MSc. Data Science and Advanced Analytics 2020/2021

Predictive Modelling – Newland project

Course: Machine Learning

Professors: Roberto Henriques | Carina Albuquerque | Lara Oliveira

Group 12:

Ana Paulino (r20170743@novaims.unl.pt)

Luís Almeida (m20200666@novaims.unl.pt)

Soraia Cunha (r20170806@novaims.unl.pt)

Index

Table of figures

Abstract

The following report exposes a predictive model that aims to make a classification for people with income below the average rate (Income = 0) and for people with income above the average rate (Income = 1) in order to help the Newland Government to start applying taxes to its residents. To perform this study were analysed 13 decision variables, having been created additionally 8 variables. Several processes were applied, including outlier's detection, data normalization and encoding, feature selection and the election of the best model based on Neural Networks, Logistic Regression, Random Forest, Decision Trees, Instance-Based Learning, Support Vector Machine and Ensemble methods. As a result, it was corroborated that the best model was the Ensemble Gradient Boosting Classifier.

Keywords: Project, Machine Learning, Predictive Modelling, Neural networks, Decision Trees, Logistic Regression, Instance-Based Learning, Support Vector Machine, Ensemble, Random Forest, Feature Engineering

1. Introduction

In 2044, living on planet Earth started to be unfeasible, thereby people started to prepare to move to another planet located in our Milky Way galaxy. Thereafter, the mission “Newland” was launch in 2046 and, in the first phase, the people selected were divided into three categories: Group A (volunteers that passed the selection process), Group B (citizens who were paid to participate since they were considered as essential people by the state) and Group C (individuals who were rejected from the selection but give a money offer to participate).

In 2048, the second phase started and 100 new spaceships with people were coming to the Newland. To be financially sustainable, the government decided to apply a binary tax rate, where for people with income above average would be applied a 30% rate over their income and a 15% rate to the ones below the average.

Thus, this research aims to create a predictive model to apply to the individuals who are coming to Newland, based on the current inhabitants. Therefore, according to each individual´s class from the first phase, it was developed a model using this past data to have a model that best predicts the class that the individuals from the second phase will belong. Or, in other words, we are trying to predict what will be the suitable tax rate to be applied to each person. Saying that, we are trying to answer the question, “What will be the class of the new individuals? What tax rate will be applied to their income?”.

2. Background

In this section, we will abord the methods used in the project not mentioned in class.

2.1. Feature Selection

Chi-Square

Whilst exploring methods to assess the importance of features in explaining our model, we opted to include the chi-squared method into the equation. This method is often used to, with categorical variables as input, discern the best features among them to explain the categorical target variable.

Getting more into the mechanics of this method, it is often used to test the independence of two events. As such, with a high chi-squared value, means that the dependence is high and therefore, the variable in question can explain the target variable.

ANOVA F-values

ANOVA is a feature selection method normally used with numeric inputs with desired categorical output. ANOVA stands for “analysis of variance” and it does exactly that, to assess the best features, it analyses the difference in variances to discern if they follow the same distribution or not, and therefore can be calculated both from the same distribution. This shows that they share dependency and that will help in determining the describing power of the input variable on the output variable.

2.2. Imbalanced Learning

Combined Undersampling and Oversampling with SMOTE

When doing several tests and doing several kinds of hyper parametrization, due to the unbalanced train set provided, the models showed to be better at predicting one class than the other. Upon further research, we ran into methods of undersampling, oversampling and a combination of both with SMOTE (Synthetic Minority Oversampling Technique) for the oversampling part.

From this, one thing was clear, despite the improvement of the accuracy in both classes and making the dataset balanced, this proved to be disastrous to the safeguarding of generality. As such, overfitting was apparent in all models as of the use of these methods.

Since partitioning methods are applied after this transformation, this made hyper parametrization biased and the model itself got fitted with artificial and not very relevant data since it closely mimicked the data already in the dataset.

This made it costly to apply to our final model since it compromised by a large margin the inexistence of overfitting for a little increase in the desired scoring method.

Adaptive Synthetic Sampling (ADASYN)

This method envisions to make the disparity in class proportion less prominent, mitigating possible bias that might appear from improper class proportion, leading to bias modelling as well as make records more distinguishable so that the classification is clearer cut, leading to fewer errors in the attribution of classes. It operates by looking at the minority class and separating easily to learn training data from the more difficult training data and it.

Afterwards, it makes it difficult to learn data more abundant for the model to better distinguish future inputs and have better class separation.

2.3. Predictive Modelling

Voting

Voting can be split into two methods, the hard and soft voting ensembles. Whereas the first involves summing the votes for crisp class labels from other models and predicting the class with the most votes, the latter involves summing the predicted probabilities for class labels and predicting the class label with the largest sum probability.

Ridge Classifier

The ridge classifier is a model based on Ridge regression method, which converts the label data into [-1, 1] and solves the problem with a regression method. From this, if the value is greater than 0 then it is considered positive. If the opposite occurs, then it is considered negative. In turn, it uses regression to classify the class.

One limitation of this model is that if a record is difficult to assess (meaning that if it is situated close to zero), then it will be more prone to making bad decisions since class separation is not so clear.

3. Methodology

The approach used in this project was based on the main steps of a Machine Learning process. All the analyses were made through python on the Jupyter Notebook using the following packages: NumPy, Pandas, Sklearn, Matplotlib and Seaborn.

The model built throughout this study has the main objective of predicting if an inhabitant of the Newland planet has an income below or above the average, based on a database of 22.400 inhabitants. This process is achieved based on some patterns of the records, and then foreseeing the classification for their income.

Therefore, it was required to explore the given data, so that it was possible to identify the behaviour of the variables, including, for instance, their distribution, extreme values, and incoherencies. Additionally, we did data cleaning and pre-processing where we filled the missing values, removed the duplicated values, corrected errors, and standardized the values. Furthermore, from the given variables, there were transformed and created new variables to enrich the future model.

Subsequently, there was the phase of the feature selection, where there were employed several techniques, as Lasso regression, Ridge regression, Recursive Feature Elimination (RFE), ANOVA F-values, Decision Trees Feature Importance and AdaBoost Feature Importance, in order to select the most important variables to the final model.

Then, after selecting the most important variables to include in the model, it was proceeded the creation of the model, applying Neural Networks, Decision Trees, Instance-Based Learning, Random Forest, Support Vector Machine and Ensemble methods. In this step, there were tested several parameters in each algorithm to achieve the best possible result, avoiding overfitting.

4. Results

In this section, we will see step by step the results obtained in our research.

4.1. Exploratory analysis

When looking at the problem at hand, after knowing what the objective and the goal of the problem is, it is always necessary to look at the initial variables provided. Getting an intuition of the data presented can go a long way in spotting possible incoherencies and patterns in data. This, in turn, will make further procedures run smoother and will help prioritize some procedures with the help of theoretical background.

One example of this is concerning feature selection. Looking at the type of data specifically and categorizing it (either categorical or numerical) allowed us to run several feature selection models and better interpret the results depending on the type of data that we had provided to said models, following the type of target variable.

Aside from this manual view, there are ways to get a look at that in a different lens. One example of this is the pandas profiling, which shows a comprehensive and detailed view of the data, with correlations and distributions on each variable, as well as other statistics like missing values and such which give a powerful insight into the way our data behaves.

The function ". describe()" similarly helps in achieving the aforementioned goal, as well as ".info()" to look at the data types of our dataset.

```
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Name                   22400 non-null  object
 1   Birthday               22400 non-null  object
 2   Native Continent       22400 non-null  object
 3   Marital Status         22400 non-null  object
 4   Lives with             22400 non-null  object
 5   Base Area              22400 non-null  object
 6   Education Level        22400 non-null  object
 7   Years of Education     22400 non-null  int64
 8   Employment Sector      22400 non-null  object
 9   Role                   22400 non-null  object
 10  Working Hours per week 22400 non-null  int64
 11  Money Received         22400 non-null  int64
 12  Ticket Price           22400 non-null  int64
 13  Income                 22400 non-null  int64
```

Figure 1 - Variables data type

Through the initial exploration of the data, we could collect some pieces of information about our dataset, like that the Preschool education level only has value 0, in other words, all people with education level correspondent to preschool have income

below the average. Additionally, we can see that we do not have mismatches between the years of education and the educational level.

Regarding the missing values, we noticed the existence of NaN´s (Not a Number) in the categorical variables, so we proceed the filling of these values with the mode. Similarly, we use the mode of the Train dataset to fill the missing values of the Test dataset.

```
Name                      0
Birthday                  0
Native Continent          0
Marital Status            0
Lives with                0
Base Area               395
Education Level           0
Years of Education        0
Employment Sector      1264
Role                   1271
Working Hours per week    0
Money Received            0
Ticket Price              0
Income                    0
dtype: int64
```

Figure 2 - Missing values

Furthermore, we did further analysis and we observed that most people com e from Europe and Oceania is the continent with fewer people; that the most recent bi rthday is on 11th of October of 2031; concerning the marital status, the majority is mar ried, and the minority is a widow; also, most of the resident live with their wife and exis ts a minority that lives with other relatives; and finally the biggest amount received fro m a participant to be part of this expedition was 122999 monetary units.

Other ways of representing data visually are histograms and boxplots to identify distributions and irregularities in the pattern of data. The latter helped, in our project, in specifically thresholding and removal of outliers since the IQR method did not seem to be effective and was removing a large portion of our data. As such, the visual component, associated with an intuition of the pattern of the variables and the meaning behind them made it possible to successfully remove some outliers whilst keeping data integrity.

Regarding the weights view as a tool which can be useful to determine feature importance, we can view that there are some peculiarities in the proportion of certain classes within the variables on the target variable “Income”.

It is noteworthy to say that since our dataset is unbalanced (one class is three times the size of the other – there is one 1 for every three 0s), there are features that stray away from the expected proportion, meaning that there is some room for interpretation.

More specifically, when looking at the positive effect of the following features on the target - Married, Husband and Wife, Years of Education, Public Sector, Management, Working Hours per week, Age, Money Received and Ticket Price – these are variables that have a proportion of over or around 50%, meaning that even though they less populated, their positive effect of the target variable on income is above expectation, which would be around 33% if there was no correlation.

4.2. New Variables

Apart from the original variables, there were created 8 variables either in the train and in the test dataset, which are described below.

Variable	Formula
Age: age of the inhabitant, calculated from the year of his/her birthday (Birthday)	(2048 - pd.DatetimeIndex(df['Birthday']).year)
Children: binary variables that states if the person has children, based on who he/she 'Lives with' variable	(df['Lives with'] == 'Children').astype(int)
Group A: binary variable that says if the person belongs to group A, based on the values of 'Money Received' and 'Ticket Price' variables – 'Money Received' == 0 & 'Ticket Price' == 0	((df['Money Received'] == 0) & (df['Ticket Price'] == 0)).astype(int)
Group B: binary variable that says if the person belongs to group B, based on the values of 'Money Received' and 'Ticket Price' variables – 'Money Received' > 0 & 'Ticket Price' == 0	((df['Money Received'] > 0) & (df['Ticket Price'] == 0)).astype(int)
Group C: binary variable that says if the person belongs to group C, based on the values of 'Money Received' and 'Ticket Price' variables – 'Money Received' == 0 & 'Ticket Price' > 0	((df['Money Received'] == 0) & (df['Ticket Price'] > 0)).astype(int)
Unemployed: binary variable that states if the inhabitant is unemployed, supported by the value of 'Employment Sector' variable	(df['Employment Sector'] == 'Unemployed').astype(int)

HigherEduc: binary variable that says if the individual has higher education, based on 'Education Level' variable	(df['Education Level'].str.contains('PhD|Masters|PostGraduation|Bachelors')).astype(int)
Female: binary variable regarding the gender of the individual, based on the gender-neutral titles of the 'Name' variable	(df['Name'].str.contains('Miss|Mrs')).astype(int)

Table 1 - Description and formulas of the newly created variables

4.3. Coherence Checking

Based on the input variables, there might be some incoherencies that need to be checked before proceeding with the analysis, so that the data to be presented to the final model is reliable. Therefore, when such incoherencies occur, the best solution that was considered was to correct the correspondent values, instead of excluding the records, since there might be relevant data in such records that just need a small revision. Accordingly, the 4 incoherencies and respective solutions are presented in the following table, wherein it was assumed that the Role of each individual can be related to their previous experience in Earth, when the Employment Sector of the citizen is set as Never Worked and Unemployed, assuming this last condition is related to the Newland planet.

Incoherence	Number of incoherent records	Solution
It is not possible to have in the same record values different from zero for both columns 'Money Received' and 'Ticket Price'.	0	-
It is not possible to have an individual that in 'Employment Sector' has Never Worked and shows a value different from zero in 'Working Hours per week'.	7	It was assumed that 'Working Hours per week' was zero.
When 'Employment Sector' is Unemployed, such a person cannot have a value above zero in	12	It was assumed that 'Working Hours per week' was zero.

'Working Hours per Week'.		
In the study, there are not considered valid individuals younger than 18 years old.	272	As the 272 records show an age of 17 years old, it was assumed that these individuals were 18 years old.

Table 2 - Description of the Incoherences, number of incoherent records and respective solution

4.4. Outliers & Standardization

Concerning the outliers, two methods were applied to the metric features, the manual one and the Inter Quartile Range (IQR); however, due to the results obtained, it was only considered the outliers manual method, since in the IQR method approximately 40% of data were excluded. This indicates that such observations are not indeed outliers, but that there are many observations out of the Inter Quartile Range defined, that should not be judged as isolated points, but rather in general as a sparse dataset, as it can also be seen through the histograms and boxplots for the metric features.

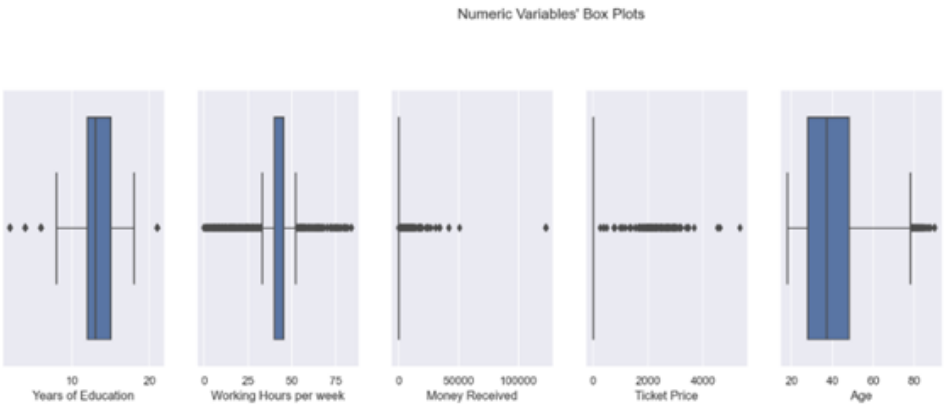

Figure 3 -Metric variables box plots

More specifically, in the manual check, it was considered that observations that have a value greater than 84 on Working Hours per week should not be included in the analysis. More than 84 working hours per week implies that an individual has to work more than 12 hours per day of the week every day, which is not physically or psychologically feasible, for this reason, the 114 observations that fit this condition were excluded from the process, so 99,49% of the data was kept.

CITIZEN_ID excluded:

12656, 12954, 13110, 13162, 13213, 13701, 14014, 14037, 14266, 14443, 15165, 15258, 15394, 16296, 16454, 16644, 16663, 16736, 16868, 17213, 17372, 17594, 17854, 18067, 18099, 18311, 18680, 18885, 18946, 19111, 19151, 19346, 19878, 20365, 20535, 20578, 20855, 20946, 21221, 21231, 21709, 21939, 22116, 22138, 22146, 22656, 22915, 23379, 23987, 24061, 24114, 24133, 24242, 24436, 24673, 24685, 24980, 25091, 25103, 25114, 25118, 25155, 25346, 25589, 25610, 25749, 25773, 25862, 26168, 26230, 26339, 26787, 27072, 27188, 27264, 27289, 27333, 27348, 27372, 27445, 27645, 27905, 27940, 28048, 28202, 28267, 28510, 28612, 28846, 29140, 29444, 29720, 29972, 30067, 30253, 30445, 30709, 30885, 31017, 31570, 31702, 32369, 32815, 32958, 33313, 33535, 33753, 33789, 33847, 34061, 34779, 34791, 34809, 34884.

Regarding the standardization, we performed two methods MinMaxScaler and StandardScaler for Train and Test dataset. Being the last one selected to the model phase.

4.5. Feature Selection

After preparing the data to build the model, we must first perform a feature selection of the variables. The reasons that lead us to make a feature selection are that it allows the algorithm to train quicker, it diminishes the complexity of the model and makes it simpler to analyse the result, diminishes overfitting, and despite all that it improves the model accuracy if the right set of variables are selected (Kaushik, 2020).

First, we check the correlation with the variables before the encoding, to see if we have problems of multicollinearity and to find out how much the variables are associated with each other.

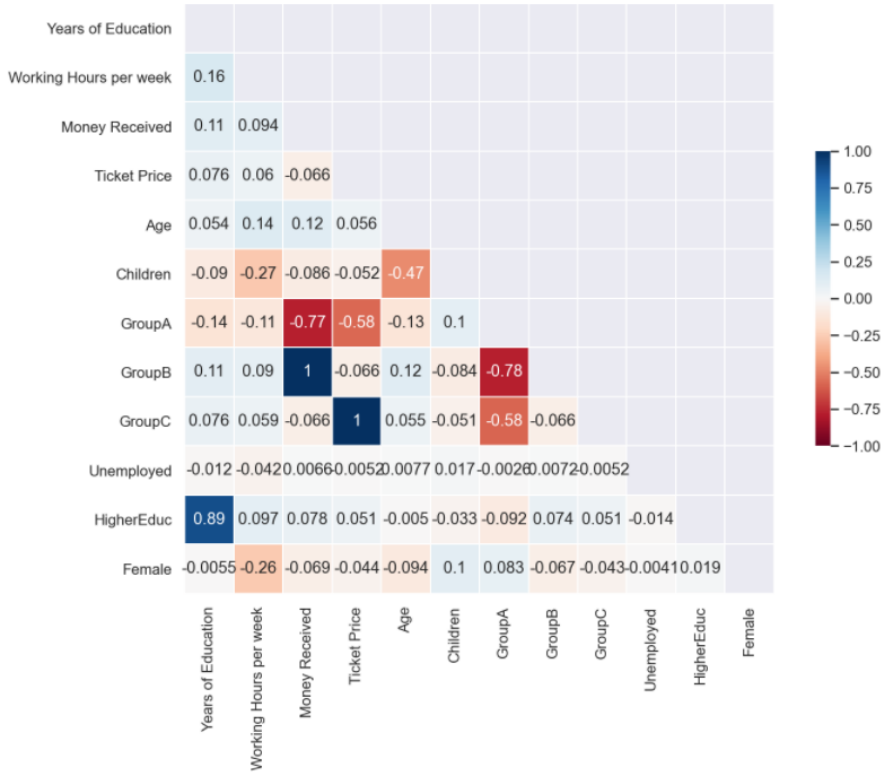

Figure 4 - Correlation Matrix

From the correlation matrix, we can analyse that if we consider a threshold of |0.70| for Spearman correlation, we have: Either we have 'Group C' or 'Ticket Price'; Either we have 'Group B' or 'Money Received'; Either we have 'Years of Education' or 'HigherEduc'; Either we have 'Group A' or 'Money Received'; Either we have 'Group A' or 'Group B'.

Then, we tried to use Principal Components Analysis, a dimensional reduction technique, in the metric features, to reduce the number of variables to apply in the predictive model. By looking at the Elbow graphic, the number of principal components to use would be two.

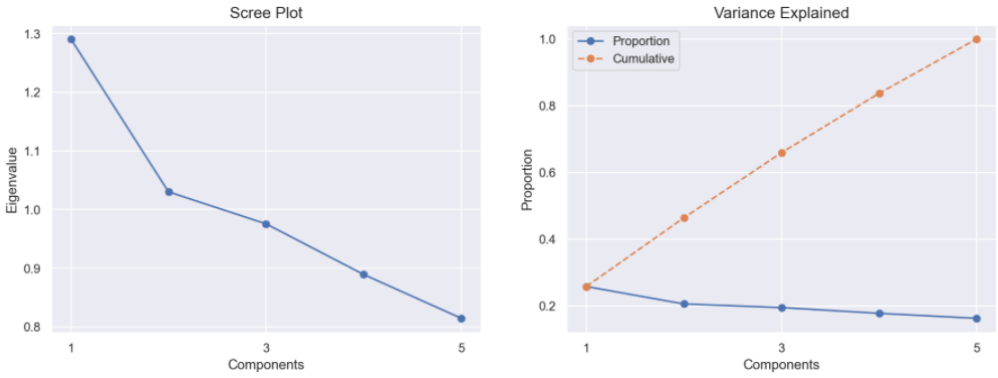

Figure 5 - Scree plot

However, the characterization of the principal components is very subjective and can be hard to define so we decided to use the original variables by making the selection using other methods for the metric features.

	PC0	PC1	PC2	PC3	PC4
Years of Education	0.646634	-0.006598	-0.393083	0.260986	-0.599327
Working Hours per week	0.598858	0.017479	-0.224893	-0.730186	0.239406
Money Received	0.505149	-0.601431	0.103114	0.406832	0.454929
Ticket Price	0.321633	0.812088	-0.021100	0.331742	0.355763
Age	0.394261	0.092367	0.871400	-0.110827	-0.253784

Figure 6 - Loadings

Consequently, we applied the RFE, LASSO, Ridge that are feature selection methods to all our variables (including the new variables created and encoded), except in RFE that we did only for the metric features, to see which one would give the best subset to improve the model accuracy. Looking at the scores of which methods, we got a score of 0.82, 0.38 and 0.38, correspondently. So, towards the goal to obtain the best model we also check the ANOVA F-values method to all variables and the Chi-square method to the non-metric features to see what the recommended subset would be. Regardless, we also did the feature importance using decision trees without establishing any parameter and by using different criteria's, such as the Gini, Entropy, MSE (mean square error), MAE (mean absolute error) and Friedman and the AdaBoost Classifier.

In the end, after getting the chosen features of each method and seeing what are the features that are considered more important, we decided to join all the ways and select the features that appear in most methods and with higher importance.

RFE		LASSO (|value|>0.10)		Ridge (|value|>0.10)		Chi-square		ANOVA	
Years of Education	4	x6_Management	3	x4_Preschool	1	Age	6	Years of Education	4
Money Received	5	x4_Masters	2	x6_Management	3	GroupC	3	Working Hours per week	5
		x2_Wife	4	x4_Masters	2	HigherEduc	3	Age	6
		x4_Masters + PostGraduation	2	x1_Married - Spouse in the Army	1	x0_Europe	1	GroupA	2
		x4_PhD	2	x4_PhD	2	x3_MillerVille	1	GroupB	4
		GroupB	4	x2_Wife	4	x3_Sharnwick	1	HigherEduc	3
		x2_Husband	3	x4_Masters + PostGraduation	2	x5_Private Sector - Services	1	x1_Married	4
		GroupC	3	GroupB	4	x6_Army	1	x1_Single	2
		x6_Agriculture and Fishing	2	x2_Husband	3	x6_Household Services	1	x2_Children	1
				GroupC	3	x6_Other services	1	x2_Wife	4
				x6_Agriculture and Fishing	2				
				x3_Orilon	1				

Decision Tree feature importance		Decision Tree w/ Gini & Entropy (>0.025)		Decision Tree w/ MAE,MSE,FRIEDMAN (>0.05)		AdaBoost (>0.025)	
Female	2	x1_Married	4	x1_Married	4	Money Received	5
x6_Professor	1	Age	6	Age	6	Age	6
Ticket Price	4	Money Received	5	Money Received	5	Ticket Price	4
x4_Bachelors+PostGraduation	2	Working Hours per week	5	Working Hours per week	5	Years of Education	4
x1_Single	2	Ticket Price	4	Ticket Price	4	x4_Bachelors + PostGraduation	2
x6_Management	3					Working Hours per week	5
Years of Education	4					x2_Husband	3
HigherEduc	3					Female	2
GroupB	4						
GroupA	2						
Money Received	5						
x2_Wife	4						
x1_Married	4						
Working hours per week	5						
Age	6						

Figure 7 - Variables selected according to the feature selection methods

In that way, we ended up with the following variables: GroupB, x1_Married, Age, Years of Education, Money Received, Ticket Price, Working Hours per week, x6_Management, x2_Husband, x2_Wife, GroupC, x5_Public Sector-Others and HigherEduc. Furthermore, we did the correlation matrix to see if existed variables that had a high correlation and we could confirm that GroupB and Money Received had a

high correlation like x2_Wife with x1_Married, Ticket Price with GroupC and HigherEduc with Years of Education.

So, based on the feature importance we eliminate GroupB, x2_Wife, GroupC, x5_Public Sector-Others and HigherEduc because they had lower feature importance. The variable x5_Public Sector-Others was added, because, during the exploratory analysis, we conclude that we had good discriminatory power, being a variable that has a lot of individuals that belong to the class 1.

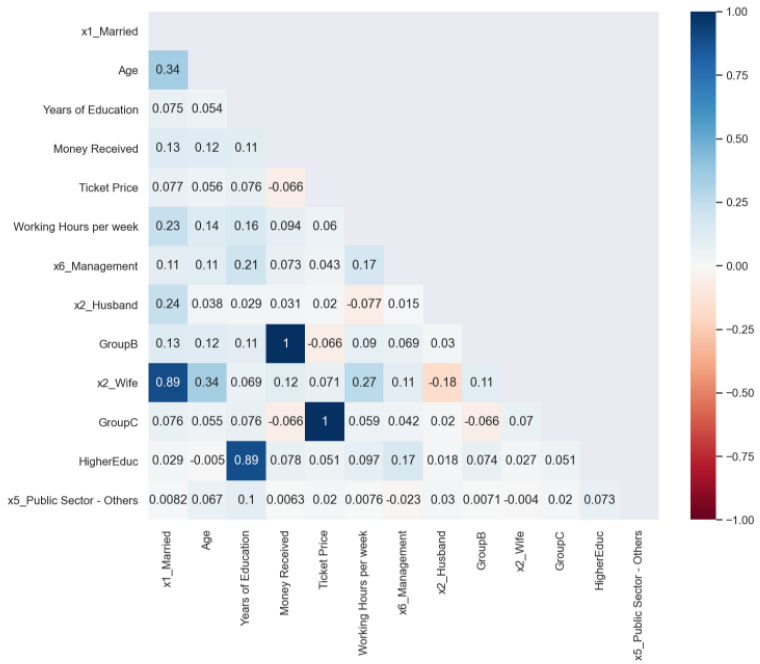

Figure 8 - Correlation matrix of selected features

4.6. Models

Next, we proceed to the partition of the dataset so that we can with one part do the Training phase and with the second part make the Validation phase. In other words, the first piece of the dataset is to build the model and the second to check how good is the developed model. We decided to use three schemas to divide the dataset, the Train-Test Split (with the parameter stratify, that is, to have the same number of 0´s and 1´s in the dataset), the K-Fold, and the Repeated K-Fold. The last schema was applied to the Neural Networks, Random Forest, and Ensemble Bagging Classifier with Neural Network. The K-fold was applied to the models mentioned previously plus Decision Trees, Instance-Based Learning and Logistic Regression. And finally, the Train-Test Split was used for all the models refer plus Naïve Bayes, Ridge, Hard and Soft Voting, Support Vector Machine, Ensemble - Bagging classifier + Trees, Ensemble AdaBoost, Ensemble Gradient Boost and Stacking. Comparing the three schemas, we notice that we obtained better results with Train-Test split and that would take less time to process the models in this schema.

Consequently, after building the different models, we moved to the tunning of the hyperparameters to increase the score of each model, and for the AdaBoost and Gradient Boosting models, we run a function in order to obtain the values for the learning rate and the number of estimators and we used them to get better scores. Also, in the Neural Networks, we performed a grid search aiming to find the best combination of hidden layers size, activation, solver, alpha and learning rate that gives the best score with the lowest standard deviation.

Besides, changing the parameters we decided to balance our dataset, in terms of zeros and ones in the target, by using different techniques like Undersampling, Oversampling, a combination of UnderSampling, Oversampling Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN). Nevertheless, balancing our dataset did not improve our model accuracy and, in some cases, we got overfitting, so we decided to leave it unbalanced.

To sum up, to find out which model is best to predict, we made a graphic with the scores of the different models and we realized that the best is the Ensemble Gradient Boosting Classifier with a score of 0.86808, a AUC of 0.923 and the following parameters: min_samples_split = 4, n_estimators = 182, random_state = 0, loss = 'exponential', learning_rate = 0.30000000000000004 and max_features = 'log2'.

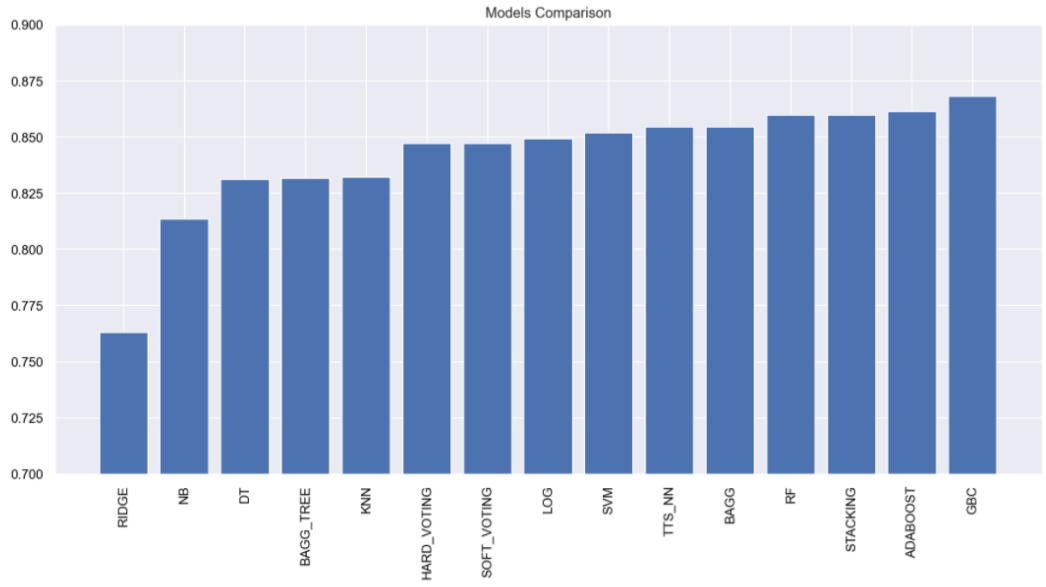

Figure 9 - Models Comparison

Also, to analyse the different models we did the ROC (Receiver operating characteristic) Curve, to help to compare the models, not in terms of score but considering the execution of the classification models at all classification thresholds, plotting two parameters: True Positive Rate, False Positive Rate. “ROC is a probability curve and AUC represent the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s” (Narkhede, 2020). However, the ROC Curve was

calculated without the Ridge, Hard and Soft Voting classification models since they do not have the attribute “predict_proba”.

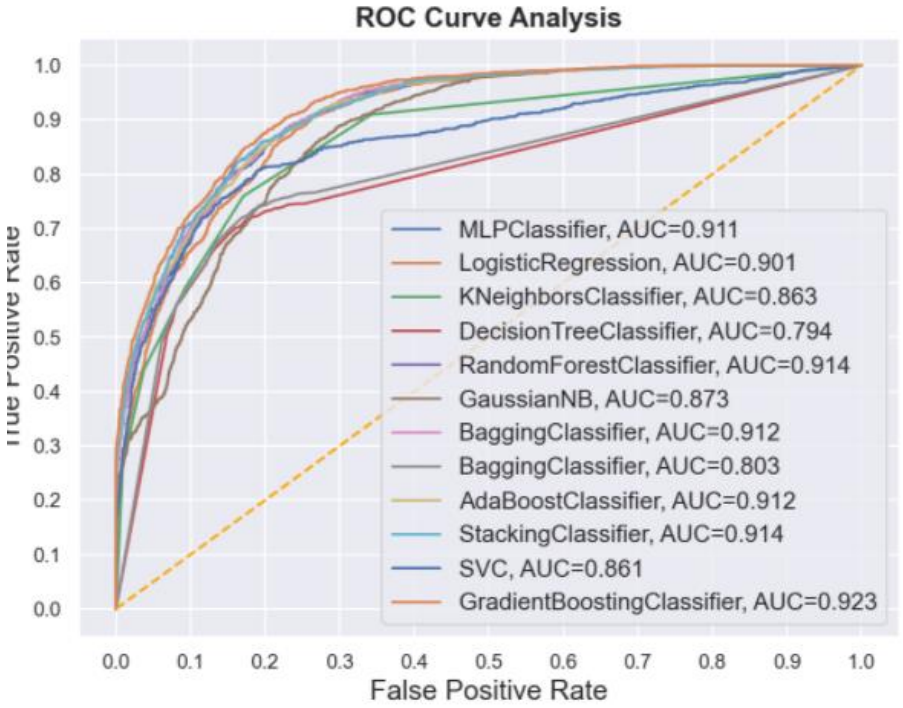

Figure 10 - ROC Curve

5. Discussion

After performing the analysis regarding the predictive modelling for the classification of income on the inhabitants of Newland planet, there could be some improvement points for future analysis. More specifically, concerning the original dataset, it is an imbalanced one, which can be problematic for classification. Concretely, "standard classifiers tend to be overwhelmed by the large classes and ignore the small ones". (Chawla & Japkowicz, 2004)

Actually, there are already many re-sampling techniques for imbalanced datasets, that some of them were tested in this project as it was the case of Oversampling, Undersampling, a combination of both, Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN); however, we faced some gaps, specifically, the results were poorer when these methods were applied, since the score was good on the training set, but not as good in the test set. This is a good point for improvement in the future since it would be beneficial to have a model that predicts as well for balanced datasets as well as good for imbalanced ones, indeed it would be the ideal case.

Therefore, a deep exploration in the matter of re-sampling techniques for imbalanced datasets as well as their hyper-parameters and parameters would be auspicious for greater performance of the model, both in the training and validation sets.

Furthermore, one limitation that was felt during the analysis was concerning the transformations on the original variables. We had a few variables that could be transformed into continuous variables, that could be more meaningful to this research question. Effectively, the most likely transformations were associated with the creation of binary variables that sometimes do not enrich the model as continuous features do, which in turn bring more variability in terms of values instead of just belonging to one or the other class.

Besides, the F1-Score in the Kaggle platform was not as good as in the Jupyter Notebook probably because our model was trained based on the unbalanced dataset, that predicts better the 0's (majority class). Also, the model that we selected as the best one was not the highest score in Kaggle; however, we considered that the elected one was better in terms of performance since Kaggle only shows the result for 30% of the data and a higher F1-Score in Kaggle could be prone to overfitting.

6. Conclusion

In our study, we started by exploring the data from the current inhabitants of Newland, to get a global grasp of the impact that the provided variables of the dataset might have in the target variable.

After carefully trimming down our dataset into optimal elements (x1_Married, Age, Years of Education, Money Received, Ticket Price, Working Hours per week, x6_Management, x2_Husband), we proceed to model and adjust the hyperparameters, whilst applying different partition methods, such as Train_Test Split, K-Fold and Repetitive K-Fold. So, based on the selected model, we obtained an AUC (area under the curve) equals to 0.923.

From the Ensemble Gradient Boosting Classifier, our best predictive model, we deem as worthy of analysis the aforementioned variables. This, in turn, will provide unequivocal evidence that the later variables will be the ones with a greater impact on the income to be analyzed in the future as such, it is a good contribution in saving recording resources since other variables will not be as impactful. This is one of the goals of Data Science as a whole, the reduction in the discrepancy between big data collected and the necessary data.

In conclusion, with this predictive model, we can answer the question about which class the new residents of the Newland will belong and consequently, what tax rate will they pay. By predicting the class of the new residents, 8228 people will have the class 0, or in other words, will have the income below average and 1872 will have the class 1, that is, will have the income above average.

7. References

Chawla, N. V., & Japkowicz, N. (2004). Editorial: Special issue on learning from imbalanced data sets. SIGKDD Explor. Newsl, 1–6.

Kaushik, S., 2020. *Feature Selection Methods | Machine Learning*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/> [Accessed 26 December 2020].

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3-24.

Narkhede, S., 2020. *Understanding AUC - ROC Curve*. [online] Medium. Available at: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5> [Accessed 26 December 2020].

Ramentol, E., Caballero, Y., Bello, R., & Herrera, F. (2012). SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. *Knowledge and information systems*, *33*(2), 245-265.

8. Appendix

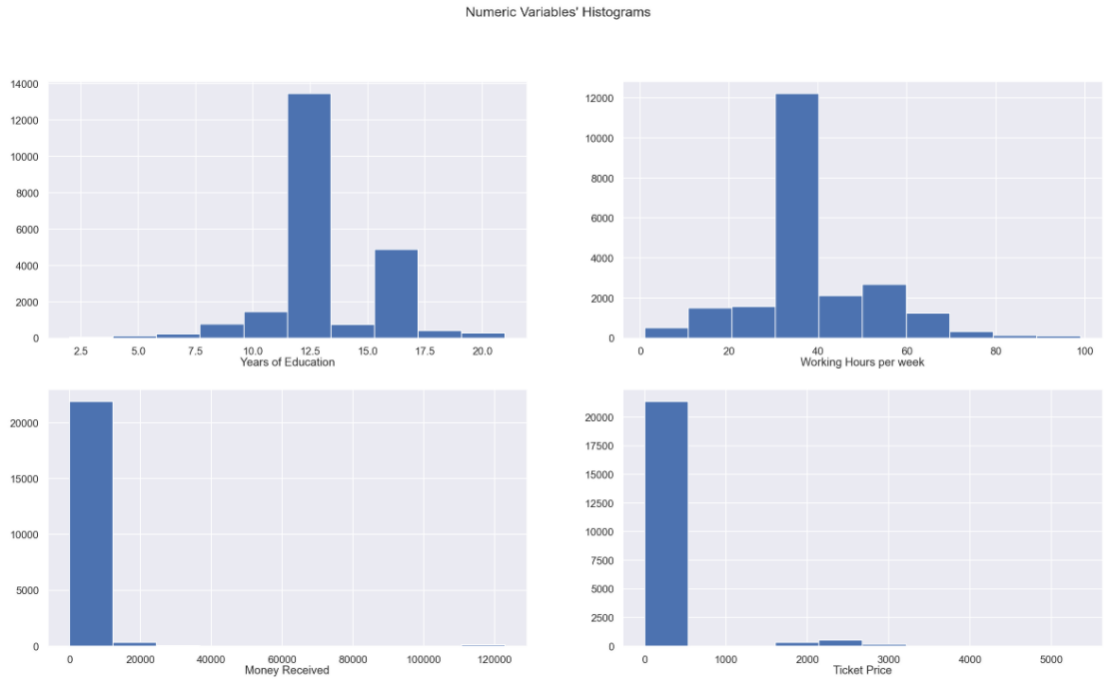

Figure 11 - Numeric Variables' Histogram

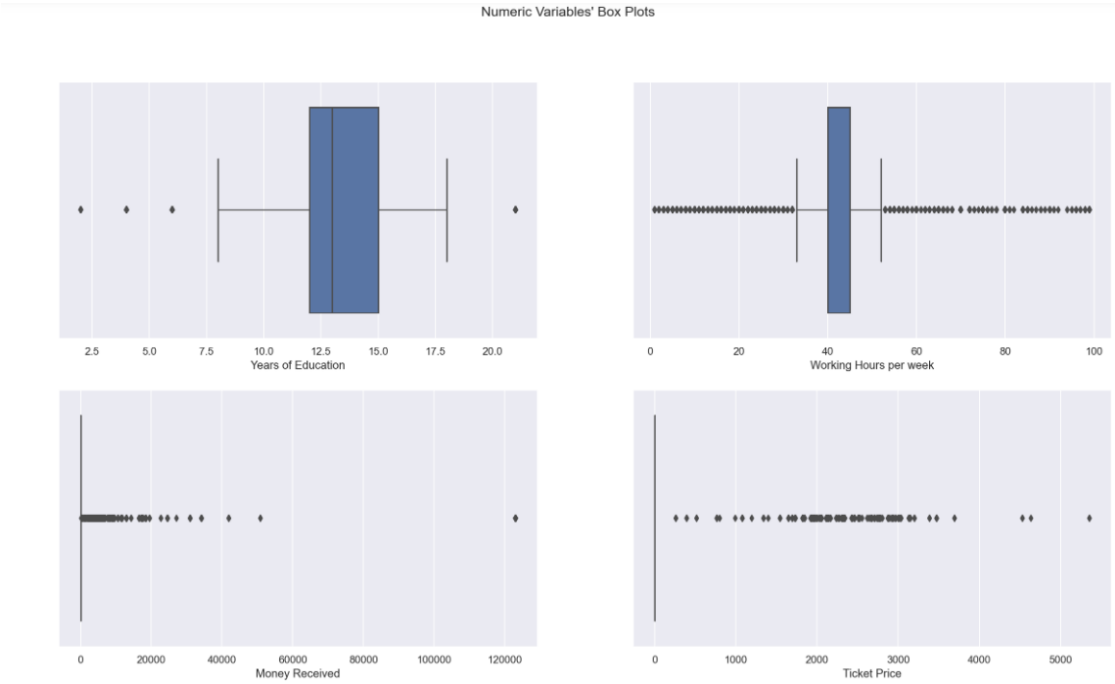

Figure 12 - Numeric Variables´Box Plots

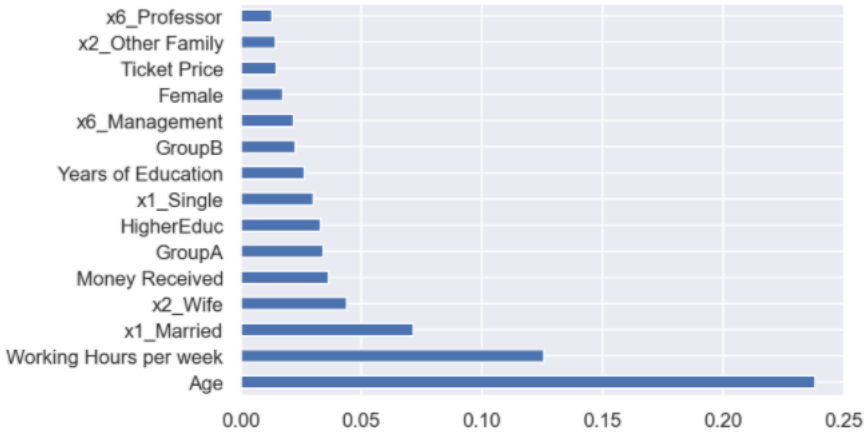

Figure 13 - Decision Tree feature importance

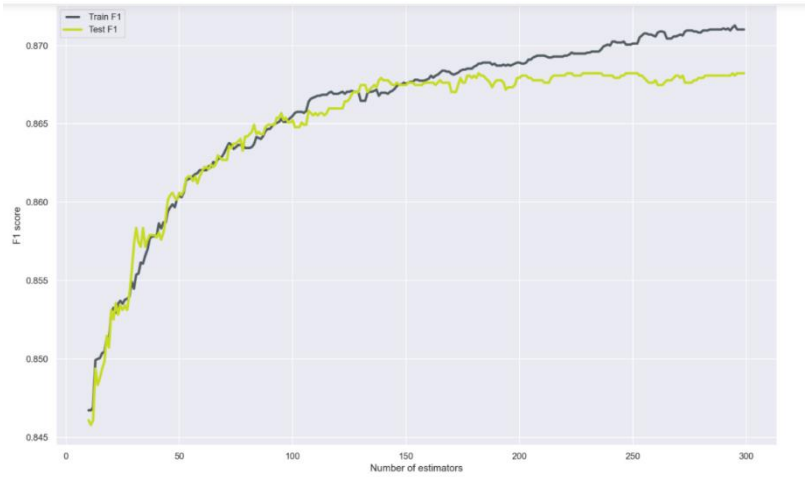

Figure 14 - Number of estimators of Gradient Boosting Classifier

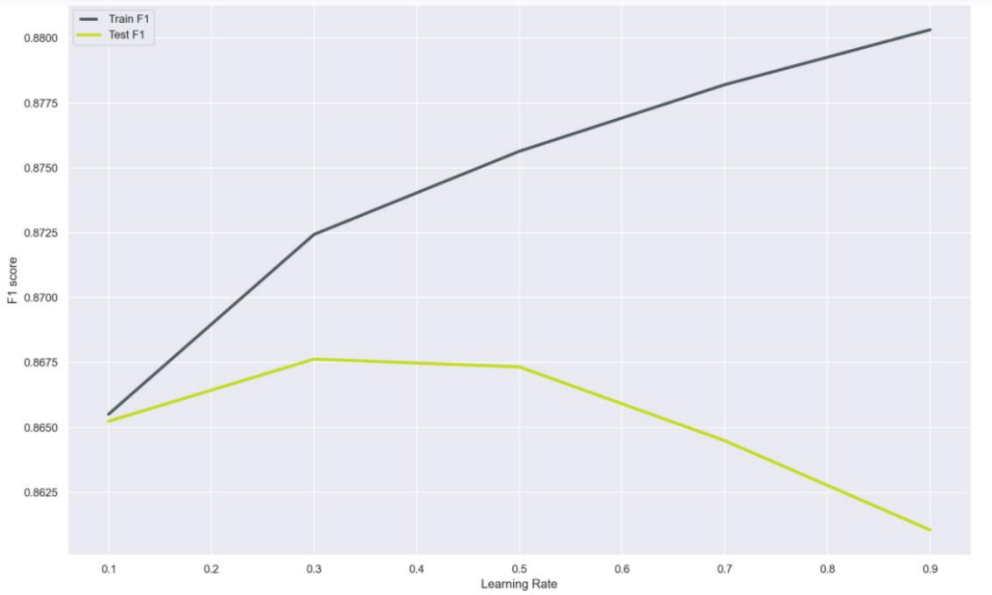

Figure 12 - Learning Rate of Gradient Boosting Classifier