EXPERIMENT REPORT

Student Name	Rui Hu Zhang (13627753)
Project Name	36114 Assignment 1 week 2 Team 3
Date	27/02/2022
Deliverables	**Jupyter notebook**: Zhang_RuiHu-13627753-week3_XGBoostCV20Folds.ipynb **Model name**: XGB_10cv_randomsearch_biggergrid_upsampled.joblib **Github repo**: JKaur1992/adsi_at1 **Repo branch**: RZ_Week3_codes **Kaggle submission**: Richard_XGB_CV_RandomSearch_week3_20220227.csv

---

1. EXPERIMENT BACKGROUND	
Provide information about the problem/project such as the scope, the overall objective, expectations. Lay down the goal of this experiment and what are the insights, answers you want to gain or level of performance you are expecting to reach.	
1.a. Business Objective	The goal of this project is to create a machine learning (ML) model that predicts likelihood whether a new NBA player will be staying in NBA after 5 years, based on their in-game performance metrics. This allows the user of this ML model the possibility of making long term investment in a new player with the individual's in-game performance, and minimise making investment on short term players.
1.b. Hypothesis	The 19 player performance variables are sufficient in making accurate prediction in the likelihood of a new player continue after 5 years. The interaction effect between these performance variables are not influencing this prediction significantly, and the higher order effects are also not influencing this prediction. These variables are descriptive of the performance of a player, hence these variables are sufficient in describing the player's skills. Therefore no other effects are necessary in describing player skills.

1.c. Experiment Objective	A ML model is created using the 19 independent variables to predict the likelihood of a new player staying in NBA after 5 years.

---

2. EXPERIMENT DETAILS	
Elaborate on the approach taken for this experiment. List the different steps/techniques used and explain the rationale for choosing them.	
2.a. Data Preparation	The variable column Id is removed from the dataframe prior to model training. This is only a descriptive variable, and is not a variable that describes the player performance. Therefore this is not included in the dataframe of the model training. For the training dataset (training.csv), the target variable column TARGET_5Yrs is also removed prior to model training to avoid incorrect or accidental inclusion of these data values by the ML models. The remaining variable values are scaled to one standard deviation about the mean, using the Scikit learn method StandardScaler. This ensures the variations in the variable values are not influencing the parameters predicted by the model. For the polynomial features, the Scikit learn method PolynomialFeatures is used to generate second order variables. All interaction variables are generated. This ensures the ML model considers For train-test split, given the target variable TARGET_5Yrs shows some imbalance between 0 and 1 classes, a stratified selection is used. In addition to the above preparations, this week examines dataset balancing. The simple upsampling is used instead of the stratified dataset splitting used in the last week. In other words, this week considers two data preparation strategies: 1. Without dataset balancing, perform stratified dataset splitting on the dataset. 2. Perform upsampling dataset balancing, then perform dataset dataset splitting.

2.b. Feature Engineering	This week the cross-validation study is reduced to 10-fold, due to the number of grid search studies with relatively large grid. In addition to cross-validation, a simple grid search study is performed for all three ML models and algorithms outlined in Section 2.c below. This grid search is performed using standard variable value variations recommended in reference books and webpages. This study incorporates cross-validation in the model training. In a separate investigation, the data balance is considered. In this week, the data imbalance is addressed by simple upsampling using the Scikit learn method resample. This brings the 1331 counts of the minority class is brought to the same count of 6669 count of the majority class. After the grid search, the optimal hyperparameters of this optimal model is the used to define a new grid of slightly larger range of some of the parameters. This is then used to define a randomised search grid. This allows to refine the parameter search around the optimal value for some of the hyperparameters. The AUC score is the only metric used in the cross-validation and grid search experiments.
2.c. Modelling	Logistic regression – simplest linear model for classification ML model, given the small number of parameters and hyperparameters, this will also be fastest to train Stochastic gradient descent (SGD) – from reference books and Internet websites, this is one of the methods for shallow ML models that allows faster approach Random forest – another shallow ML model that is relatively robust for a general situation. This will also allow the consideration of variable interaction. Given there are a number of hyperparameters, this ML model has a large scope for optimisation. XGBoost – another algorithm for ML tree models that allows the automatic tuning of weights during the epochs of the model training. This then allows potentially faster model training time. The following hyperparameters are varied in the grid search appraoch. Logistic regression: regularisation parameter, and algorithms used by Scikit learn XGBoost: number of samples before splitting, maximum depth of trees, minimum number of samples before splitting, learning rate, lambda_reg, l1 regularisation parameter, l2 regularisation parameter.

3. EXPERIMENT RESULTS

Analyse in detail the results achieved from this experiment from a technical and business perspective. Not only report performance metrics results but also any interpretation on model features, incorrect results, risks identified.

3.a. Technical Performance

Tabulated below is the AUC score for all models trained this week.

XGBoost with 10-fold cross-validation and grid search	1
XGBoost with 10-fold cross-validation and grid search, with minority class upsampled to the same size	1

	XGBoost with 10-fold cross-validation and grid search, with minority class upsampled to the same size, resetting all negative variable values to zero: 0.731202 XGBoost with 10-fold cross-validation and grid search, with no standard scaling: 0.781942 XGBoost with 10-fold cross-validation and randomised search, hyperparameter grid 1: 0.974935 XGBoost with 10-fold cross-validation and randomised search, hyperparameter grid 2: 0.585822 XGBoost with 10-fold cross-validation and randomised search, hyperparameter grid 3: 0.568441 The best model obtained from the grid search is used to determine a range of values for the hyperparameters. Based on the hyperparameter values from the optimal model of the grid searches, the randomised searches are then made around these values. From the grid search runs, it can be seen that there are a few hyperparameters that are the same between these grid searches, even when the data are preprocessed slightly differently. For example, the maximum tree depth is found to be 4. The randomised search can be seen to be worse compared to the grid search. This is possible as the same set of hyperparameter values were not used in the randomised search.
3.b. Business Impact	The highest AUC score of these models is 0.974935. This corresponds to a XGBoost model with cross-validation and randomised search.
3.c. Encountered Issues	Given the effort spent on the various algorithms this week, no improvement in the model performance is observed. This is despite the application of cross-validation and hyperparameter optimisation. This suggests that the simplest possible ML model can be used to reduce the complexity of the model explanation.

4. FUTURE EXPERIMENT

Reflect on the experiment and highlight the key information/insights you gained from it that are valuable for the overall project objectives from a technical and business perspective.

4.a. Key Learning	As shown in Section 3.a the experiments performed in this week generated ML models that have relatively weak prediction performance. Although XGBoost algorithm is powerful in terms of hyperparameters available and different forms of algorithms that can be used (e.g. different predictor algorithms, tree methods), this does not appear to provide significant improvements to the ML model prediction accuracy compared to a standard random forest ML model. The initial search for optimal hyperparameter values can take a large grid due to the number of hyperparameters in the algorithm. This then means that without any previous professional information, the hyperparameter values need to be estimated. The approach of grid search followed by random search around the optimal values is effective, as this allows the hyperparameters are further refined using a smaller grid once the optimal values are found.
4.b. Suggestions / Recommendations	As the ML models generated using XGBoost do not show significant improvement to ML models tried in previous weeks, in the final week, the attempt is to optimise hyperparmeters of these models. This reduces the amount of variables to consider in the grid search.