Newland - The City of the Future : Predicting The Income Class Based on The Average

Mariana Byrne (m20200638@novaims.unl.pt), Luisa Barral (m20201045@novaims.unl.pt), Edgardo Juarez (m20200749@novaims.unl.pt), Li-lou Dang-Thai (m20200743@novaims.unl.pt)

Abstract - To ensure a more financial and economic sustainability the government of the city Newland, is seeking support from data scientists to create a predictive model that is able to determine the social class of every newcomers. In this project we first analyzed the dataset, using the Pandas and DTale libraries. We then proceeded in doing feature engineering techniques to structure the variables of our data. Next, using over 30000 training datas, our team explored different feature selection models : Recursive Feature Elimination (RFE), and Mutual Information Classifier (MIC). After selecting the most important features, we applied Neural Networks, Logistic Regression and the Gradient Boost Classifier. We manipulated the different models and evaluated their results. Finally we chose the Gradient Boost Classifier as our best predictive model, which gave us an f1 score of 0.84554.

I. INTRODUCTION

A few years ago, when the climate change became unbearable and the Earth turned into a hostile environment for all living lives, a brand-new habitable planet was discovered in our galaxy. Following these events, a mission was launched to inhabit it. Two years after the mission 'Newland' was introduced, one hundred new spaceships were on their way to the new planet. In order to make this new home more financially sustainable, the Newland government decided to establish a tax rate — that would divide the population into two social-economic classes — individuals who earn more than the average salary will be subjected to a rate of 30%, while the remaining will have a 15% income tax.

The selection was nonetheless controversial. The participants were chosen - out of volunteers - based on certain characteristics, others were paid to participate, and some paid to participate. Our training dataset contains the income analysis of 32500 individuals from Newland, with the different evaluation features.

Due to a request from the Newland government data science department, we decided to attempt to create a prediction model, which will determine which social class each individuals belong to based on multiple variables. In our project, we apply a mixture of data mining and machine learning methods. First we will discuss about the theoretical aspect of the new technique we used, then analyze the feature selection of our data, following this we will compare the performance of the different models, and finally discuss our results.

II. BACKGROUND

Most of the work done on this project has been done through machine learning techniques learned in practical classes. However we had to look outside the scope to find new alternative to analyze data and improve our model. In order to deepen our analysis of the data, we chose to use the DTale library. The visualization of the data is similar to Pandas, although it has a more interactive interface. [1] We mainly use the describe function - which is similar to the Pandas Profiling.

Our training data is composed by mostly categorical features, in order to reduce the dimensionality of the dataset, we decided to use Mutual Information from the sklearn.feature_selection library. The Numpy library is required in order for it to work. This function [2] measures the relation between variables. A result close to 0 shows that the variables are independent, while a higher value demonstrates a strong correlation between the features. Mutual Information relies on the entropy estimation from KNN distances. It is express as follow :

$$I(X ; Y) = H(X) - H(X \mid Y)$$

where I is the Mutual Information of two variables X and Y, and H is the entropy. [3] The function mutual_info_classif takes 2 required parameters : X as the dataset and y as the target. It returns an array of the mutual information between the target and all the different features. [2]

III. METHODOLOGY

This section presents the methods and tools that were used, a brief description and their relation with the project objectives, ordered by the steps 'Data Cleansing', 'Feature Engineering' and 'Predictive Model Development'.

Data Cleansing :

The database used was provided as the Excel document 'Train.xlsx'. To facilitate the data exploration our team began to use the D-Tale Python library at first. [1] The 'Describe' function provided data visualizations that helped to identify several characteristics of the data — the null values, data distribution and variety of values — which are essential information that will help for our future decision making.

Within the features "Role", "Base Area" and "Employment Sector", we found cells that were filled with a question mark. Due to their lack of representation, we classified them as null values and then replace them with their respective category mode. After exploring the data, we discovered that some of the values appeared to be redundant or open to misinterpretation, and decided to make several modifications, those are listed below.

- 'Birthday' was modified from a string type to an integer by subtracting the year of the date of birth to the actual year.
- 'Education Level' was inferred to be a redundant variable, since the dataset has another feature called 'Year of Education'.
- From the 'Name' column, we created a binary feature : 'Gender' which replaced it.
- We assumed that the different values of public and private types of 'Employment Sector' had similar effect on the Income variable, therefore they were merged as 'Public' and 'Private' values respectively.
- Regarding the values 'Married - Spouse in the Army' in the 'Marital status' variable, we decided to merge it with 'Married' values, considering that it had a similar impact on the Income variable.
- Furthermore, we suspected a similar influence on the Income variable regarding 'Husband' and 'Wife values in the 'Lives with' column, and chose to combined the two variables as 'Spouse' values.
- Lastly, due to its distribution and low variety of value types 'Ticket Price' and 'Money Received' features were transformed into binaries : 1 meaning the event happened and 0 meaning it didn't.

Regarding the categorical features, we decided to rely on the One Hot Encoding process, as it keeps each of the values weight, and our data will not generate too many new features. Due to its high variety, 'Base Area' was analyzed independently from the other categorical features.

In order to analyze the relationship between the features, we used the Pearson's correlation. 'Lives with Spouse' presented a high correlation (higher than .9) with 'Marital Status', thus it was the first variables we dropped from the analysis.

Feature Selection :

Two algorithms were selected to support our feature selection. Recursive Feature Elimination (RFE) [4] and Maximal Information Coefficient (MIC).

RFE works by searching for the ideal subset of features, evaluating by training a specific machine learning algorithm. In the project we used Logistic Regression, because of the familiarity the team had with it. Differently, MIC evaluates, not only the features performance [5], but also the strength and direction of them, by classifying features from 0 to 1. This methods helped us to identify the statistical dependency of the features, and was also selected because of its robustness to outliers.

The 20 features with the higher scores were chosen as relevant categories for the predictive model.

Predictive Model :

To assure getting an accurate model capable of predicting each individuals income classes. Different predictive algorithms were tested and compared from each other. Logistic Regression predicts the probability of a binary outcome through the usage of a logistic function [6]. We decided to keep 50% the probability to classify 'Income' whether as 1 or 0.

On the other hand, Neural Network differentiate among the possible 'Income' results by improving the weights of the relevant features and by passing them through hidden layers with activation functions. For this case, the team selected a more conservative approach by using from scikit-learn library : the Multilayer Perceptron's Classifier [7]. We used the default parameters, because it returned better results than if we had made any modification in the parameters.

Finally, our best prediction was made by using the Gradient Boosting Classifier. The GB algorithm produces a prediction model by assembling several other prediction models. In this case it were decision trees [6]. In order to avoid overfitting the model, we modified the parameters of the process, by increasing the minimum samples split, and the number of leafs.

IV. RESULTS

Following the data cleaning, and feature engineering, we used feature selection models on our data to optimize our model. From the Pearson's correlation heat map shown in figure 1, only two variables showed a high correlation.

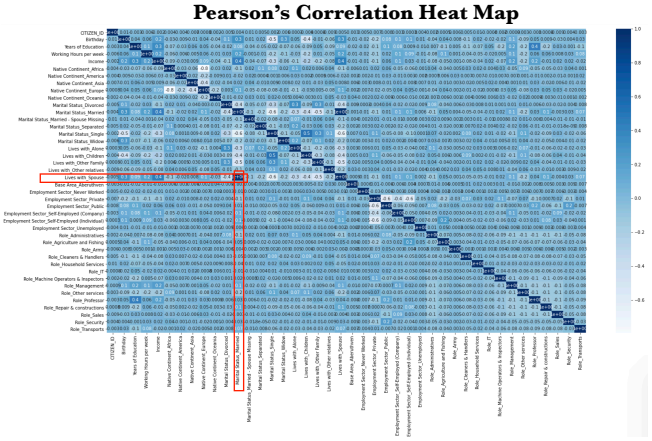

Figure 1 : Heat Map of the data features - Pearson's correlation

As it was stated in the previous part, we decided to analyze the 'Base Area' feature by itself. On the figure 2, the features are rank from 1 to 31, 1 being the variables with the highest ranking and the most importance.

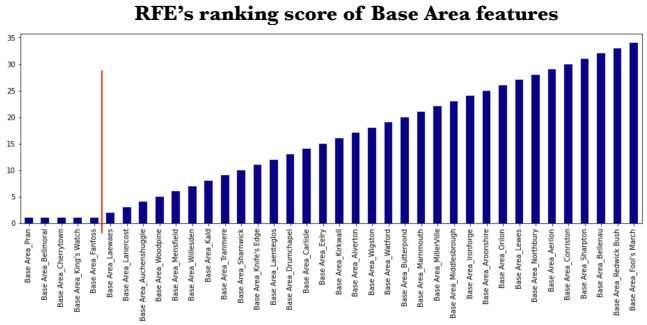

Figure 2 : Base Area - RFE.Ranking_

Finally the MIC process determine our 20 best features, as it shows on figure 3.

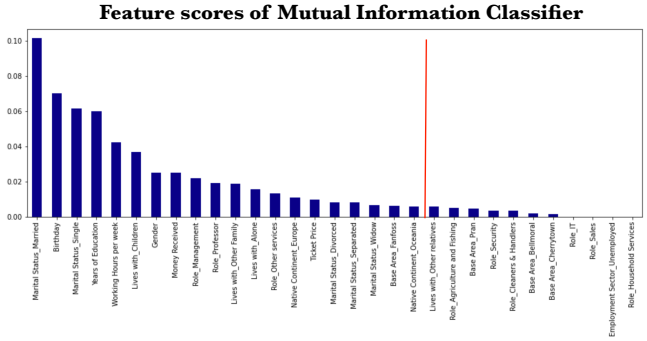

Figure 3 : Bar Chart - Mutual Information Classifier : feature scores

The three algorithms implemented yielded similar results, with all models behaving more or less in the same way for training data and validation data, therefore being neither Underfit nor Overfit. We achieved the best scores with the default parameters in the case of the Logistic Regression and Neural Network algorithms, but in the case of Gradient Boosting Classifier we tuned certain boosting and tree specific parameters, in order to control overfitting and increase score, such as min_samples_split where higher values prevent overfitting but too high values led to under-fitting so we settled on 200, min_samples_leaf which we defined as 100 because generally lower values should be chosen for imbalanced class problems, and so on. [8] The final model used for prediction, Gradient Boosting Classifier, was chosen based on overall higher F1 score, as we can see in the figure 4.

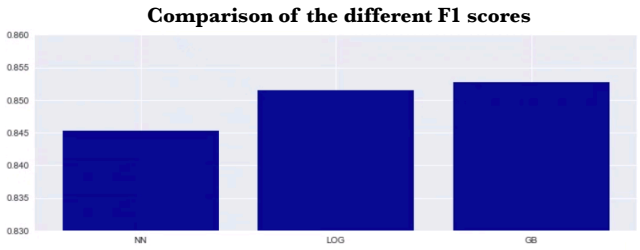

Figure 4 : Neural Network, Logistic Regression and Gradient Boosting Classifier - F1 score

IV. DISCUSSION

Although the dataset analyzed was imbalanced the three models implemented performed reasonably well. However, the Gradient Boosting Classifier was the model that had the best performance scoring an accuracy of 0.86, against 0.85 from the other models used, and an f1 score of 0.8566964285714286.

Nevertheless, there is still margin for improvement relatively to model performance such as:

- A more balanced dataset by ways of an over, or under, sampling algorithm.
- Other classifying methods, for example a Support Vector Machine, where each data point is plotted in a n-dimensional space after which, by performing classification, the hyperplane is found and used to classify each class [9]. Another possible classifier that could have been used was the Naïve Bayes which is based on the Bayes' Theorem and classifies each instance in the data by deriving the maximum posteriori. [10]

V. CONCLUSION

In summary, after comparing different classification methods, we found that the Gradient Boosting Classifier performed better than both the Logistic Regression and the Neural Network, both in terms of accuracy and in terms of the f1 score. The performance measures could also be improved by having a more balanced dataset and by using other classifiers.

VI. REFERENCES

[1] Dtale 1.29.1. pypi.org

[2] Mutual Information, scikit-learn.org.

[3] Jason Brownlee. *Information Gain and Mutual Information for Machine Learning,* machinelearningmastery.com. 2020

[4] Brownlee Jason. *Recursive Feature Elimination for Feature Selection in Python,* machinelearningmastery.com. 2020.

[5] Wint Rhondene. *On Maximal Information Coefficient: A Modern Approach for Finding Associations.in large Datasets,* Medium.com. 2019

[6] Ronaghan, Stacey. *Machine Learning: Trying to classify your data,* Medium.com. 2018

[7] Multi-layer Perceptron Classifier. scikit-learn.org

[8] Aarshay Jain. *Complete Machine Learning Guide to Parameter Tuning in Gradient Boosting (GBM) in Python,* analyticsvidhya.com. 2016

[9] Roberto Henriques. SVM.pdf

[10] Roberto Henriques. Probabilistic Classifiers.pdf