EXPERIMENT REPORT

Student Name	Archit Murgudkar
Project Name	AT1 - Kaggle Competition Week2
Date	25/08/2023
Deliverables	AT1_PartB-Binary Classification using Random Forest Model - Xgboost Classifier
Github Repo Link	https://github.com/Arch30it/NBA_Draft

1. EXPERIMENT BACKGROUND	
1.a. Business Objective	The NBA draft, an annual event where teams pick players studying in American colleges and from international professional leagues to join the rosters. These picked player based on their statistics will be selected to play in the NBA league. A binary classification model when trained will help predict whether a particular player based on his numbers will be drafted to get into NBA.
1.b. Hypothesis	Features provided will help classify if a certain player gets picked up to play for NBA by training a XGBoost Classifier. There are numerous features present in the dataset which can help predict the dependent variable named, “drafted”.
1.c. Experiment Objective	A supervised machine learning classification model predicting whether a college basketball player will make into the dream NBA league depending on the independent variables which are his game skills. The objective is to develop the model and its performance better where overfitting and underfitting to be dealt with if present using hyperparameter tuning and the model performing on the unseen data to help classify the result.

2. EXPERIMENT DETAILS

2.a. Data Preparation

In data preparation, I started with data cleaning first, I made a copy of dataset to perform data cleaning, so that we always have our original dataset if we want to refer and make the changes in the duplicate dataset. Further, I did a check on finding null/missing values in the records.The feature ‘rec_rank’ and ‘pick’ comprised a greater amount of null values, so I decided to totally drop off those columns. Adding any value like mean/median/0 etc. would create extreme bias in the predictions. After this, I deleted the following features due to valid reasons as follows:

1. num" feature as it is a player specific record
2. "ht" feature as it consists false information
3. "player_id" as it consists specific information
4. "type" as it consist of the same value in all records and would not help in making any decision.

Further, in the missing values of other features, I imuted with median, as median is robust on outliers. “Yr” had redundant values and nulls which I replaced with the mode of it. I also checked for duplicates and there were no duplicates present in the dataset.

2.b. Feature Engineering

I checked how all features are correlated with the target variable and also plotted a heatmap. From that, I understood that the features did not strongly correlate much with the target feature, “porpag” with the correlation of 0.255was the highest among all, and “mid_ratio” ranked lowest. Also I checked mutual information of dependent variable s and target variables. Mutual information tells us about how strongly independent variables are related to the independent variable. As our machine learning classifier requires data to be in the numeric format before feeding it, I decided to perform encoding on the features with object data type. Due to high cardinality in the “team” and “conf” variables, I performed encoding using leave one out encoding technique. I performed ordinal encoding on the “yr” feature and converted them into numeric datatypes. Another important factor is ensuring that we provide data to the model which is on the same scale, which will help our model make accurate predictions. For this, I performed data scaling by importing the ‘StandardScaler’ library from sklearn. After accomplishing all this, the data was finally suitable to be fed to the classifier algorithm and carry out modeling.

2.c. Modelling

Firstly splitting data into training, validation in order to model the classifier model. I decided to split the records in the following manner, 25% of the whole data for the validation set and remaining training data. The dataset was highly imbalanced with just around 9 % output to be 1 and 99% to be 0. To tackle this, I made use of the SMOTE technique which generated 80% of the synthetic data of minority class present in the output variable. This helped to balance the dataset.I then modeled the XGBoost classifier to make predictions. I made a list of different hyperparameters namely ‘n_estimators’,’max_depth’, ‘gamma’, ‘reg_alpha’, ‘colsample_bytree’, ’min_child_weight’, ‘learning_rate’ and performed a randomized search using RandomizedSearchCV on the XGBoost classifier. The best parameters obtained were as follows:

‘reg_alpha’ - 0.5
‘ n_estimators’ - 100
‘Min_child_weight’ - 1
‘ max_depth’ - 5
‘ leaninig_rate’ - 0.3
‘ gamma’ - 0.0
‘ colsample_bytree’ - 0.3

3. EXPERIMENT RESULTS	
3.a. Technical Performance	I performed cross_val_score on the classifier and got a score of 0.9993. Then I fitted xg_classifier on the training set. The accuracy score using the ‘roc_auc_score’ metric came out to be 1 and on the validation set, it came out to be 0.9986. Using the XGBoost model on the unseen data, i.e, the testing set, the score of the performance metric, roc_auc_score came out to be 0.8917 .
3.b. Business Impact	The XGBoost model achieved an accuracy rate of 89.17% on the roc_auc_score performance metric. So the predictions made by the model to classify if a player will get drafted is 89.17 %accurate.
3.c. Encountered Issues	Major issue I thought of was the high imbalance present in the distribution of the target variable and high cardinality in the features. Also, alot of null and redundant values were present in the variables of the dataset.

4. FUTURE EXPERIMENT	
4.a. Key Learning	The model would have been more accurate if the features in the dataset were more strongly correlated with the target variable and if the distribution of the target fields would have been more balanced.
4.b. Suggestions / Recommendations	There is a high need to collect more records which would contribute in balancing the dependent variable. Also, more features need to be collected which may have a stronger relation with the target variable which in turn will help the model predict with higher accuracy based on stronger evidence.