week | Algorithm Employed | Experiment No. | Hypeerparameters Utilised | Techniques Applied |
|---|---|---|---|---|
| 1 | Logistic Regression Classifier | 1 | Default parameters | |
| | | 2 | l1_ratio=0.5, solver='saga', class_weight='balanced, penalty='elasticnet' | Both L1 and L2 Regularisation |
| | | 3 | l1_ratio=1, solver='saga', class_weight='balanced, penalty='elasticnet' | L1 Regularization |
| | | 4 | solver': 'lbfgs', 'penalty': 'l2', 'class_weight': None, 'C': 4.281332398719396 | Fine-Tuned Hyperparameters |
| 2 | Random Forest Classifier | 1 | Default parameters | |
| | | 2 | class_weight': 'balanced', 'criterion': 'entropy', 'max_depth': 487, 'max_features': 'log2', 'min_samples_leaf': 14, 'n_estimators': 209 | Fine-Tuned Hyperparameters |
| 3 | AdaBoost Classifier | 1 | n_estimators=1000 | Feature Engineering & Feature |
| | | 2 | estimator=DecisionTreeClassifier(), n_estimators=50, learning_rate=1.0, algorithm='SAMME', random_state=9 | Feature Engineering & Feature Importance |
| | | 3 | estimator=DecisionTreeClassifier(max_depth=487, min_samples_leaf=14, class_weight='balanced', criterion='entropy', max_features = 'log2'), n_estimators=209, learning_rate=1.0, algorithm='SAMME', random_state=9 | Feature Engineering & Feature Importance |
| 4 | Random Forest Classifier | 1 | Note: Just added the confusion matrix and performed predict_proba() | |
| | AdaBoost Classifier | 2 | | |

Considering the business objective of determining potential players and selection, there is no good theory to map and select a suitable algorithm for this binary classification problem, so different experiments are performed to discover which algorithm and algorithm configuration results in the best performance for this binary classification task.



**Binary Classification**The target variable is binary, containing 0 signifying players who are not selected, and 1 representing drafted players. With our goal to predict outcomes within these two classes, a Binary Classification model is appropriate, aligning with the objective of categorizing players into these distinct groups.

| Approach /Week | Algorithm Employed | Rationale |
|---|---|---|
| 1 | Logistic Regression Classifier | Utilized the Logistic Regression Classifier algorithm, which is well-suited and simple yet effective algorithm for binary classification tasks like predicting whether a college basketball player will be drafted. |
| 2 | Random Forest Classifier | Random Forest Classifier was chosen because it's an ensemble method that combines multiple decision trees to make accurate predictions. Additionally, is a versatile algorithm that aligns well with the characteristics of the NBA draft prediction problem, making it a suitable choice for modeling. |
| 3 | AdaBoost Classifier | AdaBoost Classifier was employed as it's an ensemble learning method that can improve classification performance by combining multiple weak learners. It helps boost the predictive power by leveraging several decision trees and make accurate predictions about whether a college basketball player will be drafted to the NBA league. |

### Hyperparameter Tuning:

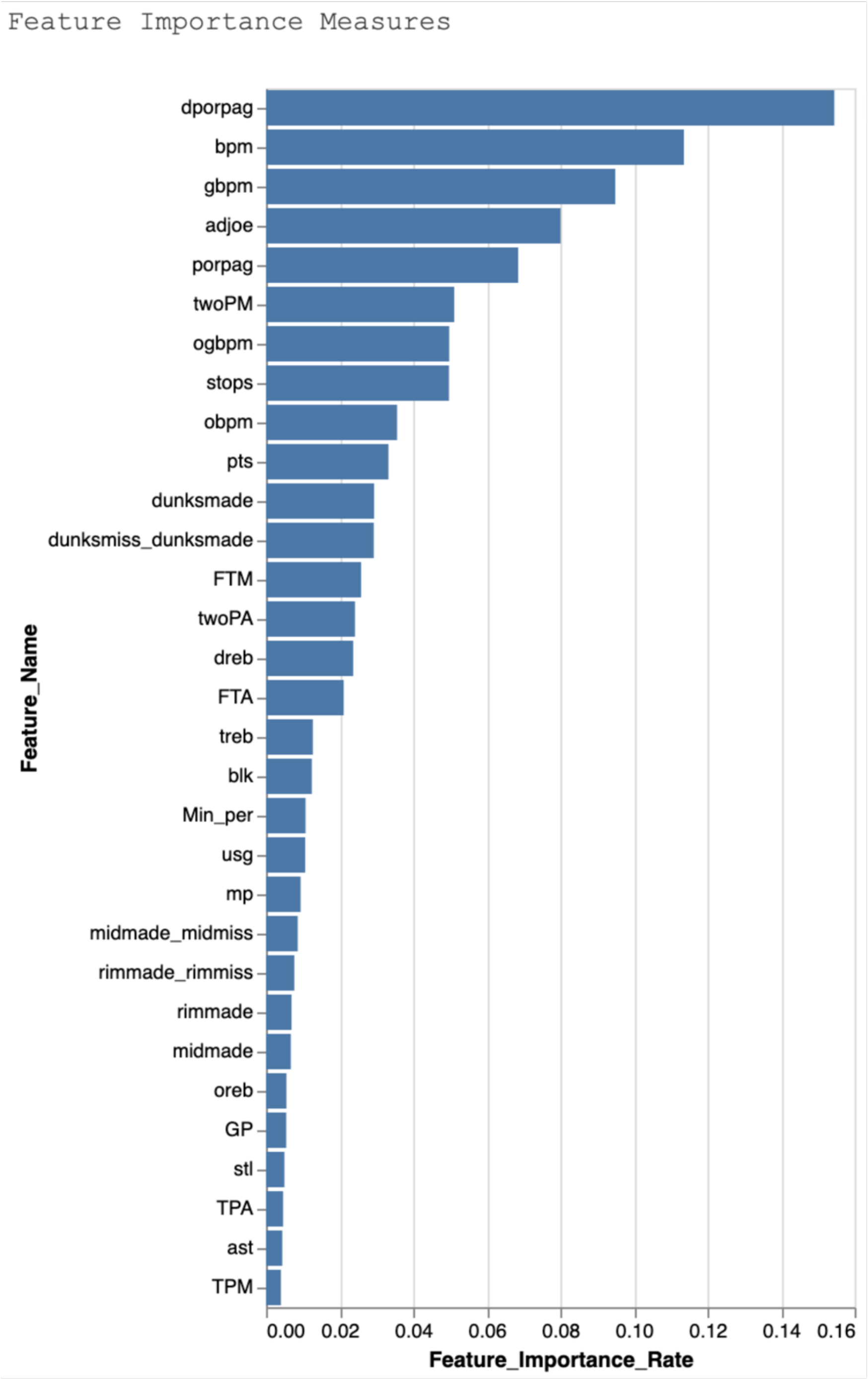
Utilized Hyperparameter Tuning to identify the best hyperparameter values for the Random Forest Classifier algorithm, addressing the slight overfitting observed in the model.

Applying the Random Search method for hyperparameter tuning, which involves incorporating randomness by selecting values from the search space. This approach can generate various combinations of hyperparameter values that lead to achieving minimal error. Grid Search's evenly spaced points may miss optimal hyperparameters.

### Feature Importance:

Derived feature importance measure that can offer valuable insights into the players' records, to help comprehend which features have the most significant impact on the model's predictions to identify prospective players for selection in the NBA league.

Furthermore, this measure is readily available as the Random Forest Classifier algorithm computes the change in purity for each feature during the splitting process. By combining these values for each feature, the algorithm determines the influence of each feature on predictions.



The bar chart above illustrates that the features making the least contribution include 'TPM', 'ast', 'TPA', 'stl', 'GP', 'oreb', 'midmade', 'rimmade', 'rimmade_rimmiss', 'midmade_midmiss', 'mp', 'usg', 'Min_per', 'blk', 'treb'.

The rest of the features have a significant impact on predicting either class 1, indicating players selected for the NBA league, or class 0, representing those not chosen. Their feature importance rates range from 0.02% to 0.15%.

# 6. Evaluation

Evaluation Metrics:

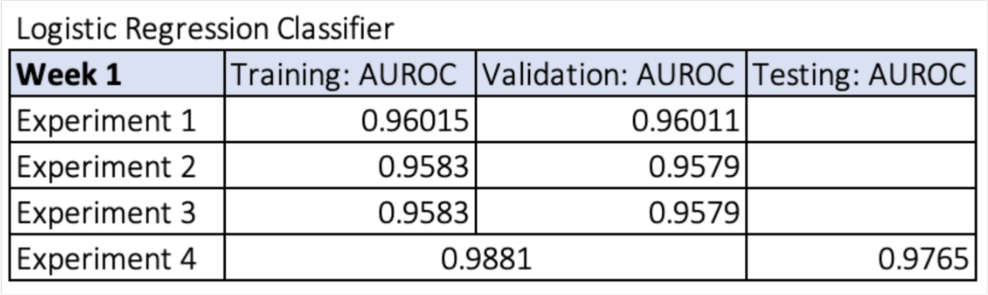
The model's performance is assessed using the AUROC (Area Under ROC) metric, which assesses a model's ability to distinguish between the two classes (drafted or not drafted) by measuring the trade-off between true positive rate and false positive rate.

The AUROC metric is relevant because it quantifies the model's capability to make accurate predictions, which directly aligns with the project goal of predicting NBA draft outcomes. A higher AUROC indicates a better ability to differentiate between drafted and non-drafted players, which is essential for making informed decisions in player selection.

Results and Analysis:

Note: The Notebooks from Week 2$^{nd}$ and Week 3$^{rd}$ are modified (*_Tuned.ipynb) to include a confusion matrix and obtain the probabilities of target classes on the test dataset using the predit_proba() function for the Kaggle result submission. So, the performance score of only the testing dataset has changed and is mentioned in the below results tables.

Week 1:

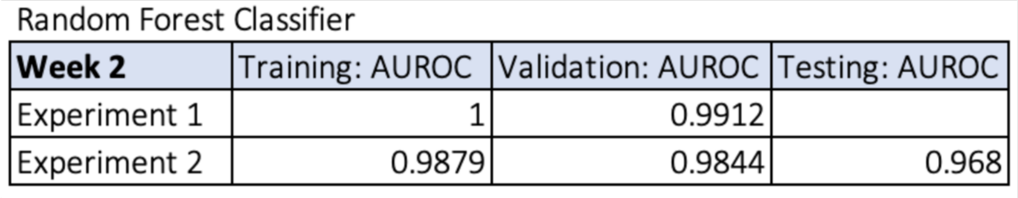


Logistic Regression Classifier

| **Week 1** | Training: AUROC | Validation: AUROC | Testing: AUROC |
|---|---|---|---|
| Experiment 1 | 0.96015 | 0.96011 | |
| Experiment 2 | 0.9583 | 0.9579 | |
| Experiment 3 | 0.9583 | 0.9579 | |
| Experiment 4 | 0.9881 | | 0.9765 |

- In the 1st experiment, the logistic regression algorithm with default hyperparameters had the AUROC performance score of 0.96015 for the training dataset and 0.96011 for the validation dataset, indicating that the model is good at identifying players to be drafted in the NBA league.
- Applying regularization techniques, the models from the 2nd and 3rd experiments have slightly dropped to the same AUROC performance score of 0.9583 for the training dataset and 0.9579 for the validation dataset, demonstrating that the model may slightly miss some potential players who are likely to be drafted in the NBA league in comparison to the 1st experiment's model.
- With Hyper-tuned parameters in the 4th experiment, the model performance has considerably increased to the AUROC score of 0.9881 and 0.9765 on the testing dataset, illustrating that the model has generalized well enough to accurately detect potential players that will be drafted on the unseen data.

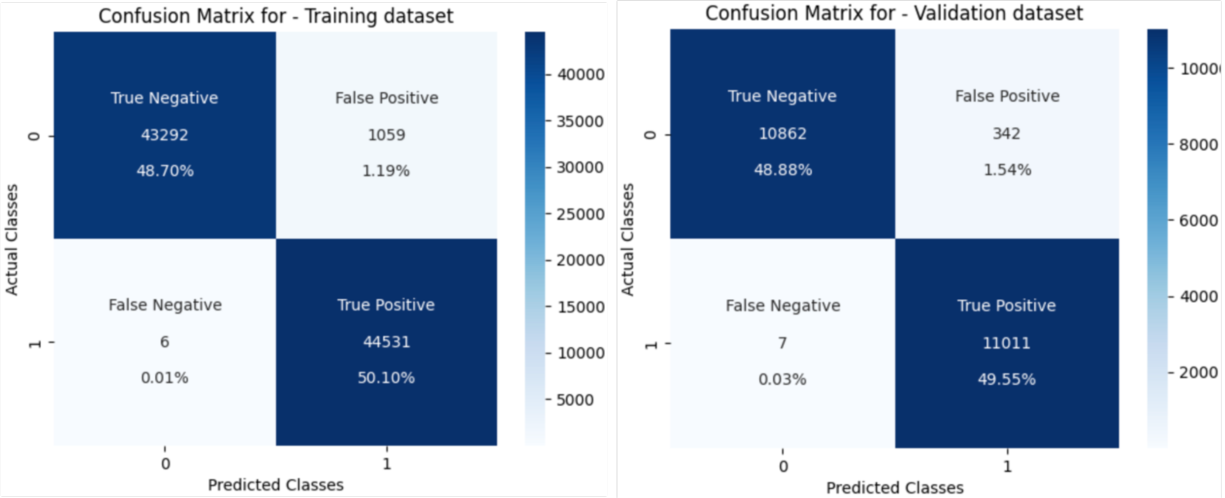
Week 2:



Random Forest Classifier

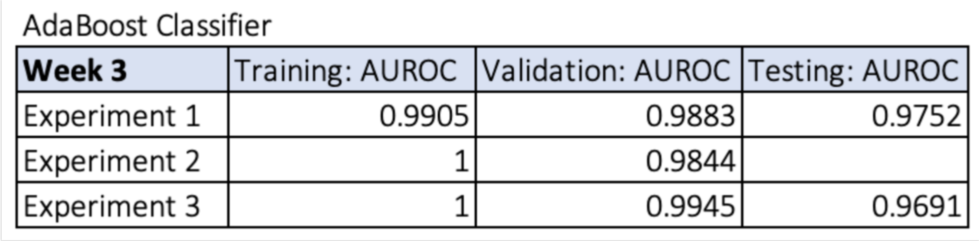
| **Week 2** | Training: AUROC | Validation: AUROC | Testing: AUROC |
|---|---:|---:|---:|
| Experiment 1 | 1 | 0.9912 | |
| Experiment 2 | 0.9879 | 0.9844 | 0.968 |

Experiment 2 Confusion Matrix



- From the 1st experiment, the AUROC performance score on the validation set 0.9912 is slightly lower compared to the ideal 1.0 score achieved on the training set, indicating that the model is narrowly overfitting.
- Therefore, in the 2nd experiment, Hyperparameter Tuning was applied to identify the best hyperparameter values for the Random Forest Classifier algorithm, aiming to mitigate the minor overfitting observed in the model.
- Utilising Hyper tuned parameters in the 2nd experiment, there is a slight variation in AUROC scores between training (0.9879), validation (0.9844) and testing (0.9680) datasets suggesting that the model with optimized hyperparameters is not maintaining consistent performance across datasets and is relatively overfitting on the unseen data.
- Additionally, the occurrence of 1059 False Positive in the training dataset and 342 in the validation dataset suggests that players who should not be selected are mistakenly included in the NBA team which might impact the team’s efficiency and shooting goal percentage.

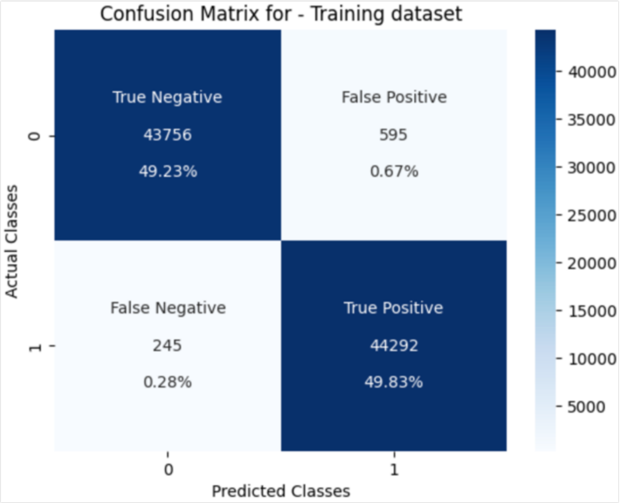
## Week 3:



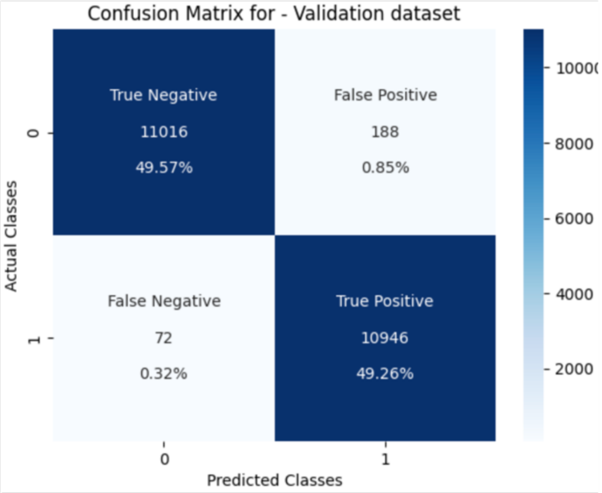
AdaBoost Classifier

| **Week 3** | Training: AUROC | Validation: AUROC | Testing: AUROC |
|---|---:|---:|---:|
| Experiment 1 | 0.9905 | 0.9883 | 0.9752 |
| Experiment 2 | 1 | 0.9844 | |
| Experiment 3 | 1 | 0.9945 | 0.9691 |

Experiment 1 Confusion Matrix

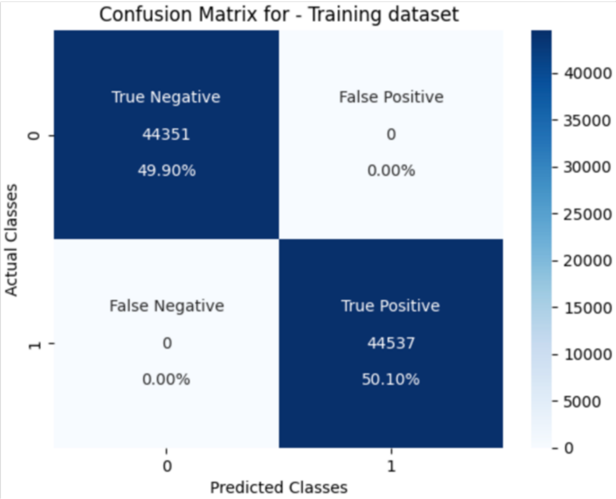


Confusion Matrix for - Training dataset

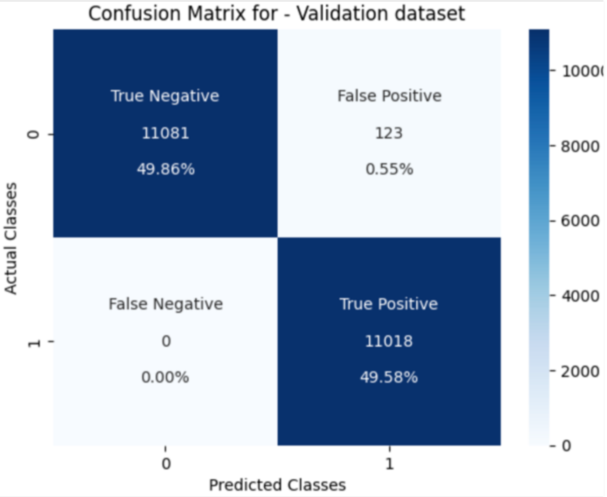


Confusion Matrix for - Validation dataset

Experiment 3 Confusion Matrix



Confusion Matrix for - Training dataset



Confusion Matrix for - Validation dataset

- In the 1$^{st}$ experiment, the AUROC performance score of 0.9752 on the testing set is slightly below the score of 0.9883 and 0.9905 achieved on the validation and training set, suggesting that the model is exhibiting a small degree of overfitting.

- Furthermore, the relatively high False Positive errors of 595 in the training set and 188 in the validation set from the first experiment stand out when compared to the False Positive errors from the third experiment with 0 for training and 123 for validation.
- However, the AUROC score from experiment 3 suggests that the model is significantly overfitting in comparison to the AUROC score of experiment 1 informing that the model has not generalised enough on the new unseen data.

Thus, when assessing the performance metrics of all the models in the experiments conducted over consecutive weeks, the Logistic Regression Classifier model with tuned hyperparameters from the 4th experiment in week 1 stands out. It demonstrates better performance compared to the others, achieving an AUROC score of 0.9881 on cross-validation data and 0.9765 on the testing dataset, which surpasses the performance of models in other experiments.

Additionally, it suggests that the model is generalised enough on the unseen data to predict prospective players and could be deployed in the operational environment.

Business Impact and Benefits:

The Logistic Regression Classifier model (of experiment 4 of week 1) outperforms other models with a performance score of 0.9881 on training, and validation data and 0.9765 on the testing data and contributes by providing relatively highly accurate predictions of whether a college basketball player will be drafted into the NBA. It helps in solving the challenge of making informed draft decisions and exploiting the opportunity to select promising players. The improved accuracy ensures that the right talent is selected for NBA teams.

The quantifiable improvement in model performance, as demonstrated by the performance scores, indicates that it adds substantial value to the draft selection process. The potential value generated includes better team performance with the selection of more suitable players, increased fan engagement, and potentially higher revenues for NBA teams.

Below are a few risks from a business point of view.

- Overfitting: The model may not work well with new data.
- Data Quality: Inaccurate or biased data may affect predictions.
- Privacy Concerns: Legal and ethical issues related to player data.

Also, subsequent are some recommendations for business.

- Regular Model Updates.
- Improve Data Quality.
- Ensure Privacy Compliance.
- Integrate Model into Decision-Making.
- Collaborate with Experts.

Data Privacy and Ethical Concerns:

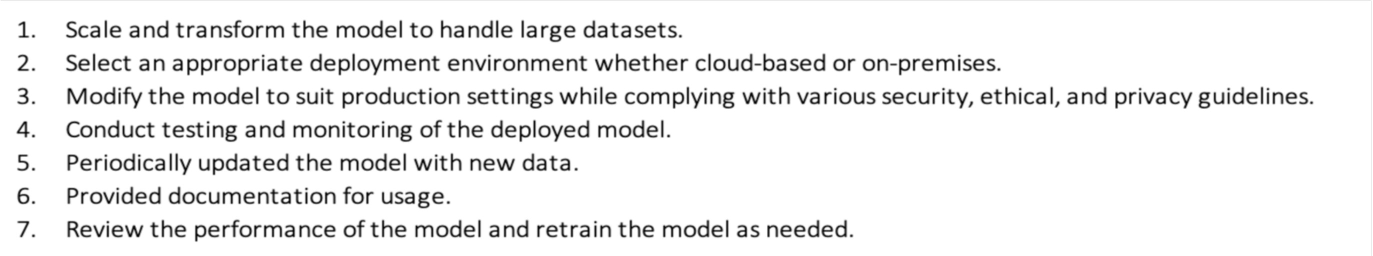
- Data privacy was carefully considered in this project. Since the dataset was obtained from a university portal as a student, there were no concerns regarding copyright or privacy issues. However, the dataset contained unique identifiers like player IDs and numbers, which were removed to ensure the privacy of individuals associated with the data.

- Ethical concerns in this project primarily revolve around fairness and bias. When using data for predicting NBA draft selections, it's crucial to ensure that the model doesn't preserve biases related to race, ethnicity, or other factors.

- Ethical data collection and preprocessing techniques, like removing personally identifiable information, were employed to mitigate these concerns. Data was used solely for the purpose of predicting NBA draft selections, and all efforts were made to ensure fairness, transparency, and privacy in the modeling process.

# 7. Deployment

Effective implementation of a binary classification algorithm in a production setting, it's recommended to follow these steps.



1. Scale and transform the model to handle large datasets.
2. Select an appropriate deployment environment whether cloud-based or on-premises.
3. Modify the model to suit production settings while complying with various security, ethical, and privacy guidelines.
4. Conduct testing and monitoring of the deployed model.
5. Periodically updated the model with new data.
6. Provided documentation for usage.
7. Review the performance of the model and retrain the model as needed.

Challenges and considerations in deployment might include version control for models, managing dependencies, ensuring low latency for predictions, and handling unexpected errors or downtime.

## 8. Conclusion

In conclusion, this project successfully achieved its goals by developing a predictive model for NBA draft selections based on college basketball player statistics. The Logistic Regression Classifier model from the 4th experiment in week 1 emerged as the top performer, with impressive AUROC scores on both cross-validation and testing datasets.

The project's impact on business use cases is significant. It provides NBA teams with a data-driven tool to make more informed draft selections, improving their chances of recruiting top talent. It also aids sports analysts and fans in predicting which college players are likely to transition to the professional NBA league.

In the future, I would like to explore other techniques like under sampling, and experimenting with other algorithms, refining the model further by considering additional features, for optimizing the predictions. Overall, this project has demonstrated the value of machine learning in the context of NBA draft selections.

# 9. References

So, A., & Tith, R. (2023, August 8). *Machine Learning Engineering* [Lab]. https://docs.google.com/presentation/d/1zNzRPkJua_qJn5ZM0Eku5BBrk3EX12jI0-9BUeyADG0/edit#slide=id.gb753713d85_0_643

So, A., & Tith, R. (2023, August 8). *Custom Package* [Lecture]. https://canvas.uts.edu.au/courses/28052/pages/2-dot-0-module-2-overview?module_item_id=1465624

So, A., & Tith, R. (2023, August 8). *Custom Package* [Lab]. https://colab.research.google.com/drive/1HzN0kcWIFmApxJFDYQoiBZVurVZd9HJk#scrollTo=XH7B4qftdese

Salvatierra, J. (n.d.). *How to use pyenv to manage Python versions*. Teclado. Retrieved August 21, 2023, from https://blog.teclado.com/how-to-use-pyenv-manage-python-versions/

Rigoulet, X. (n.d.). *Your Guide to pyenv*. LearnPython. Retrieved August 21, 2023, from https://learnpython.com/blog/change-python-versions/

Krishna, A. (n.d.). *Your Guide to pyenvHow to Build and Publish Python Packages With Poetry*. Freecodecamp. Retrieved August 25, 2023, from https://www.freecodecamp.org/news/how-to-build-and-publish-python-packages-with-poetry/

(n.d.). *Packaging Python Projects*. PyPA. Retrieved August 27, 2023, from https://packaging.python.org/tutorials/packaging-projects/

(n.d.). *Create and Access a Python Package*. Tutorialspoint. Retrieved August 26, 2023, from https://www.tutorialspoint.com/create-and-access-a-python-package

Brownlee, J. (2021, May 1). *How to Develop an AdaBoost Ensemble in Python*. Machine Learning Mastery. Retrieved August 27, 2023, from https://machinelearningmastery.com/adaboost-ensemble-in-python/

Polzer, D. (2022, August 4). *AdaBoost, Step-by-Step*. Towards Data Science. Retrieved August 29, 2023, from https://towardsdatascience.com/adaboost-in-7-simple-steps-a89dc41ec4