EXPERIMENT REPORT

Student Name	Kai-Ping Wang
Project Name	Kaggle Competition
Date	07/02/2021
Deliverables	1. Wang_Kai-Ping-Week1_assignmentA.ipynb 2. kpw_best_model_assignmentA 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	There are many positions and situations in basketball games that a player with niche skills and experiences is considered highly valuable. The assumption here is if a player is failing on stats across multiple disciplines, then he is less likely to stay in the League for long time. Hence, the experiment here is to look through all stats and find poor performing players in general and not in specific stats.
1.c. Experiment Objective	• Players with higher stats across the board would stay for 5+ years • Players with lower stats across the board are less likely to stay 5+ years • Players with special skills (really high in some stats) are still likely to stay 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

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. **Ensure all "made" <= "attempt"** – It doesn't make sense if made > attempt

Not taken:

1. **Resampling** – Have briefly tried oversampling and undersampling, and it has huge negative impact on the score. Left this preparation out for this round.

Important for future:

1. **Resampling** – There is dark art in mastering resampling to not overfitting or underfitting. It does have potential to bring the prediction further.

2.b. Feature Engineering

Taken:

1. **StandardScaling** – As I intend to utilize coefficient to find out which features are more important, and also using LogisticRegression.
2. **Recalculated percentage related features** – These values are off compared to the actual calculation.
3. **Added "BadStats" feature** – When selected stats are in the lower 25%, each stat is counted as 1, and average it to be BadStats. If a player is in lower end across the board, then this feature will be 1; on the other hand, it will be 0. This is to test our assumption in this experiment.
4. **Selected features** – Based on a number of iterations of LogisticRegression, removing some features with really low coefficient. After reevaluation, confirm the model performance is about the same.

2.c. Modelling

Trained:

1. **LogisticRegression** – Based on the assumption, lower stats across the board will attribute to 0, and higher stats across the board will attribute to 1. The concept of good and bad in each stat is linear, although the overall performance may not be, it has potential.
   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.*

2. **RandomForestClassifier –** Reference model to compete against. If there are deciding features in non-linear form, it shall get picked up by random forest, and can prove the assumption is incorrect.
    a. **Max_depth** – `[2, 3, 5]`
        - *To reduce overfitting in training.*
    b. **Min_samples_leaf –** `[2, 3, 5]`
        - *To reduce overfitting in training.*

<u>Skipped:</u>

1. **KNN –** Team member has tried it and it didn't go very well.

<u>Future Potential:</u>

1. Xgboost
2. ANN
3. Ensemble (combination of different classifiers)

5. 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 - 0.71038

Potential Root Cause:

1. **Imbalanced data** – The 0 class is around 16% of the whole dataset. This makes the prediction on 0 class harder.
2. **Assumption is incorrect** – There is a ceiling on this experiment, which can mean the assumption on what determines a player stay or leave in 5 years is incorrect.
3. **Missing decisive features** – There might be other factors in determining if a player would stay or leave, and also if it's willingly or unwillingly can be important for prediction.

3.b. Business Impact

As AUROC is 0.71, it can be interpreted that the model has 70% chance to correctly distinguish between stay or leave after 5 years. It is an okay performance, and it will be great if it can be around 80%.

In terms of business impact, there is 30% chance the model cannot correctly distinguish between stay or leave after 5 years. This is still better than assuming most of them will stay (50%). It can save the business some investment, but it may not reach the ideal state, which depends on the business requirement/decision.

3.c. Encountered Issues

Solved:

1. **Train-Test-Split random state affects the final model** – Use cross validation on model selection, and then use the whole training set as input to train the classifier.
2. **How to feature engineer to test assumption –** It was challenging to convert the idea of "perform poorly across the board" into feature engineering at first. However, using flag of each stat, and then put them together as an overall indication seems fair. Although, it didn't greatly improve the model prediction performance.

Unsolved:

1. **Oversampling** – Tried oversampling with SMOTE and sklearn.resampling. Although, the local result (cross validation) is great, it actually has poorer performance in submission.
   - Not using it at the moment
2. **Undersampling –** Tried oversampling first, then random undersampling. The performance is greatly worsened.
   - Not using it at the moment

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	Resampling sounds impressive on paper, but it is very hard to master. It has the risk of overfitting and data leakage, and also the risk of removing important observations. It’s easy to spend a lot of time to tune there with much poorer performance. Although a lot of experiments have gone in, there are still a number of models can be tried. I think we should still continue with more experiments.
4.b. Suggestions / Recommendations	As explained above, based on AUROC score 0.71, the model’s performance has 70% chance to correctly distinguish stay from leave after 5 years. Which is much better than baseline, assuming most rookie players will stay. **The model can be put into Production to start saving investment cost.** Further experiments are recommended to improve the AUROC score to closer to 0.8.