Assignment 1

Kaggle Competition

Ivan Cheung
Student ID: 13975420

07/09/2023

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

Table of Contents

1. Business Understanding

a. Business Use Cases

The business of online betting is based on anticipating the likelihood of events and controlling the payout of dividends to ensure that the business returns an overall profit against the aggregate. Poor predictive analytics is likely to negatively impact the business revenue and put the business at risk of insolvency.

Being drafted to an NBA (National Basketball Association) professional basketball league team is a highly competitive process, with a basis on merit and individual past performance. Given the high correlation between a player being drafted and their collegiate performance on the court there is an opportunity to apply predictive analysis to draft potentials based on on-court performance statistics.

NBA teams can draft players from college basketball, NBA G League and overseas leagues. Individual player information is highly detailed and includes information on games played, minutes on court, passes, shots, on-target, misses and intercepts. The volume and variety of information that can inform draft potential makes human interpretation difficult and unintuitive. This analysis is most suited to machine learning processes which is able to ingest and evaluate the large datasets with ease.

The prediction of which players will be drafted and ability to prescribe predictive odds of being drafted will allow the business to allocate matching dividends to each player which will favor a net positive return to the business. This will enable the business to enter a new market of betting, bringing in new business and grow overall business revenue and profits.

b. Key Objectives

This project aims to develop a machine learning algorithm that can interpret information on individual player performances to determine the likelihood that player will be drafted into an NBA team (with no consideration to draft pick rankings). The algorithm will be expected to interpret statistical information as well as descriptive information such as the player team and their league conference.

The outputs from this project will support the business analyst and betting operations in prescribing dividend payout odds to each player. For this project to be successful the prediction algorithm must be highly confident in it's estimations. A poor prediction score can result in higher than expected dividend payout for the business, reducing the return on investment opportunity and potentially placing the business into capital loss. Lack of confidence in predicting players that will not be drafted is likely to also result in lower dividend odds from betting operations. This uncertainty will reduce risk vs reward opportunity for customers, resulting in lower customer base. The consequence will be a diminished business potential on this market opportunity.

To address the requirements of business stakeholders, the project will tackle the prediction of player drafts using machine learning classification models. These models are designed to draw some conclusion based on a set of given inputs. In this project, the model will predict the likelihood of a player being drafted based on the player performance statistics over the last season.

The validity of the machine learning algorithm models at each iteration will be evaluated based on the AUROC (Area Under Receiver Operating Characteristic) performance metric. AUROC is a best-practice approach to evaluating classification models and an iterative approach of constant improvement will be taken to identify the most robust model, based on AUROC performance scores.

2. Data Understanding

Individual player performance statistics will be used as the input data that will be used by the classification models. This data has been supplied by the various collegiate conferences that make up the eligible player pools for the NBA draft. The data has been collected by the project team as a one-off bulk data capture in an excel format. This data has been provided as-is from the source and does contain some data inconsistencies and missing information. The project team has taken this into consideration when working with the data provided, as outlined in section 4.

a. Input Data Features

The individual player performance statistics available include:

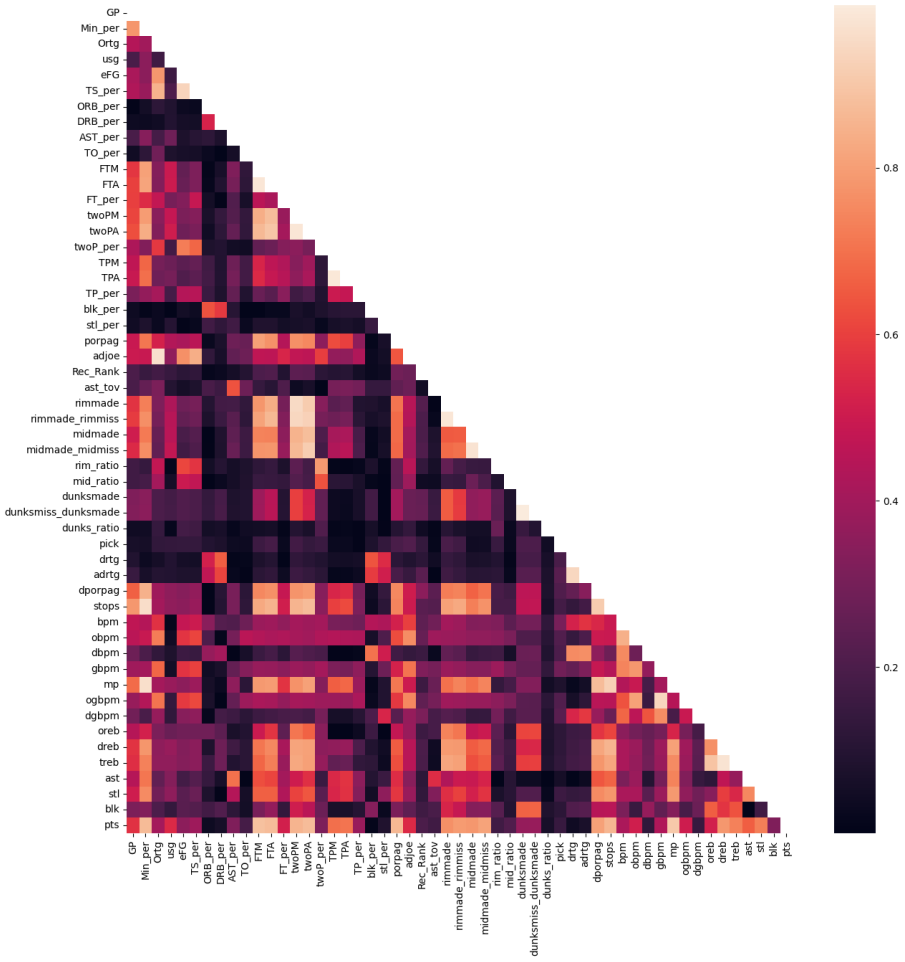
Feature Name	Description
Player Information	
Unique identifier of player	
Name of team	
Name of conference	
Student's year of study	
Height of student	
Player's number	
Player game performance	
Games played	
Player's percentage of available team minutes played	
Usage percentage is an estimate of the percentage of team plays used by a player while he was on the floor.	
Turnover percentage, an estimate of turnovers per 100 plays.	
Ratio Assists against Turnovers	
Estimate the player's contribution in points above league average per 100 possessions played (BPM)	
BPM 2.0, alternative calculation to bpm.	
Minutes played.	
Player's total points	
Player offensive performance	
Offensive rating, points produced per 100 possessions.	
Effective field goal percentage. This statistic adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal	
True shooting percentage. A measure of shooting efficiency that takes into account field goals, 3-point field goals, and free throws.	

Feature Name	Description
Offensive rebound percentage, an estimate of the percentage of available offensive rebounds a player grabbed while he was on the floor.	
Assist percentage, estimate of the percentage of teammate field goals a player assisted while he was on the floor.	
Free throws, attempts made, and percentage of attempt conversions.	
2-Point field goals, attempts made, and percentage of attempt conversions.	
3-Point field goals, attempts made, and percentage of attempt conversions.	
Points Over Replacement Per Adjusted Game	
Adjusted offensive efficiency, an estimate of offensive efficiency (points scored per 100 possessions) against an average defense.	
Shots made at or near the rim, misses and attempt conversions.	
Ratio of on goal shots vs shots missed	
Dunks attempts made, and percentage of attempt conversions.	
Offensive BPM	
Offensive rebounds	
Player defensive performance	
Defensive rebound percentage, an estimate of the percentage of available defensive rebounds a player grabbed while he was on the floor.	
Block percentage, an estimate of the percentage of opponent two-point field goal attempts blocked by the player while he was on the floor.	
Steal percentage, an estimate of the percentage of opponent possessions that end with a steal by the player while he was on the floor.	
Defensive rating, points allowed per 100 possessions.	
Individual defensive stops.	
Defensive BPM	
Defensive rebounds	

From available player statistics, the on court performances (offensive, defensive and overall) is likely to play a significant factor in draft potential. However, player information such as their year of study may also have an influence. For this reason all features from the data set were included with few exceptions. The features that were excluded have been explained in the following section.

b. Understanding the data

Exploratory data analysis steps were carried out to gain an insight into the quality of the dataset and the correlation between features. A correlation heatmap revealed that features within the dataset were not wholly independent:



This correlation between features is not unexpected, particularly in high correlation between defense features and offense features, as the underlying statistic that informs these features are often duplicated. While there were plans to address highly correlated features prior to training the machine learning models, this work did not end up in the final models.

Data contains empty values

A count of blank values within the dataset reviewed that there were a number of columns with missing values (30 columns in total), and 190,205 out of 3,399,619 cells were blank (5.6% of the training data). These missing values will need to be addressed prior to the modelling.

Dropping invalid and inconsequential columns

Several columns were discarded from the source data. The columns dropped and their reasons are described below:

Feature Name	Reason for exclusion
ftr	This feature did not contain any contextual information in the data dictionary. Given the richness of data already available, this unknown column was discarded.
ht	Player height information was corrupt in the source data. There was no ability to restore this information. In future iterations, access to this feature is likely to improve model quality.
num	The player number on court is unlikely to influence draft pick potential into an NBA team and therefore was excluded.
type	This metadata field adds no significant value to the model.
pfr	This feature did not contain any contextual information in the data dictionary. Given the richness of data already available, this unknown column was discarded.

Cleaning player year information

```
yr
Jr      14923
Fr      14906
So      13252
Sr      12711
0           5
57.1        1
42.9        1
Name: count, dtype: int64
```

The 'yr' feature identifies the player's year in college. This information is highly relevant as the NBA is most likely to pick a player in their senior year.

A small number of players had invalid values, this issue is taken into account during the data cleaning steps.

Imbalanced target

The total players that were drafted in the training data were 536 (out of 56091 total players) with an average of 45 players drafted each season year. With less than 1% of players being drafted, this dataset is highly imbalanced. Consideration for this has been addressed in the feature engineering steps.

3. Data Preparation

a. Data cleaning

As identified in the Exploratory Data Analysis, the following steps were carried out to clean the source data prior to pre-processing:

1) remove columns: columns previously identified as invalid or not significant were dropped.

2) apply placeholder values for numerical features, all blank values were replaced with a '0' placeholder.

3) In the categorical features, only 'player year' had invalid values not matching valid options. Each invalid value was replaced with a placeholder value of 'Other'.

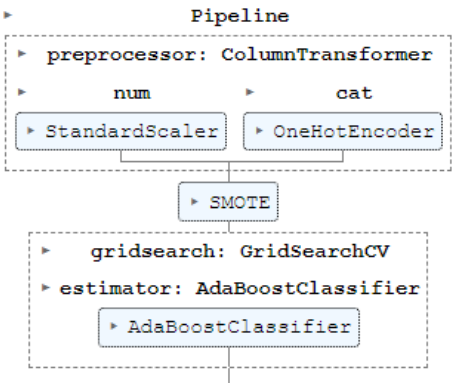
b. Preprocessing and feature processing

To improve the performance of the model, a standard scalar was applied to numerical features. The inclusion of categorical features required the application of OneHotEncoding to be able to be processed by the model. Experiments in fitting the test datasets revealed some category features contained additional values that were not present in the train set. Therefore, the OHE step was set to ignore and discard unknown values.

Due to the imbalanced nature of the target, the Synthetic Minority Oversampling Technique (SMOTE) algorithm was applied to leverage the benefits of oversampling in counteracting the imbalance.

a. Processing pipeline

A pipeline was developed, incorporating both modelling and preprocessing steps to support experiments by the project team. An overview of this pipeline is below:

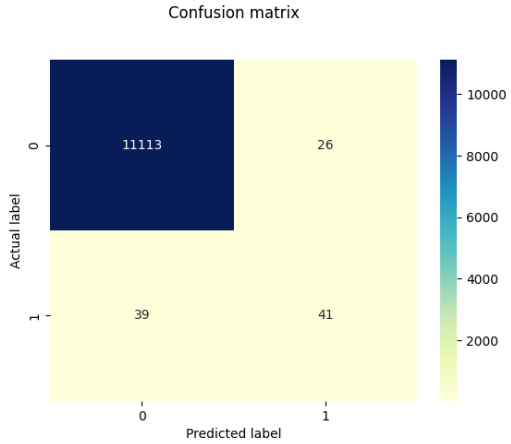


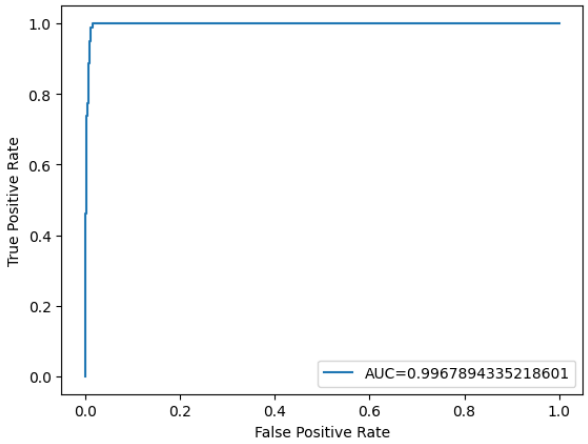
4. Modeling

a. Approach 1

Model	Logistic Classification
Hyperparameters	max iterations: 1000
Training/Test split	80/20 split
Pipeline:	
- Numerical Features, scaled:	✓
- Numerical Features, replaced blanks:	✓
- Categorical Features, OHE:	*n/a. category features were dropped.*
- Categorical Features, cleaned invalid values:	*n/a.*
- SMOTE	*n/a.*

The first approach utilised a simplistic algorithm, the logistic classification model. This model is ideal as a first experiment due to it's simplicity and short processing time. The output values from this experiment were:

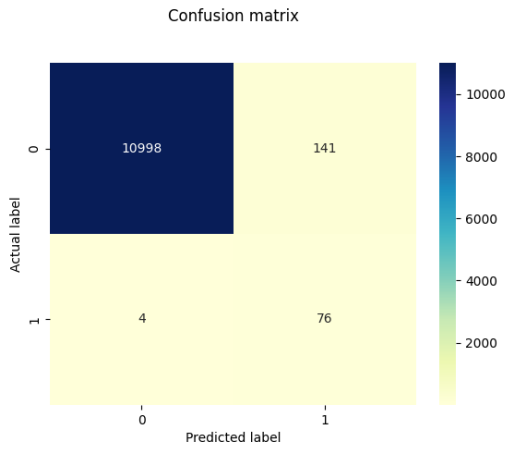


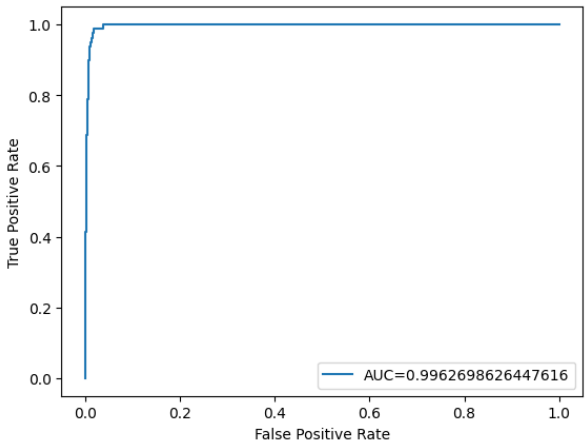


b. Approach 2

Model	Logistic Classification
Hyperparameters	max iterations: 1000
Training/Test split	80/20 split
Pipeline:	
- Numerical Features, scaled:	✓
- Numerical Features, replaced blanks:	✓
- Categorical Features, OHE:	✓
- Categorical Features, cleaned invalid values:	*n/a.*
- SMOTE	*n/a.*

In the next experiment, categorical features were restored back to the training data, applying OneHotEncoding to convert category values into numerical features.





c. Approach 3

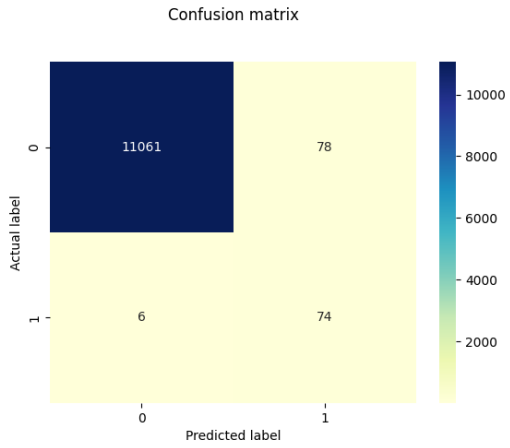
Base estimator	Decision Tree Classification
Meta estimator	Adaptive Boosting
Meta estimator Hyperparameters	learning rate: 1.0 no. base estimators: 100
Training/Test split	80/20 split
Pipeline:	
- Numerical Features, scaled:	✓
- Numerical Features, replaced blanks:	✓
- Categorical Features, OHE:	✓
- Categorical Features, cleaned invalid values:	*n/a.*
- SMOTE	✓

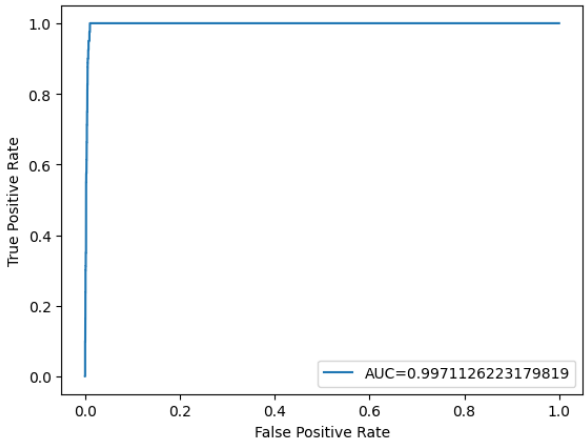
Due to the highly imbalanced nature of the target, two approaches were taken to address this.

1) The application of up sampling, through SMOTE, and
2) Switching to the iterative learning technique of Adaptive Boosting (AdaBoost)

Both methods have been shown to improve model output against imbalanced classification problems. The switch from a logistic classifier to a Decision Tree Classification (DCT) necessary to support the AdaBoost algorithm, as DCT performs better under this method vs a logistic regression.

Experiment results:





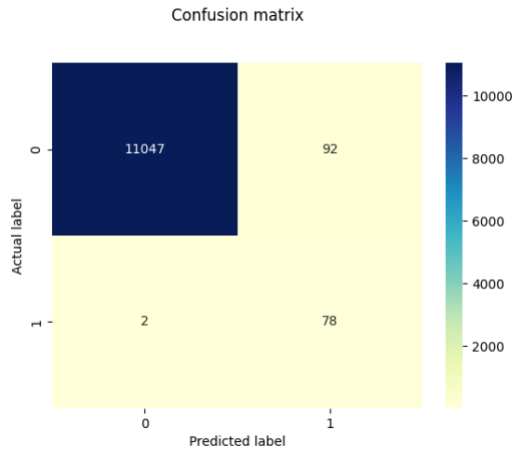
d. Approach 4

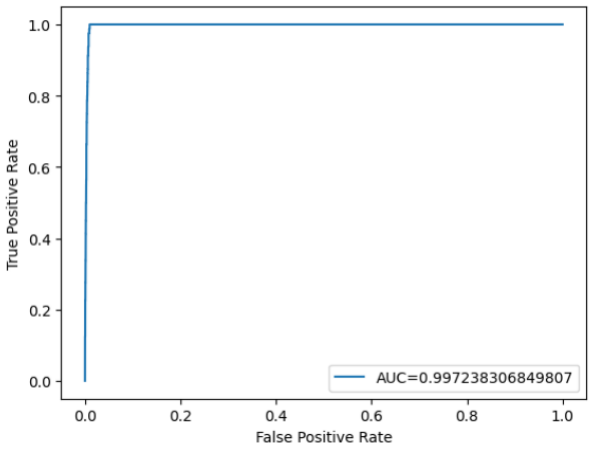
Base estimator	Decision Tree Classification
Meta estimator	Adaptive Boosting
Model tests / selection	GridSearchCV
Model selection	Number folds: 2 Scoring metric: auroc
Hyperparameter tests	learning rate: [0.5, 1.0, 2.0] no. base estimators: [100, 200]
Training/Test split	80/20 split
Pipeline:	
- Numerical Features, scaled:	✓
- Numerical Features, replaced blanks:	✓
- Categorical Features, OHE:	✓
- Categorical Features, cleaned invalid values:	✓
- SMOTE	✓

In this experiment, a hyperparameter testing algorithm was applied on top of the modelling pipeline, using the GridSearchCV library. This test allowed rapid evaluation of a number of different hyperparameter values, with the best model identified by the AUROC metric. The result of the model tests were:

	params	rank_test_score	mean_test_score	std_test_score
1	{'learning_rate': 0.5, 'n_estimators': 200}	1	0.999610	0.000111
3	{'learning_rate': 1, 'n_estimators': 200}	2	0.999551	0.000038
2	{'learning_rate': 1, 'n_estimators': 100}	3	0.999542	0.000099
0	{'learning_rate': 0.5, 'n_estimators': 100}	4	0.999488	0.000120
4	{'learning_rate': 2, 'n_estimators': 100}	5	0.995200	0.000809
5	{'learning_rate': 2, 'n_estimators': 200}	6	0.995048	0.000789

This experiment identified that the best hyperparameters were **learning rate: 0.5, no. estimators: 200**. The results of the best model were:

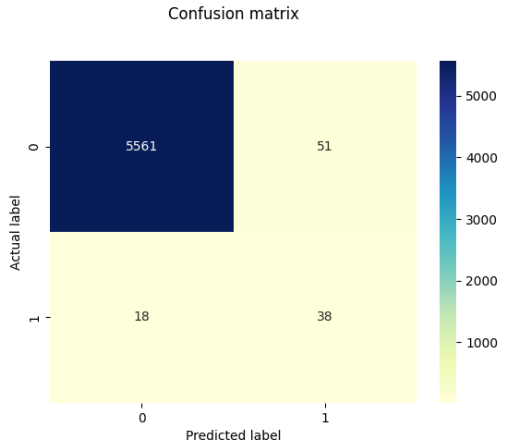


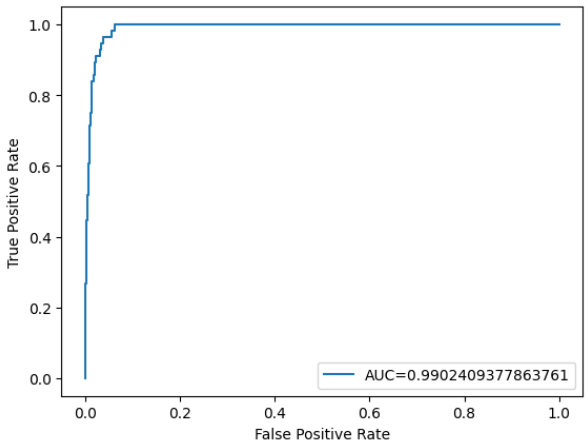


e. Approach 5

Base estimator	Decision Tree Classification
Meta estimator	Adaptive Boosting
Hyperparameters	learning rate: 0.5 no. base estimators: 200
Training/Test split	80/20 split
Data Processing:	
- seasons fitted	2020, 2019, 2018, 2017, 2016

A final experiment was carried out to evaluate if training data from more recent years would outperform historic data (2015 and older). The source data was segmented prior to performing standard preprocessing and feature engineering steps. The results of this experiment were:





5. Evaluation

a. Evaluation Metrics

For each experiment, a confusion matrix was generated, predicting against the 20% test data. However, the primary evaluation metric used was the AUROC (Area under the receiver operating characteristic) score. The AUROC score priorities the a high true positive rate and a low false positive rate, with no consideration for the failure type (false positive or false negative). This is a balanced approach to evaluating the predictive performance of a classification model.

AUROC is an ideal metric to meet the needs of the business, as neither failure type has higher impact than the other.

1) False positive: The dividend odds for this player will be lower than the business can support. This may result in lower customer engagement and a reduction of revenue.

2) False negative: The dividend odds for this player will be higher than the business can support. This will result in lower return on investment for the business for this market, a reduction in profitability.

b. Results and Analysis

The evaluation scores for each experiment were:

Experiment	Accuracy	Precision	Recall	F1-score	AUROC
Approach 1	0.99421	0.61194	0.51250	0.55782	0.99421
Approach 2	0.98708	0.35023	0.95000	0.51178	0.99627
Approach 3	0.99251	0.48684	0.92500	0.63793	0.99711
Approach 4	0.99162	0.45882	0.97500	0.62400	0.99724
Approach 5	0.98783	0.42697	0.67857	0.52414	0.99024

The high AUROC scores for experiments support the use of machine learning algorithms as a valid approach to predicting player drafts in the NBA. While the first experiment showed promise with a high accuracy and precision, the propensity for this model to predict false negatives warranted further experiments. Approach 5 is the worst performing model, this revealed that historic drafts are still relevant to modern draft methodology and is worth keeping in the model training. With such high base AUROC scores, further fine-tuning will start to offer diminishing returns. However, the cost of refinement may still offer good return on investment for the business over the longer-term outlook. This project recommends moving forward with the 4th approach as an initial deployment.

c. Business Impact and Benefits

The final model (Approach 4) is recommended for deployment to the business in a trial capacity. The predictions from this model will allow the business to enter an untapped market on NBA draft picks. The model is expected to provide high quality outputs in support of the betting operations team, to assigning dividend odds to individual player draft picks not previously possible. From this opportunity, the business will have the ability to grow revenue, improve user engagement from existing customers and potentially target a new cohort of customers.

d. Data Privacy and Ethical Concerns

The use of this model to assign betting odds to individual players use publicly available data, provided by each collegiate league and by the NBA. The predictive output from this model is intended only for internal use to support business operations. There is the potential for misuse of the model outputs from third parties, looking to identify stars or for draft candidates. To avoid liability in misuse of our information, the project team suggest clear messaging is displayed to customers viewing or making decisions based on output from the algorithm.

6. Deployment

The trained prediction model has been saved as a Python joblib file and can be reinstated in a python environment using joblib without needing to be retrained. When data from a new season becomes available, the data can be feed directly into the model pipeline without additional preprocessing.

A sample of this process is available below:

```
from joblib import load
import pandas as pd

# load data and model
df = pd.read_csv(<new data>)
model = load(<saved joblib file>)

# run preprocessing
df = remove_unwanted_cols(df)
df = fillna(df)
df = clean_yr(df)

# run prediction
pred = model.predict_proba(df)
```

Further work is possible to improve integration of this model into business processes, including:

1) development of a workflow to communicate model outputs directly to the operations team

2) development of a data pipeline to automatically extract new data as soon as the season ends.

3) The model currently expects a full season worth of data to make predictions. If it is possible to obtain live statistics of players as games happen, the prediction of draft odds during the season may represent a greater opportunity for the business to secure more customer engagement and stronger market capitalisation.

7. Conclusion

The results of the experiments provide strong evidence to support the use of machine learning algorithms to predict NBA draft picks based on performances of players in previous seasons. The target evaluation for **the best model returned a score of 0.99724** (out of 1). The results show that predictions from the model will give us high confidence in our ability to match betting dividend odds and capitalise on our profit margins for this market. The score also gives confidence that the model will be able to satisfy the business needs in minimising risk to the business through capital loss on poor dividend setting.

The project team recommends commencing a trial run of this model to gain real world insights into the success potential of applying machine learning to this market opportunity. The ongoing success of this trial may also highlight other opportunities for machine learning to be applied to the business, either improving existing operations or to enter new markets.

Continued effort to further improve the model will benefit the business greatly in the long term by allowing operations to maximise profit margins through higher confidences in predictions. An investment in technical resources will allow the project team to conduct more intensive model training approaches and experiment on more complex algorithm that may improve performance.