The Prospection of the Taxation System in Newland

Gustavo Brito (r20170760@novaims.unl.pt), Mariana Albernaz (r20170785@novaims.unl.pt), Marta Santos (r20170770@novaims.unl.pt), Miguel Mateus (r20170752@novaims.unl.pt)

Abstract: *Newland's goal to find the optimal taxation structure depends on the discernment taken on the salaries' of the citizens, labelling its citizens (1 - salary above average and 0 otherwise). Therefore, this study aimed at the training of a model to predict the desired categories. Through the usage of several data cleaning and transformation procedures and heterogeneous Machine Learning algorithms, it was possible to train such models, estimate and calibrate their hyperparameters, compare them and see which one provided the best outcome. With the results, a correct and easy-to-implement estimation of the requirements for the tributation system into the new planet becomes now possible.*

Keywords: Machine Learning, Multilayer Perceptron, Ensemble Methods, Adaboost, Python

1. INTRODUCTION

It's 2048 and planet Earth has experienced massive climate changes that have become untenable for all life forms. Newland was the first spaceship with 40000 individuals in mission to bring people into a recently-found planet where life is now possible. The selection was made by 3 main groups:

A: Unpaid people carefully chosen through an extensive selection process.

B: People paid to participate in the mission (considered by the state has essential to have on an initial phase of populating a planet.

C: People who paid to participate in the mission (those who were rejected in a selection process but entered by making a money offer).

With another 100 new spaceships on the way, the government of Newland agreed that to make the new city more financially competitive, people should start paying taxes. It was decided to apply a binary tax rate at which the rate would be 15% of for individuals with an income below or equal to the average and 30% of taxable income for the remaining people. Identifying the individuals that belong to each class is the first stage of this process. With this in mind, a first income study was carried out on 32500 individuals from which the government plans to develop a predictive model to apply to the people on the way to Newland. The variables available for the development of the predictive model were the following:

Variable	Description
CitizenID	Unique identifier of the citizen
Name	Citizen's Name
Birthday	Date of Birth of the Citizen
Native Continent	The continent the citizen was born on Earth
Marital Status	Marital status of the citizen
Lives with	The household environment of the citizen
Base Area	Neighbourhood of the citizen in Newland
Education Level	Education level of the citizen
Years of Education	Number of years of education of the citizen
Role	Job role of the citizen
Working Hours per week	The number of working hours per week of the citizen
Money Received	The money paid to the elements of Group B
Ticket Price	The money received by the elements of Group C
Income	Target (Where 1 is income greater than the average and 0 otherwise)

Table 1: Variables used on the Project

2. BACKGROUND

The techniques used in the development of this report that were not explicitly taught in Machine Learning Course fall under three main categories:

- Statistical Tests*
- Oversampling Techniques
- Feature Importance

2.1 Statistical Tests

2.1.1 Shapiro-Wilk Test for Normality

The Shapiro-Wilk test examines if a variable is normally distributed in some population.This test quantifies the similarity between the observed and normal distributions as a single number: it superimposes a normal curve over the observed distribution as shown below. It then computes which percentage of our sample overlaps with it: a similarity percentage.

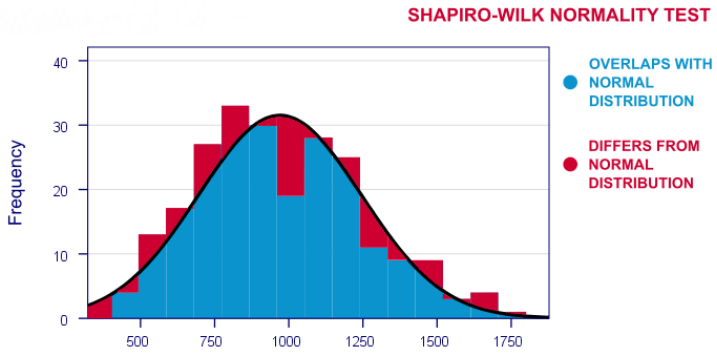

Figure 1: Shapiro-Wilk Normality Test

Finally, the Shapiro-Wilk test computes the probability of finding this observed -or a smaller- similarity percentage. It does so under the assumption that the population distribution is exactly normal. As so, the hypothesis are as follows:

H0: The data is normally distributed

H1: The data isn't normally distributed

When the p-value is smaller than 0.05 the null hypothesis is then rejected, meaning that the variable is not normally distributed.

2.1.2 Chi-Squared Test for Independence

The Chi-Square test of independence is used to determine if there is a significant relationship between two nominal (categorical) variables. The frequency of each category for one nominal variable is compared across the categories of the second nominal variable. The data can be displayed in a contingency table where each row represents a category for one variable and each column represents a category for the other variable. This statistic will be used to examine the relationship between the target variable (binary) and the rest of the binary variables.

$$\tilde{\chi}^2 = \frac{1}{d}\sum_{k=1}^{n}\frac{(O_k - E_k)^2}{E_k}$$

*All statistical tests were made with an $\alpha = 0.05$

The hypothesis are as follows:

H0: The target and the variable are independent

H1: The target and the variable are not independent

2.1.3 One-Way ANOVA Test

The one-way analysis of variance (ANOVA) is used to determine whether there are any statistically significant differences between the means of two or more independent (unrelated) groups.

H0: The mean of the target and the variable are equal

H1: The mean of the target and the variable are different

The formulas to reach the F-statistic for critical values evaluation are as follows:

$$SST = \sum (y_{ij} - \bar{y}_{..})^2$$

$$SSE = \sum (y_{ij} - \bar{y}_{i.})^2$$

$$SS_{group} = SST - SSE.$$

$$\begin{aligned} MS_{group} &= SS_{group}/df_{group} \\ MSE &= SSE/df_{error}. \end{aligned} \tag{1}$$

$$F = MS_{group}/MSE. \tag{2}$$

2.2 Oversampling Techniques

2.2.1 Random Oversampling

Random Oversampling multiplies rows of the minority class, without changing any of its features. It's important to highlight that the observations from the minority class are not multiplied equally, meaning they are randomly chosen with replacement. For instance, some observations may be duplicated once, while others are duplicated thrice.

2.2.2 Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique (SMOTE) first selects one observations from the minority class, then finds its k nearest neighbours, typically k=5, randomly picks one neighbour, draws a line between the two observations in the feature space and a synthetic new observation is created at a random point in that line.

2.2.3 Adaptive Synthetic Sampling Method

Adaptive Synthetic Sampling Method (ADASYN) is very similar to SMOTE, the main differences are that the number of synthetic samples to be generated from an observation belonging to the minority class are decided by a density distribution, whereas in SMOTE every minority class observation has an uniform weight, and the synthetic observation features have a small random component added to them, allowing them not to be linearly correlated with their parent observations.

2.3 Feature Importance Techniques

Elastic Net is a penalized linear regression that includes both the L1 penalty, used in Lasso regression and the L2 penalty, used in Ridge Regression. The L1 penalty penalizes the model based on the sum of the absolute coefficient values and the L2 penalty penalizes the model based on the sum of the squared coefficient values, both penalties minimize the size of all coefficients, although only the first one allows the coefficient to be zero, effectively removing a predictor from the model.

$$\begin{aligned} L(\beta, \alpha, \lambda) &= \|X\beta - y\|_2^2 + \lambda \left((1-\alpha)\|\beta\|_1 + \alpha\|\beta\|_2^2 - t\right) \\ &= \|X\beta - y\|_2^2 + \lambda(1-\alpha)\|\beta\|_1 + \lambda\alpha\|\beta\|_2^2 - \lambda t, \end{aligned}$$

$\alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2}$.

3. METHODOLOGY

3.1 Materials

The greater helper in this study was Python and its enormous variety of libraries and packages that contained all the statistical methods, functions, and models crucial to this project. It's important to emphasize that some of the most important modelling procedures were made with the usage of the scikit-learn library. The disclosure of such methods will be available on GitHub after the evaluation period as an opensource notebook that can be opened in a variety of notebook environments. Particular emphasis should be given to the developed functions along the report that were of great use during the development of such, and might be useful in some similar analysis.

3.2 Procedures

The research was conducted in 3 main steps: data preparation, feature selection and exploration, model development. As for the first step, the data was already collected and so there was not much to worry about it.

3.2.1 Data Preparation

The data preparation part was mainly divided into 3 larger groups, and on the scope of the first, there was the need to correct some errors from data entry, coherence checking the data, and treat missing values, outliers, and typos.

As so, the most alarming makeover on this step was the transformation of "?" on the categorical variables, that indeed represented missed values.

To make sure that the data contained only physically possible values, the next step was to make sure that there were no negative values in the variables that could only allow positive or null ones, confirmed that every person could not have paid to go to Newland and at the same time been able to pay for it too. Regarding the education level the individuals had, it was assured that the levels of education had the correct corresponding number of years, and lastly, no one could work more hours per week than the week contains.

The subsequent step consisted of transforming the data to make it valuable to any modelling technique. As such, the first logical step was to transform the Birthday of the individuals into a numerical variable, which contained the Age of the individual as of 2048. As of the variable "Base Area" there was the need to introduce a dummy variable that contained information whether the individual was from the Northbury base or not since this was the case for the great majority of the observations and there were 39 different base areas. Then, the usage of the person's name on the model didn't seem possible, and so it was transformed, with the help of the title of the person, to a dummy variable that introduced the Gender information in the model. On the variables Lives With, Marital Status, and Profession an aggregation of some logical categories was made. The Education Level variable was perhaps the trickiest one. As it is, a naturally ordered variable, even if it is categorical, the difficult part was to understand how to order it with the given labels. As such, the joining of some labels began from the highest education to the lowest. But further, it was recalled that most of the citizens from Newland

were Europeans and the EU provided a naturally ordered classification system for its citizen's education level, and so that was the one that was used. This solved a greatly huge concern that was whether the "Professional School" label should be at the same level as a High School Graduate or above. The "Role" variable was also divided into the 4 sectors of the economy.

The next step was to impute the missing values on the dataset, and two paths were followed. Firstly the missing values were imputed by the mode of their respective categorical variable. This is far from the perfect solution but was fast and also temporary, which allowed us to progress with the project and not be stuck in this step. After having an almost complete project the missing values were imputed by a predictive model, a Neural Network. A model was estimated for each variable with a missing value. This required the partition of a train and validation dataset from the observations where there were no missing values, so that the predictive model could be properly trained and assessed. Only two categorical variables had missing values. While in the first one, Employment Sector, the accuracy of the prediction on the train and validation wasn't any different than the accuracy of the mode in the same partitions, 0.78, the accuracy of the predictive model of the second variable, Role, was substantially greater than the accuracy when using the mode, 0.57 when compared to 0.47.

Without the presence of missing values the variables could now be one-hot and label encoded. The Education Level variable, seeing as it is the only ordered categorical variable, was the only one who was label encoded, all of the other categorical variables were one-hot encoded.

Some extra variables were also created, by transforming and simplifying some of the numerical variables, but they didn't replace them. The purpose of their creation was to later see in the model training faze if the models performed better with the full continuous variables or with "simplified" versions of them. No one paid to go to Newland and also received money to go there, which was also confirmed by some of the coherence checks made, so a variable was created, Money Ticket, which is simply the sum of minus Money Received and Ticket Price. Binary variants of these variables were also created, where a 0 represented someone from the A group and a 1 represented someone from the group B or D respectively. All of the numerical variables were also binned.

To be able to advance further, there were only 2 steps left: the outlier removal and the multicollinearity analysis.The outliers were identified and removed by analyzing the histograms and boxplots of the numerical variables. This step deleted around 2.3% of the observations.

3.2.2 Feature Importance and Selection

As referenced, the multicollinearity step was crucial to the understanding of the general picture of the data. Multicollinearity was found by analysing the correlation between features using the Spearman coefficient (although Pearson is the most widely used, the numeric variables did not seem to be normally distributed and most of the variables were binary, and so the Spearman rank correlation method became the most reliable). To make sure the numeric variables were not normally distributed, it was important to make a Shapiro-Wilk test. The p-value for all numeric variables was always close to 0 and so, the null hypothesis was rejected and it is concluded that there is no statistical evidence of the presence of normality distribution in the analyzed variables.

As the p-value was smaller than the chosen alpha, the null hypothesis was not rejected so there is statistical evidence that the data is not normally distributed. When there was a correlation index greater than 0.8, the criterion to decide which variable to delete was the one least correlated with the dependent variable, except in one of the cases. Both variables, Europe and Africa, had a really low and almost equal correlation with Income. As both are binary variables, the criterion to choose which to eliminate was the one with the most unbalanced proportion of zeros and ones. So the variables Education Level, Married and Africa were eliminated. Of course, this criterion of a correlation above 0.8 wasn't applied to the extra variables created, since they obviously are highly correlated with their "parent" variables but won't be used together in the same model, and will also not be considered during the feature selection.

One of the methods of feature selection that was prioritized was the usage of statistical tests to grasp a better understanding of the relationship between the independent variables and the target and as such, it was performed a Chi-Squared test to the binary variables and the ANOVA testing to the numeric variables.

Analyzing the p-value for each variable in both tests, 2 variables failed to reject the null hypothesis of the Chi-Squared Test: Northbury and Asia, so they were eliminated. All the numerical variables were able to reject the ANOVA's null hypothesis.

Since some of the feature selection algorithms and models used require that no multicollinearity exists among the variables, and there were still some "complete groups of one-hot encoded variables", one variable from each of these groups was eliminated, the criteria to chose which one to eliminate was the one that had the least correlation with the dependent variable. The variables Alone, 3rd Sector and Not Working were eliminated.

Before the feature selection the data was split into the train (60%), validation (20%) and test (20%), always with the dependent variable stratified, making sure that the proportion of both classes remained the same among all sets, and only performed the different methods and algorithms on the train data.

The feature selection was conducted by ranking the different features by importance, separating the training set into numeric and binary variables as in most feature selection methods employed, if they were used with the numerical features, the latter would always be ranked highest due to their better discretization power. As so, in the end, the highest the score of each variable, the worst its feature importance compared with its peers.

The feature importance methods were divided in Wrapper Method (RFE with Logistic Regression), Intrinsic Methods (Decision Tree Classifier and Random Forest Classifier, using Gini index and entropy on both) and Embedded Methods (Lasso and Ridge Regressions and Elastic Net Technique, all of the alphas, and l1 to l2 ratio in the Elastic Net, were estimated using cross-validation of 10).

The selection of techniques was made due to detailed planning and understanding of each one's advantages. Firstly, gathering techniques that make as heterogeneous decisions will allow the selection to be made efficiently and with no bias.

After analysing every one of these feature importance methods, the final ranks were as follows:

Variable	Rank
Spouse	1
4th Sector	2
Never Married	3
Male	4
Children	5
3rd Sector	6
1st Sector	7
Public Sector	8
Not Married	9
Oceania	10
Private Sector	11
Europe	12
America	13
Other Family	14
Self Employed	15

Variable	Rank
Money Received	1
Age	2
Working Hours	3
Years Education	4
Ticket Price	5

Tables 2 and 3: Feature Importance Ranking on Binary Variables (2) and Numeric Variables (3)

3.2.3 Model Development

After the feature selection, it was time to reflect on the balance of the dataset and as so, the target variable had unbalanced classes (0-16933, 1-4947 on the whole dataset) and when looking at the confusion matrices of some models it was clear that there was a problem with the**False Negatives** (observations with an income above average classified as below average). The natural path to follow was to try out some oversampling techniques: Random Oversampling, SMOTE and ADASYN, on the train partition data to produce some balanced datasets.

The chosen models were divided into 3 main categories: **Simpler Models** (Logistic Regression, Gaussian Naive Bayes, Decision Trees, and Two Types of Instance-Based Learners), **Elaborate Models** (Support Vector Machine, Neural Network) and **Ensemble** (Random Forests, Bagging using KNN, AdaBoost, Gradient Boost and Stacking of GaussianNB, Decision Tree and KNN).

The technique used to better estimate the hyperparameters of the previously mentioned models was a **Grid Search**, preferably making sure the cross-validation was 10 to give us more robust results and using a combination of hyperparameters that was decided mostly on common practices, related to both the specific models and data, and constant adaptation. It is important to highlight that in some models this was an unnecessary step, for example in a Logistic Regression, and so it was not done.

Several steps were fine tunned along the way. The combination of variables was a crucial step in the development of the models, always having in mind the previously stated feature selection. It was also important to adjust different methods of standardizing (MinMax Scaler and Standard Scaler), use of outliers, and using the Oversampled data, towards trying to increase the predictive ability of the model's minority class. As the data was split into 3 partitions (train, validation, and test), in the model development phase only the train and validation partitions were used, logically the 1st to train the model and the 2nd to check if there was any over or underfitting.

The test partition was used only on the best models in terms of both accuracy (more specifically f1-score with a micro average, as this was the measure used on the Kaggle activity), precision and recall, to check once again if there was any overfitting or if the accuracy of the models remained equal. These metrics also allowed us to be able to distinguish between the best models that gave very similar results and be able to define which one should be chosen for the final one.

4. RESULTS

4.1 Data Adjustments

The first logical thought followed on this project was to use first the "original" variables to train the models, and the first things noticed was that with the usage of any of the extra variables (the bins and binary equivalent of the variables Money Ticket, Ticket Price, Money Received), in detriment of its "parent" variables, the models did not improve. Afterwards, it was necessary to experiment with different combinations of variables in the various models estimated, considering the feature selection ranking, reaching an optimal number of features, achieved by iteratively removing variables until the ten worst binary variables were removed. Removing the worst numeric variable was also tried, however, the models always performed worst. The models experimented always performed better with this specific combination of variables, except for the Logistic Regression, the Support Vector Machine and the Stacking, that still performed slightly better with the original dataset.The combination of variables that gave us the best results was: **'Years of Education', 'Working Hours per Week', 'Money Received', 'Ticket Price', 'Age', 'Male', 'Never Married', 'Spouse', '1st Sector' and '4th Sector'**.

As usual, the outlier removal was one of the first steps taken, but the models were also trained with outliers and did not provide better results with this specific combination of variables in the most robust models.

Due to the mentioned unbalance of classes in the dependent variable, all models found it difficult to predict observations with an income above average (1). This was easily shown on, for example, the recall metric that was consistently below 0.6. One of the solutions the authors came across was using oversampling.

The models capabilities of predicting 1s improved, however it reduced the model's ability to predict 0s, which had as consequence the appearance of some overfit. Let's now assess the problems in each of the Oversampling methods: **Random Oversampling:** as it duplicates observations of the minority class (which is clearly not an optimal solution), it does remove the problem of having fewer observations, though in return increases the weight given to the observations of the minority class due to their duplication. On **SMOTE and ADASYN** the estimation of 0s also got worst. Trying to assess why this would happen, it was later found that the binary variables were being interpreted as numeric, and there was no specific "parameter" to change this, which means that the new observations presented values different from zero and one in binary features. One solution found that was considered relevant was to round the values, which fixed the issue since the wrong new values were between zero and one. This means that in fact SMOTE and ADASYN were correctly performed only for the numeric variables and, when it came to the categorical ones, the new values were simply numeric approximations, which might explain the experienced problems. Although the first scaling method used throughout the model was the MinMax Scaler, this was not a static decision, and the StandardScaler was also tried out, although it didn't provide better results. At first, and as a quick fix, the imputation of missing values was made through

the mode. However, afterwards, it was developed a more robust method of imputation (using MLP) as stated on the Methodology section. Most of the models' accuracies slightly decreased after this change, however, this transformation was kept as it is known that the imputation with the predictive model provided more robust results than the one with the mode.

4.2 Model Tuning

The variance of the models' scores was not considerable, but there is a clear distinction between what was previously called **simpler** and more **complex** models. Nonetheless, it is important to highlight that there are some key differences in the recall scores, both in train and validation of AdaBoost and Gradient Boost. These were the only models that scored higher than 0.6 and had the best f1-scores with a micro average. These are the best candidates to the final model selection. A summary of the models used and its respective scores can be found in the table below:

Model	Train Score	Validation Score
Logistic Regression	0.846	0.853
Gaussian NB	0.827	0.839
Decision Trees	0.854	0.851
K Nearest Neighbour	0.827	0.839
K Nearest Centroid	0.827	0.839
Support Vector Machine	0.851	0.854
Multilayer Perceptron	0.862	0.859
Random Forest	0.868	0.866
Bagging KNN	0.86	0.856
AdaBoost	0.869	0.875
Gradient Boost	0.875	0.878
Stacking	0.858	0.86

Table 4: Models Made and Its F1 Scores(avg:micro)

4.3 Final Model

The MLP and Random Forest, although not having as good of a recall score as the Boosting models, still have considerable f1-scores and so will also be analyzed with the test partition of the data.As the test partition was still unused, it became a key in this final assessment, and so, the following graphic was plotted:

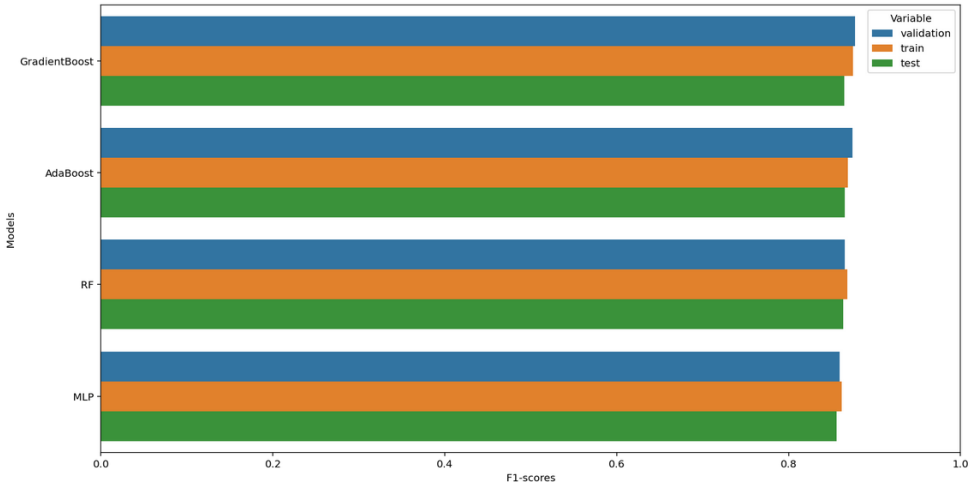

Figure 2: Scores for the Final Models

All models show very similar results; however, some differences can be observed. The Gradient Boost model, that before showed the best results, seems to show some overfitting when comparing the train value with the test.The MLP, although with very similar scores across all partitions, shows the lowest values. The Random Forest and AdaBoost seem like the most balanced models. The Random Forest has slightly more consistent results but a lower recall than the AdaBoost.

The AUC of each of these four models was also analyzed, as seen on the table below:

Model	AUC
AdaBoost	0.9178
Gradient Boost	0.9171
Random Forest	0.9149
Multilayer Perceptron	0.9049

Table 5: Area Under the ROC Curve

Once again, the values are very similar, however when comparing only AdaBoost with the Random Forest, the first one comes out on top, and so AdaBoost was selected as the final model.

It was also important to check the recall and the precision with the test dataset of the AdaBoost model to make sure that the values remained consistent. The model showed a precision of 0.77 on train and validation and 0.74 on test, a recall of 0.61 on train and test and 0.64 on validation and a micro average f1-score of 0.865 on test.

5. CONCLUSIONS

The main objective of this study was to train a model that would be able to categorize the citizens that are on the way to Newland with "Income Above Average" or "Income Equal or Below Average" and therefore be able to create the binary taxation system of the new planet.

The prediction of the final assessed model allowed us to correctly identify 87% of the observations. Assuming the Government of Newland has information about its resident's salary, they can now estimate confidence intervals for the revenue of its citizens' taxation contributions and, therefore, begin conducting studies for the realization of its state budget for 2049.

To further conduct this analysis it is recommended that a study should be conducted on the probabilities behind each one of the labels, to better assess the differences on the observations that have more balanced probabilities in each label. With the new upcoming spaceships and in case they come temporally spaced, the integration of these individuals' information should be beneficial, especially when the citizens with an income above average come first. This would allow the prediction of 1s to be made more accurately and would be a quick fix on the unbalanced target issue, that is believed to be one of the causes for the reduced scores on the models.

REFERENCES

[1] Brownlee, J., "Random oversampling and undersampling for imbalanced classification." https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/.

[2] Brownlee, J., "Smote for imbalanced classification with python." https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/.

[3] Brownlee, J., "How to develop elastic net regression models in python." https://machinelearningmastery.com/elastic-net-regression-in-python/.

[4] Europass(n.d.), "Description of the eight eqf levels." https://europa.eu/europass/pt/description-eight-eqf-levels/, (Retrieved 4 December 2020).

[5] Europass(n.d.), "Diagramasespeqfpt1." http://internacional.ipvc.pt/sites/default/files/Diagrama_SESP_EQF_PT1.pdf, (Retrieved 5 December 2020).

[6] Wikipedia, I., "Education in the united states." https://en.wikipedia.org/w/index.php?title=Education_in_the_United_States&oldid=996090808, (Retrieved 5 December 2020).

[7] gajawada, s. k., "Anova for feature selection in machine learning." https://towardsdatascience.com/anova-for-feature-selection-in-machine-learning-d9305e228476.

[8] gajawada, s. k., "Chi-square test for feature selection in machine learning." https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223.

[9] Johnson, M. K. and K., "Recursive feature elimination — feature engineering and selection: A practical approach for predictive models." https://bookdown.org/max/FES/recursive-feature-elimination.html.

[10] Vala, K., "Adasyn: Adaptive synthetic sampling method for imbalanced data." https://towardsdatascience.com/adasyn-adaptive-synthetic-sampling-method-for-imbalanced-data-602a3673ba16.

[11] (n.d.), "Chi-square test of independence." https://www.statisticssolutions.com/non-parametric-analysis-chi-square/.

[12] (n.d.), "One-way anova in spss statistics—step-by-step procedure including testing of assumptions." https://statistics.laerd.com/spss-tutorials/one-way-anova-using-spss-statistics.php.

[13] (n.d.), "Shapiro-wilk test—quick tutorial with example. (n.d.)." https://www.spss-tutorials.com/spss-shapiro-wilk-test-for-normality/.

[14] (n.d.), M., "How to appropriately plot the losses values acquired by from mlpclassifier." https://stackoverflow.com/questions/48123023/how-to-appropriately-plot-the-losses-values-acquired-by-loss-curve-from-mlpcl.

[15] (n.d.), M., "How to plot roc curve in python.." https://stackoverflow.com/questions/25009284/how-to-plot-roc-curve-in-python.