EXPERIMENT REPORT

Student Name	Kai-Ping Wang
Project Name	Kaggle Competition
Date	14/02/2021
Deliverables	1. Wang_Kai-Ping-Week2_assignmentB.ipynb 2. kpw_best_model_assignmentB 3. https://github.com/ronvoluted/kaggle-nba

---

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	Being able to predict if a rookie player will stay in the league for 5 years based on player's stats. If a player has longer career in the League, he can generate better popularity and stronger fan base, which has more commercial value for the company to invest and market. The ability to correctly predict a player's career longevity means the company wastes less money and can have better investment return. Failure to predict so accurately means worse investment return.
1.b. Hypothesis	Based on result from week 1, the BadStats approach doesn't improve the performance greatly as expected. When looking into the wrongly predicted records, it is noticed that some negative cases are very hard to be separated from positive cases. The assumption in this experiment is if the model focuses more on distinguishing the negative cases, then the general performance will be better. So far the feature importance shows that "GP" dominates above all other features. If we can group the dataset into smaller groups, and take away feature GP, other features will stand out further and potentially assist in model performance.
1.c. Experiment Objective	• Binned GP into smaller groups • Drop GP after it's binned • Train the model using each binned group, so other features shall stand out and assist further on the prediction By taking away the dominant feature, other features shall assist on the prediction performance.

---

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

Taken:

1. **Converted all values to positive** – This is a player's stats and not year to year tracking performance, so negative values don't make sense.
2. **Group GP into 5 groups** – Each group becomes a dataset to train one model, the idea is within this group, the GP difference shall be minimal enough to be ignored, and other features shall stand out and assist model performance.

Not taken:

1. **Resampling** – Have briefly tried SMOTE oversampling, and it has some minor improvement on the score. However, not significant enough to be included in the final submission.

2.b. Feature Engineering

Taken:

1. **Binned GP** – Use pandas.cut to bin GP into 5 groups, and can see that group 1, 4 and 5 don't have much negative cases, while group 2 and 3 have more negative cases. In stead of oversampling, the concept here is to limit the training set to a more balanced group, while GP is not as dominant as in general (whole dataset)

2.c. Modelling

Trained:

1. **XGBRFClassifier** – XGBoost has a number of features embedded in, and very powerful; hence, try it in this experiment to handle imbalanced data.
    a. **Max_depth** – `[2,3,4,5,6]`
        - *Minimum tree depth for base learners*
    b. **Learning_rate** – `[0.001, 0.03, 0.1, 1, 5, 100]`
        - *Update along with Max_depth to control performance, when the depth is larger and learning rate is smaller, the performance is generally better in this experiment.*
    c. **Scale_pos_weight** – `[0.2, 0.4, 1, 100]`
        - *According to documentation, this hyperparameter is used to control the positive cases weights, and the ideal value is (negative cases)/(positive cases)*
    d. **Subsample** – `[0.1, 0.3, 0.4, 0.6, 1]`
        - *This is to control how many observation being passed in before each tree grows, and to avoid overfitting.*

	2. **XGBClassifier** – XGBoost has a number of features embedded in, and very powerful; hence, try it in this experiment to handle imbalanced data. a. **Hyperparameter Tuning as above** 3. **Blending** – Create a collection of weak XGBoost classifiers with different datasets and the hyperparameters, then save the predicted probability as stage 1. Use the stage 1 result as input of stage 2 RandomForestClassifer to form a stronger classifier.

2. 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	AUROC score on Kaggle submission is - 0.70331 (lower than week 1 result) <u>Potential Root Cause:</u> 1. **More TN also causes more FT –** *Although more True Negative under this experiment, the False Negative has also grown, and it has dragged down the overall performance.*
3.b. Business Impact	As AUROC is 0.703, which is lower than last week's model. This experiment is not considered as an improvement, and should be reviewed on the approach and hypothesis. **In short, this model is not ready to replace last week's model.** Further review and improvement is required.
3.c. Encountered Issues	<u>Solved:</u> 1. **Want to focus on binned GP group 3, which has the most negative cases** • The idea is straight forward, but not sure how to include this, or how to use it to make final prediction/submission. Was trying to process the data row by row, and if the GP is in this group A, use model A. If GP is in group B, then use model B…etc. • Tried blending approach to tackle this. Have a model that is only trained using Group 3 data. Once it's fit, then use it to predict all dataset, and use the result as stage 2 input/weight for another classifier to come up a stronger performance. <u>Unsolved:</u> 1. **Scale_Pos_Weight doesn't seem to help out much in imbalanced data –** Tried setting the value to be (negative cases)/(positive cases), but the overall performance doesn't seem to improve as expected. 2. **Blending approach has too many moving parts and very sensitive** - I like the concept of blending and able to have different trained models contributing to the final weights. However, it's so flexible that it's easy to get lost in all possible combinations.

---

2. 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	Although the concept of blending is promising, and the idea of limiting dataset to a degree that dominate feature is no longer dominating, it doesn't help out too much on the model performance too much. The ensemble approach has a lot of moving parts, and it can have surprising effect to avoid the model overfitting. However, it's quite hard to decide what models to include in the collection, and hard to imagine how they would work together.

	It is still worth exploring in the future as many people online recommending it, but it definitely needs some more experiences and better knowledge to master this skill. XGBoost is also an interesting and powerful tool, there are so many parameters can be used. However, no a silver bullet in this experiment as it doesn't have dramatic improvement with this dataset.
4.b. Suggestions / Recommendations	As explained above, based on AUROC score 0.70, which is worse than week 1, this model should not be considered an update to last week's model. During researching, I found there is a GAN (Generative Adversarial Network) that has a generator and discriminator concept, which self-train each other to improve the accuracy rate of each other in classification problems. It is also a good candidate to tackle imbalanced data. I would recommend to explore further into this field as it seems to have great potential.