Machine Learning

NEWLAND|THE CITY OF THE FUTURE

NOVA INFORMATION AND MANAGEMENT SCHOOL

MASTER DEGREE IN DATA SCIENCE AND ADVANCED ANALYTICS

// MACHINE LEARNING

MASTER’S DEGREE IN DATA SCIENCE AND ADVANCED ANALYTICS

1st year 1st semester

2020/2021

Teachers: Roberto Henriques, Carina Albuquerque

CAROLINA NEVES // [20200049]

DIOGO GONÇALVES // [20200632]

HENRIQUE COSTA // [20200652]

INÊS PIRES // [20200757]

NUNO PAIS // [20200576]

Total of pages: 22

Total of words: 5227

1. Abstract

On the behalf of the project “Newland”, the mission with the purpose of colonizing a planet with conditions necessary to sustain human life, our group is tasked with creating a predictive model in order to predict the exact rate of income tax that “Newland’s” colonizers will be subjected to, to make the new city more financially sustainable. With that purpose, a series of feature selection processes and supervised learning algorithms were used in varying combinations in order to achieve the best possible f1-score in correctly predicting whether the new participants in the program (the test data) had income above or below the average.

Through this process, it was concluded that the best model to apply to this project was the Random Forest Classifier, when paired with an outlier removal that employed both visualization removal and the Local Outlier Factor, a feature selection that englobed several methods as well as the Permutation Importance, the imputing of missing values using the Random Imputer, a categorical data encoding using the ordinal Encoder and the One-Hot Encoder, and several processes of data engineering.

TABLE OF CONTENTS

2. List of figures

3. Introduction

This project comes as requested by the Newland Government, given the context of a newly established taxation system. As is common every year, hundreds of space voyagers, sent by the International Orbital Transportation Services, arrive at the many Newland arrival ports, filled with newcomers looking for better lives and opportunities.

It is important for the governmental agencies, for matters of policy, strategy and decision-making, to predict, given a specific set of information about the immigrants, whether an individual will be taxed at 15% of his income or 30%. This binary tax system considers whether a citizen earns an income above or below the societal average. This is particularly useful, as it may have implications regarding the following year's governmental budget, and thus predicting how the coming individuals may contribute to the planet's economy. Also, the predictions can be utilized as means of selecting which individuals can or cannot live and work in Newland, for example, if the government needs higher-earning citizens.

Scientifically, the attempt was to apply a machine-learning algorithm, training it to a given dataset with 22.400 records and solve what can be seen as a "classification problem"- classifying individuals with 0 or 1 in which 0 represents an income below average and therefore a tax rate of 15%, and 1 represents an income above average and thus a tax rate of 30%. This model will then be applied to a different dataset with 10.100 records, predicting the classification for those individuals, of which this scientific group does not know the ground-truth, the true results are however known to the government, as a way to test the accuracy and precision of the model produced.

The model's prediction quality will be evaluated through the F1-Score, which is a harmonic mean between out of the whole predicted positives, how many were correctly predicted and recall out of all actual positives, how many the model got right, that is, respectively, precision and recall.

The information about the individuals, used to train and build the model, can be separated in two categories:

Categorical Features- Citizen ID, Name, Birthday, Native Continent, Marital Status, Lives With, Base Area (where the individual will settle), Educational Level, Employment Sector and Role (profession).

Numeric Features- Years of Education, Working Hours per Week, Money Received and Ticket Price.

The variable Income is the target variable the model will attempt to predict, based on all other independent variables.

The variables 'Money Received' and 'Ticket Price' refer to the citizen types that exist in Newland. These can be summarized as:

Citizen type A- Volunteers selected through an extensive process.

Citizen type B- Highly valuable people, having received money to work in Newland (the variable 'Money Received refers to these citizens).

Citizen type C- Rejected individuals during the selection process, but that offered money in exchange for the opportunity to live and work in Newland (the variable 'Ticket Price' refers to these citizens and how much they paid for the ticket).

3.1 Project Strategy Overview

The structure of the model building process will start with, after defining and understanding the objectives inherent to the data science activity, the Exploration and Preprocessing. During this part, the dataset given will be cleaned, missing values imputed, conclusions will be taken from the data distribution, outliers checked and removed, new variables will be created from the initial information and a relevancy/ redundancy analysis will occur, as means of selecting the only meaningful variables (deleting highly correlated ones). These measures are thus the backbone of feature selection, a process which was deepened with the usage of Chi-Squared, Pearson, Lasso, Ridge, Mutual Information, RFE and Permutation Importance procedures.

Afterwards, categorical data needs to be encoded, turned into numeric, so that it can be read and used in the classifier, making use of several encoding methods, such as OneHotEncoding (with dummy variables).

The data is split into a Training and Validation set (used for the fine-tuning of several model's hyperparameters, before achieving the final optimal model). This was followed by a normalization process, scaling all metric data, so that every metric information is read in an equal manner and follows the same standard distribution (the standard scaler centers distributions at 0 with a standard deviation of 1).

Then, several models are built, applied on the data and through continuous trial and error, the best model parameters and most important features are selected.

When the final model is achieved, scoring the highest F1 score, the model is ready to be applied on the Test dataset, which requires it to be preprocessed in the exact same way the initial Train dataset was, with the main differences being that any measures fitted to the Train set (such as the Random Simple Imputer, One Hot Encoder, Standard Scaler) are now not fitted to the Test set, but merely used to transform it.

4. Background

In order to complement the practical sessions and improve the efficiency of the data preparation, it was used a more robust method to remove outliers – the Local Outlier Factor (LOF). This method was chosen since it is able to find Multivariate outliers, as opposed to the traditional method of manually removing outliers, through data visualization methods, which only allows for a removal of the Univariate outliers.

While a univariate outlier is a data point that consist of an extreme value on one variable, a multivariate outlier is, on the other hand, a combination of unusual scores on at least two variables. therefore, a method that can operate on multivariate outliers, such as the LOF, is more robust than the standard methods applied that only operate on univariate outliers.

The LOF is based on a concept of a local density, where locality is given by k-nearest neighbours, whose distance is used to estimate the density. By comparing the local density of an object to the local densities of its neighbours, one can identify regions of similar density, and points that have a substantially lower density than their neighbours, which are considered to be outliers.

Another important strategy used on data engineering was filling missing values through the application of the Random Sample Imputer (RSI). In fact, RSI replaces missing data with a random sample extracted from the variable. It was chosen for its capability for filling both numerical and categorical variables. The imputer automatically selects all the variables with missing values, allocating a random sample after it.

5. Methodology

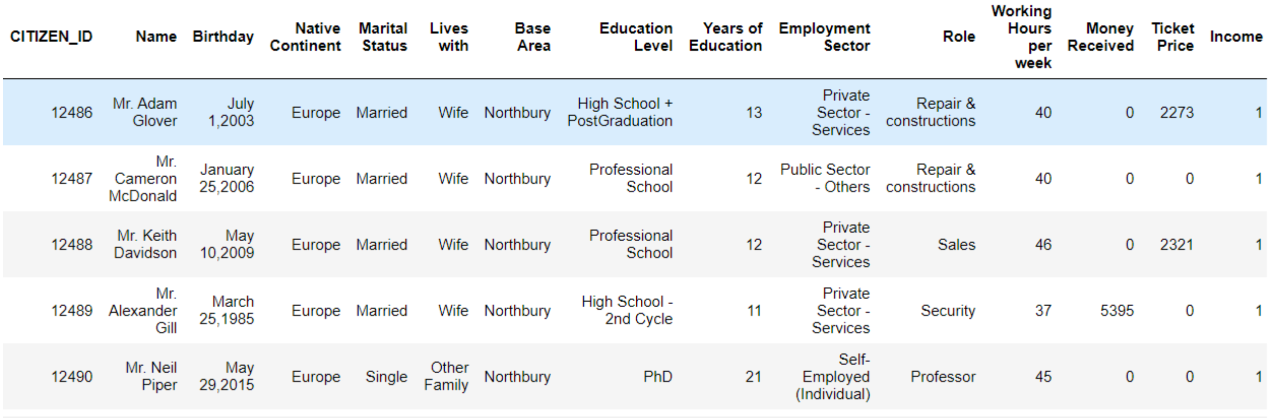
In the pursuance of the most suitable combination of data preparation procedures and the selected predictive model, it was decided to pursue a methodology that enabled the group to react to the modelling results and thus adapt the previous steps, to obtain the best results possible. Therefore, a SEMMA methodology was used. This methodology follows the ensuing steps:

- Sampling the data observations, importing the consequent file, and performing initial data visualizations, so as to gain a better understanding of the dataset at hand.
- Performing initial data modifications, and coherence checks, thusly altering or eliminating uncoherent variables, executing data imputation, separating categorical from numerical data, as well as the target variable from the independent variables, and finally encoding the categorical data.
- Implementing the initial prediction models, prior to feature selection, evaluating the ensuing f1-scores.
- Applying the suitable feature selection models and choosing both the 15 and 40 most important features.
- Performing the train-test split procedure, removal of the outliers and normalization of the data.
- Modelling subsequent to feature selection, evaluating the f1-score of the test subset of the data, which will indicate the performance of the model.
- Adjust the procedures as expected to achieve a higher performance from the prediction model. Once the weighted average of the f1-score is as desired, export the results into a csv file and promptly submit it so Kaggle.

Regarding the materials used for the carrying out of this project, the materials provided in both theoretical and practical lessons of Machine Learning, such as the PowerPoints, Jupyter Notebooks and additional information, were benefited from. The sklearn documentation was also consulted as well as a number of articles regarding the techniques used in this project.

6. Data exploration

To better understand the project's main idea, it was crucial to become acquainted to the dataset. After uploading it to the Jupyter notebook, it became visible that the dataset had 22400 rows distributed among 15 columns (Figure 1).



CITIZEN_ID	Name	Birthday	Native Continent	Marital Status	Lives with	Base Area	Education Level	Years of Education	Employment Sector	Role	Working Hours per week	Money Received	Ticket Price	Income
12486	Mr. Adam Glover	July 1,2003	Europe	Married	Wife	Northbury	High School + PostGraduation	13	Private Sector - Services	Repair & constructions	40	0	2273	1
12487	Mr. Cameron McDonald	January 25,2006	Europe	Married	Wife	Northbury	Professional School	12	Public Sector - Others	Repair & constructions	40	0	0	1
12488	Mr. Keith Davidson	May 10,2009	Europe	Married	Wife	Northbury	Professional School	12	Private Sector - Services	Sales	46	0	2321	1
12489	Mr. Alexander Gill	March 25,1985	Europe	Married	Wife	Northbury	High School - 2nd Cycle	11	Private Sector - Services	Security	37	5395	0	1
12490	Mr. Neil Piper	May 29,2015	Europe	Single	Other Family	Northbury	PhD	21	Self-Employed (Individual)	Professor	45	0	0	1

Figure 1 Data.head()

As it is possible to observe in Figure 1, the variables can be distinguished between numerical and categorical, as already detailed in the Introduction. Besides, it was also important to understand each variable through an analysis of its distribution, which facilitated the perception of the variance and the presence of potential outliers. Therefore, boxplots and histograms were used to visualize the variables' distribution and get a sense of the quality of the dataset.

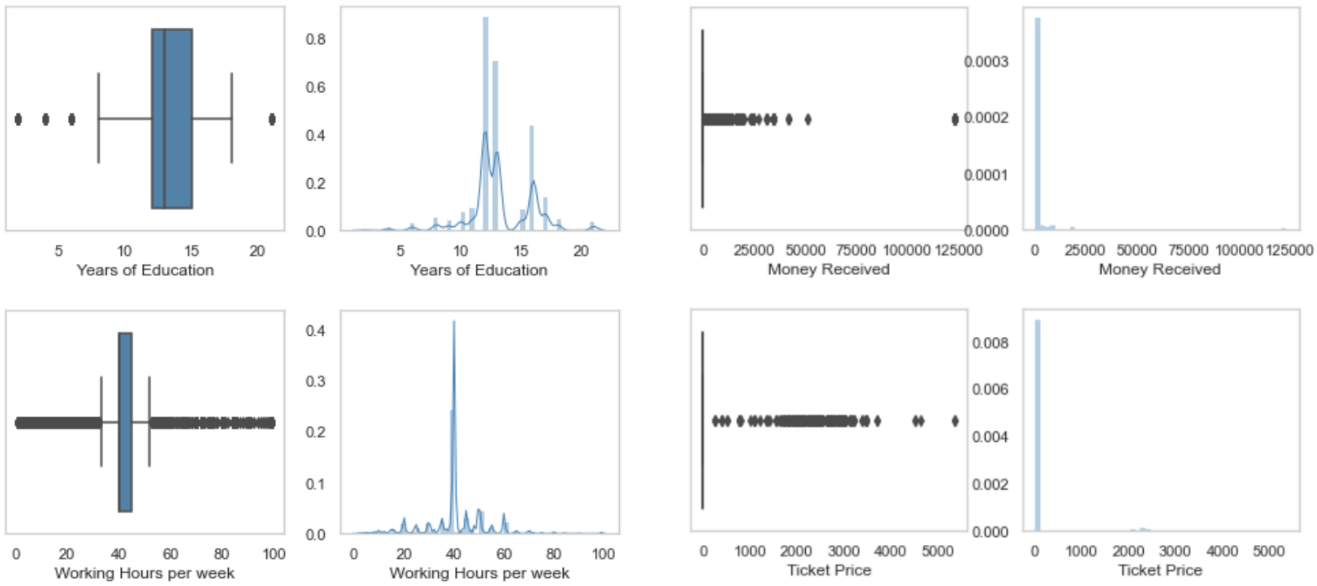


Figure 2 Visualizations through Boxplot and Histograms

As seen in Figure 2, the dataset is not highly exposed to outliers on Years of Education and neither on Working Hours per Week once. Although they have data points outside the box, both variables seem to follow a semblance of a Gaussian distribution. The range of values on Money Received and Ticket Price (alongside with Income) can be explained for being specific variables for the project's task. Their quite low variance might assume nearly any point other than 0 to be an outlier.

The correlation matrix (Figure 3) shows which features could be more powerful and useful on the modelling process. This first correlation matrix, before the data treatment, indicates that Income has a high correlation with Years of Education, Woking Hours per Week and Money Received.

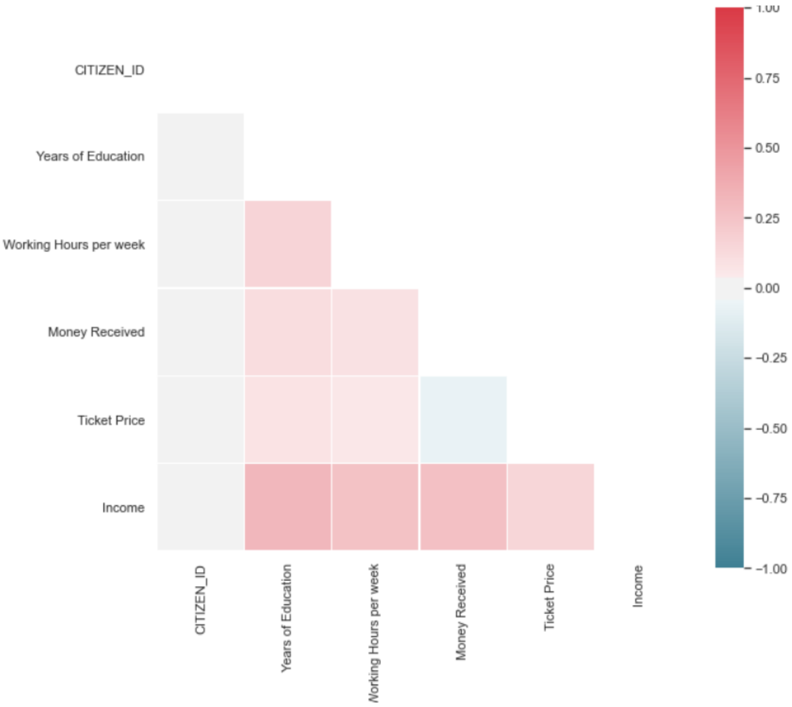


Figure 3 Correlation matrix

Once the data exploration is done, the dataset is now ready to deal with incoherencies, missing values and outliers on the next sub-chapters.

7. Data preparation

7.1.Missing values

Training a predictive model with a dataset with missing values can significantly influence the model's performance quality. Additionally, some models, such as the *sklearn* estimators used in this project, assume that all values are both numerical and significant. As such, there were three possible ways considered to dealing with the missing values present in the data: the least preferred of which would be to eliminate all the observations which contain missing data. However, that would put the dataset at risk of losing valuable information and, as such, the remaining two other methods were given more weight in consideration.

The remaining two methods considered were to use the mean or the median value of the non-missing values of the corresponding columns, or to use the Random Imputer. The Random Imputer was considered the best overall choice since the method of imputing the missing values with mean or median is only used for numerical variables and, in the case of the Newland project, the only variables to have missing values were categorical variables, which also exclude other robust methods like KNN Imputer.

The Random Imputer works for both categorical and numerical variables replacing the missing values with a random sample obtained from the variables in the training set, and if more than one variable are showed to seed the random sampling per observation, the values will be added. The figure 4 allows to conclude that Role and Employment sector had both 1200 missing values, which could be considered as low values, representing only around 5% of the raw dataset.

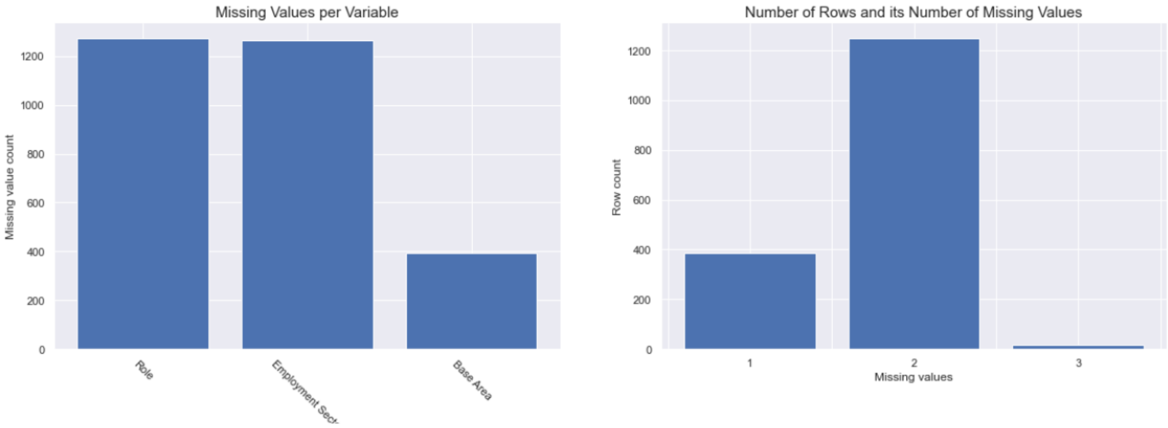


Figure 4 Missing values distribuition

7.2. Data engineering

Originally, when first running the model to obtain the first F1-scores, no data alteration was performed. Only when stiving to obtain a better performance, some variables were changed so as to achieve this goal.

As such, the variable Education Levels, was changed due to its high cardinality into a variable with less unique values, condensing some value into a single value with a broader meaning. For example, “Middle School- 1st Cycle”, “Middle School- 2nd Cycle” and “Middle School Incomplete” were replaced by a single value “Middle School”, and the same was done for other values – Figure 5.

Middle School- 1st Cycle	Middle School
Middle School- 2nd Cycle	
Middle School Complete	
High School- 1st Cycle	High School
High School- 2nd Cycle	
High School Complete	
High School PostGraduation	PostGrad Low
Professional School + PostGraduation	
Bachelors PostGraduation	PostGrad High
Masters PostGraduation	

Figure 5 Changes in Education Level Variable

Additionally, further alterations were made on existing variables. The variable title was replaced by Gender, by replacing all the “Miss”, “Mrs” with the value “Female” and the “Mr” occurrences with “Male”, as well as change the variable’s name. Through the variable “Birth Date”, the variable “Age” was created, after which the variable “Birth Date” was promptly eliminated. Furthermore, through the variables “Money Received” and “Ticket Price” the variable “Citizen Type” was created, describing which citizen belonged to which category of citizen established by the Newland Government. Finally, the variable “Marital Status” had some values that were considered by the group to be redundant, such as the values “Spouse Missing” and “Spouse in the Army”, and therefore these were both replaced by the simpler value “Married”.

7.3. Data encoding

Most predictive algorithms cannot operate in categorial data directly, without any type of transformation required. Therefore, in order to increase the number of possible models to use in this project, the categorical variables were encoded.

In order to encode the ordinal variable both the Label Encoder and the Ordinal Encoder were considered, but the Ordinal Encoder was deemed to be a better choice since it is usually used for independent variables, as is the case with the ordinal variable at hand, Education Level, while the Label Encoder is habitually used for encoding target variables.

The remaining nominal variables could be encoded in a series of differing methods that were interchanged with the predictive models used. Thus, in order to discover the best performing encoder, two different encoders were trialled jointly with several predictive models. The Target Encoder was not performing to satisfaction since it was overfitting the model, possible due to the target leakage, thus the One-Hot Encoder was chosen since it showed the most promising results.

7.4. Outlier removal

In order to perform a correct feature selection, and to prevent the outliers to influence the most important features chosen, the outlier removal should occur prior to the feature selection. Besides, both univariate and multivariate outliers can put bias on the dataset (see Background). As such, it was used a combination of two methods in order to eliminate outliers from the dataset.

The first approach dealt with multivariate outliers, spotted by the local outlier factor). The LOF graphics (Figure 5) gives information that, for the vast majority of the observations, the score is quite normal.

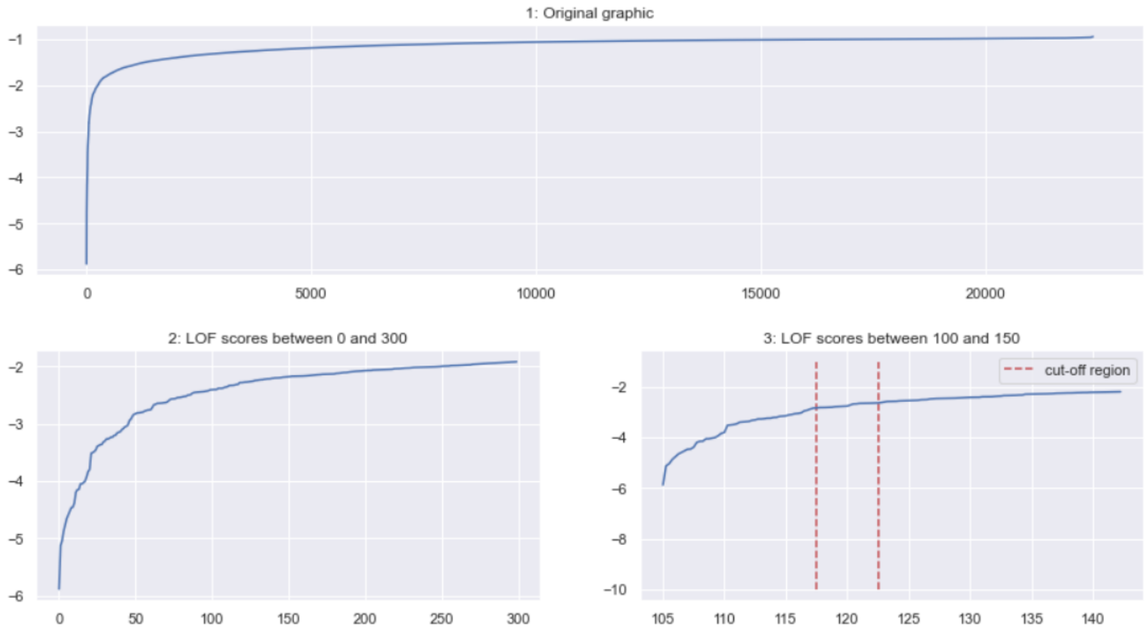


Figure 6 Outlier Factor Plot

However, there is around 120-130 observations that will be considered outliers, as their scores differ from the norm.

After that, outliers were removed through data visualization, eliminating rows by observing the boxplots of each numeric variable, resulting on the elimination of nearly 3% of the dataset – figure 6.

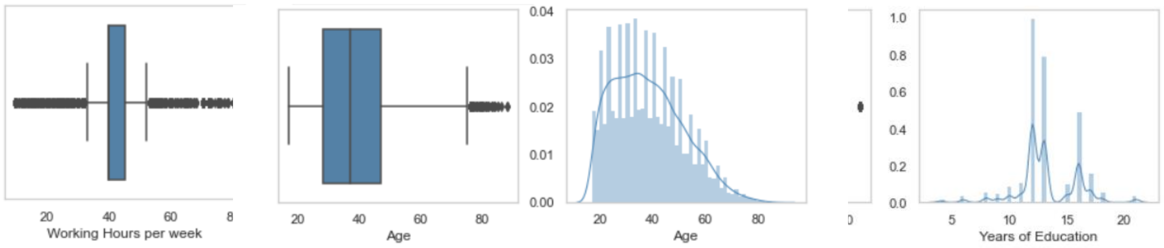
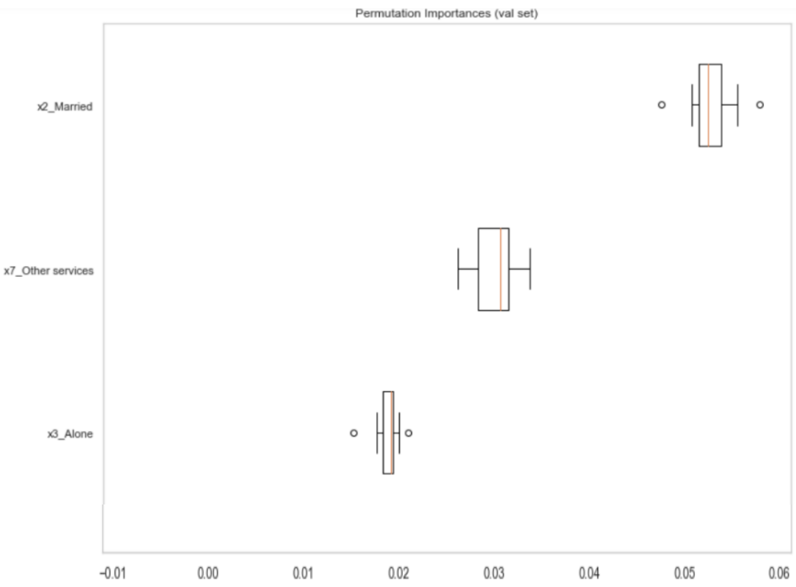


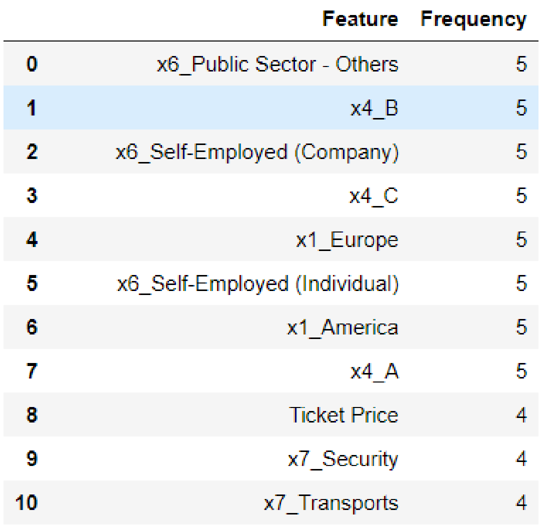
Figure 7 Visualization after Data Preparation

7.5. Variable importance and feature selection

The chosen model can only learn and work at its best if the training data contains enough relevant features and not many irrelevant ones. To achieves this, feature selection was done to select the most useful variables to train on among all of the existing features.

To find the most important variables in the dataset, two different approaches were used. The first one combined many different methods in which the selection of the variables was conducted by choosing those who were chosen by the models most frequently. The models used, in this first approach, were the following: Chi Squared, Pearson, Mutual Information, RFE, LassoCV and RidgeCV – figure 8. All of these models are available in the packages Linear Models and Feature Selection in *sklearn*.





	Feature	Frequency
0	x6_Public Sector - Others	5
1	x4_B	5
2	x6_Self-Employed (Company)	5
3	x4_C	5
4	x1_Europe	5
5	x6_Self-Employed (Individual)	5
6	x1_America	5
7	x4_A	5
8	Ticket Price	4
9	x7_Security	4
10	x7_Transports	4

Figure 8 Permutation Importance and Top 10 Feature Importance

Although the first approached included different feature selection methods, it was found that *permutation_importance* from *sklearn* would give the best results – figure 8. This method allows to find itself the most important features, based on the model and type of score it was used, as further detailed in the Modelling chapter. Thus, the first approached was used to make an initial decision and the last one to make the final choice on the best features for the model.

7.6. Data splitting and normalization

In order to have the least biased evaluation of the model's performance possible, it is necessary to use data observations never seen by the model during training. Thus, it is essential to split the dataset in such a way that there is a subset of data that is used to calculate the models' parameters and a different and independent subset of data used to assess the performance of the model, respectively the train and the test data. In order to split this project's dataset into train and test data, the *sklearn'*s *train_test_split* was used.

Additionally, in order to avoid data leakage, the dataset is normalized after the train-test split, since doing otherwise would lead to the standard deviation used to normalize the data being based on the entirety of the dataset, including the test data, leaking information about the test into the train set. The *sklearn*'s Standard Scaler was used to normalize the train data and subsequently applying, then apply the mean and standard deviation of the train data normalization to the test data normalization. The Standard Scaler was chosen in opposition to the Min Max Scaler since it delivered the best performance when paired with several models.

8. Modelling

8.1. Deep Learning Neural Network

Artificial Neural Networks is a Machine Learning model inspired by the networks of biological neurons found in every brain. It is very versatile, robust and scalable, having been successfully applied to many different problems from interpreting visual scenes to learning automatic control strategies.

This model has many advantages for prediction problems, such as, it works very well with a lot of data and with many variables, it does non-linear interpolation and can perform universal approximations. Thus, this model was one of the chosen to try with our dataset. However, it did not give as good results as other models did.

8.2. Random Forest

Random forest is a supervised learning algorithm that can be used for both classification and regression purposes. A simple way to explain how this algorithm works is that random forest builds multiple decision trees and merge them together to get a more accurate and stable prediction. This model produces a great result most of the time, in fact it was the best performing algorithm between the ones tested. Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction so, though this advantage, only the most importance features were used.

8.3. SVM

The Support Vector Machine can be used for both classification and regression purposes. A such, it was one of the models that was considered for this project. While this learning algorithm usually provides a higher accuracy than most algorithms, it also works best on smaller datasets and tends to not perform so well on larger datasets when the training time with SVM can be high. Additionally, it was not the best performing algorithm out of the ones that were tried.

8.4. Decision Tree

This model can also be used for both classification and regression, predicting the value of the target variable by learning simple decision rules extrapolated from the data features, which would also make it a suitable model to use on the project at hand. However, when the final results of this model were compared to the Random Forest Model, the Decision Tree’s performance fell short. While this model is considered to be simple to understand and interpret, they are also considered to be overall unstable and inaccurate, especially when compared to other models.

8.5. Ridge Classifier

Another classifier that was used was the Ridge Classifier, this method converts the label data into [-1, 1] and solves the problem with a regression method. This is a type of an embedded method; it learns which features best contribute to the accuracy of the model while the model is being created. The model creates some bias to lower complexity, to have fewer coefficients, by introducing additional constraints into the optimization of the predictive algorithm.

Although this model has many advantages, such as it prevents overfitting, it is quite simple to apply and it is very computationally efficient, it creates bias error leading to difficulties to have a balanced bias-variance trade-off.

This model was very good in predicting 1s but acted very poorly when predicting the 0s, leading to a low score.

8.6. K-Neighbours Classifier

K-Neighbours Classifier is a type of instance-based learning. This type of model predicts new data by training the available data to search for the instances that most closely resemble to them. There are three requirements for this type of model: it needs the stored records, the distances between the records must be calculated and the number of nearest neighbours to retrieve has to be defined (value of k). Then, a neighbour is created, and each neighbour will belong to a certain class label.

To be able to achieve a good classification, the data must be normalized to prevent distance measures from being dominated by one of the attributes, can be quite computational efficient, since it requires a lot of memory and time to calculate the distances, it is very sensitive to outliers and can be influenced by irrelevant attributes. The last two are dealt in the project with the elimination of outliers and by doing feature selection; additionally, the dataset was normalised before applying any model.

9. Results

The model that provided the best results, after fine-tuning the hyperparameters, was the **Random Forrest Classifier**. For purposes of the relevancy of the results, the following measures will be focused on the predictions of ‘1s’ in the data, and not ‘0s’ (the majority class).

In the Training set, the precision measure was of 0.71, recall of 0.81 and F1-Score 0.76. In the Validation set, the most important, precision was 0.69, recall 0.78 and F1-Score 0.73. Afterwards, using cross-validation scoring, the mean F1 for training was 0.70 (+/- 0.02) and for the validation 0.71 (+/- 0.03). Applying the function f1_score (from sklearn) and specifying the average to: “micro” outputs 0.8647, ‘weighted’ outputs 0.8673, “macro” outputs 0.8214.

Other important results were the Permutation Importance for the Validation set, which, out of the variables used during the modelling process, plots each variable's contribution to the overall F1_score of the model (as seen in 6.2.4). This was the basis for the data_redux, a smaller selection of variables that all positively contribute towards the model's overall score. It is important to choose the variables for their conceptual contributions, for instance, if a variable is meaningful for the study of a particular subject matter, despite not being extremely relevant in terms of scoring.

Some of the most important features in terms of their F1 scoring impact:

- Married (whether the individual is married or not)
- Role - Other services, Management, Machine Operators and Inspectors, Cleaners and Handlers, Professor, Agriculture and Fishing, Repair and Constructions
- Private Sector - Services, Private Sector- Others, Self-Employed Individual, (Employment Sector)
- Money Received
- Ticket Price; Alone (related to Lives With)
- Working Hours per week
- Years of Education
- Age
- A (Citizen Type)

		F1 - score			
		Train		Validation	
		0	1	0	1
Models	Random Forest Classifier	0,92	0,76	0,9	0,73
	Support Vector Machine	0,89	0,70	0,89	0,69
	Ridge Classifier	0,9	0,6	0,9	0,61
	KNeighbours Classifier	0,92	0,71	0,91	0,69
	Neural Networks	0,92	0,72	0,90	0,64

Figure 9 Different Models' F1-scores

10. Discussion

In Machine Learning there is a great range of different models to choose from. However, it is imperative to know what kind of dataset is presented to work with and what type of problem there is in hand.

The dataset of this project had both metric and non-metric features, and the main goal was to predict 0s and 1s, predicting if people will have an income below or equal to the average population. With that in mind, it is possible to know that it is a classification problem that can be dealt with supervised learning.

Random Forest Classifier is one of the best models to tackle this kind of problem. The building blocks for this type of algorithm are Decision Trees, i.e., Random Forest is an ensemble of Decision Trees generally trained via the bagging method. It is much more efficient to use this model than to build a bagging classifier and pass it a decision tree classifier, because it will be immediately evaluating each individual tree and defining a class prediction, becoming the class with the most votes the model's prediction. In other words, the Random Forest allows for many trees to work together and to create the best possible model, outperforming any individual tree model.

For Random Forests to perform well, the features used for the model must be better than random guessing and the errors made by the individual trees must not be correlated with each other. The first prerequisite is overcome with feature selection. The second difficulty is handled with bootstrap aggregation, a parameter of this model that allows for each individual tree to randomly sample from the dataset with replacement, resulting in distinct trees, thus, it allows to exist trees trained on different sets of data.

Another important procedure that could have led the model to decrease its F1 Score it's the number of outliers removed, which was nearly 3%. Even though it was applied a combination of visualization with local outlier factor, making the data more clean, these processes might have deleted data relevant to predict the binary tax rate.

11. Conclusion

As requested by the Newland government, the group created a model to predict whether an individual will be taxed at 15% of his income or 30%.

After an extensive research, it was concluded that the best model to build this kind of predictive model was the Random Forest Classifier, working in tandem with remaining algorithms chosen as the most effective throughout the project, such as the two different approaches used in the feature selection stage, the combination of algorithms and the Permutation importance, the outlier removal through both visualization methods and the Local Outlier Factor, and the categorical data encoding with both the Ordinal Encoder and the One-Hot Encoder. This combination of algorithms and techniques was proved to be the most effective one and obtained the best overall results. Hence, using our project, the Newland government can correctly classify the validation dataset and predict whether an individual will be taxed at 15 % or 30 % of income tax with a success rate of 86 %, as evaluated through the F1-Score.

However, while the project at hand could only be accomplished through the sole use of sklearn models, due to the project rules, it is the belief of the group of the project that, if allowed, non-sklearn algorithms could possibly obtain better results and thus, predict with a bigger accuracy what would be the taxation rate of each individual.

Concluding, a highly satisfactory success rate of 86% in identifying the taxation rate for each individual was obtained by using the *sklearn* methods such as the ones detailed above as well as a successful data engineering.

12. References

Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Machine Learning*, 20(3), pp.273-297.

Deng, H., Runger, G. and Tuv, E., 2011. Bias of importance measures for multi-valued attributes and solutions. In: *Proceedings of the 21st International Conference on Artificial Neural Networks*.

Feature-engine.2020.*RandomSampleImputer– Read de Docs.* [online] Availabe at: https://feature-engine.readthedocs.io/en/latest/imputers/RandomSampleImputer.html [Acessed 26 December 2020].

Feature-engine.readthedocs.io. 2020. *Randomsampleimputer — Feature-Engine 0.6.1 Documentation*. [online] Available at: <https://feature-engine.readthedocs.io/en/latest/imputers/RandomSampleImputer.html> [Accessed 26 December 2020].

Géron, A. 2017. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems* , O'Reilly Media , Sebastopol, CA .

Medium. 2020. *Understanding Random Forest.* [online] Available at: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> [Accessed 26 December 2020].

Medium. 2020. *6 Different Ways To Compensate For Missing Data (Data Imputation With Examples)*. [online] Available at: <https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779#:~:text=They%20are%20often%20encoded%20as,the%20machine%20learning%20model's%20quality.&text=A%20better%20strategy%20would%20be%20to%20impute%20the%20missing%20values.> [Accessed 26 December 2020].

Medium. 2020. *Target Encoding And Bayesian Target Encoding*. [online] Available at: <https://towardsdatascience.com/target-encoding-and-bayesian-target-encoding-5c6a6c58ae8c> [Accessed 22 December 2020].

Micci-Barreca, D., 2001. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, 3(1), pp.27-32.

Scikit-learn.org. 2020. *1.1. Linear Models — Scikit-Learn 0.24.0 Documentation.* [online] Available at: <https://scikit-learn.org/stable/modules/linear_model.html> [Accessed 26 December 2020].

Scikit-learn.org. 2020*. 1.13. Feature Selection — Scikit-Learn 0.24.0 Documentation.* [online] Available at: <https://scikit-learn.org/stable/modules/feature_selection.html> [Accessed 26 December 2020].

Scikit-learn.org. 2020*. Sklearn.Ensemble.Randomforestclassifier — Scikit-Learn 0.24.0* Documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> [Accessed 26 December 2020].

Scikit-learn.org. 2020*. Sklearn.Inspection.Permutation_Importance — Scikit-Learn 0.24.0 Documentation.* [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.inspection.permutation_importance.html> [Accessed 26 December 2020].

Scikit-learn.org. 2020*. Sklearn.Linear_Model.Ridgeclassifier — Scikit-Learn 0.24.0 Documentation.* [online] Available at: <https://scikit-

learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html> [Accessed 23 December 2020].

Scikit-learn.org. 2020. *Sklearn.Neighbors.Kneighborsclassifier — Scikit-Learn 0.24.0 Documentation.* [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html> [Accessed 23 December 2020].

Scikit-learn.org. 2020*. Sklearn.Neighbours.LocalOutlierFactor — Scikit-Learn 0.24.0* Documentation. [online] Available at: <*https://scikitlearn.org/stable/modules/generated/sklearn.neighbours.LocalOutlierFactor.html*> [Accessed 21 December 2020].

Scikit-learn.org. 2020. *Sklearn.Model_Selection.Train_Test_Split — Scikit-Learn 0.24.0 Documentation*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html> [Accessed 25 December 2020].

Scikit-learn.org. 2020. *Sklearn.Preprocessing.Labelencoder — Scikit-Learn 0.24.0 Documentation*. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html> [Accessed 20 December 2020].

Tingle, M., 2020. *Preventing Data Leakage In Your Machine Learning Model*. [online] Medium. Available at: <https://towardsdatascience.com/preventing-data-leakage-in-your-machine-learning-model-9ae54b3cd1fb> [Accessed 10 November 2019].