EXPERIMENT REPORT

Student Name	Roger Yu
Project Name	MDSI ADSI Assignment 1
Date	2020-02-07

Deliverables	`yu_roger-10906675-week1_randcv_xgb_69051.ipynb`
	`randomised_xgb.joblib`
	https://github.com/roger-yu-ds/assignment_1/tree/master

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	Predict the probability of a rookie NBA player, given certain traits, having a career in the NBA that is greater than 5 years.
1.b. Hypothesis	The target variable is not balanced (83% positive) so initial naive classifiers (untuned) might have a hard time correctly predicting the two classes.
1.c. Experiment Objective	1. Calculate a baseline score, which has an AUC of 0.5. 2. Create an untuned classifier that beats this score. 3. A small extent of hyperparameter tuning using RandomizedSearchCV.

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

Steps

Removal of the ID columns

This is because

- they should not be predictive
- they are unique to each row and hence would not have predictive power on out of sample data
-

Profile Report (pandas-profiling)

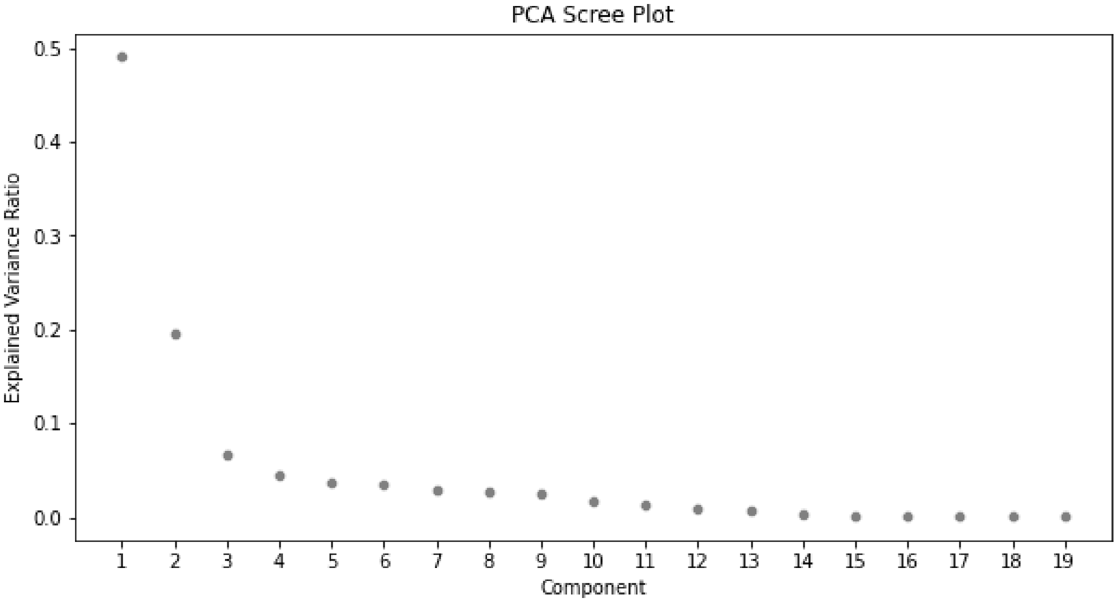
- Basic individual statistics of the columns
- Noticed several right skewed columns, which suggests a log transform
- The target column doesn’t seem to have a high correlation with any of the feature columns
- Several columns are highly correlated with each other (e.g. FGM-MIN, and FGM-PTS), which suggests that omitting these columns could improve the model performance

Scaling

Column means range from as low as 0.24 to as high as 72. This difference is likely to cause some algorithms to underperform. The sklearn.preprocessing.StandardScaler is used to ensure that all features are of comparable values.

Dimension reduction

From the profile report above, dimension reduction could improve model performance due to highly correlated columns. PCA was and visualisation of the explained variance vs number of components, the elbow seems to be at the 4th component, and 8 components explain about 95% of the variance.



Checking for Consistency

The percent columns are defined by the values made divided by the values attempted, e.g. the Field Goals Percent is Field Goals Made divided by the Field Goals Attempted. However, upon dividing these two columns, almost all the numbers differ from the percent column. Furthermore, some of the attempted columns contained zeros. How to handle this will be investigated in future experiments.

Train Test Split

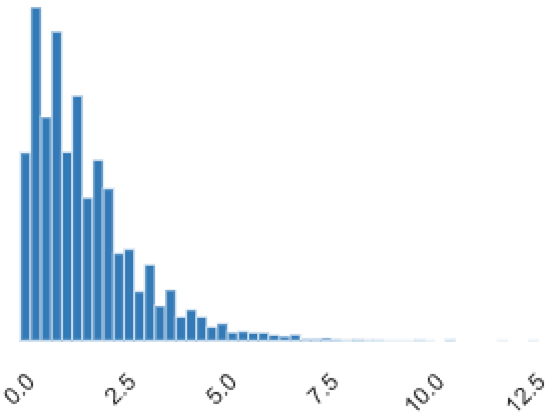
More accurately, train validation split. The training data was split into a training (80%) and validation set (20%).

Future

The following were not executed due to time contrainsts but are planned for future experiments:

- Additional techniques (other than PCA) to remove highly correlated columns, e.g. L1/Lasso regression, K-means clustering
- SMOTE: results from an XGB classifier shows that the most instances in the negative class were misclassified.

- Log transform of some of the right skewed columns, e.g. AST



- Investigate further that the percentage columns doesn’t seem to be consistent with the made and attempt columns and decide how to handle the problem.

2.b. Feature Engineering

This iteration is simply to produce a quick set of baseline results to improve upon in future experiments.

PCA

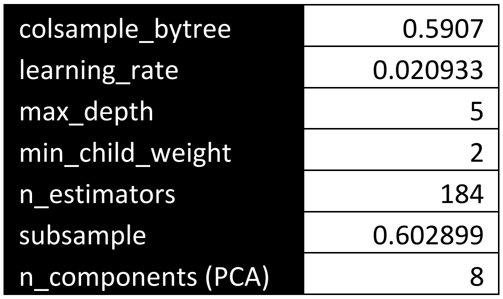
The PCA step was included in the pipeline. The pipeline was fed through a RandomizedSearchCV object and the number of components in the PCA step was made searchable.

Future

- GLM with L1 regularisation to detect important features.
- Clustering
- Adversarial Validation (maybe, see section 3.a)

2.c. Modelling

The model chosen for the first iteration is an XGB classifier with the following parameters.



colsample_bytree	0.5907
learning_rate	0.020933
max_depth	5
min_child_weight	2
n_estimators	184
subsample	0.602899
n_components (PCA)	8

These were the parameters that resulted in the best average AUC from a 5-fold CV out of 100 iterations in a RandomizedSearchCV.

The number of PCA components chosen was 8, which is the number that explains about 95% of the variance.

Future

- Tuning the pos_scale_weight parameter of the XGB classification algorithm: making this value smaller than 1 will make the model weight the errors from the negative class more, hence reducing the false positive rate. This is to be balanced with SMOTE.
- Run larger iterations of the random search to cover more of the parameter space
- Random Forest Classifier
- Logistic Regression with L1 regularisation
- ensemble: create classifiers that do well on the negative classes and include these predictions in the ensemble
- Investigate probability calibrations

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

Score of the relevant performance metric(s). Provide analysis on the main underperforming cases/observations and potential root causes.

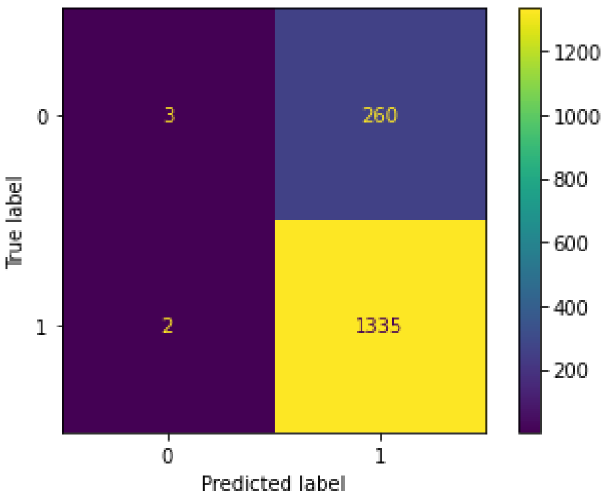
Scores

Data	ROC AUC Score
Validation	0.678522
Test (Kaggle)	0.69051

The scores are comparable, which indicates that the distributions of the training and test sets are also comparable, i.e. an adversarial validation exercise should not be able to distinguish the two sets with high score. In other words, there aren't particular sets of features in the test set that are fundamentally different to those in the training set.

Underperforming cases

Below is a confusion matrix on the validation set. The predictions are predominantly of the positive class, with only 5 predictions of the negative class. Naturally, there is a lot of false positives (260).



As mentioned in the previous section, tweaking the pos_scale_weights is likely to improve on the false positive rate.

3.b. Business Impact

Interpret the results of the experiments related to the business objective set earlier. Estimate the impacts of the incorrect results for the business (some results may have more impact compared to others)

Sponsors and teams would like to support players that are likely to have a career greater than 5 years. There is likely to be a lot of upfront costs in training that are not recuperable if that player stops their career early, so the cost of a false positive is high. Too many of such cases could cause the company to shutdown, as initial investments are not recuperated.

On the other hand the cost of a false negative is foregone chance of hiring a well performing player for basketball teams or a player that produces a lot of marketing income for sponsoring companies. While this is unlikely to bankrupt companies/teams, they are also unlikely to overcome their competitors.

3.c. Encountered Issues

List all the issues you faced during the experiments (solved and unsolved). Present solutions or workarounds for overcoming them. Highlight also the issues that may have to be dealt with in future experiments.

	Issue	Evaluation
1	Unacceptably large false positive rate	unsolved: handle in future experiments by tweaking the pos_scale_weight parameter
2	Imbalanced data	unsolved: use SMOTE in future experiments
3	High correlated features	partial: • used fewer PCA components • future: use L1 Logistic regression for feature selection • future: use K-Means clusters as features
4	Inconsistent columns	unsolved: future: consider ignoring the either the **Percent** column or the **Made** and **Attempted** columns

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	I didn’t realise there was an XGB classifier parameter called pos_scale_weights to tweak the extent to which positive class errors impact the algorithm. This could be used to produce more balanced predictions. However, it’s unlikely that this alone will produce satisfactory overall results. Different stakeholders will have different tolerances for false positives and false negatives, this parameter can be adjusted to the business needs. Although for this particular case the business impact of a false positive seems to be higher than that of a false negative, so more emphasis needs to be put on the negative (minority class).
4.b. Suggestions / Recommendations	In order of expected uplift, and hence actions to take: 1. Add pos_scale_weights to the parameter space for random search 2. SMOTE is potentially another good path to pursue with the pos_scale_weights tweaking to overcome the imbalanced data problem. 3. Log-transform the skewed columns 4. Feature selection using Lasso Regression. It is difficult to assess the uplift resulting from these experiments, judging from the scores of other Kaggle submissions, assuming that other teams have attempted these techniques, the uplift is expected to be up to 0.02 of the ROC AUC score.