**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:** | 05/01/2024 |
| **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.

To the effect we run a PCA model with all the null values changed into an obvious outlier (-999), this is possible since only a small percentage of each column’s values are null, this was chosen as a model that tends to deal well with outliers if they exist in small quantities.

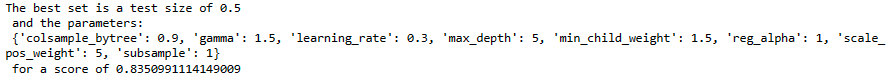
From the PCA we can see:

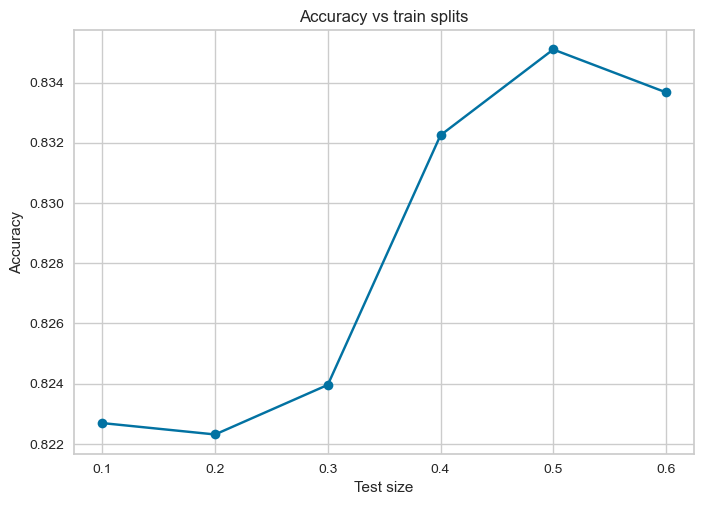
* Job satisfaction is mostly related with salaries (monthly and daily rate), employee number and standard hours.
* Attrition the same factors are related, alongside with the number of companies they worked what salary hikes they have and how many years have they worked.
* For productivity I went up to a total of 17 parameters and it seemed not to appear in any high related set, so as would be guessed from the high concentration of the values seen in the profiling there seems not to be any factor set that relates more or less to it.

As such attrition seems to be the best option for the supervised algorithms is to go with Attrition, and if we need to deal with null values we could drop the columns other than the 9 columns found in the pca and then work reconsider what to do

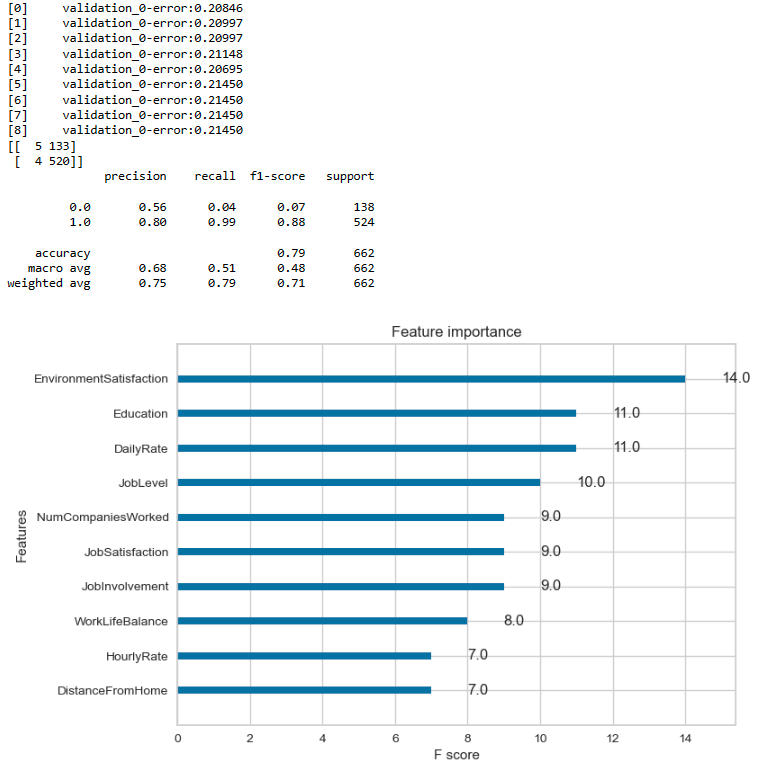
Supervised learning:

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, after selecting the objective as binary and evaluation of the model as merror I use a gridsearch to find the set of parameters with the best accuracy, we ended up with:

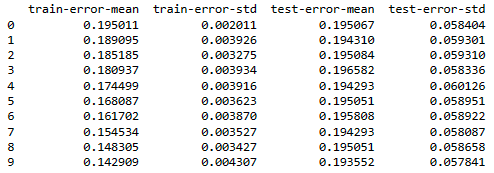
Fig1. Output from Gridsearch

Fig2. Accuracy vs test/train

With using this we get a result of:

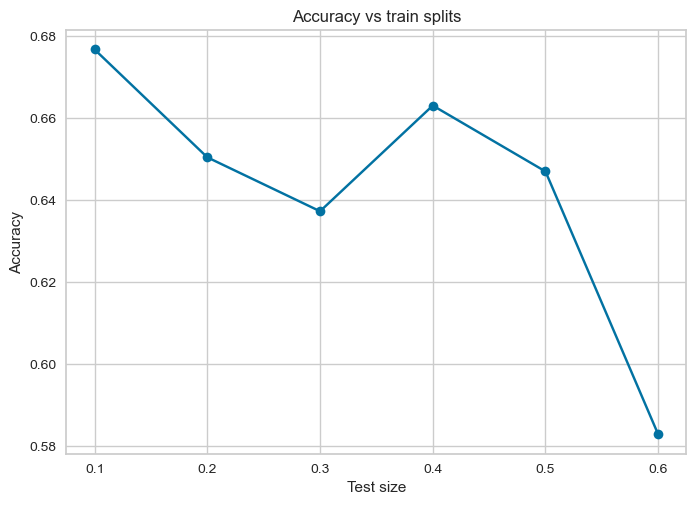
Fig3. Confusion matrix, classification report and Feature importance from unbalanced classes set

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 tho 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 , it is very good at guessing if they will leave, having a 80% precision for this, and an overall 79% accuracy.

Fig4. Folds tests for unbalanced classes

As we can see the model found with unbalanced classes also tends to have a low error

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

Fig 4. 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.

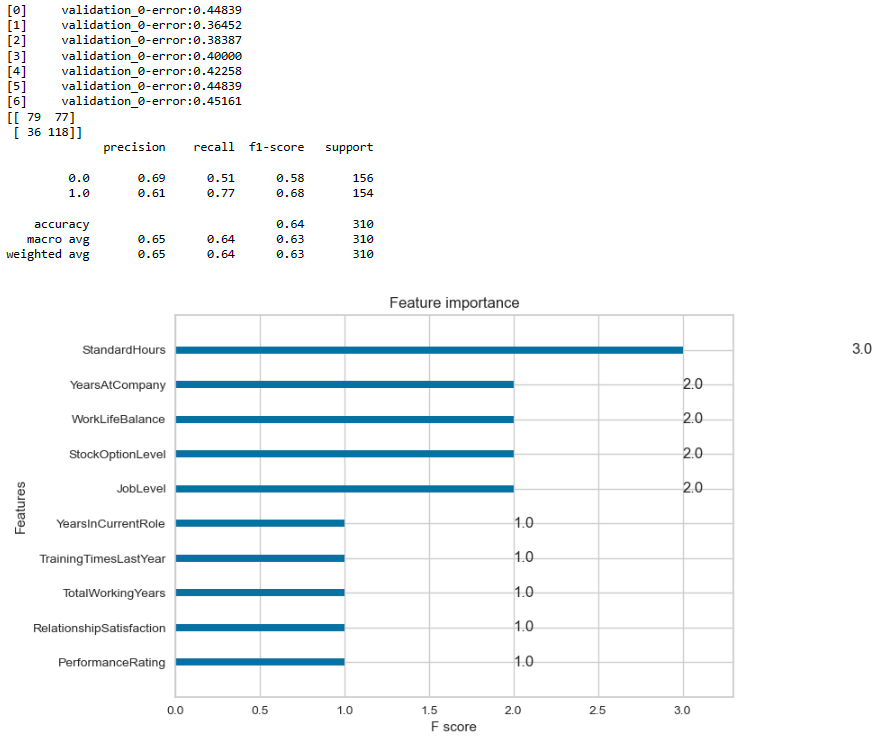


Fig.5 The results of the under sampled test

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.