Assignment 1

Kaggle Competition

Daniel Alexander
Student ID: 24591214
05/09/2023

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

Table of Contents

1. Executive Summary **2**
2. Business Understanding **3**
a. Business Use Cases 3
3. Data Understanding **4**
4. Data Preparation **5**
5. Modeling **6**
a. Approach 1 6
b. Approach 2 6
c. Approach 3 6
6. Evaluation **8**
a. Evaluation Metrics 8
b. Results and Analysis 8
c. Business Impact and Benefits 8
d. Data Privacy and Ethical Concerns 9
7. Deployment **10**
8. Conclusion **11**
9. References **12**

1. Executive Summary

XYZ, a sporting apparel company is competing hard with our competitor ABC for share market especially in the Basketball category. This project aimed to develop a predictive model to predict the likelihood of a young player to be drafted for the NBA. The success of this project will affect our company's profitability through better business decisions which eventually will lead us to gain share market without sacrificing our profit margin. We hope out of this project, our company will have a competitive advantage compared to our competitors from the talent acquisitions point of view.

2. Business Understanding

a. Business Use Cases

NBA is one of the most prestigious basketball leagues in the world, and an important part of it is the event called NBA Draft. The NBA Draft is an annual event in which NBA teams select eligible players to join the league. The draft can make or break the future of a team, a poor choice can set a team back for years and a good pick could lead to future Michael Jordan.
Our company is one of the leading sporting apparel companies continuously trying to find a way to increase our profit. One of the ways to increase our profit is to invest early in highly potential young players because the cost of investment will be significantly cheaper and hence higher profit margin. This endeavor requires a systematic way for our company to screen the pool of young players so that we can increase our chance to make better player picking. Predicting the NBA draft has never been an easy task, and if not addressed systematically, instead of increasing profitability, we might end up inducing more loss.

b. Key Objectives

The goal of this project is to assist the business to build a systematic way to make investment decisions relating to the NBA draft. The stakeholders would be the shareholders of the company who are directly impacted by the profitability of the business will require realization of increase in profit, and the end user which is the talent acquisition team who will run the model as part of their business as usual and will make decision guided by the model which require ease of use and interpretation.

Our department, the data science team will utilize machine learning algorithms to build a predictive model to predict the likelihood of a potential player to be drafted for the NBA which arguably will help the company to increase its profitability. On top of that, we are going to prepare and produce a model in a way such that a non-technical end user can use the model as a part of their business process.

create a prediction model that estimates the likelihood of a young player's chances of being selected in the NBA based on their current season stats. The Area Under ROC (AUROC) score is the project's primary performance indicator. Ultimately provide insights for the talent acquisition team to make informed decisions before the NBA drafting.

3. Data Understanding

We have sourced our data from Kaggle https://www.kaggle.com/competitions/advmla-2023-spring/data. Kaggle is an online platform and community that is primarily focused on data science, machine learning, and artificial intelligence. It is one of the largest and most popular platforms for data science competitions, collaboration, and learning.
There are 3 different datasets that we used, the training set for experimentation, the test set for Kaggle evaluation, and the supplemental information about the data. All the datasets are stored in csv format. The quality of the data however is not perfect, there are plenty of missing observations and irrelevant values stored in some of the columns.
Below is the list of features present in the dataset:

1	team	Name of team
2	conf	Name of conference
3	GP	Games played
4	Min_per	Player's percentage of available team minutes played
5	ORtg	ORtg - Offensive Rating (available since the 1977-78 season in the NBA); for players it is points produced per 100 posessions, while for teams it is points scored per 100 possessions. This rating was developed by Dean Oliver, author of Basketball on Paper. Please see the article Calculating Individual Offensive and Defensive Ratings for more information.
6	usg	Usg% - Usage Percentage (available since the 1977-78 season in the NBA); the formula is 100 * ((FGA + 0.44 * FTA + TOV) * (Tm MP / 5)) / (MP * (Tm FGA + 0.44 * Tm FTA + Tm TOV)). Usage percentage is an estimate of the percentage of team plays used by a player while he was on the floor.
7	eFG	eFG% - Effective Field Goal Percentage; the formula is (FG + 0.5 * 3P) / FGA. This statistic adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal. For example, suppose Player A goes 4 for 10 with 2 threes, while Player B goes 5 for 10 with 0 threes. Each player would have 10 points from field goals, and thus would have the same effective field goal percentage (50%).
8	TS_per	TS% - True Shooting Percentage; the formula is PTS / (2 * TSA). True shooting percentage is a measure of shooting efficiency that takes into account field goals, 3-point field goals, and free throws.
9	ORB_per	ORB% - Offensive Rebound Percentage (available since the 1970-71 season in the NBA); the formula is 100 * (ORB * (Tm MP / 5)) / (MP * (Tm ORB + Opp DRB)). Offensive rebound percentage is an estimate of the percentage of available offensive rebounds a player grabbed while he was on the floor.

10	DRB_per	DRB% - Defensive Rebound Percentage (available since the 1970-71 season in the NBA); the formula is 100 * (DRB * (Tm MP / 5)) / (MP * (Tm DRB + Opp ORB)). Defensive rebound percentage is an estimate of the percentage of available defensive rebounds a player grabbed while he was on the floor.
11	AST_per	AST% - Assist Percentage (available since the 1964-65 season in the NBA); the formula is 100 * AST / (((MP / (Tm MP / 5)) * Tm FG) - FG). Assist percentage is an estimate of the percentage of teammate field goals a player assisted while he was on the floor.
12	TO_per	TOV% - Turnover Percentage (available since the 1977-78 season in the NBA); the formula is 100 * TOV / (FGA + 0.44 * FTA + TOV). Turnover percentage is an estimate of turnovers per 100 plays.
13	FTM	Free Throws
14	FTA	Free Throw Attempts
15	FT_per	Free Throw Percentage; the formula is FTM / FTA.
16	twoPM	2P - 2-Point Field Goals
17	twoPA	2PA - 2-Point Field Goal Attempts
18	twoP_per	2P% - 2-Point Field Goal Percentage; the formula is 2P / 2PA.
19	TPM	3P - 3-Point Field Goals (available since the 1979-80 season in the NBA)
20	TPA	3PA - 3-Point Field Goal Attempts (available since the 1979-80 season in the NBA)
21	TP_per	3P% - 3-Point Field Goal Percentage (available since the 1979-80 season in the NBA); the formula is 3P / 3PA.
22	blk_per	BLK% - Block Percentage (available since the 1973-74 season in the NBA); the formula is 100 * (BLK * (Tm MP / 5)) / (MP * (Opp FGA - Opp 3PA)). Block percentage is an estimate of the percentage of opponent two-point field goal attempts blocked by the player while he was on the floor.
23	stl_per	STL% - Steal Percentage (available since the 1973-74 season in the NBA); the formula is 100 * (STL * (Tm MP / 5)) / (MP * Opp Poss). Steal Percentage is an estimate of the percentage of opponent possessions that end with a steal by the player while he was on the floor.
27	ftr	
28	yr	Student's year of study: `Fr` for freshmen, `So` for sophomores, `Jr` for juniors, `Sr` for seniors

29	ht	Height of student
30	num	Player's number
31	porpag	Points Over Replacement Per Adjusted Game
32	adjoe	AdjO – Adjusted offensive efficiency – An estimate of the offensive efficiency (points scored per 100 possessions) a team would have against the average D-I defense.
33	pfr	
34	year	Season's year
35	type	Type of metrics displayed: `All` for all types, `C` for conference`, `NC` for non-conference, `PC` for pre-conference tour, `R` for regular season, `P` for post-season, `T` for NCAA
36	Rec_Rank	Recruiting rank i.e. what the player was ranked as a recruit coming out of high school
37	ast_tov	Ratio Assists against Turnovers
38	rimmade	Shots made at or near the rim
39	rimmade_rimmiss	Sum of Shots made at or near the rim and Shots missed
40	midmade	Two point shots that were not made at or near the rim
41	midmade_midmiss	Sum of Two point shots that were not made at or near the rim and Shots missed
42	rim_ratio	Ratio between Shots made at or near the rim against Shots missed
43	mid_ratio	Ratio between Two point shots that were not made at or near the rim and Shots missed
44	dunksmade	Dunks made
45	dunksmiss_dunksmade	Sum of Dunks made and Dunks missed
46	dunks_ratio	Ratio between Dunks made and Dunks missed
47	pick	Order of NBA draft
48	drtg	DRtg - Defensive Rating (available since the 1973-74 season in the NBA); for players and teams it is points allowed per 100 posessions. This rating was developed by Dean Oliver, author of Basketball on Paper. Please see the article Calculating Individual Offensive and Defensive Ratings for more information.
49	adrtg	Adjusted DRtg
50	dporpag	Asdjusted porpag

54	stops	Stops - Stops; Dean Oliver's measure of individual defensive stops. Please see the article Calculating Individual Offensive and Defensive Ratings for more information.
55	bpm	BPM - Estimate the player's contribution in points above league average per 100 possessions played
56	obpm	Offensive BPM
57	dbpm	Defensive BPM
58	gbpm	BPM 2.0
59	mp	MP - Minutes Played (available since the 1951-52 season)
60	ogbpm	Offensive BPM 2.0
61	dgbpm	Defensive BPM 2.0
62	oreb	ORB - Offensive Rebounds (available since the 1973-74 season in the NBA)
63	dreb	DRB - Defensive Rebounds (available since the 1973-74 season in the NBA)
64	treb	TRB - Total Rebounds (available since the 1950-51 season)
65	ast	AST - Assists
66	stl	STL - Steals (available since the 1973-74 season in the NBA)
67	blk	BLK - Blocks (available since the 1973-74 season in the NBA)
68	pts	PTS - Points
69	player_id	Unique identifier of player
70	drafted	Target - Was the player drafted at the end of the season

Due to the large number of features present in the dataset, instead of listing those which are significant for our project, we will list the insignificant ones instead. We can safely deem these features are insignificant without doing EDA due to logical reasoning.

- Team - Because team name does not affect the performance and electability of a player.
- Year - Because Season's year does not affect the performance and electability of a player.
- Pick - Because we would not have this feature available for the future unseen data, as it is there only because this is a past data where the drafting has been done.
- Player_id - As this is a unique ID which would not affect electability of a player, instead will introduce inaccuracies to our model.
- Num – Just like player_id, this feature is insignificant.

Exploratory Data Analysis

Below is the list of features from the training data and their proportion of missing value respectively. Features with proportion of missing value > 15% are highlighted in red, and we

avoid to remove observations with a missing value for those features since it will remove large proportion of the dataset.

Features	Proportion of missing value
yr	0.52%
ht	0.17%
num	8.36%
Rec_Rank	69.63%
ast_tov	7.47%
rimmade	10.84%
rimmade_rimmiss	10.84%
midmade	10.84%
midmade_midmiss	10.84%
rim_ratio	16.87%
mid_ratio	17.27%
dunksmade	10.84%
dunksmiss_dunksmade	10.84%
dunks_ratio	54.90%
pick	97.53%
drtg	0.08%
adrtg	0.08%
dporpag	0.08%
stops	0.08%
bpm	0.08%
obpm	0.08%
dbpm	0.08%
gbpm	0.08%
mp	0.07%
ogbpm	0.08%
dgbpm	0.08%
oreb	0.07%
dreb	0.07%
treb	0.07%
ast	0.07%
stl	0.07%
blk	0.07%
pts	0.07%

We will not do EDA for all features; we will only perform EDA for the features that we as the data team are unsure whether those features are significant and usable or not. Those features are conf, yr, and ht.

Conf	Drafted
A10	0%
ACC	5%
AE	0%
ASun	0%
Amer	1%
B10	3%
B12	4%
BE	3%
BSky	0%
BSth	0%
BW	0%
CAA	0%
CUSA	0%
GWC	0%
Horz	0%
Ind	0%
Ivy	0%
MAAC	0%
MAC	0%
MEAC	0%
MVC	0%
MWC	1%
NEC	0%
OVC	0%
P10	4%
P12	4%
Pat	0%
SB	0%
SC	0%
SEC	4%
SWAC	0%
Slnd	0%
Sum	0%
WAC	0%
WCC	1%
ind	0%

EDA for feature Conf

The figure on the left is the proportion of players who were drafted from specific conference. Looks like the distribution is not uniform, we assume that conf could be a significant feature based on our EDA.

yr	drafted
0	0%
42.9	0%
57.1	0%
Fr	1%
Jr	1%
So	1%
Sr	1%

EDA for the feature yr

There are illegitimate values in this feature and the distribution of drafted players are relatively uniform. We could safely assume this feature is insignificant.

ht	Drafted
0	0%
1-Jul	2%
1-Jun	0%
1-May	0%
10-Jun	2%
10-May	0%
11-Jun	3%
11-May	0%
2-Jul	1%
2-Jun	1%
2-May	0%
3-Jul	3%
3-Jun	1%
3-May	0%
4-Jul	0%
4-Jun	1%
4-May	0%
5-Apr	0%
5-Jul	0%
5-Jun	1%
5-May	0%
6'4	0%
6-Jul	0%
6-Jun	1%
6-May	0%
7-Jun	1%
7-May	0%
8-Jun	1%
8-May	0%
9-Jun	1%
9-May	0%
Apr-00	0%
Fr	0%
Jr	0%
Jul-00	4%
Jun-00	0%
So	0%

EDA for the feature ht

Most of the values here are illegitimate values in this feature and the distribution of drafted players are somewhat uniform. We could safely assume this feature is unusable.

4. Data Preparation

To prepare the data for modeling we had to deal with the missing observations, remove insignificant features from the dataset, perform resampling to overcome the imbalance dataset, apply feature engineering for some of the features, and apply transformations to help the model to predict better especially for the algorithms that are scale sensitive.

1. Removing insignificant features

The features that we deemed insignificant based on the EDA and our logical interpretation are firstly removed from our dataset.

- Team
- Year
- Pick
- Player_id
- Num
- Type as it only contains 1 value
- Yr as discussed in EDA part
- Ht as discussed in EDA part

2. Removing observations which has missing values

We deem that the features below are useful for us to reach our objective, however missing values are present in these features. Since the proportion of observations with missing values in these features are relatively small, then it is safe for us to remove the observations having missing values in these features.

- Rimmade
- Drtg
- Mp

However, we do not want to remove any observations from the test data since we can only use the complete test dataset due the target variable is on hide in Kaggle.

3. Filling in the value 0 to missing values.

Logically these features are significant for electability of one player, and the state of missing is physically applicable. Therefore, to capture the state of missingness we want to fill the null values with 0

- dunks_ratio

  caused by no succesful dunks were made

- mid_ratio

  caused by no midmade_midmiss were made

- rim_ratio

  caused by no succesful rimmshot were made

- Rec_Rank

  to capture the state of unranked

4. Filling in the average values to missing values.

On contrary of point number 3 where the state of missingness is physically possible, these variables should not be missing. Therefore, we are imputing the average values on the missing values for these features.

- ast_tov

Below is the summary of the dataset after we dealt with all the missing values, now we have 56 features with each 50006 non null observations:

	Feature	Non null data	Data Type
0	conf	50006	object
1	GP	50006	int64
2	Min_per	50006	float64
3	Ortg	50006	float64
4	usg	50006	float64
5	eFG	50006	float64
6	TS_per	50006	float64
7	ORB_per	50006	float64
8	DRB_per	50006	float64
9	AST_per	50006	float64
10	TO_per	50006	float64
11	FTM	50006	int64
12	FTA	50006	int64
13	FT_per	50006	float64
14	twoPM	50006	int64
15	twoPA	50006	int64
16	twoP_per	50006	float64
17	TPM	50006	int64
18	TPA	50006	int64
19	TP_per	50006	float64
20	blk_per	50006	float64
21	stl_per	50006	float64
22	ftr	50006	float64
23	porpag	50006	float64
24	adjoe	50006	float64
25	pfr	50006	float64
26	Rec_Rank	50006	float64
27	ast_tov	50006	float64
28	rimmade	50006	float64
29	rimmade_rimmiss	50006	float64
30	midmade	50006	float64
31	midmade_midmiss	50006	float64
32	rim_ratio	50006	float64
33	mid_ratio	50006	float64
34	dunksmade	50006	float64
35	dunksmiss_dunksmade	50006	float64
36	dunks_ratio	50006	float64

37	drtg	50006	float64
38	adrtg	50006	float64
39	dporpag	50006	float64
40	stops	50006	float64
41	bpm	50006	float64
42	obpm	50006	float64
43	dbpm	50006	float64
44	gbpm	50006	float64
45	mp	50006	float64
46	ogbpm	50006	float64
47	dgbpm	50006	float64
48	oreb	50006	float64
49	dreb	50006	float64
50	treb	50006	float64
51	ast	50006	float64
52	stl	50006	float64
53	blk	50006	float64
54	pts	50006	float64
55	drafted	50006	float64

5. Scaling the numerical features using standard scaler arguably to avoid scale sensitivity issues for some algorithms.

6. Applying one hot encoding for conf feature as the only categorical feature that we are using in this experiment.

7. Resampling the training dataset

To mitigate the issue on imbalanced training dataset we had to perform resampling, specifically under sampling to mitigate the imbalance dataset. The minority class represented are less than 2% of the raw dataset. We resample the dataset with a 70:30 ratio, hoping that by capturing the scarcity of the minority will produce a better performing model.

8. Apply similar treatment to the test dataset.

Since the test dataset comes separately and we do not want to remove any observations from it then we have to apply similar but not the same treatment as the training set. All of the removed features from train should be removed from the test as well.

Which then we impute the average values for the missing observations for these features as missing values here are physically impossible:

- Drtg
- Adrtg
- Dporpag
- Stops
- Bpm
- Obpm
- Dbpm
- Gbpm
- Ogbpm
- Dgbpm
- ast_tov

And fill the missing observations for these columns with 0 since we want to capture the state of missingness here:

- Rimmade
- Rimmade_rimmiss
- Midmade
- Midmade_midmiss
- Dunksmade
- Dunksmiss_dunksmade
- Rim_ratio
- Dunks_ratio
- Mid_ratio
- Rec_Rank

5. Modelling

a. Approach 1

The algorithm that we used for our first experiment was the

```
RandomForestClassifier(random_state=42)
```

as this is a versatile algorithm which performs well negating effects of outliers and is a good starting point to build our project. However, due to time constraint, we use the default hyperparameter as a starting point.

At the time we performed our first experiment we did not have the data definition for ftr and pfr, therefore we did not use those features for our modelling. The other data pre-processing and data engineering including resampling due to imbalance data are aligned with what was discussed in previous section.

On top of it, we further split the under sampled training dataset to train and validation dataset with 90:10 ratio.

b. Approach 2

In experiment B we spent most of our effort in hyperparameter tuning because the result from approach 1 was relatively well performing, albeit overfitting then we end up settling with

```
RandomForestClassifier (n_estimators=700, max_depth=5, min_samples_split=50)
```

All data processing, feature engineering, and resampling are the exact same as approach 1 including the splitting of training and validation with 90:10 ratio.

We tried to jump straight using grid search for our hyperparameter tuning, however it took hours to run and we ended up manually adjust the hyperparameters and stop at this sweet spot.

c. Approach 3

In experiment C we want to compare how other algorithm performs compared the model from experiment B. We decided to try adaptive booster since in general adaptive booster performs better than random forest. We used

```
AdaBoostClassifier(random_state=42,
                   n_estimators= 468,
                   learning_rate=1,
                   base_estimator=DecisionTreeClassifier (max_depth=7),
```

```
                    algorithm='SAMME')
```

In this experiment, we take ftr and pfr into our model, which make the data processing, feature engineering, and resampling fully aligned with the previous section. We also tried random search hoping that it would run quicker than grid search.

6. Evaluation

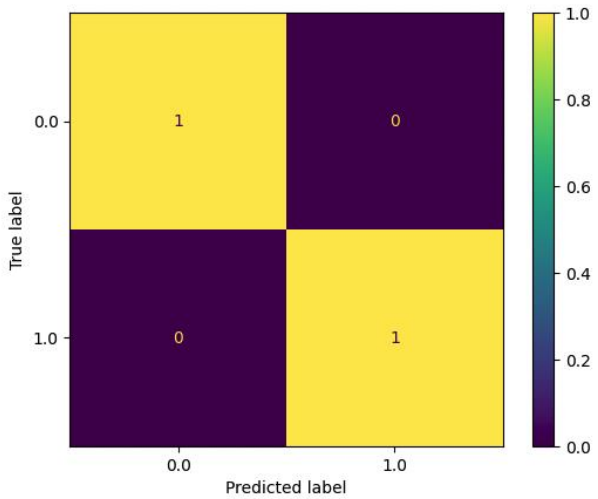
a. Evaluation Metrics

We are using the ROC AUC score because it is useful for comparison across models and rank them based on their ability to distinguish between the 2 classes considering that we run multiple experiments. We also use the recall score to supplement the ROC AUC score as recall focus on identifying the positives and to deal with the imbalance dataset.

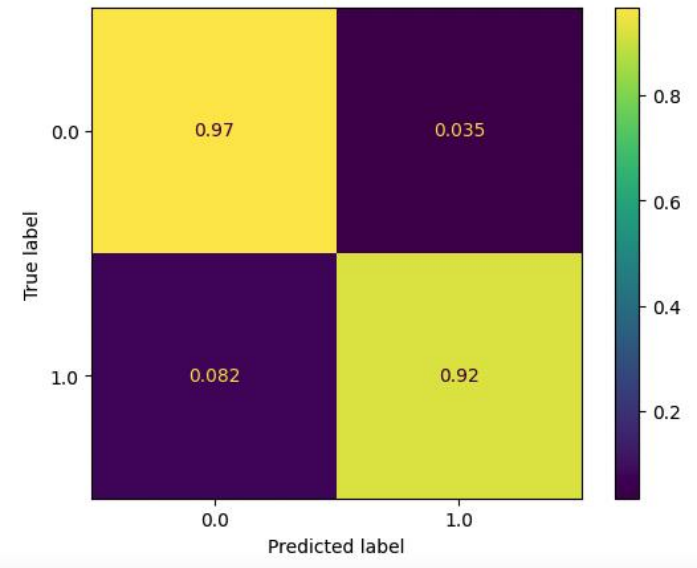
b. Results and Analysis

Experiment A

Confusion matrix for experiment A's training set || ROC Score= 1.0 || Recall =1



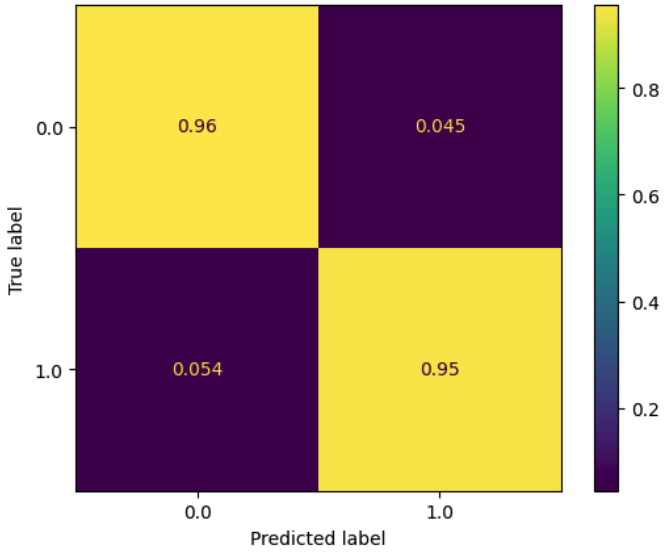
Confusion matrix for experiment A's validation set || ROC Score= 0.988021 || Recall =0.92



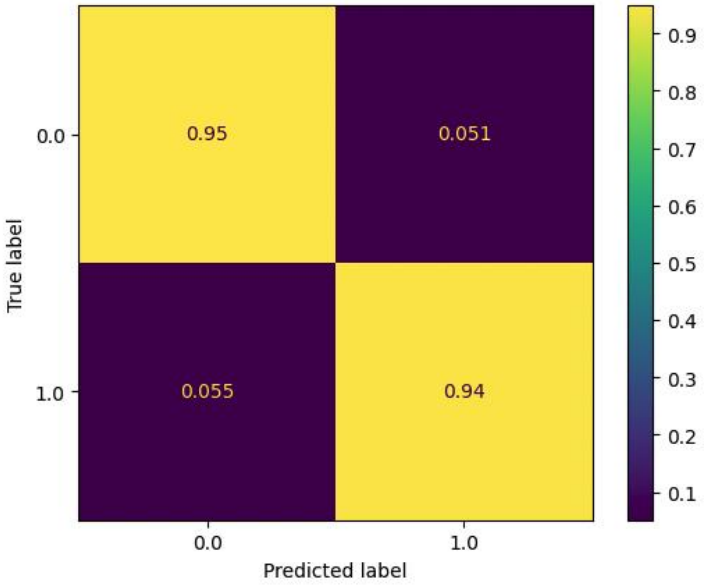
Looks like the ROC score is great but recall is not consistent indicating overfitting. That's the reason we pursue experiment B

Experiment B

Confusion matrix for experiment B's training set || ROC Score= 0.989 || Recall =0.95



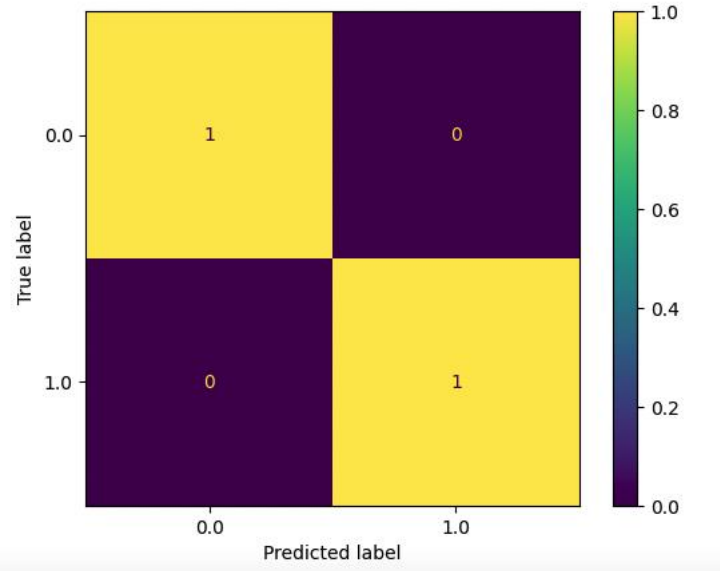
Confusion matrix for experiment B's validation set || ROC Score= 0.986 || Recall =0.94



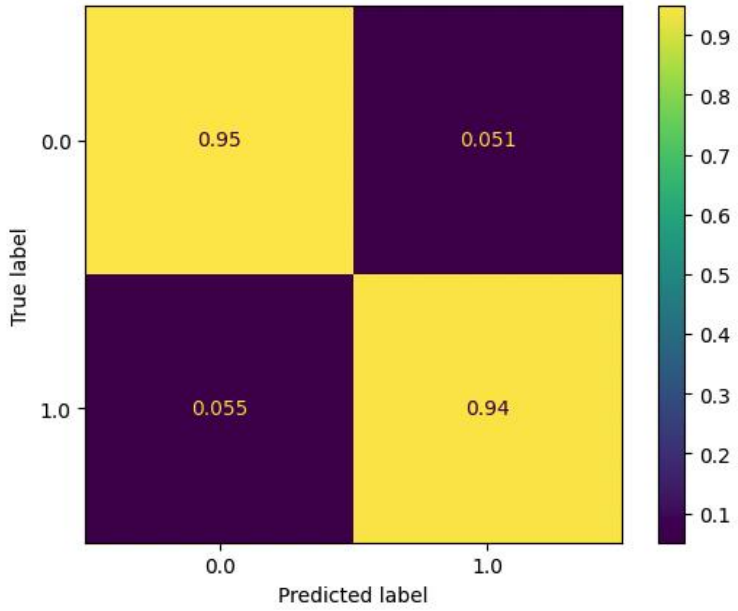
The ROC score and recall are both great and somewhat consistent, we should consider this model for production if no model is better.

Experiment C

Confusion matrix for experiment C's training set || ROC Score= 1 || Recall =1



Confusion matrix for experiment C's validation set || ROC Score= 0.99 || Recall =0.94



The ROC score is the best amongst the 3 experiments, however the recall score is relatively inconsistent.

To summarize:

	Experiment A		Experiment B		Experiment C	
	ROC	Recall	ROC	Recall	ROC	Recall
Train	1	1	0.989	0.95	1	1
Validation	0.988021	0.92	0.986	0.94	0.99	94

The ROC score from all 3 experiments is relatively close and high. However, the recall score indicates overfitting for experiment A and C. We think that the slight improvement in ROC score from model C does not justify the inconsistency on the recall score and thus we suggest model from experiment B be used for production

c. Business Impact and Benefits

If model B is pushed to production, then XYZ should be able to make more accurate and informed targeted endorsement deals decision, it would help XYZ to allocate its resources more efficiently leading to better ROI on sponsorship contracts.

For example:

Without the Model (Historical Approach):
Total endorsement budget: 10 players per year
Number of drafted players: 2 players on average per year (Historically)
Success Rate: 2/10 or 20%

With the model:
Total endorsement budget: 10 players per year
Successful Predicted Number of Drafted Players: 9 players per year (90% recall)
Expected Success Rate: 9/10 or 90%

Ultimately will increase XYZ's ROI.

d. Data Privacy and Ethical Concerns

There are no data privacy concerns from this project as all data are considered as public data. However, we should monitor the pattern of our model's prediction as it might algorithmically exclude players with certain stats and hence blocking some player's chance to get sponsored by XYZ. Manual adjustment could be made if the discussed concerns are the issue, i.e. assigning slot for few players manually.

7. Deployment

The model would be deployed via API to be used internally by XYZ company. The future data should first be pre-processed to the compatible forms with the training data.

8. Conclusion

To conclude, we had performed 3 experiments during the life of the project. All 3 experiments produced great models which have high ROC score, however based on the recall scores we favour the model from experiment B and suggest XYZ to push this model to production to help XYZ making better investment decision. Overall, this project has met its objective and met all stakeholder's expectation.

9. References

Kaggle. (2023). AdvMLA 2023 Spring Competition Data. https://www.kaggle.com/competitions/advmla-2023-spring/data