**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Machine Learning |
| **Assessment Title:** | CA2 for AI |
| **Lecturer Name:** | Muhammad Iqbal |
| **Student Full Name:** | André dos Santos Calheiros |
| **Student Number:** | sba23032 |
| **Assessment Due Date:** | 07/01/2024 |
| **Date of Submission:** |  |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

Understanding the data

First we need to understand the problem at hand, we are asked to prepare and analyse the data set, and we are asked to have job satisfaction and productivity as goals. Looking at the columns and their descriptions in the attached document we may consider what are the 3 important factors, job satisfaction (this would be the main factor for “employee satisfaction”), performance (this shows how productive an employee is) and attrition (it is more likely an employee will either leave the company if they are unsatisfied or be fired if unproductive).

After this understanding we use pandas profiling to get a wide view of the dataset, looking at the parameters we can see:

* Job satisfaction has a rather even distribution, although we can see a high concentration of low satisfaction results and a wide variation of higher values and the average value is in the mid values;
* Performance rating tends to be on the middle, with most values falling close to this average;
* Attrition is an unbalanced class with just 2 classes, if we chose this as the target in supervised models it is a binary option.

We can also see there are plenty of columns (including the ones above) with about 10% of their data missing, but if we try removing all rows with null values, we verify those are not the same rows and the dataset becomes too small should we drop all the rows containing such.

As such we have 3 options: replace values, drop columns/rows or use an algorithm that can deal with those.

Of the initial dataset I also decided to drop:

* “Over18”, row as only the true value exists there and if relating to the age only 7 values would be switched to false;
* “EmplyeeCount”, as no proper reason is given in the data dictionary and it seems somewhat constant as mentioned there;
* “MonthlyIncome”, redundant with either daily or monthly rate and standard hours mix, could have been used instead of that set
* “YearsSinceLastPromotion” redundant with current role, as current role includes non-promoted roles.

All other columns will have to be sorted as we do the models.

Models choices:

The first choice after data cleaning one can do is to select the models, supervised models tend to do better on many cases as we often want to find out how likely a thing is when other factors are true. On the other hand unsupervised models are better at finding general relations, those may not show the data with interest points we are looking for although this can be improved by modifying hyperparameters until we get our targets.

Since we did decide on a target on the dataset, the focus will be the the supervised, but doing an unsupervised model can also be interesting in understanding the factor connections.

As for the type of model thee choice of main factor as attrition makes it most interesting to do a prediction model, classification would be better fit for something as JobSatisfaction, or other linear values. Clustering could also be another interesting option that could show relation between multiple factors creating groups of similar rows.

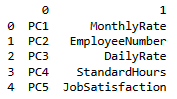
Any model we choose has multiple hyperparameters that the ideal results vary by dataset, as such they needed tuning as seen bellow.

Unsupervised model: PCA

On unsupervised models I have decided to run PCA, for this we modified all the null values to be a faraway outlier (-999), so it would not be fit into the model as they are only a small percentage of the values, to see if within the first half of the relational importance we would get any of 3 parameters we may want. The choice of this model is to try and understand what are the main relational factors in the dataset.

The results we got by increasing the hyperparameter of “n\_components” in order to view our targets:

* Job satisfaction

Fig1. Important factors, cluster of 5, the min n\_components for Job satisfaction

We can see for an employee to be satisfied with the job the most important factors seem to be monetary.

* Attrition

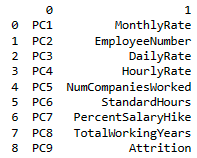
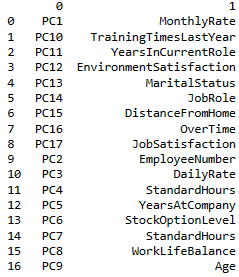


Fig2. Important factors, cluster of 9, the min n\_components for Attrition

As we can see here the important factors are still monetary, however the number of years working on a company and how the salary hike occurs over a total of working years, this shows employees like to see recognition of their work and seniority.

* Productivity

Fig.3 Important factors, cluster of 17, a failed experiment to fit productivity in a cluster

We could see that considering more than half the factors in the dataset, no better set explained cluster would include the productivity, so we can conclude it is a more complicated result and widely varies. This would be the predicted outcome as we saw in the pandas profiling the values of productivity were highly concentrated around the average.

From here we can see how difficult performance is hard to evaluate and job satisfaction is a rather shallow evaluator, but we can advise what factors lead to it. This also points us to the option that we’ll use in supervised learning, with attrition.

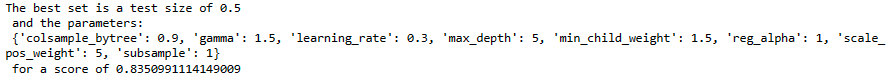
Supervised learning: XBoost

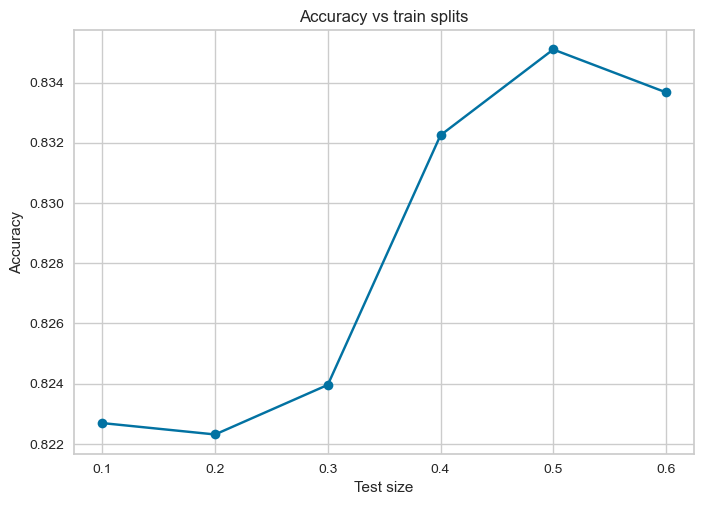
While there are many options for supervised learning not all of them can deal with null values or imbalanced classes, this is the reason why I chose to use the xgboost classifier for a random tree algorithm with a binary option, this algorithm accepts null values in the features columns and allows us to set ‘scale\_pos\_weight’ to deal with unbalanced classes, so we just need to deal with the null values in the target “Attrition” column, since there is less than 50% I’ve decided to drop those rows.

We initially set 2 hyperparameters, objective as binary and evaluation of the model as error.

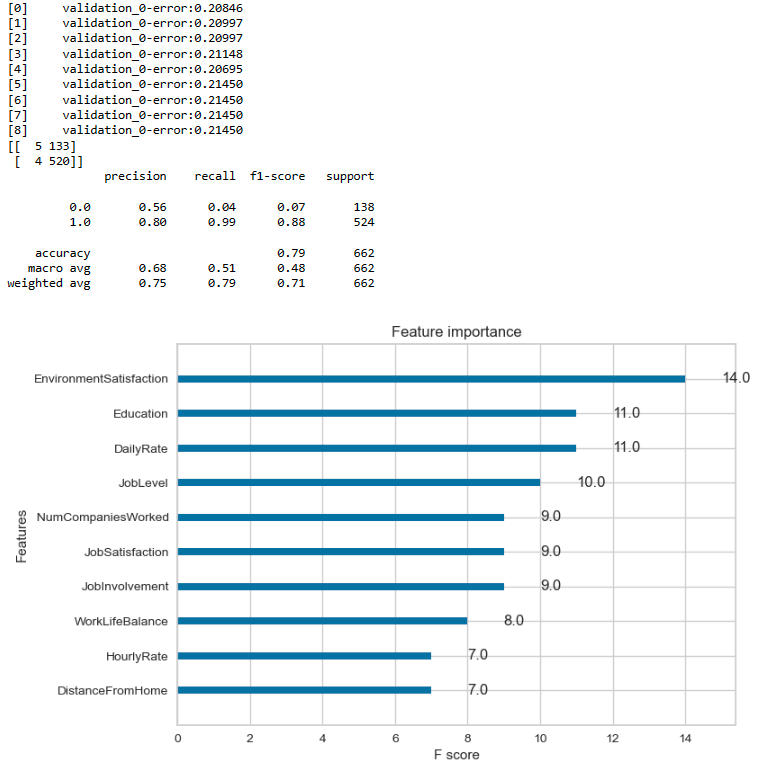
Unbalanced classes

I use a gridsearch to find the set of parameters with the best accuracy, we ended up with:

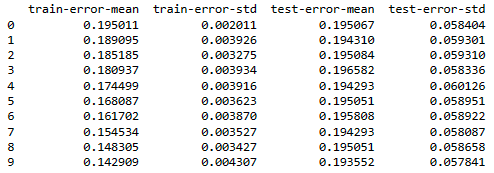
Fig4. Output from Gridsearch

Fig5. Accuracy vs test size

With using this we get a result of:

Fig6. Confusion matrix, classification report and Feature importance from unbalanced classes set

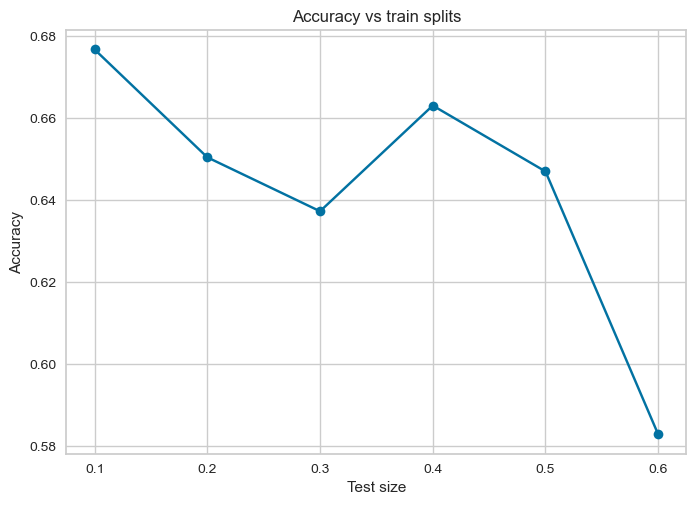
In this we can see the most important factor why one should leave is job satisfaction, Education and daily rates come next, this makes it, the number of companies someone worked, how satisfied they are with their job and how involved they are comes next, therefore offering good should be the focus as we can see although the chances for someone not leaving tend to be harder to guess than those of someone leaving as seen in the confusion matrix, having only 56% precision makes It a slightly better than random guess , it is very good at guessing if they will leave, having a 80% precision for this, and as a binary option this is still gives us reasoning to guess the opposite option, and an overall 79% accuracy.

Fig7. Folds tests for unbalanced classes

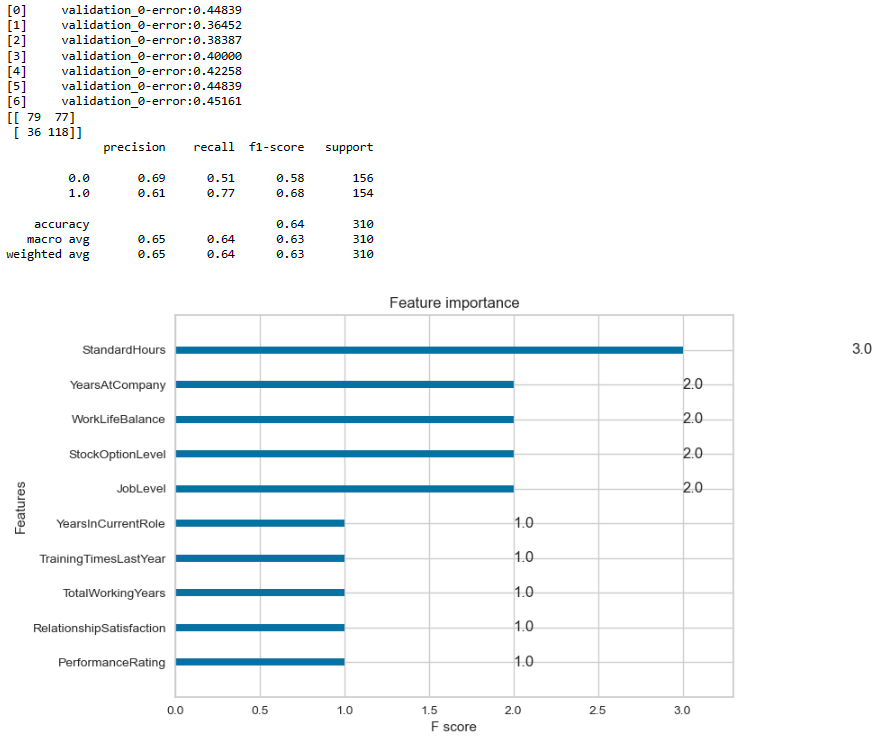
As we can see the model found with unbalanced classes also tends to have a somewhat low test error, although it has some variance on the training.

Balanced Classes

We also considered rebalancing the classes but the results of such led to a lower accuracy as can be seen bellow:

Fig 8. Balanced classes by undersampling accuracy vs splits

I chose undersampling as due to plenty of missing data I could choose to input it but would decrease the accuracy and the algorithm used is able to deal with missing data in the features set.

Fig.9 The results of the under sampled test

Something interesting in this under sampling, and the reason why I keep it here, is that despite its lower accuracy and lower precision guessing if a person will leave the company, it has a higher precision in guessing if they will stay in the company, so depending on the costumer this may be a more desirable model.

Conclusion