EXPERIMENT REPORT

Student Name	Kai-Ping Wang
Project Name	Kaggle Competition
Date	21/02/2021
Deliverables	1. Wang_Kai-Ping-Week3_assignmentC.ipynb 2. kpw_best_model_assignmentC 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 2, the binned_GP feature and blending approach wasn't generating enough improvement. Although more TN has been identified, but also created more FP; hence, no better performance. As the course mentioned KNN clustering and distance to the cluster center, which is in sync with the binned_GP concept, while it's a better grouping approach. This experiment is trying to use cluster as grouping approach, and use if-else to determine which model to use when predicting each observation.
1.c. Experiment Objective	• Instead of using binned_GP, we use cluster grouping to shrink down the scope. • Possible that each group has different focuses in determining if a player is worthy. • By finding better classifier in each group, in conjunction, they can improve the general 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.b. Feature Engineering

Taken:

1. **Use KNN clustering to group data** – This is to use natural/factual grouping and so the dedicated classifier can be specialized in understand this group better.
2. **TP and TN classifier predict probability result** – These two features are generated using classifiers that specialized/focused on TP or TN better, and expect this combined insights can be utilized by stacking approach to improve the overall performance.

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. **LogisticRegression** – It is still the best performed model so far, and it has better performance in certain cluster groups. a. **Penalty** – `[Elasticnet]` • *There are a number of features seem correlated, so want to use regularization to balance them off.* b. **L1_Ratio** – `[1~0]` • *Try the balance between L1 and L2 to see which one performs better.* c. **Solver** – `[Saga, liblinear]` • *Only Saga supports elasticnet, choose liblinear as it stats better performance on small dataset.* *d.* **C** – `[0.01~20]` • *Use this to control the strength of regularization.* 3. **Blending** – Create an if-else condition by using **cluster** feature, and allow the observation to use different model for prediction based on cluster grouping.

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	AUROC score on Kaggle submission is – 0.633 (worst result so far) <u>Potential Root Cause:</u> 1. **Not enough TN being identified** – This is completely against the original assumption unfortunately. The data set in each cluster group turns out to be harder to separate Positive case and Negative case. 2. **The TP and TN features not strong enough** – The selected TP and TN classifiers are not strong enough in their field, and the stacking is not providing good performance as the action of reducing overfitting is also turning the model to be more conservative. This caused more TP and less TN.
3.b. Business Impact	As AUROC is 0.633, which is worst so far, and much closer to 0.5. This experiment is considered as failure and a dead end. **There shall be no further experiment in this area/direction.**
3.c. Encountered Issues	<u>Solved:</u> 1. **Blending approach has many moving parts -** • This was last week's encountered issue, and based on the feedback, I have started with simple approach with one strong in TP and one strong in TN. Although the result isn't as good as expected, the issue was resolved and the experiment was able to proceed to generate insight. 2. **Use different dedicated models for each cluster group** – • This is also an issue from last week, and based on the feedback, I have updated the prediction process to be a for-loop. Handle each observation one by one, and with the id-else condition, I am able to resolve this issue. However, the overall result was far worse from expected.

---

3. 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	Tried a different approach with blending, which is allow dedicated models to handle dataset that's suitable for them. Original assumption was able to create better performance by having small specialized models working together. However, the main issue here is when splitting training set into many smaller groups, each group has a lot less observation. More groups to be created, less data in each group to form an effective classifier. Also when the data is highly imbalanced in a group, it can easily cause a false impression that the model is performing well in that group by having small amount of TP or TN. This can be very sensitive to unseen data, and have high risk in Production.

4.b. Suggestions / Recommendations	As explained above, the performance in this experiment is so bad that this path shall not continue further. During last week 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.