EXPERIMENT REPORT

Student Name	Roger Yu
Project Name	MDSI ADSI Assignment 1 Part B
Date	2020-02-014

Deliverables	`yu_roger-10906675-week1_early_stopping.ipynb`
	`xgb_top_8_features_early_stopping.joblib`
	https://github.com/roger-yu-ds/assignment_1/tree/roger

---

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 previous model was overfit, due to the high training AUC of 0.83 and the validation AUC of 0.70. This round of experiments tries to reduce the overfitting.

The hypotheses are that the following would improve the validation AUC:

1. Limiting the features
2. Early stopping by lessening the overfitting
3. SMOTE and undersampling
4. Calibration of probabilities

1.c. Experiment Objective

1. Improve the validation AUC beyond 0.70
2. Reduce overfitting as indicated by the difference between the training and validation AUC, i.e. 0.13.

---

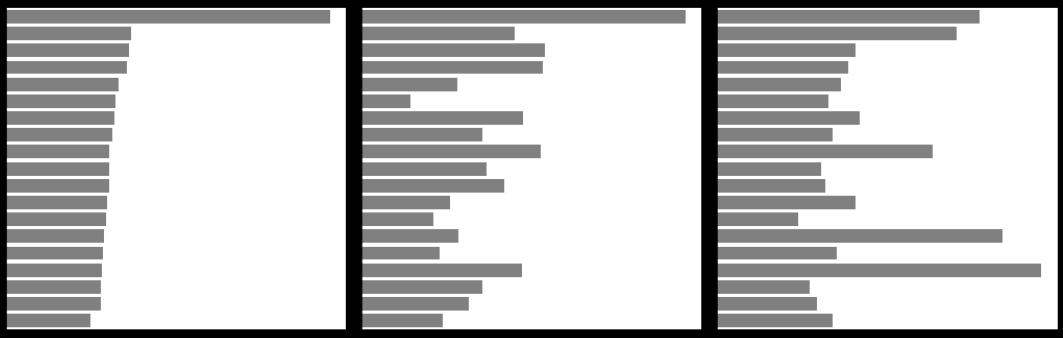
2. EXPERIMENT DETAILS

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

Feature Selection

Using the feature importance results (by gain) from the last experiment, the top 8 features were selected to create a **reduced data set**.



Early Stopping

Both the full and **reduced data set** were used in the calibration process.

SMOTE

Several combinations of upsampling of the minor class and downsampling of the major class were performed and the validation AUC of each resulting data was calculated.

Calibration

Both the full and **reduced data set** were used in the calibration process.

Future

- Gridsearch the different number of features

2.b. Feature Engineering

NA

2.c. Modelling

Early Stopping

The data with only the top 8 features were used to train an XGB Classifier with early stopping; `early_stopping_rounds=10`. This means that if the AUC on the validation set does not improve after 10 iterations then the fitting stops.

The early stopping is implemented by first getting the best_ntree_limit parameter from the booster object: `clf.get_booster().best_ntree_limit`, then applying this in the predict step `clf.predict(X_train, ntree_limit=best_ntree_limit)`.

SMOTE

The XGB Classifier used early stopping to fit on the training set that has been augmented by over and under sampling. The `best_ntree_limit` parameter is calculated for each augment data set.

Calibration

The calibrated classifier was done using the XGB classifier as the base estimator

1. with fitting from scratch
2. with fitting using the early stopping `best_ntree_limit` parameter value

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 (problematic with XGB early stopping)
- 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 (did not improve)

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

Early stopping

Early stopping, with the reduced features, achieved a respectable validation AUC, while also decreasing the different between the training and validation AUCs; dropping from 0.18 to 0.056.

	Training AUC	Validation AUC
Early Stopping	0.862	0.687
Early Stopping & Reduced Features	0.743	0.685 (this was used to submit to Kaggle)

SMOTE

SMOTE and under sampling, with the early stopping, did not seem to help. The top five iterations were the down sampling of the positive class by 50% and the up sampling of the negative class by 120%, which only achieved a validation AUC of 0.634.

n_positive	n_negative	best_ntree	train_auc	val_auc
2650	2525	5	0.776968	0.633741
3025	2882	5	0.789681	0.632650
3025	2277	5	0.775853	0.625792
3524	2829	5	0.783247	0.625418
2650	2392	10	0.819528	0.625356

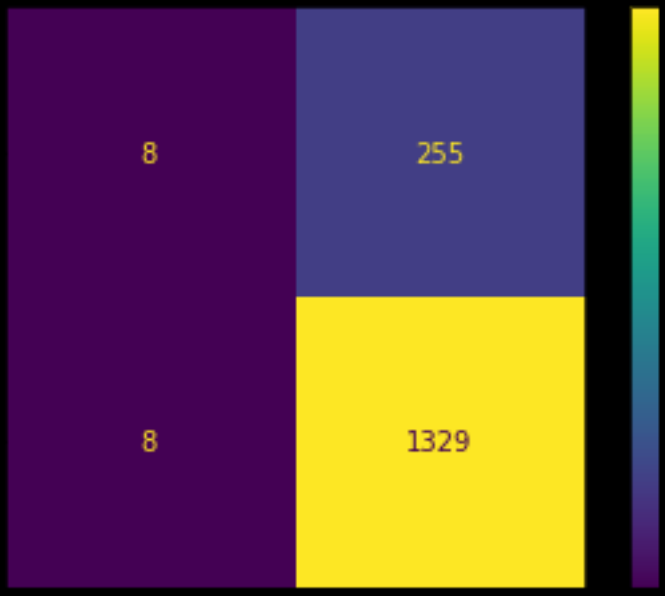
Calibration

Calibration did not perform very well. The validation AUC (0.677) is not as good as early stopping with reduced features (0.685) and the overfitting problem is also worse (difference between training and validation AUC of 0.14).

	Training AUC	Validation AUC
Calibration	0.994	0.638
Calibration & Reduced Features	0.817	0.677

Confusion Matrix

The result of the XGB classifier with early stopping and limited features still indicates a heavy bias towards the prediction of the positive class, with only very few predictions of the negative class.



Future experiment

It would be insightful to consider how the different values of the `pos_scale_weight` changes the results. Decreasing this parameter would result in more predictions of the negative class.

Test Set

The AUC in the test set was 0.654, which is considerably lower compared with the 0.685 of the validation set. Which suggests that the distribution of the test set is significantly different to that of the training set.

	## Future Confirm if this is the case through adversarial validation.
3.b. Business Impact	The issues from the previous experiment still persists, namely, misclassifications on the positive side. 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	A technical issue arose with the inability of the sklearn's interface to accommodate XGB Classifier's implementation of early stopping, which requires an evaluation set (`eval_set`) as a **fitting parameter**. sklearn's Pipeline object performs all the data transformations prior to the fitting step, whereas the evaluation set is used only after the initial preprocessing is done. The implication is that the interface cannot be used as conveniently as usual. This means that the all the steps that Pipeline performs need to be written manually.

---

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	# Modular design Modularising boilerplate functions in the src has turned out to be very useful indeed, this reduces cognitive load by a. keeping the notebook cleaner, due to less code b. keeping consistency between notebooks

Naming convention

The output of one notebook is an input of another notebook. Data sets and models need to be identifiable to the processes that produced them. Having a good naming convention helps in identifying which object is which.

Importance of user interface

Function design is quite important. Our team already had a `make_classification_report` that takes as argument the classifier, X, and y. I didn't realise its limitation until I needed to pass some arguments to the predict method of the classifier, e.g. the `best_ntree_limit`. This is not possible in the current function because it calls the predict method with X as an argument by default.

So I created another one that takes y, preds, and probs, which solved the problem because the prediction is done outside of the function.

Furthermore, from a software engineering point of view, the original function didn't adhere to the single-responsibility principle, i.e. the function is doing two things: predicting and producing the classification report.

4.b. Suggestions / Recommendations

Data investigation

Adversarial validation to discover differences in the training and test sets.

Modelling

Search through the space of `scale_pos_weight` for an XGB Classifier.

Best practices

1. Agree on a naming convention for different types of objects
2. Save objects often
3. Consider good function design