EXPERIMENT REPORT

Student Name	April Nommesen
Project Name	NBA Career Prediction
Date	16/11/2022
Deliverables	notebook names: • nommesen_april-week2-prepare_data.ipynb • nommesen_april-week2-logreg0-allfeat.ipynb • nommesen_april-week2-logreg2-resampled.ipynb • nommesen_april-week2-svc0-allfeat.ipynb • nommesen_april-week2-randomforest1.ipynb • model name: • logreg0 • logreg2 • scv0 • randomforest1 GitHub repo: https://github.com/aprilgum/adv-dsi-2022-at1-grp4/tree/master/nba-career-prediction

---

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	Explain clearly what is the goal of this project for the business. How will the results be used? What will be the impact of accurate or incorrect results? Rookie players joining in the first round of NBA draft are given four-year contracts where the first two years of the contract are guaranteed with the NBA team and the third and fourth years can change. Rookie players joining in the second round of NBA draft and undrafted players can sign contracts that can be anything from one year to four years and that are either fully guaranteed or not guaranteed at all. The value of rookie players contract is tied to the salary cap and the team they will play

	for. A rookie player who lasts for at least five years are considered successful and a player who does not last at least five years is considered risky. Being able to have a data-based decision to decide whether a rookie is a potential success or a potential risk greatly impacts the team's budget to pay the players' salary and also impacts the team's performance.
1.b. Hypothesis	Present the hypothesis you want to test, the question you want to answer or the insight you are seeking. Explain the reasons why you think it is worthwhile considering it, I would like to begin with the question, who are the rookie players who are likely to last at least 5 years? It is worthwhile to check both sides – whether to predict who are risky (ie not last 5 years) or who are potentially successful.
1.c. Experiment Objective	Detail what will be the expected outcome of the experiment. If possible, estimate the goal you are expecting. List the possible scenarios resulting from this experiment. To try different methods of handling imbalanced data. I expect that the model would be able to better detect the cases where y = 0 (career years played < 5). I expect the accuracy could be lower but I am hoping the AUROC would be at par or better. This week, I would only focus on imbalanced data and not experiment on different hyperparameters or feature selection or transforming features.

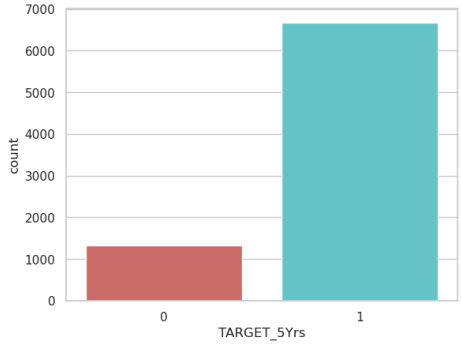
2. EXPERIMENT DETAILS

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

Describe the steps taken for preparing the data (if any). Explain the rationale why you had to perform these steps. List also the steps you decided to not execute and the reasoning behind it. Highlight any step that may potentially be important for future experiments

The graph below shows the frequency of the classes in the target variable.



In Logistic Regression - Resampled:

I split the working data (excluding the separate test data) into 80% training and 20% validation.

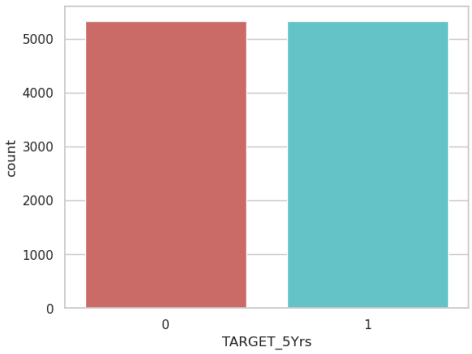
I upsampled the y = 0 only in the training data.

Original:

```
1 5326
0 1074
Name: TARGET_5Yrs, dtype: int64
```

Resampled:

```
1 5326
0 5326
Name: TARGET_5Yrs, dtype: int64
```



In SVC, I used the parameter class_weight='balanced'

2.b. Feature Engineering	Describe the steps taken for generating features (if any). Explain the rationale why you had to perform these steps. List also the feature you decided to remove and the reasoning behind it. Highlight any feature that may potentially be important for future experiments. For this week, I used all the features/independent variables. I did not transform any of the variables except resampling the target variable.

2.c. Modelling	Describe the model(s) trained for this experiment and why you choose them. List the hyperparameter tuned and the values tested and also the rationale why you choose them. List also the models you decided to not train and the reasoning behind it. Highlight any model or hyperparameter that may potentially be important for future experiments
	
	1. Logistic Regression All Features (LOGREG 0)
	No standardisation
	Default parameteres in logistic regression
	No resampling
	I started with this to see how it would perform accuracy and AUROC. I expect that this model would see more and predict more of the y = 1 (career years played >= 5).
	
	`# instantiate the model (using the default parameters)`
	`Logreg0_train = LogisticRegression(random_state=16)`
	
	`# fit the model with data`
	`Logreg0_train.fit(X_train, y_train)`
	
	https://github.com/aprilgum/adv-dsi-2022-at1-grp4/blob/master/nba-career-prediction/notebooks/nommesen_april-week2-logreg0-allfeat.ipynb
	
	2. Logistic Regression Resampled (LOGREG 2)
	All features
	No standardisation
	Default parameteres in logistic regression
	Up-sample the minority class as described in section 2.a
	
	`# instantiate the model (using the default parameters)`
	`logreg2_train = LogisticRegression(random_state=16)`
	
	`# fit the model with data`
	`logreg2_train.fit(X_train, y_train)`
	
	https://github.com/aprilgum/adv-dsi-2022-at1-grp4/blob/master/nba-career-prediction/notebooks/nommesen_april-week2-logreg2-resampled.ipynb
	
	3. SVC All Features (SVC0)

No standardisation
Using SVC to handle imbalanced data

```
# instantiate the model (using the "balanced" mode)
svc0_train = SVC(kernel='linear',
                 class_weight='balanced',
                 probability=True)

# fit the model with data
svc0_train.fit(X_train, y_train)
```

https://github.com/aprilgum/adv-dsi-2022-at1-grp4/blob/master/nba-career-prediction/notebooks/nommesen_april-week2-svc0-allfeat.ipynb

4. Random Forest Classifier with Balanced Weights
No standardisation
Using Random Forest Classifier to handle imbalanced data

```
# instantiate the model (using the "balanced" mode)
randomforest1_train=RandomForestClassifier(n_estimators=900,
class_weight = 'balanced')

# fit the model with data
randomforest1_train.fit(X_train, y_train)
```

Experimented on the following parameters:

- `RandomForestClassifier(n_estimators=900, class_weight = 'balanced')`
- `RandomForestClassifier(n_estimators=100, class_weight = {1:0.60082614, 0: 2.97951583})`
- `RandomForestClassifier(n_estimators=1000, class_weight = {1:0.60082614, 0: 2.97951583})`
- `RandomForestClassifier(n_estimators=2000, class_weight = {1:0.60082614, 0: 2.97951583})`
- `RandomForestClassifier(n_estimators=2000, class_weight = 'balanced')`

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

Score of the relevant performance metric(s). Provide analysis on the main underperforming cases/observations and potential root causes.

Model	Accuracy	AUROC	
Logistic Regression All Features (LOGREG 0)	0.84	0.707	Though this looks like the "best", I don't think this is a good model because it tends to predict most of the test data as y=1.
Logistic Regression Resampled (LOGREG 2)	0.62	0.699	
SVC All Features (SVC0)	0.63	0.706	This is my best output for this week. I will still continue experimenting on the SVC after a tried a few more in the "pipeline"
Random Forest Classifier with Balanced Weights			I will still continue experimenting on the Random Forest Classifier after a tried a few more in the "pipeline". The accuracy is good and maybe there could be a way to increase the AUROC since I have only used default parameters this week.
`RandomForestClassifier(n_estimators=900, class_weight = 'balanced')`		0.666	
`RandomForestClassifier(n_estimators=100, class_weight = {1:0.60082614, 0: 2.97951583})`		0.661	
`RandomForestClassifier(n_estimators=1000, class_weight = {1:0.60082614, 0: 2.97951583})`		0.676	
`RandomForestClassifier(n_estimators=2000, class_weight = {1:0.60082614, 0:`		0.676	

	2.97951583})			
	RandomForestClassifier(n_estimators=2000, class_weight = 'balanced')	0.84	0.678	

3.b. Business Impact	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)
3.c. Encountered Issues	List all the issues you faced during the experiments (solved and unsolved). Present solutions or workarounds for overcoming them. Highlight also the issues that may have to be dealt with in future experiments. There are too many ideas to try. I need to prioritise what I'd would look at first.

---

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	Reflect on the outcome of the experiment and list the new insights you gained from it. Provide rationale for pursuing more experimentation with the current approach or call out if you think it is a dead end. My python programming has improved. There is definitely more our team can do with the experimentations. It would like to produce a model that looks better than Logistic Regression All Features (LOGREG 0)
4.b. Suggestions / Recommendations	Given the results achieved and the overall objective of the project, list the potential next steps and experiments. For each of them assess the expected uplift or gains and rank them accordingly. If the experiment achieved the required outcome for the business, recommend the steps to deploy this solution into production. Next week: I will no longer experiment on models where the target is imbalanced. I can look at changing hyperparameters for SVC and Random Forest Classifier. I would try XGBoost. Next couple of weeks:

	I would like to think about re-framing the problem and predicting who are risky ie y = 0 (career years played < 5). This requires some transformation so I can still submit the probability of y=1 for Kaggle.