EXPERIMENT REPORT

Student Name	Sampath Pitchandi
Project Name	NBA Career Prediction
Date	14-Feb-2021
Deliverables	Pitchandi_Sampath_Week2_01_01-data-analysis-and-clean.ipynb [Mandatory] Pitchandi_Sampath_Week2_01-02-train-RF-iteration1.ipynb Pitchandi_Sampath_Week2_01-03-predict.ipynb Pitchandi_Sampath_Week2_02_01-data-sampling.ipynb [Mandatory] Pitchandi_Sampath_Week2_02-02-train-RF-Upsample-iteration2.ipynb Pitchandi_Sampath_Week2_02-03-train-RF-Downsample-iteration2.ipynb [Scored] Pitchandi_Sampath_Week2_03-01-train-XG-iteration1.ipynb Pitchandi_Sampath_Week2_03-02-train-XG-Upsample-iteration2.ipynb Pitchandi_Sampath_Week2_03-03-train-XG-Downsample-iteration2.ipynb [Scored] https://github.com/ronvoluted/kaggle-nba/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	----- From the Previous Experiment ----- The main objective of this project will be to determine/predict if an NBA rookie will last at least 5 years from the given set of data. With this experiment, we are trying to see if a rookie player will be able to survive the first 5 years of the NBA league. Joining the NBA league is a big deal for any basketball player. And this prediction model can be used by various sets of people to track the performance of each player through their journey. For sports commentators & fans, they can keep track of the most important players. And for the teams, they can focus on picking the right players from the very beginning based on the predictions.

1.b. Hypothesis	In Assignment 1, I did notice that the data was imbalanced, but I didn't realize that the prediction class is abundant. This is something that I have not encountered so far. In most cases, the class that needs to be predicted is always the minority. So, in this experiment, I'll be manipulating data to change that and run a few tests. **Choice of Model**: Random Forest and XGBoost I believe I have not trained them well in the first experiment. XGBoost to see how it compares with the Random Forest. **List of Changes:** Data Manipulation: - Try to train a model to predict class zero, meaning players who won't last five years in the NBA league. - Up-Sample - Down-Sample Set of Model Training: - Train a normal RF to get a baseline the score - Train an RF with Up-sampled Data - Train an RF with Up-sampled Data - Train a normal XG to get a baseline the score - Train an XG with Down-sampled Data - Train an XG with Down -sampled Data I'll be reversing the class and trying to identify players who will not last 5 years in the league.
1.c. Experiment Objective	In the previous experiment, I trained an RF for binary classification problems. The data was imbalanced, so I played with the CLASS_WEIGHT hyperparameter. I got a score of 0.69437. The objective of this week's experiment is to up and downsample the data to get a better prediction than the previous experiment.

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

The CSV was loaded into a data frame without any issues. There are 8000 rows and 22 columns. Checked the following to see if there were any anomalies in the data

- duplicate check using `duplicated()`
- descriptive statistics to check data issues using `describe()`
- datatype check using `info()`
- null value checks using `isna().sum()`

----- UPDATES -----

Looking at descriptive status found negative values in the following columns.

- 'GP' , '3P Made', '3PA', '3P%', 'FT%' , 'BLK'

Investigating further I found that the percentages of these columns are incorrect. FG%, 3P%, FT% were not representative of the actual data. Also, there were 0 and negative values in '3P Made', '3PA', which would result in incorrect (or inf) values when calculating percentages.

To fix these issues, I applied the following formula

- Update all negative column values as zero
- If values of FGM, FGA, 3P Made, 3PA, FTM, FTA is zero then update all the column values as zero, the **new percentage column is also set as zero**
- Set the new percentage column for each category of data for all non-zero records using the formula.

CALCFG% = FGM/ FGA*100
CALC3P% = 3P Made/3PA*100
CALCFT% = FTM/ FTA*100

- Create a new target column called **'Target_5Yrs_Inv'** to store the inverse of **'Target_5Yrs'**. I made this change as I was not able to tell if the model is doing well based on the RECALL and PRECISON scores.

Stats of Rows & Features used for the experiment
Number of Rows: 8000
Number of Features: 22

2.b. Feature Engineering	Using Tree-based algorithms. Did not perform any feature engineering.
2.c. Modelling	Selected to use Random Forest and XGBoost to predict the career of NBA players to start with.

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

A score comparison between the models trained in Assignment 1 (using Random Forest + CLASS_WEIGHT tunning) and Assignment 2 (Random Forest & XGBoost + Data Sample Up and Down) is given in the table below.

I've only mentioned details of the model that resulted in the highest score.

Model	Accuracy	F1 Score	Precision	Recall	ROC_AUC	Kaggle Score
Baseline	0.83	0.9	0.83	1		
Assignment 1 - Random Forest + CLASS_WEIGHT						
Train	0.74	0.83	0.92	0.76	0.79	
Validate	0.66	0.77	0.87	0.71	0.66	**0.69437**
Test	0.68	0.79	0.87	0.72	0.65	
Assignment 2 - Random Forest + Down Sample						
Train	0.99	0.99	0.99	0.99	0.99	
Validate	0.7	0.63	0.63	0.62	0.63	**0.70416**
Test	0.69	0.62	0.59	0.56	0.63	
Assignment 2 - XGBoost + Down Sample						
Train	0.71	0.65	0.67	0.7	0.64	
Validate	0.66	0.64	0.64	0.65	0.64	**0.69223**
Test	0.66	0.62	0.61	0.6	0.63	

As suspected, the downsampled data performed better than the upsampled data. Having said that, these are not the best score. I think this algos can do better than this.

3.b. Business Impact

<u>Observed result based on the initial hypothesis:</u>

The model is getting better. However, it can be tuned a lot better than this. Feature importance has changed with the updated datasets.

- Random Forest top five feature are:
    1. GP,
    2. CALC3P%,
    3. MIN,
    4. CALCFT%,
    5. FTA.
- XGBoost top five feature are:
    1. CALCFT%,
    2. GP,
    3. CALCFG%,
    4. MIN
    5. FGA

	----- From the Previous Experiment ----- Observed result based on the initial hypothesis: As expected, the most important features are games played and there is a high importance on the number of field goals actual or attempted. Players with high numbers in these two areas are predicted to have at least a 5-year NBA career. There are other important features however these two stood out by a huge margin. The model was able to learn this relationship. Impact on the result if the model could not learn this relationship: If the model had learned incorrect patter, then the model would have not picked the right players. If these predictions were just used by fans and commentators to track the players' growth, then it would have only resulted in disappointment (no serious implications). However, one potential issue would be if this result was being used by a PRO team in picking the payer to invest in their future then it would have resulted in reduced ROI compared to the correctly predicted player.
3.c. Encountered Issues	Hyperparameter Tuning: - There are too many hyperparameters to keep track of. - Hyperopt is not an easy beast to tame. Model Training: - I don't have an efficient training process. I think I made a lot of mistakes during the training. It is hard to keep track of the variables through the process. - A lot of model training results in a lot of outputs. It gets hard to look at the results scrolling the page up and down. Metric - There are too many metrics that evaluate the results in different ways.

---

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	Hyper Parameter Tuning: - Tried to create a dataframe with a list of all the hyperparameters to keep track of all values that are in play. - Model Training: - Tried to create a function to loop through the dataframe with the list of hyperparameter values. It has helped me logically organize the results bit better. - I need to find a way to consolidate the results in a more concise manner that is easy to understand and interpret. - A small change in one of the parameters can cause the model to train better. Case in point, with XGBoost, I was missing the OBJECTIVE hyperparameter. Once I added that to the mode, the results were better. Metric - It is essential to understand what metric to use. - Both RF and XGBoost, there are many metrics to evaluate. Picking the right one can be the key to solving the puzzle.

	Result: - While writing this report, I see the RF model for Assignment 2 looks like the Training model is overfitting and the Validate model is underfitting. - It is a repeat of what I already know, understanding data is most crucial to prediction. - There are many ways to go about training a model. In the first assignment, I tried to train a model to predict the CLASS =1, in this assignment I tried to predict CLASS=0. The results are a bit better than the previous assignment. - I'll try to streamline the process using python functions to handle the number of variable assignments. - I'll try to streamline the results for easier comparison and interpretation.
4.b. Suggestions / Recommendations	1. Perform Global Interpretation to understand the results. And see if this information can be used to train models. 2. Try SMOTE to see if performance gets any better 3. Train Random Forest and XGBoost