Course: Advanced Data Science for Innovation

Group Name: AT1 Group 4

* April Nommesen
* Hitoshi Fukuda

Date: 30 November 2022

**Github Link:** <https://github.com/aprilgum/adv-dsi-2022-at1-grp4/tree/master/nba-career-prediction>

# ****Assessment 1: Kaggle Competition Final Report****

This report states our experimental steps on a machine learning project following the CRISP-DM methodology, and presents achievement that we found, issues encountered, and contribution that our members made to the project.

## Business Understanding (Background & Objective)

In the National Basketball Association (NBA), a rookie is defined as a player who has played a game in the NBA for the first time at that season. Rookie players who belong to teams in the NBA are expected by the league, teams, sport commentators and fans to perform well and build their brilliant careers in the future. However, the NBA is the most competitive basketball league in the world and hence it is highly challenging for rookies to stay in the league for long time.

Rookie players joining in the first round of NBA draft are given four-year contracts where the first two years of the contract are guaranteed with the NBA team and the third and fourth years can change. Rookie players joining in the second round of NBA draft and undrafted players can sign contracts that can be anything from one year to four years and that are either fully guaranteed or not guaranteed at all. The value of rookie players contract is tied to the salary cap and the team they will play for. A rookie player who lasts for at least five years is considered successful and a player who does not last at least five years is considered risky. Being able to have a data-based decision to decide whether a rookie is a potential success, or a potential risk greatly impacts the team’s budget to pay the players’ salary and also impacts the team’s performance.

The **main business objective** of this project is to determine whether their rookie players are likely to perform well beyond five years from the start of their career. Assuming that these “successful” rookies are to be given guaranteed contracts for their third and fourth years, the team’s financial management can better plan or understand future budgets for all the players salary within the team.

The result of the model’s prediction is not the only criteria for offering guaranteed contracts for the rookies therefore a very high accuracy is not essential. The **technical objective** of the modelling is to produce the model with the best AUROC and least False Positives within the bounds of the project timeline allocated while addressing the existing data challenges. For example, there is a high number of rookie players that are deemed successful (career years played >= 5) in the available training data whereas our knowledge of the business is that the NBA is the most competitive basketball league in the world and hence it is highly challenging for rookies to stay in the league for more than five years. Therefore, one data challenge is that the successful players are over-represented. We will not always use the Accuracy as a main criterion for assessing the model because when we address the imbalanced data, it will always produce a lower Accuracy.

## Data Understanding

### Datasets

We were provided with two CSV files for this project, and so each dataset can be input into a form of dataframe by using a method of “read\_csv” provided by pandas.

* train.csv: a dataset for training machine learning models
* test.csv: a dataset for testing machine learning models

#### Observations/Rows

Each row is a rookie player and has an ID.

#### Target Variable

* TARGET\_5Yrs = 0 if career years played < 5
* TARGET\_5Yrs = 1 if career years played >= 5

#### Features

Each column is performance measure of each player.

The datasets given for the experiment include following features or performance measure on rookie players.

| No. | Feature | Description | Type |
| --- | --- | --- | --- |
| 1 | Id | Player Identifier | int64 |
| 2 | GP | Games Played | int64 |
| 3 | MIN | Minutes Played | float64 |
| 4 | PTS | Points Per Game | float64 |
| 5 | FGM | Field Goals Made (field goals are the sum of 2-Points and 3-Points) | float64 |
| 6 | FGA | Field Goals Attempts | float64 |
| 7 | FG% | Field Goals Percent (FG% = FGM/FGA) | float64 |
| 8 | 3P Made | 3-Points Made | float64 |
| 9 | 3PA | 3-Points Attempts | float64 |
| 10 | 3P% | 3-Points Percent (3P% = 3P Made/3PA) | float64 |
| 11 | FTM | Free Throw Made | float64 |
| 12 | FTA | Free Throw Attempts | float64 |
| 13 | FT% | Free Throw Percent (FT% = FTM/FTA) | float64 |
| 14 | OREB | Offensive Rebounds | float64 |
| 15 | DREB | Defensive Rebounds | float64 |
| 16 | REB | Rebounds (REB = OREB + DREB) | float64 |
| 17 | AST | Assists | float64 |
| 18 | STL | Steals | float64 |
| 19 | BLK | Blocks | float64 |
| 20 | TOV | Turnovers | float64 |
| 21 | TARGET\_5Yrs | Target variable: outcome is 1 if career length >= 5 years, 0 otherwise | int64 |

## Data Exploration

<https://github.com/aprilgum/adv-dsi-2022-at1-grp4/blob/master/nba-career-prediction/notebooks/at-group4-week1-eda.ipynb>

#### Structure

The dimensions of the datasets are below:

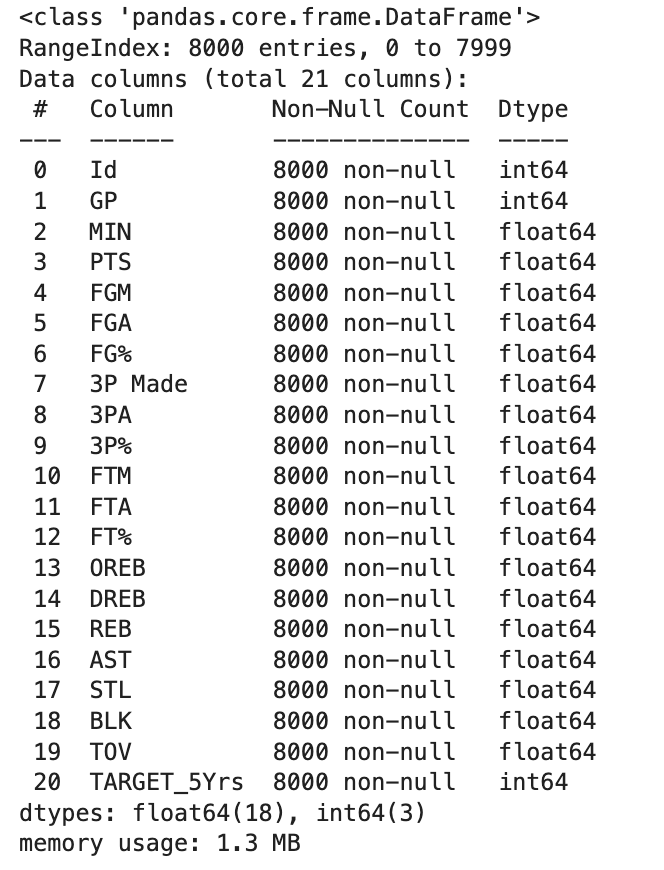
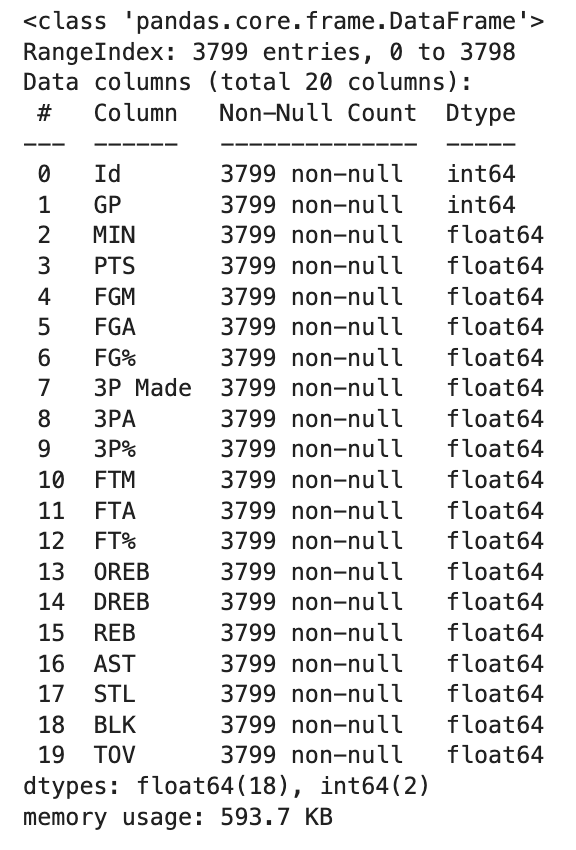
* train dataset: 8,000 rows and 21 columns
* test dataset: 3,799 rows and 20 columns (not having the “TARGET\_5Yrs” column)

#### Duplicates

There were no duplicate IDs and rows in the training and test data.

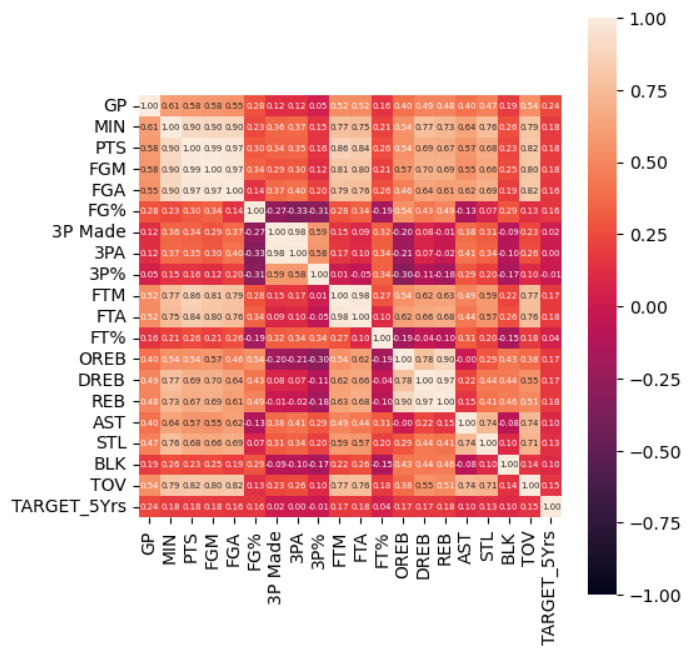
#### Missing Values

There were no missing values in the traning data and test data.

#### Correlation between the features (variables)

The correlations between the features are demonstrated in the matrix below:



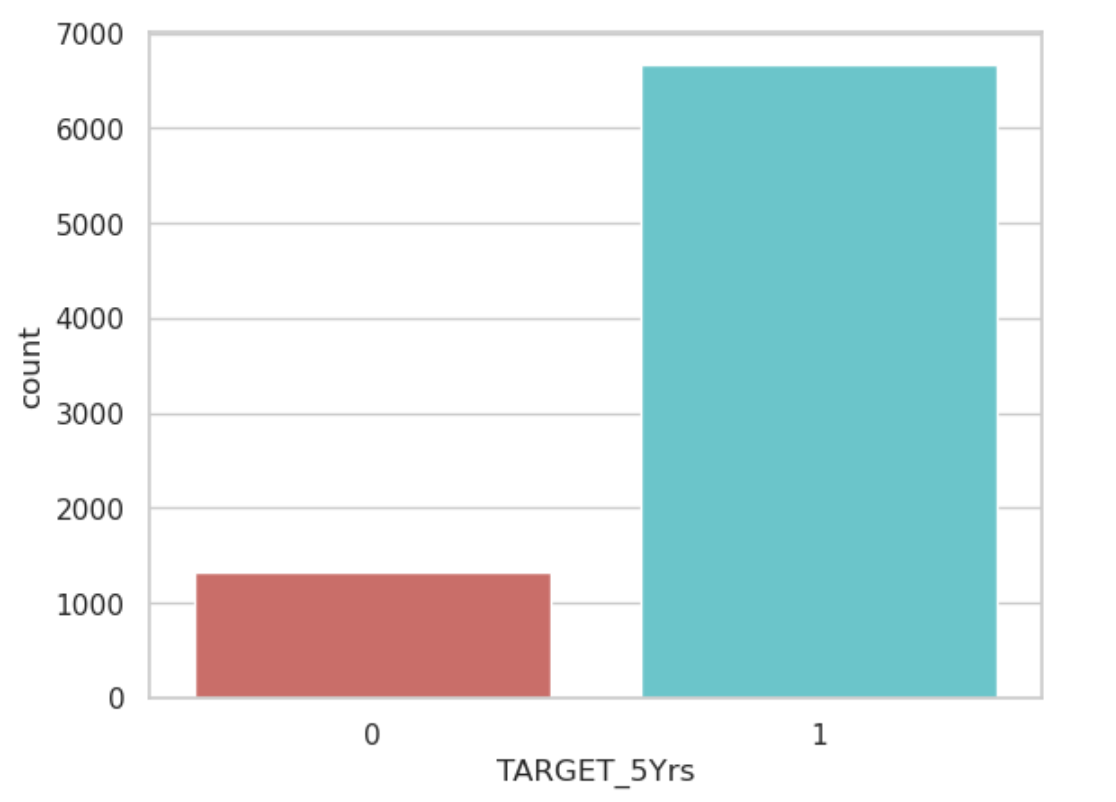
As shown above, following pairs of features are strongly correlated (close to 1.00).

* “MIN” Minutes Played and “PTS” Points per Game: 0.90
* “MIN” Minutes Played and “FGM” Field Goals Made: 0.90
* “MIN” Minutes Played and “FGA” Field Goals Made: 0.90
* “FGM” Field Goals Made and “FGA” Field Goals Attempts: 0.97
* “3P Made” 3-Points Made and “3PA” 3-Points Attempts: 0.98
* “FTM” Free Throw Made and “FTA” Free Throw Attempts: 0.98
* “REB” Rebounds and “OREB” Offensive Rebounds: 0.90
* “REB” Rebounds and “DREB” Defensive Rebounds: 0.97

In a case of there being a tightly correlated pair of variables, then removing either of them from the dataset will sometimes improve accuracy of prediction in regression model since it would mean preventing multicollinearity between variables. Integrating variables into one feature will also have possibility to improve accuracy if it explains better for the target variable.

#### Target Variable

One data challenge is that the successful players are over-represented in the training data.



## Data Preparation

The training data set was used for modelling and the test data set was set aside and used only to implement the model on unseen data.

Loaded the training data to Python.

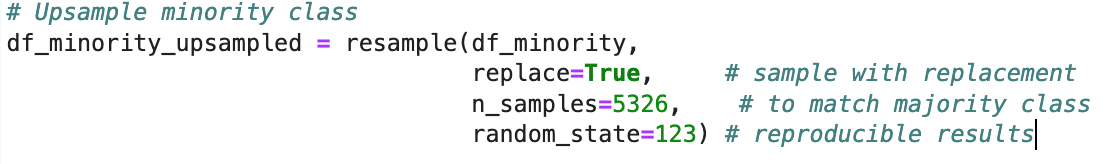
Stored the target variable (y) to a separate data frame.

The following preparations was applied to the training data. These were applied as needed in experiments.

1. Derived new features to add into the choices of features
   1. 2-Points Made = Field Goals Made - 3-Points Made
   2. 2-Points Attempts = Field Goals Attempts - 3-Points Attempts
2. Identified the features that are similar or mixtures as a result of the same calculations. When selecting features to put in the model, we would be mindful of putting features that are similar. During the process of modelling, we may choose to keep or drop some features based on similarity or collinearity.
   1. Points per game (PTS), Field Goals Made (FGM), Field Goals Attempts (FGA), Field Goals Percent (FG% = FGM/FGA)
   2. 3-Points Made, 3-Points Attempts, 3-Points Percent (3P% = 3P Made/3PA), 2-Points Made
   3. Free Throw Made, Free Throw Attempts, Free Throw Percent (FT% = FTM/FTA), 2-Points Attempts
   4. Offensive Rebounds, Defensive Rebounds, Rebounds (REB = OREB + DREB)
3. Applied scaling for use in experiments with models such as Logistic Regression where it helps to put the different features on the same scale. Used two kinds of scaling
   1. Standardization
   2. Normalization

For every experiment, we partitioned the training data into 80% training set and 20% validation set.

During handling of imbalanced data, we experimented on:

1. Oversampling by boosting the minority group ( career years played < 5) to the same number as the majority group (career years played >= 5) using sampling with replacement. 
2. Using the model’s hyperparameter class\_weight = 'balanced'. This parameter was used with Logistic Regression, SVC, and Random Forest Classifier.

## Modelling

We experimented on the following algorithms from scikit-learn that can predict a two-category outcome.

1. Logistic Regression - This ML algorithm is used to predict categorical variables using independent variables which are continuous.
2. Support Vector Classifier – This ML algorithm is typically used for classification tasks. SVC works by mapping data points to a high-dimensional space and then finding the optimal hyperplane that divides the data into two classes.
3. Random Forest Classifier – This ML algorithm can be used in both classification and regression problems. One desirable trait of this algorithm is that you can get the relative feature importance, which helps in selecting the most contributing features for the classifier.
4. XGBoost

### Hitoshi’s Highlights

I have made some attempts on feature engineering to enhance the accuracy of estimation with my model.

* Implementing XGBoost
* Implementing Partial Dependence Plot
* Then, I try to do feature engineering based on the result of XGBoost
* Check how does the result change

This time, I reached a model with features engineering such as:

<1st model>

* Points Per Game, Field Goals Percent, 3-Points Percent, Free Throw Percent, and Rebounds were dropped because they are just mixture (results of calculation) of other columns
* 2-Points Made and 2-Points Attempts were derived/calculated and inserted as new columns because Field Goals Made and Field Goals Attempts are mixed variables which include 3-points scores

Train data accuracy: 0.835

Validation data accuracy: 0.837

AUC: 504803

<2nd model>

* 3-Points Attempts and 2-Points Attempts were dropped because they were thought to be less important based on the result of XGBoost and Partial Dependence Plot.
* This model scored highest in Kaggle as of 30 Nov 2022.

Train data accuracy: 0.835

Validation data accuracy: 0.838

AUC: 0.505176

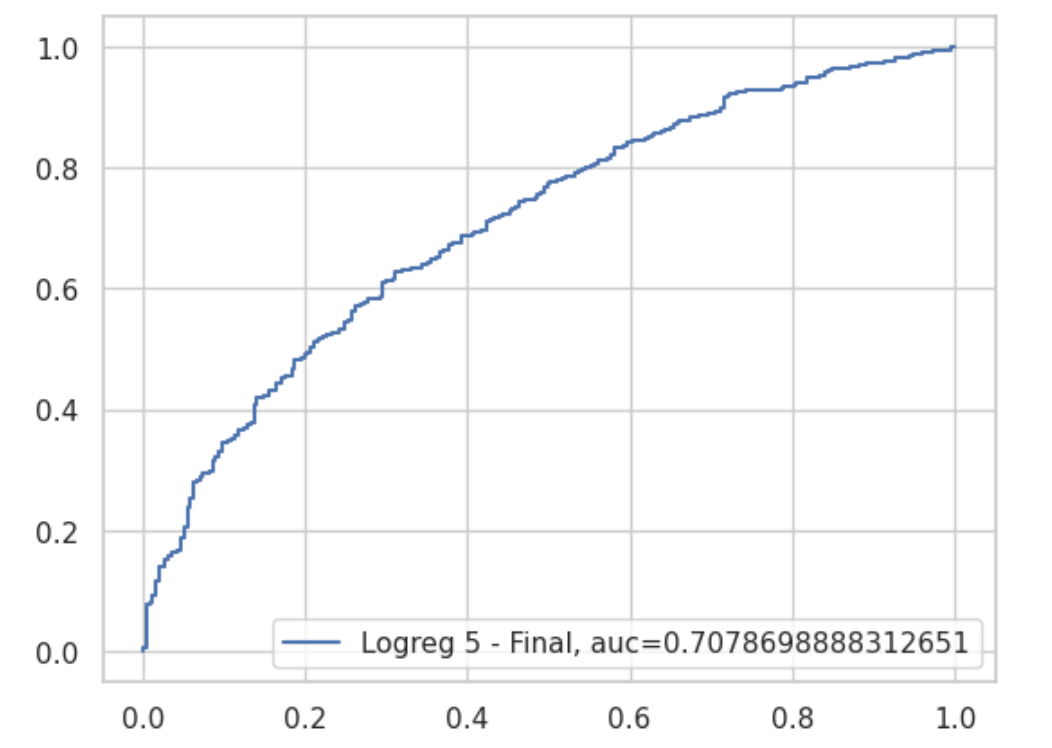
### April’s Highlights

I did not do any feature engineering during the model building. I spent the first week doing exploration and feature selection. Then the week 2, I decided that imbalanced data has to be addressed. The succeeding weeks are spent using the remaining time to improve the model or look at new algorithms.

Logistic Regression

With Logistic Regression, I scaled the features using normalization, handled the imbalanced target variable and experimented on different hyperparameters. For feature selection, I used backward stepwise selection where I started with all (scaled) variables under consideration and used the p-value of each coefficient as a criterion which features to drop. I also tried to select features based on the highest F-scores of XGBoost Feature Importance, but those features did not yield to a better model performance than the backward stepwise selection.

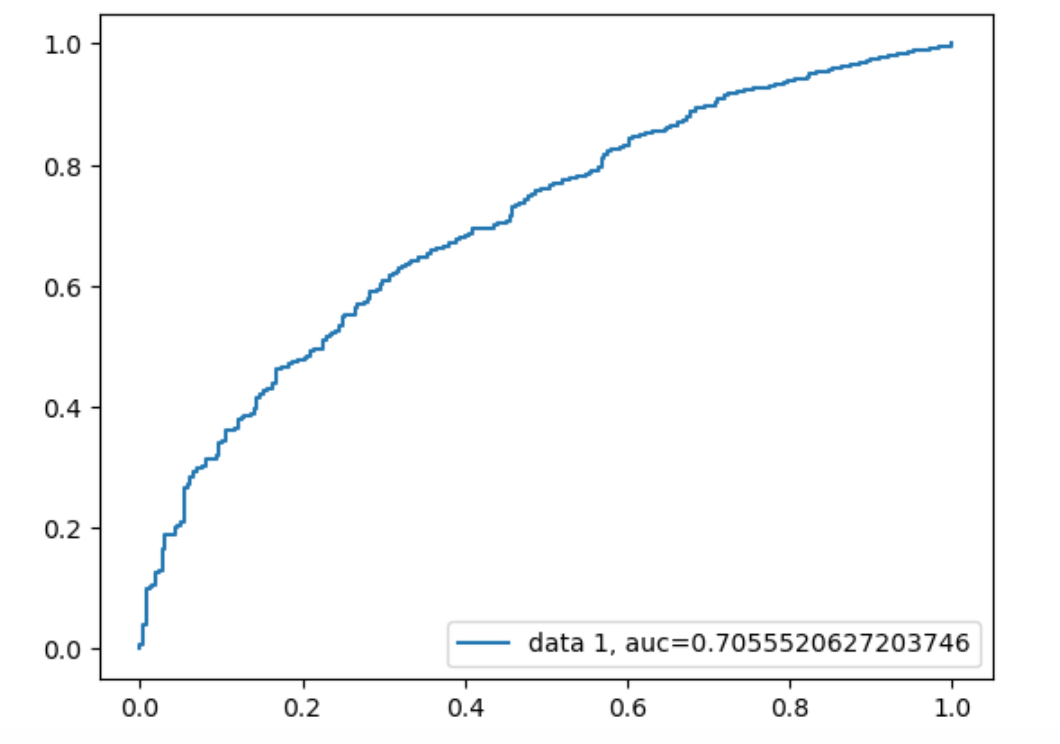
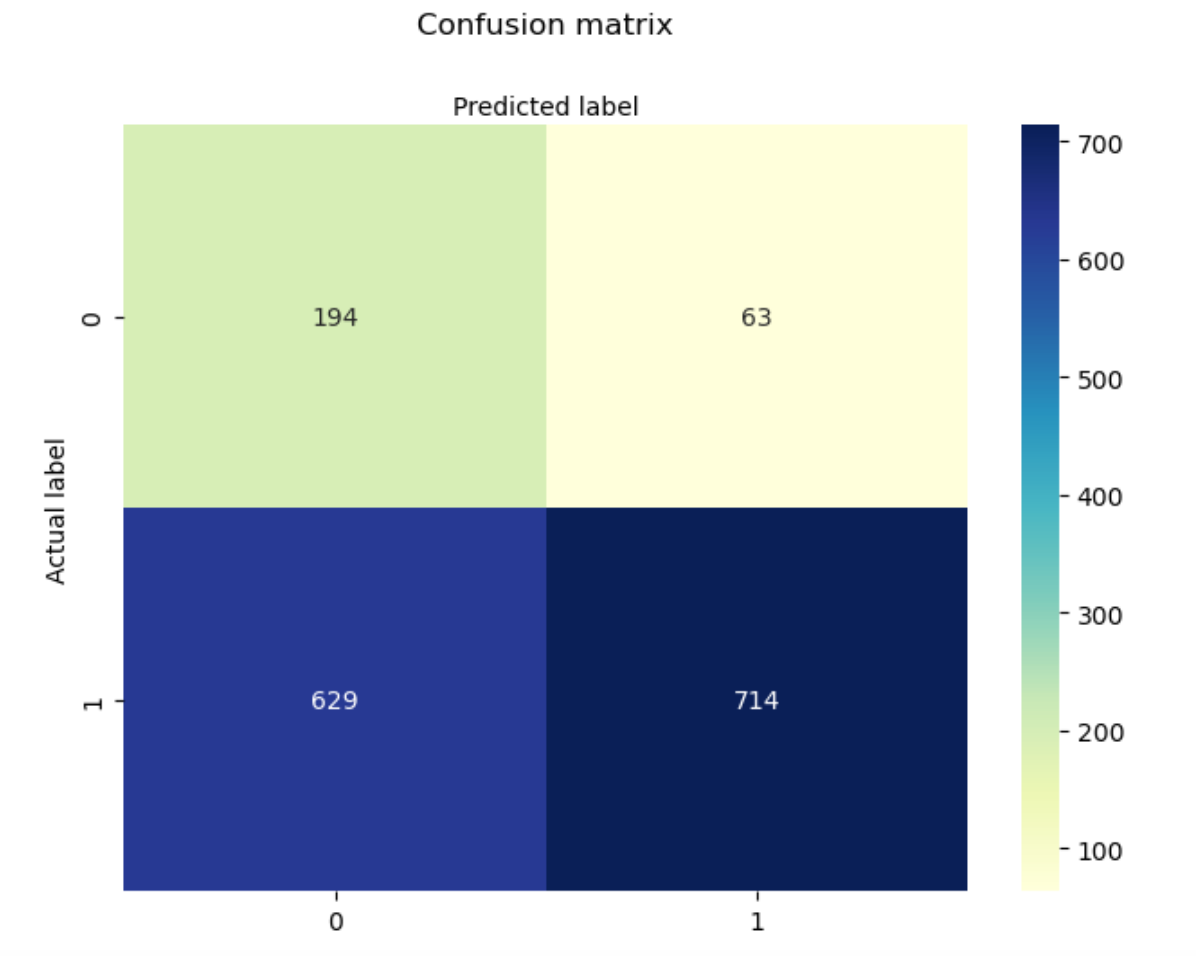
Chart, treemap chart

Description automatically generated

Logistic Regression was the first model that the team has decided to focus on because of its popularity with problems of predicting between two outcomes.

SVC

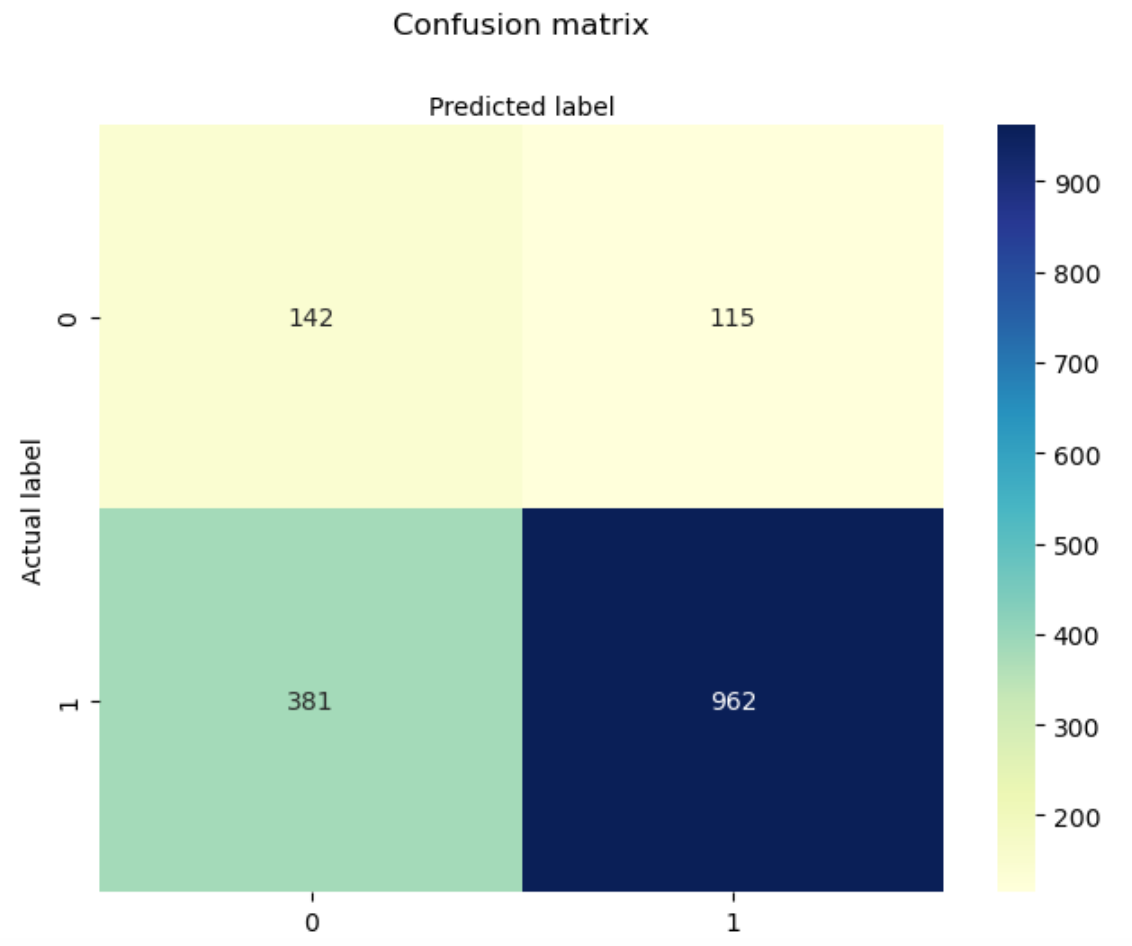
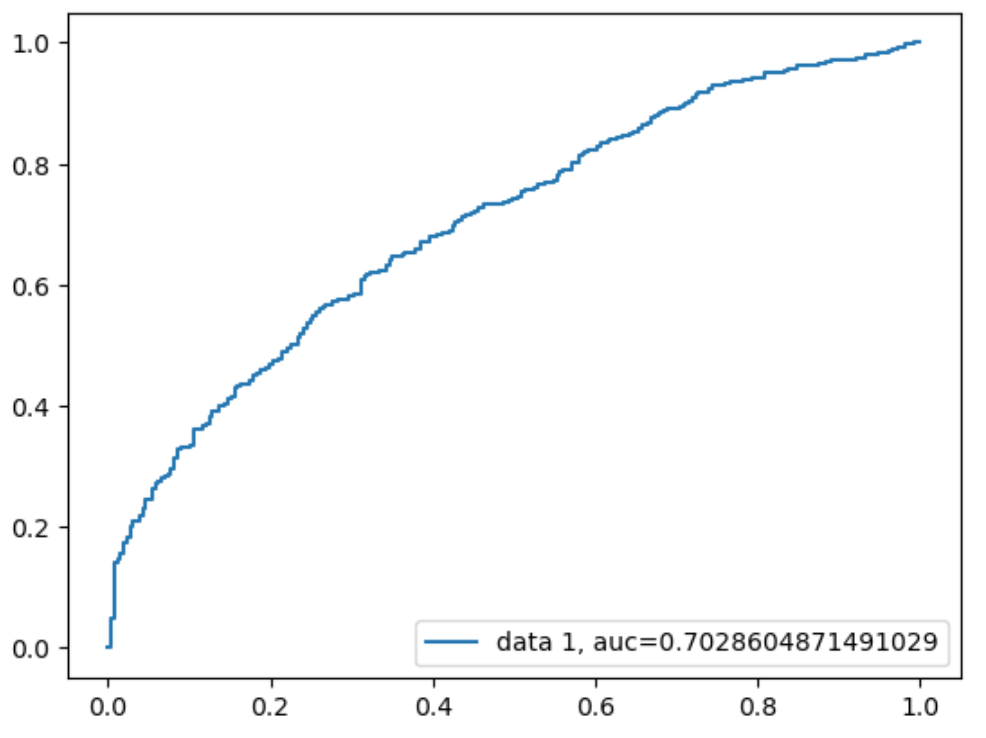
To challenge Logistic Regression, I experimented on SVC because one of the issues I find in the training data set is the very high percentage of “success” (career years played >= 5). SVC offer a specific hyperparameter to add some weights to each class, thus simplifying the data handling. Furthermore, the SVC algorithm has offered some of the best performing models I had seen.



SVC1

Random Forest Classifier

To challenge Logistic Regression and SVC, I experimented on Random Forrest Classifier because like SVC, it also offers a specific hyperparameter to add some weights to each class to handle the imbalanced data. Furthermore, the Random Forrest Classifier algorithm also has a “natural” feature selection property, thus simplifying the experiment.

XGBoost

To challenge Logistic Regression, SVC and Random Forrest Classifier, I experimented on XGBoost to take advantage of using an automated hyperparameter tuning (Hyperopt). XGBoost was the only algorithm where I was able to move away from manual hyperparameter tuning because it is fast and does not render my machine in a stand-still. However, the model it produced fell short in the AUROC even though it has the lowest false positives.

Chart, treemap chart

Description automatically generated Chart, line chart

Description automatically generated

## Evaluation

The two best model to compare are:

##### Model A from Hitoshi:

* Logistic Regression, L2 penalty term, lbfgs solver, C = 1.0, tol = 0.0001 (hyperparameters were determined by XGBoost)
* Features used: Games Played, Minutes Played, Free Throw Made, Free Throw Attempts, Offensive Rebounds, Assists, Steals, Blocks and Turnovers. Experimented on feature engineering based on the result of XGBoost.
* AUROC = 0.5055
* Kaggle Score = 0.71121

##### Model B from April:

* Logistic Regression, L2 penalty term, saga solver, C = 0.01, tol = 0.0001 (hyperparameters were determined manually)
* Used *class\_weight = 'balanced'* to handle imbalanced data
* Used MinMax normalisation to put the features on the same scale
* Features used: Normalised: Games Played, Field Goals Attempts, Field Goals Percent, 3Points Made, 3Points Attempts. Used backward stepwise selection using p-value as criterion.
* % False Positives = 5%
* AUROC = 0.7079 – highest of all models seen in this team
* Kaggle Score = 0.70499

Based on the business and technical objective set at the start of the project, we choose Model B as the one to pass to the team Finance Department and Recruitment/Contracts Department who will analyse the team’s budget and player performance.

With the highest AUROC, Model B has the best predictive potential for all classes most of the time (based on both AUROC and Kaggle Score). Model B also has a low potential of predicting a rookie to have at least 5 years if career when in fact will have a short career.

## Deployment

It has been decided that this model is a once-a-year exercise and does not need to predict new data / new rookies frequently (i.e. monthly). The model prediction, of whether a rookie will have at least 5 years of successful career in NBA, will be used as one of the business criteria to decide whether a rookie will potentially have a guaranteed contract in the future.