NOVA IMS Information Management School

Newland

Project Report

Maximilian Kurt Maukner - m20200645@novaims.unl.pt

Ehsan Meisami Fard - m20201050@novaims.unl.pt

Franz Michael Frank - m20200618@novaims.unl.pt

Steffen Hillmann - m20200589@novaims.unl.pt

Date of submission: 26th of December 2020

Table of Contents

A comparative study of algorithms for a binary classification problem

Abstract

In a case where a new government imposes a new two-fold income-tax, we used numerous classification algorithms to classify every observation of a given dataset. Subsequent to data cleaning, data transformation, data rebalancing and exploratory data analysis of the dataset which proves that the given dataset represent the true population, this report reviews multiple models and interprets their outcome in mainly two different stages. The first stage is prior to utilizing techniques such as k fold cross validation, hyperparameter tuning and feature selection where the second stage is after implementing the techniques mentioned. Iteratively and incrementally, the Gradient boosting model has yielded the best outcome out of the following classification algorithms: Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Gaussian Naïve Bayes, Decision Tree Classifier, Random Forest Classifier, Neural Network, SVC, K Neighbors Classifier. The vast majority of this report evaluates the performance of classification models using the F1 as the main evaluation criterion while the train accuracy and the duration of model execution act as subsidiary evaluation criteria.

Keywords: Classification, Gradient Boosting, Random Forest, Predictive Models, Machine Learning.

I. Introduction

In this report, the new government of Newland has passed a new income-tax of 15% for citizens below and equal to the average as well as a 30% income-tax for citizens who are earning above the average. This report is concerned with identifying citizens who belong to each class given the population data sourced from NOVA IMS. This can be modeled through a binary 1 and 0 classification implying above and below or equal average income, respectively. In this case, we are dealing with a classification problem and need to find the appropriate model to estimate the classification of each observation. Just like many other classification problems in Machine Learning, this challenge requires an iterative approach of tying multiple algorithms and techniques and using other methods on the dataset to improve the produced models. In this case, we have two distinct classes (0,1) which simplifies the challenge faced slightly as many algorithms function poorly on a multi-class classification problem.

The features of the dataset include 6 numeric and 9 categorical features while the full training dataset contains 22400 observations. Examining such classification problem requires finding methods to deal with unbalanced data set, experimenting with ways to clean the dataset and apply data engineering as well as feature engineering to extract the maximum amount of information from the given data and ultimately applying numerous classifications techniques to generate the best model score. For this classification problem, the following algorithms will be used:

Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Gaussian Naïve Bayes, Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, Neural Network, SVC, K Neighbors Classifier. The vast majority of this report evaluates the performance of classification models using the F1 as the main evaluation criterion while the train accuracy and the duration of model execution act as subsidiary evaluation criteria. These algorithms will be evaluated upon its F1 Score which is a function of Precision and Recall.

Firstly, this paper gives a brief theoretical introduction of some the techniques and algorithms that have not been discussed in practical classes. Secondly, it dives into the methodology section to provide a comprehensive overview of the dataset as well as the data- cleaning and engineering process followed by a description of the methods applied. Thirdly, it presents the key findings in two fashion. First by showing the initial results prior to the pruning phase and second by showing the ultimate final results after the pruning accompanied by an examination of their significance and further implications in the discussion section. Finally, it concludes the steps taken in this process and the final results.

II. Background

Out of the techniques we have applied for our classification Linear Discriminant (LDA) Analysis, Quadratic Discriminant Analysis (QDA), Light Gradient Boosting Machine (LGBM) are algorithms that have not been discussed in theoretical- or practical classes. Here is a brief introduction to each method:

LDA & QDA

LDA and QDA are two classic classifiers are two well-known supervised classification methods in statistical and probabilistic learning and are a linear and a quadratic decision surface, respectively [1]. These techniques become quite interesting for our problem solution as they don't require hyperparameter tuning and can be easily computed. LDA is mostly used due to its simplicity but also because it is not limited to only two-class classification problems. For instance, logistic regression is intended for two-class or binary classification problems and rarely used for multiple-class classification problem. In terms of differences, LDA and QDA, as their names suggest, are a linear and a quadratic decision surface.

The figure below helps to understand difference how these model function when used to test a binary classification. LDA (on the left side) shows a linear boundary while the QDA (on the right side) shows a rather flexible quadratic boundary:

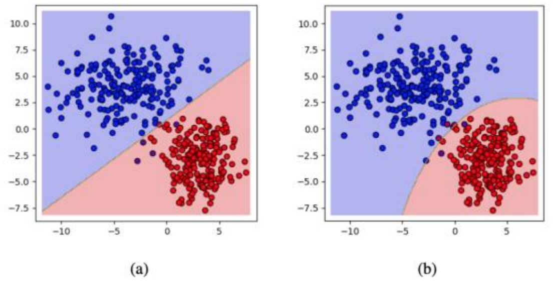


Figure 1. (a) LDA for two classes, (b) QDA for two classes

Moreover, where logistic regression is a classification technique commonly limited to only two-class classification problems, LDA and QDA are able to distinguish between multiple classes. Yet it is understood that the LDA is a foundation of QDA and many more algorithms such as the Flexible Discriminant Analysis and Regularized Discriminant Analysis. Concerning LDA, although it often used for dimension reduction similarly to Principal Component Analysis (PCA), it focuses on maximizing the separability among different classes. Simply said, LDA uses the data of all classifications, which in this case would be (k=0,1), to create a new axis in a way to maximize the separation between the two classifications. Similarly, QDA functions the same way yet the new axis will be a flexible quadratic decision.

Light Gradient Boosting Machine

Lastly, Light Gradient Boosting Machine, or in short LGBM, is tree-based learning algorithm developed at Microsoft that has been used to speed up the implementation of the gradient boosting algorithm which leads to a more effective model. The main difference between LGBM and other tree-based algorithms is that it operates vertically while other algorithms grow tree horizontally. Alternatively stated, it grows tree leaf-wise while others grow level-wise. Moreover, LGBM is able to distinguish multiple-class classifications and is not limited to only two classes. In a comparative classification model evaluation, LGBM classifier is the winner in accuracy and F-Score among six other well-known classifiers for sentiment analysis: Logistic Regression, SVM, Random Forest, Multinomial Naïve Bayes, Extreme Gradient Boosting, and Deep Learning [2]. In addition, it has demonstrated its ability to converge faster than Extreme Gradient Boosting and Deep Learning classifiers. More importantly, it has shown its strength in handling imbalanced dataset classes which become important due to the imbalance in our Newland dataset.

After explaining the theoretical background of the LDA, QDA and LGBM, we dive into the Methodology section where we go through the process of data cleaning, data engineering as well as methods used to improve the models created.

III. Methodology

Data preparation

First of all, we apply a prevalent examination of the data to obtain a general overview. The given train dataset consists of 22,400 rows and 15 columns, free of any undefined values or generally known as *Nan* values. The dependent variable is the binary field *Income* in which 1 corresponds to an income higher than the average and 0 to an equal or lower one. Six out of the 15 columns are integers while the rest are objects. However, there are several values that only represent a question mark in the columns *Base Area*, *Employment Sector* and *Role*. This illustrates a lack of information available for the respective column of a specific row which is equivalent to a *Nan* value. All of these values are replaced by the most frequent value of the corresponding column. This is a better way to approach this issue than the alternative option, which is to drop all affected rows, as the latter would lead to a major loss of information, in such case 1,649 rows. These observations represent more than seven percent of the whole train dataset. Furthermore, a scan for duplicates among the rows has confirmed that the data consists of unique rows only.

A closer look at the columns reveals that the column *Citizen_id* will not be relevant for the upcoming model as it has no relation to the target variable and is only supposed to be used for the unique identification of citizens. As a result, this field gets dropped from the data frame which supports our goal to simplify and reduce dimensionality.

Feature engineering

To start the process of feature engineering, the *Birthday* column needs to be converted into a usable format. One reasonable method to achieve this is to extract the age in years as a continuous variable from the information given on the birthdate. Hence, the ninth of September of 2048 is chosen as the current date, as the youngest person in both the train as well as the test dataset were born on this day and both are already over 17 years of age at the current time. Consequently, the column is renamed to *Age*.

Furthermore, the gender of the respective citizen can be retrieved from the *Name* field. For this purpose, the titles of the respective persons are required. There are only three different ones in total. The title *Mr.* stands for the male gender and the titles *Mrs.* and *Miss* represent the female one. This information is used to replace the original field with a new binary column called *Gender* in which the value 0 corresponds to female and the value 1 to male. Thus, two columns of the type object, namely *Birthday* and *Name*, are already transformed into columns of numerical type by the feature engineering process. To complete the process, we check the data for constant features, yet a test shows that a constant feature does not exist in our data frame and thereby we do not take any further action in that respect.

Up and down-sampling

Upon closer examination of the target variable *Income* in the train data frame, we clearly observe a discrepancy in the distribution of the binary values. Meaning the frequency of each value is highly disproportional. There are 17,089 rows assigned to the value 0 and 5,311 to the value 1, hence it is an imbalanced dataset.

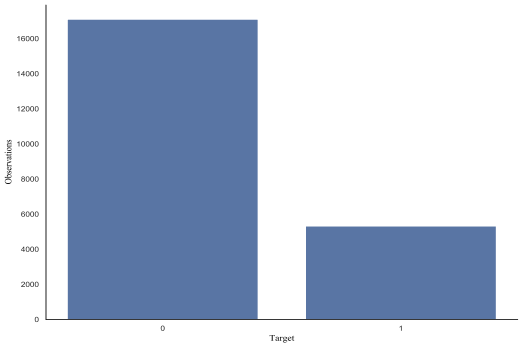


Figure 2. Distribution of the Target Variable

This issue must be resolved, because continuing with unbalanced data can result in overfitting and wrong correlations. If an unbalanced dataset is to be investigated, it is of great importance that it gets first into balance. In addition, literature [3] suggests that having balanced dataset would improve the learning process which gives us sufficient incentive to solve this issue at hand. Despite numerous algorithms, such as LGBM, do not penalize imbalanced classification, most machine learning technique will ignore this issue and perform poorly. The performance discount is specially observed on the minority class which in most cases is the most important class. For instance, when developing a cancer detection classification, one is interested to find true positives which is the minor class given majority testing negative for cancer.

Generally, there are three main techniques to achieve this, which are under-sampling, also called down-sampling, oversampling, also known as up-sampling, or hybrid sampling. With down-sampling the majority class of the dataset is reduced in such way that the number of rows assigned to the majority class is equal to the number of rows assigned to the minority class. This has the advantage that no data points are duplicated, but the disadvantage that the information of the deleted rows of the majority class is lost. With up-sampling, on the other hand, it is exactly the opposite. Hereby the number of rows of the minority class is adjusted to the number of rows of the majority class. This means that the minority class may be displayed slightly blurred, however, there is no loss of information in the majority class. The number of rows is therefore usually several times higher by using the up-sampling technique than by using the down-sampling one.

In this scenario, the simplest approach involves up-sampling which is duplicating examples from minority class. However, this approach does not add any new information to the model. Therefore, we use a new method which is referred to as the Synthetic Minority Oversampling Technique, or simply SMOTE [4]. According to literature [5], this technique first selects a sample from the minority class at random and finds its k nearest class neighbors. Furthermore, it creates a synthetic instance by randomly choosing one of the k nearest neighbors (typically k=5) at a randomly selected point between two examples in feature space.

As a result, it is possible to continue with 34,178 data points instead of 10,622 that would be left by using the down-sampling technique, which represents more than triple the amount [6].

An additional obstacle to overcome is the remaining 7 non-numerical features in the data frame which have more than two distinct values and a nominal scale. Thus, the concept of one-hot-encoding becomes a suitable solution which creates binary dummy variables for all different values of all columns [7]. Consequently, the data frame is free of columns with the object type, however the number of columns increases by a total of 88.

As all fields are in a numerical format, yet not all of them are binary, the data requires normalization in order to train certain models properly. The module *preprocessing* from the *sklearn* library is useful for this purpose [8]. Applying this method results in a transformation of all remaining continuous features, namely *Age*, *Years of Education*, *Working Hours per week*, *Money Received* and *Ticket Price*, into a scaled and normally distributed format.

At this stage, to subsequently train and test the model, the dataset must be split into a train as well as a validation dataset. The given test dataset cannot be used as the validation dataset due to the missing dependent variable, which is essential for the model evaluation. The training dataset holds 70% of the original data and the validation dataset 30%, as this is a reasonable trade-off between enough training data for an accurate model and sufficient test data for precise testing [9].

Model creation and selection

Once the data is suitably prepared, the model creation process can be initiated. It is very helpful to write a function that calculates four important details for a given model, namely the train accuracy, the test accuracy, the F1 score and the running time. This allows the models to be compared with each other in an efficient and transparent way. At the beginning of the model creation and comparison process, it is advisable to compare as many models as possible, whose parameters are initially all default. This gives a first overview of which models perform better for the given problem. Since the F1 score is a good indicator at this point, the models with the highest F1 scores should be examined more closely afterwards. These models can then be examined and improved in more detail. For instance, a major problem that generally needs to be solved is overfitting. A good measurement for detecting overfitting is the accuracy score in the train dataset, where a high score hints at a likely overfitting issue [10].

The robustness of the model should also be verified. To reach this goal, cross validation is a suitable technique which investigates whether the performance of the model on different folds is consistent and does not vary excessively.

Hyperparameter tuning

To further improve the models examined, it is recommendable not to use the default parameters, but to identify the best ones instead. The respective best parameters for a specific task can be determined by a hyperparameter tuning [11]. For this purpose, it is useful to write a function which tests a base model with given different parameter ranges and then captures the parameters with which the model performs best. However, a better train accuracy score does not necessarily mean that the overall performance of the model is also better, as this might indicate that the model is overfitting. Therefore, all four important variables, i.e., train accuracy, test accuracy, F1 score and running time, should be taken into account when evaluating a model.

Feature Importance

Feature importance refers to a technique that determines how useful the input features are for predicting a target variable. Applying such technique can have numerous merits such as dimensionality reduction which can improve the efficiency as well the effectiveness of our classification models. In addition, it gives insights concerning which features may be most relevant to the target. This may be interpreted and be used as foundation for collecting more and different data. In this case, we use the Random Forest algorithm for Feature Importance implemented in scikit-learn.

Feature Selection

Once the optimal model with the ideal parameters is found, another major decision arises, namely the feature selection. By applying feature selection, we are able to reduce model complexity, training time and overfitting while increasing the generalization ability. However, this might lead to an information loss which causes the models to perform worse than before.

Despite the advantages of such technique, it is quite vital to consider the performance of the model in terms of accuracy and thus a certain reasonable trade-off should be considered. A useful tool for this is the *RFECV* module of the *sklearn.feature_selection* library, which identifies the most important features and the optimal amount. Furthermore, it can be insightful to compare these results with the feature important scores of the respective model.

Having completed the final model, it can then be applied to the given test dataset to assign the respective rows to one of the binary values for *Income*. The prerequisite for this is that the test dataset must be edited in the same way as the train dataset. This ranges from data cleaning, feature engineering and normalization to feature selection. Only if the test dataset is in exactly the same format as the train dataset with which the model was trained, meaning it has to have the exact same columns with normalized values, the model can be successfully applied to the test dataset. An overview of the whole process is shown below:

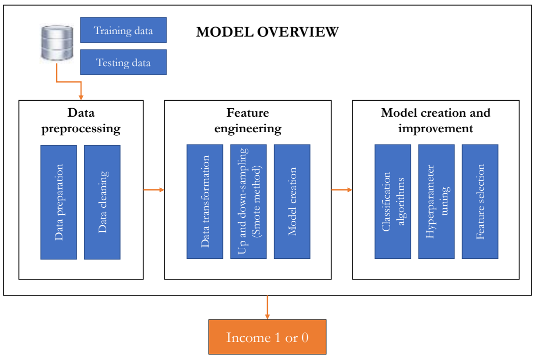


Figure 3. Model overview

VI. Results

Exploratory data analysis

Before jumping into the classification algorithms and other techniques we have implemented, it is important to understand the dataset first and look out for unanticipated findings. It is critical to analyze features that could have an impact on our results and investigate whether our sample is representative which is an assumption for certain classification algorithms. Thus, we focus on several features and try to plot them against our dependent variable.

As illustrated in the figure below, we observe a higher income as the years of education increases.

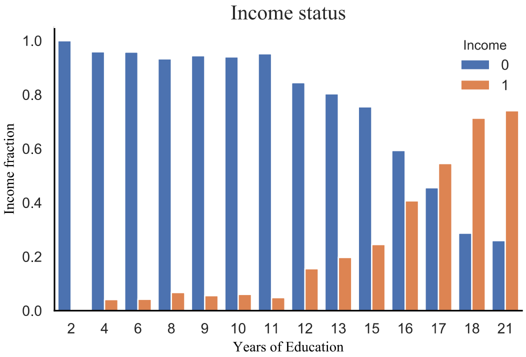


Figure 4. Income Status

However, by further examining the data, the figure below encapsulates the fact that the years of education of our sample is representative of the true population.

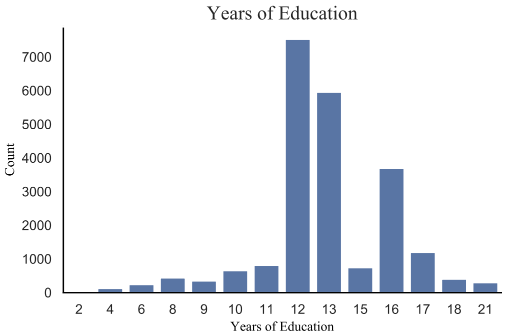


Figure 5. Year of education

Similarly, we observe a rise in income as the age increase which raises the question whether the age distribution in our sample is representative of the true population.

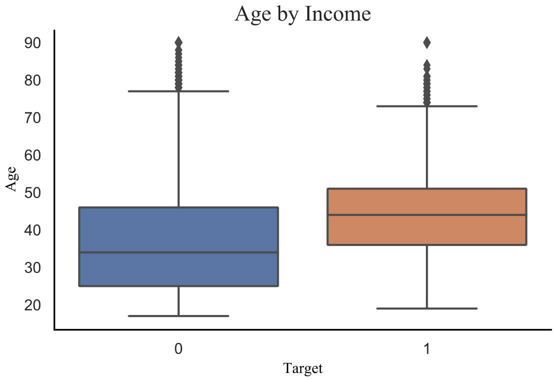


Figure 6. Age by income

Yet, we have plotted the age frequency table in the figure below which confirms the age distribution of the sample is representative of the true population.

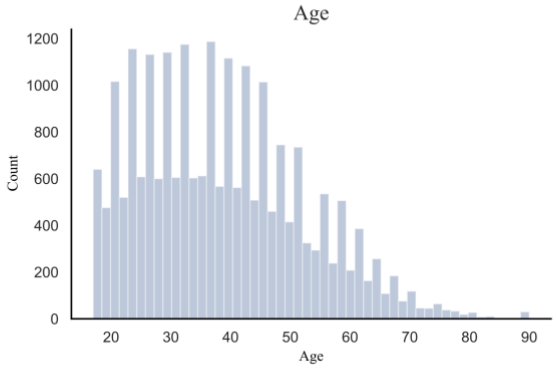


Figure 7. Distribution of variable Age

Last but not least, the amount of weekly working hours of the train data is important to explore as well, as it should also display a reflection of the true population.

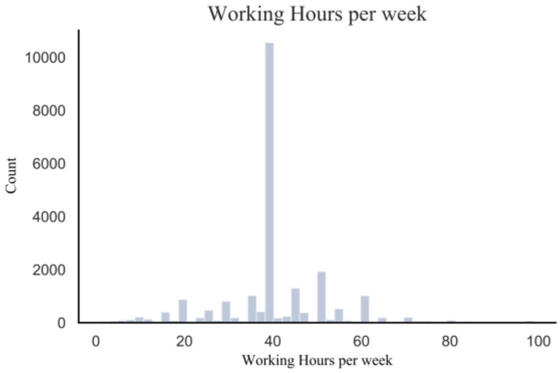


Figure 8. Distribution of variable Working Hours per week

Initial results

In this section, we illustrate and describe the results of the models we have implemented prior to any pruning or any other methods. Finding the best performing model is an iterative approach which requires multiple attempts with the same exact model, therefore we show the initial results as well as the final results both in this section of the paper. The figure below shows the numerous models we have implemented and displays their corresponding F1-Score.

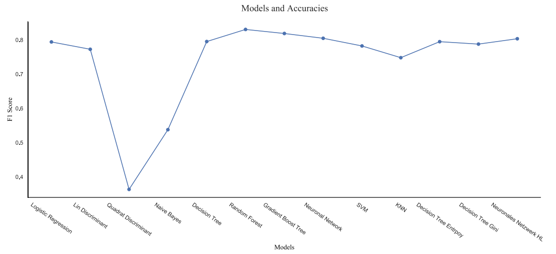


Figure 9. Initial F1 scores

Although the illustration above is poor when it comes to comparing models that have a slim margin of F1 score differences, yet for our initial analysis, it shows us the "outliers". Thus, as seen in the figure above, we can conclude that QDA and Naïve Bayes have the lowest F1-Score among all techniques implemented with QDA performing worse by a huge margin. Thus, we would not consider a model improvement for them from here onwards. In the following table we can see the corresponding train accuracy of each model:

Model	Acc. Train
Logistic Regression	0.8081
Linear Discriminant Analysis	0.7862
Quadratic Discriminant Analysis	0.3511
Gaussian NB	0.5459
Decision Tree Classifier	0.9844
Random Forest Classifier	0.9844
Gradient Boosting Classifier	0.8367
Neuronal Network	0.8853
SVC	0.8089
K Neighbors Classifier	0.8471

Table 1. Initial train accuracies

The results of the previous table show us that on multiple occasions the models might have an overfitting problem due to their high train accuracy. Therefore, we enter the *pruning* phase which enables us to improve the existing models and prevent them from overfitting.

Pruning

To begin with, in respect to the Decision Tree model, we chose the default value for the maximum depth which will often result in overfitted decision trees. Similarly, Neural networks are often over-parameterized with significant redundancy in the number of required neurons which does not only lead to overfitting but also unnecessary computation and memory usage [12]. In the figure below, we compare the test accuracy (in orange) with the train accuracy (in blue).

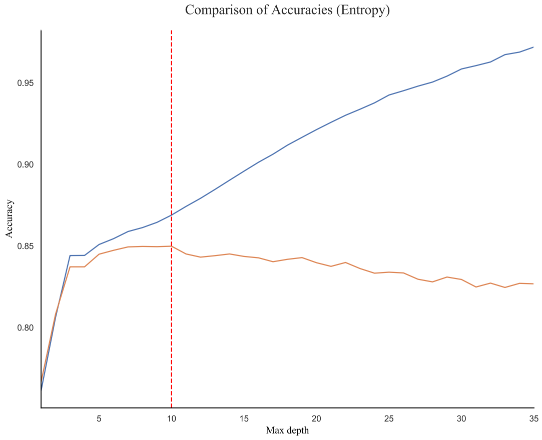


Figure 10. Decision tree - Max depth

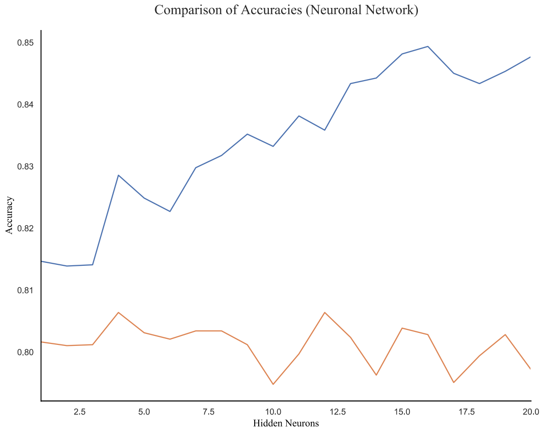


Figure 11. Neural networks - Hidden neurons

Consequently, we set the maximum depth of our Decision Tree model to 10 as higher depth result in a negative effect on the test accuracy. Similarly, we achieve the highest test accuracy when setting the hidden neuron to 8 for the Neuronal Network model.

Model	Acc. Train	F1 Score	Time
Decision Tree Classifier	0.9844	0.7949	0.3523
Neuronal Network	0.8853	0.8115	44.3151
DecisionTree Gini & depth 10	0.8274	0.8119	0.22
Neuronal Network & 15 HN	0.8480	0.8079	11.01

Table 2. Decision tree and Neural network

As the results show, the overfitting issue has been solved however we have not obtained a significantly better F1 score as desired. Moreover, we implement k fold cross validation to ensure the models are preventing the noise in data along with low variance and bias. Performing a k fold cross validation on every model where *k=10* as it is recommended as a general rule and empirical evidence [13].

Model	Acc. Variance	F1 Variance
Logistic Regression	0.0001	0.0004
Linear Discriminant Analysis	0.0001	0.0004
Quadratic Discriminant Analysis	0.0014	0.0001
Gaussian NB	0.0005	0.0001
Decision Tree Classifier	0.0001	0.0002
Random Forest Classifier	0.0000	0.0003
Gradient Boosting Classifier	0.0000	0.0003
Neuronal Network	0.0000	0.0008
SVC	0.0001	0.0004
K Neighbors Classifier	0.0000	0.0002

Table 3. K fold Cross Validation (k=10)

The consistent low variances observed in the table above shows us the robustness of our models and the fact that the models are not prone to overfitting.

Applying hyperparameter tuning

Furthermore, we use hyperparameter tuning, specifically the Randomized Search CV, for Gradient Boosting classifier as well as Random Forest classifier which can have a significant impact on the performance of the model that is being trained. As a result, with respect to Random Forest, despite a decrease of 15% in train accuracy, the F1-Score has improved slightly. However, the Gradient Boosting classifier has boosted its F1-Score by 4% which can be seen in the figure below:

Model	Acc. Train	F1 Score	Time
Random Forest Classifier	0.9844	0.8302	4.0630
Gradient Boosting Classifier	0.8367	0.8188	5.7561
Random Forest Classifier Tuned	0.8585	0.8461	8.45
Gradient Boosting Classifier Tuned	0.8338	0.8500	0.90

Table 4. Gradient Boosting and Random Forest tuned

Random Forest Classification Feature Importance

By applying Feature Importance with Random Forest Classification, several features stand out as very important for predicting our target variable such as *Age, Married (Marital status), Working hours per week, Year of education, Lives with wife* and *Single (Martial status)* ordered after their corresponding Feature importance score.

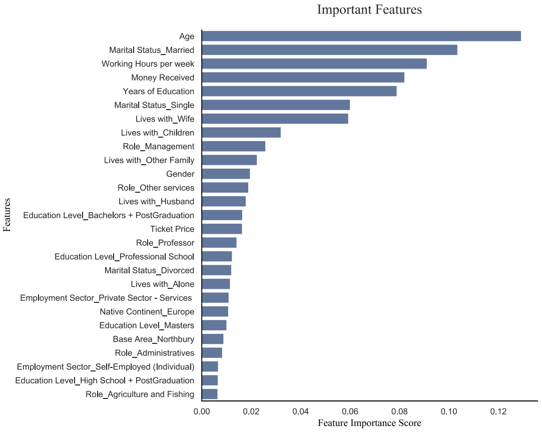


Figure 12. Feature importance overview

Feature Selection

Moreover, we have implemented *RFECV*, the abbreviation of recursive feature elimination and cross-validated, which is tuning the number of features selected with cross-validation and reduces model complexity as well as the dimensions of the dataset along with its running duration.

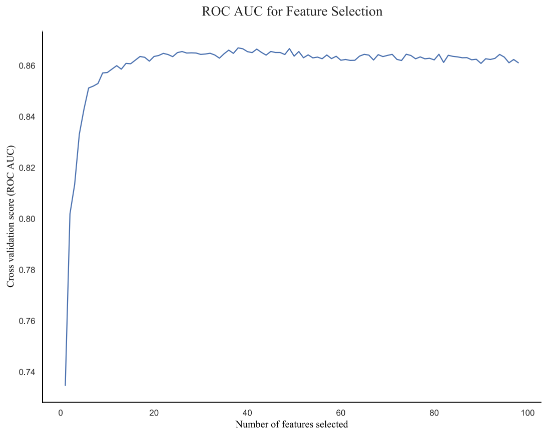


Figure 13. ROC AUC for Feature selection

As a result, 38 features have been selected out of the initial 99 features as the optimal number of features according to their cross validated ROC AUC accuracy.

Running the Grading Boosting Classifier and the Random Forest Classifier with 38 features selected, we observe a slight improvement in all of our model evaluation metrics in respect to the Gradient Boosting Classifier, however, a striking significant improvement is shown in the duration of the model execution due to the reduction in features. The model evaluation metrics as well as the confusion matrix for the Gradient boosting classifier are depicted below:

Model	Acc. Train	F1 Score	Time
Random Forest tuned	0.8577	0.8207	9.6760
Gradient Boosting tuned	0.8338	0.8219	1.3077
Random Forest feature selection	0.8885	0.8204	7.2466
Gradient Boosting feature selection	0.8993	0.8368	1.8448

Table 5. Feature selection for Gradient Boosting and Random Forest

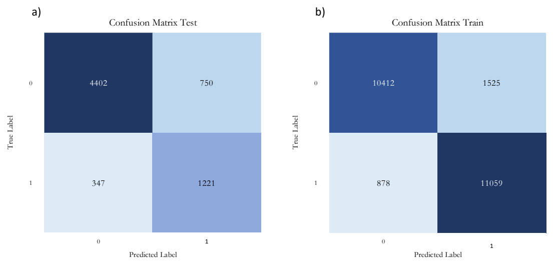


Figure 14. Confusion Matrix (a) Test Data (b) Train Data

By comparing the results of Feature Selection and Feature Importance, which is shown in Appendix A we come to the conclusion that RFECV selects features which the Random Forest Classifier would not take into account. Due to the more accurate calculation of RFECV, we decide to go with the selected features instead of choosing the feature importance of the Random Forest Classifier.

Model evaluation based on ROC/AUC

As feature selection leads to an information loss ultimately, it's vital to test the significance of this consequence. Thereby, we use the ROC AUC score which shows the model performance in terms of True Positive rate and False Positive Rate. Despite applying feature selection, the previous model and the new model distinguish themselves only by a -0.00049 which is quite insignificant. Thereby, we continue with the feature selection technique as this has barely affected our model performance.

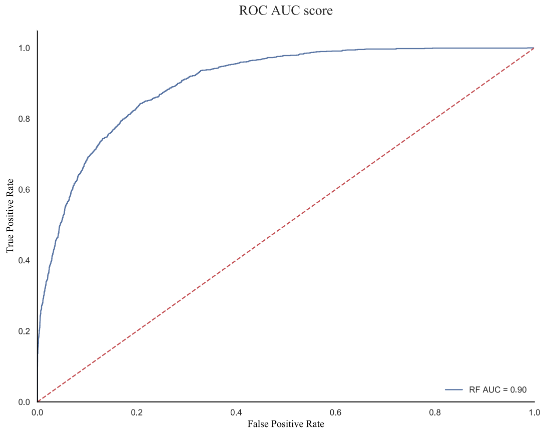


Figure 15. Model evaluation (ROC AUC score)

Model selection

At last, following the implementation of numerous classifiers and other techniques to improve the existing models, we approach the last model overview. It's important to mention that we have used 0.733 as our classification threshold for all the models where a value above 0.733 indicates the dependent variable, income, to be 1 and a value below indicates 0. This specific threshold has been chosen after testing all possible thresholds in a for loop to ensure we have the optimum threshold to generate the highest micro F1 score for our models.

Model	Acc. Train	F1 Score	Time
Logistic Regression	0.8081	0.7940	0.8962
Linear Discriminant Analysis	0.7862	0.7726	0.6608
Quadratic Discriminant Analysis	0.3511	0.3427	0.1577
Gaussian NB	0.5459	0.5378	0.0645
Decision Tree Classifier	0.9844	0.7949	0.3523
Random Forest Classifier	0.9844	0.8302	4.0630
Gradient Boosting Classifier	0.8367	0.8188	5.7561
Neuronal Network	0.8853	0.8115	44.3151
SVC	0.8089	0.7823	46.8215
K Neighbors Classifier	0.8471	0.7481	0.9367
DecisionTree Entropy & Depth 10	0.8248	0.7948	0.18
DecisionTree Gini & depth 10	0.8274	0.8119	0.22
Neuronal Network & 15 HN	0.8480	0.8079	11.01
Random Forest Classifier Tuned	0.8577	0.8207	9.6760
Gradient Boosting Classifier Tuned	0.8338	0.8219	1.3077
RandomForestClassifier fs	0.8885	0.8204	7.2466
GradientBoostingClassifier fs	0.8993	0.8368	1.8448

Table 6. Final results

V. Discussion

As mentioned in the *methodology* section, throughout the entire process, we have utilized the *sklearn* machine learning library. Although *sklearn* is a framework for a rather general machine learning models, other frameworks, such as *TensorFlow,* have their merits and purposes. For instance, *TensorFlow* could be used for a more complicated deep learning model with much a higher degree of freedom compared to *sklearn*. On the other hand, *sklearn* is optimal for relatively small sample size as the dataset with an average computing power while TensorFlow requires a better computing power, especially an external GPU. Utilizing this work as the groundwork for implementing increased complicated model could lead to a more accurate classification and ultimately much better anticipated result. Furthermore, there are also other models that could be both out of scope of the practical classes yet lead to a higher classification accuracy such as XGBoost and Gradient Boosting Machines. GBM is one type of ensemble learning which merges a set of weak models to build one strong model. It functions based on minimizing the errors of earlier leant models. Furthermore, as many algorithms do not function properly when the data is unbalanced, LGBM, as discussed earlier, does not necessarily require a balanced data to perform correctly. For the future work it would be to try out LGBM by itself prior to data balancing.

In respect to the feature engineering, we chose to although Label Encoder is inferior to the One hot encoder method as Label Encoder imposes ordinality and our categorical features have a nominal scale, there are numerous alternatives to the One hot encoder technique.

Furthermore, in this report, we used the SMOTE method to handle to unbalanced classification dataset that we initially had. Despite many merits that the SMOTE method has, one could argue that there are better approaching to deal with our minority class than a synthetic sample.

Lastly, despite using the RFECV Feature Selection method over Random Forest Classification Feature Importance, results of both methods give us a massive insight on what kind of data is important for predicting our target variable. This can does not only offer insights regarding the current dataset but could also be used as a foundation to gather further data which supports predicting the target variable better.

VI. Conclusion

To summarize, we have implemented numerous algorithms and techniques to generate the best outcome for our classification problem.

Prior to our model creation and selection, we have analyzed the entire dataset and implemented data cleaning by handing missing data as well as data transforming by turning impractical features into meaningful and purposeful information. Moreover, we handled the imbalanced data given and used the SMOTE method to synthesize new observations for the minority class while minimizing overfitting which alternative solution might have caused. Moreover, we implemented hyperparameter tuning as well as feature selection techniques to diminish the under- and overfitting problem that arises when training model which ultimately led to our model performance enhancement. Throughout the whole report, the model evaluation was based on its accuracy, F1 score and lastly the time it took to execute. Despite the initial results showing a high train accuracy which indicates an overfitting problem, we observe a significant incremental improvement over the course whilst applying multiple techniques to address overfitting and bias. We used k fold cross validation (k=10), hyperparameter tuning and feature selection to differentiate the superior model from the rest by exposing the effectiveness of the models. Consequently, we were successful to increase the F1-Score while decrease the model execution duration in conjunction with model complexity. Subsequent to model validation and pruning, Gradient boosting and Random forest classifier were the best performing models, while Gradient boosting classifier was the best of both of them with a F1-Score of 0.8368 and an execution time of 1.8448 seconds relative to Random Forest with a F1-Score of 0.8204 and an execution time of 7.2466 seconds. In respect to Gradient Boosting and Random Forest, comparing the initial model outcome with the latest results we observe a significant improvement of the models by only executing cross validation as well as feature selection and parameter tuning.

VII. References

[1] B. Ghojogh, M. Crowley. Linear and Quadratic Discriminant Analysis: Tutorial, June 2019, p.1

[2] F. Alzamazami, M. Hoda, A. E. Saddik. Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evalution. IEEE Access. 2020, p. 101840 - 101841. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9099543

[3] G. M. Weiss and F. Provost. Learning when training data are costly: The effect of class distribution on tree induction. J. Artif. Intell. Res., vol. 19, pp. 315–354, Oct. 2003.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Interlligence Reasearch, 2002, 16, p. 321-357

[5] H. He, Y. Ma. Imbalanced Learning: Foundations, Algorithms, and Applications, 1st edition. Wiley—IEEE Press, 2013

[6] Sachin S. Patil, and Shefali P. Sonavane. Improved classification of large imbalanced data sets using rationalized technique: Updated Class Purity Maximization Over_Sampling Technique (UCPMOT). In Journal of Big Data, volume 4, 2017.

[7] Patricio Cerda, Gaël Varoquaux, and Balázs Kégl. Similarity encoding for learning with dirty categorical variables. In Machine Learning, volume 107, 2018.

[8] Felix Mohr, Marcel Wever, and Eyke Hüllermeier. ML-Plan: Automated machine learning via hierarchical planning. In Machine Learning, volume 107, 2018.

[9] H. Liu, M. Cocea, Semi-random partitioning of data into training and test sets in granular computing contex, 2017, p. 357–386.

[10] Thomas Groensfelder, Fabian Giebeler, Marco Geupel, David Schneider, and Rebecca Jaeger. Application of machine learning procedures for mechanical system modelling: capabilities and caveats to prediction-accuracy. In Advanced Modeling and Simulation in Engineering Sciences, volume 7, 2020.

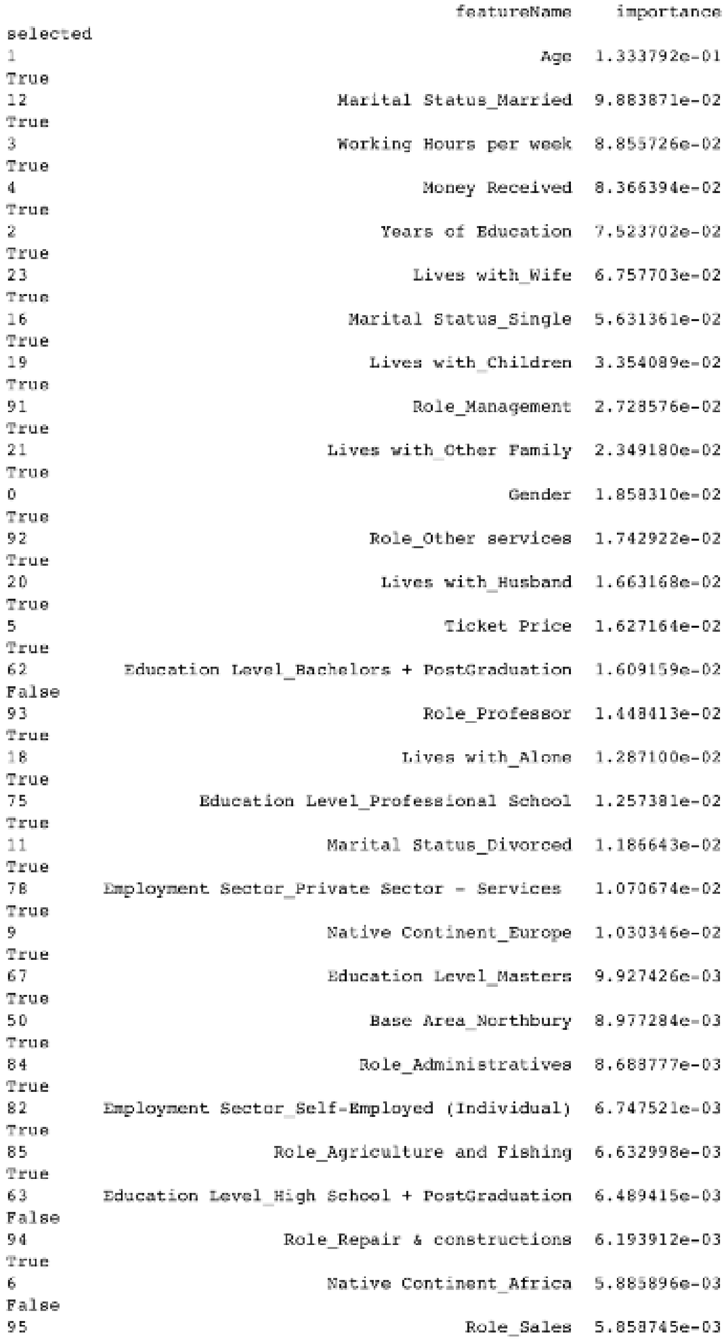
[11] Leandro L. Minku. A novel online supervised hyperparameter tuning procedure applied to cross-company software effort estimation. In Empirical Software Engineering, volume 24, 2019.

[12] M. Babaeizadeh, P. Smaragdis and R. H. Campbell. A simple yet effective method to prune dense layers of neural networks, 2017.

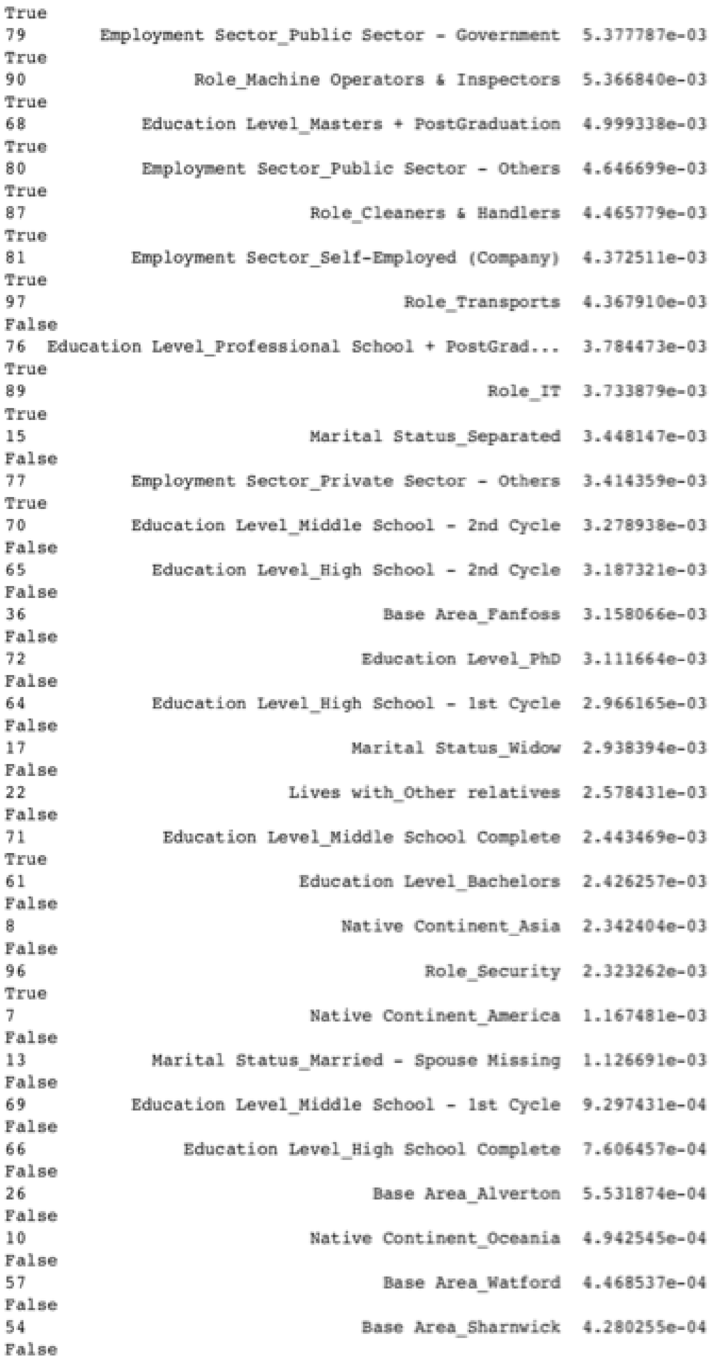
[13] T. Hastie, R. Tibshirani. Cross-validation and Boostrap. 25 February 2009. Statistical Leaning and Data Mining - Stanford University, Available from: http://statweb.stanford.edu/~tibs/sta306bfiles/cvwrong.pdf

VIII. Appendix

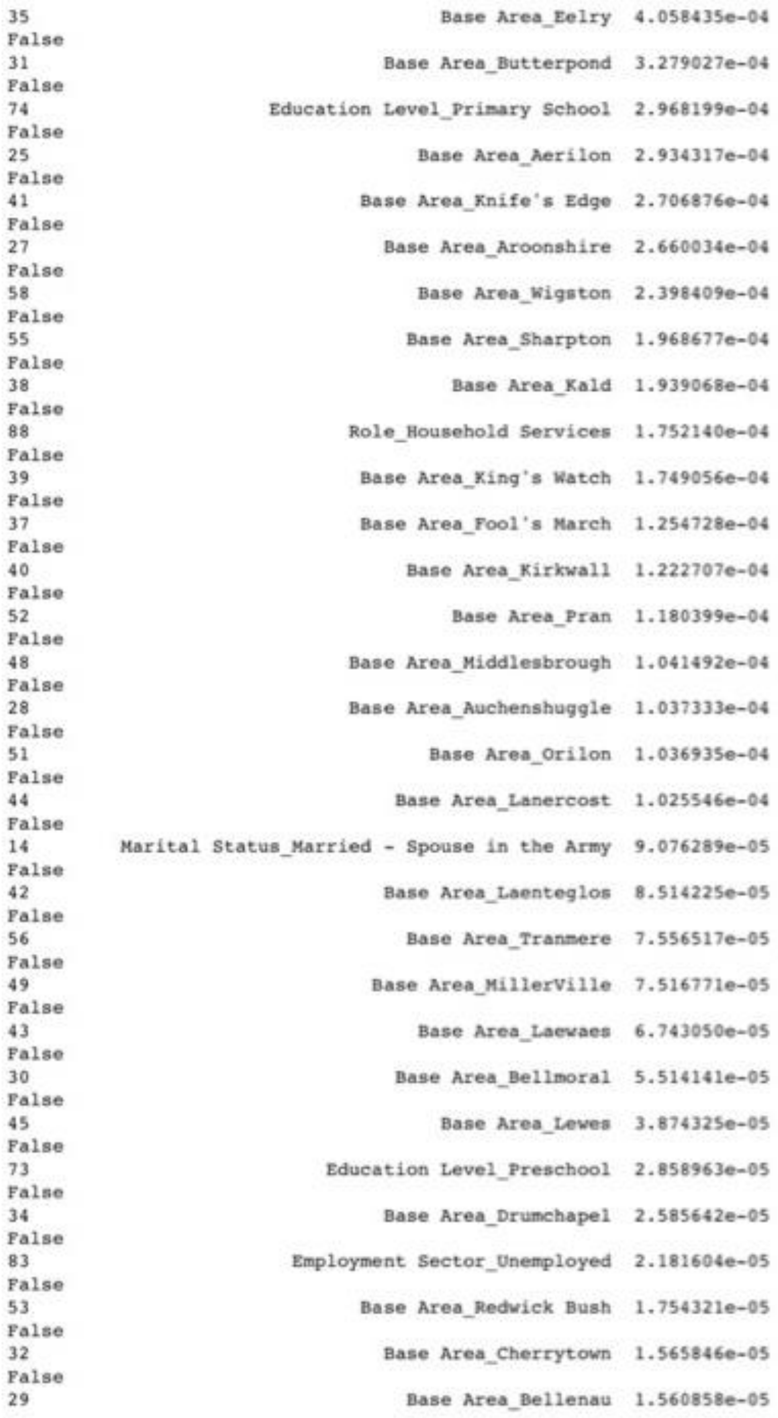
Appendix A:



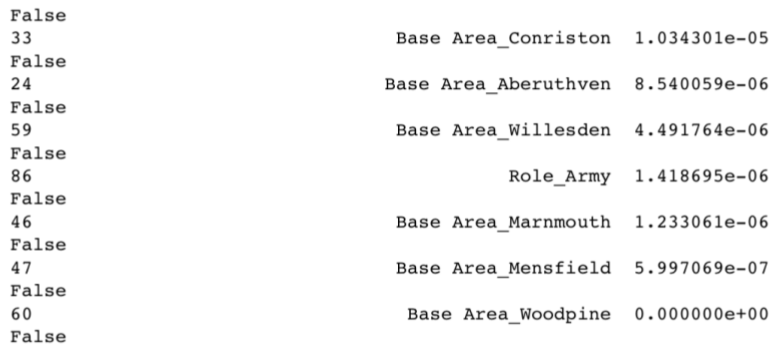
```
                                          featureName    importance
selected
1                                                 Age  1.333792e-01
True
12                             Marital Status_Married  9.883871e-02
True
3                              Working Hours per week  8.855726e-02
True
4                                      Money Received  8.366394e-02
True
2                                  Years of Education  7.523702e-02
True
23                                    Lives with_Wife  6.757703e-02
True
16                              Marital Status_Single  5.631361e-02
True
19                                Lives with_Children  3.354089e-02
True
91                                    Role_Management  2.728576e-02
True
21                            Lives with_Other Family  2.349180e-02
True
0                                              Gender  1.858310e-02
True
92                                Role_Other services  1.742922e-02
True
20                                 Lives with_Husband  1.663168e-02
True
5                                        Ticket Price  1.627164e-02
True
62         Education Level_Bachelors + PostGraduation  1.609159e-02
False
93                                     Role_Professor  1.448413e-02
True
18                                   Lives with_Alone  1.287100e-02
True
75                Education Level_Professional School  1.257381e-02
True
11                            Marital Status_Divorced  1.186643e-02
True
78       Employment Sector_Private Sector - Services   1.070674e-02
True
9                             Native Continent_Europe  1.030346e-02
True
67                            Education Level_Masters  9.927426e-03
True
50                                Base Area_Northbury  8.977284e-03
True
84                               Role_Administratives  8.688777e-03
True
82       Employment Sector_Self-Employed (Individual)  6.747521e-03
True
85                       Role_Agriculture and Fishing  6.632998e-03
True
63       Education Level_High School + PostGraduation  6.489415e-03
False
94                        Role_Repair & constructions  6.193912e-03
True
6                             Native Continent_Africa  5.885896e-03
False
95                                         Role_Sales  5.858745e-03
```



```
True
79       Employment Sector_Public Sector - Government  5.377787e-03
True
90                Role_Machine Operators & Inspectors  5.366840e-03
True
68           Education Level_Masters + PostGraduation  4.999338e-03
True
80           Employment Sector_Public Sector - Others  4.646699e-03
True
87                           Role_Cleaners & Handlers  4.465779e-03
True
81          Employment Sector_Self-Employed (Company)  4.372511e-03
True
97                                    Role_Transports  4.367910e-03
False
76  Education Level_Professional School + PostGrad...  3.784473e-03
True
89                                            Role_IT  3.733879e-03
True
15                           Marital Status_Separated  3.448147e-03
False
77          Employment Sector_Private Sector - Others  3.414359e-03
True
70          Education Level_Middle School - 2nd Cycle  3.278938e-03
False
65            Education Level_High School - 2nd Cycle  3.187321e-03
False
36                                  Base Area_Fanfoss  3.158066e-03
False
72                                Education Level_PhD  3.111664e-03
False
64            Education Level_High School - 1st Cycle  2.966165e-03
False
17                               Marital Status_Widow  2.938394e-03
False
22                         Lives with_Other relatives  2.578431e-03
False
71             Education Level_Middle School Complete  2.443469e-03
True
61                          Education Level_Bachelors  2.426257e-03
False
8                               Native Continent_Asia  2.342404e-03
False
96                                      Role_Security  2.323262e-03
True
7                            Native Continent_America  1.167481e-03
False
13            Marital Status_Married - Spouse Missing  1.126691e-03
False
69          Education Level_Middle School - 1st Cycle  9.297431e-04
False
66               Education Level_High School Complete  7.606457e-04
False
26                                 Base Area_Alverton  5.531874e-04
False
10                           Native Continent_Oceania  4.942545e-04
False
57                                  Base Area_Watford  4.468537e-04
False
54                                Base Area_Sharnwick  4.280255e-04
False
```



```
35                                Base Area_Eelry  4.058435e-04
False
31                           Base Area_Butterpond  3.279027e-04
False
74                Education Level_Primary School  2.968199e-04
False
25                              Base Area_Aerilon  2.934317e-04
False
41                         Base Area_Knife's Edge  2.706876e-04
False
27                           Base Area_Aroonshire  2.660034e-04
False
58                              Base Area_Wigston  2.398409e-04
False
55                             Base Area_Sharpton  1.968677e-04
False
38                                 Base Area_Kald  1.939068e-04
False
88                       Role_Household Services  1.752140e-04
False
39                         Base Area_King's Watch  1.749056e-04
False
37                         Base Area_Fool's March  1.254728e-04
False
40                             Base Area_Kirkwall  1.222707e-04
False
52                                 Base Area_Pran  1.180399e-04
False
48                        Base Area_Middlesbrough  1.041492e-04
False
28                        Base Area_Auchenshuggle  1.037333e-04
False
51                               Base Area_Orilon  1.036935e-04
False
44                            Base Area_Lanercost  1.025546e-04
False
14   Marital Status_Married - Spouse in the Army  9.076289e-05
False
42                           Base Area_Laenteglos  8.514225e-05
False
56                             Base Area_Tranmere  7.556517e-05
False
49                          Base Area_MillerVille  7.516771e-05
False
43                              Base Area_Laewaes  6.743050e-05
False
30                            Base Area_Bellmoral  5.514141e-05
False
45                                Base Area_Lewes  3.874325e-05
False
73                     Education Level_Preschool  2.858963e-05
False
34                           Base Area_Drumchapel  2.585642e-05
False
83                  Employment Sector_Unemployed  2.181604e-05
False
53                          Base Area_Redwick Bush  1.754321e-05
False
32                           Base Area_Cherrytown  1.565846e-05
False
29                             Base Area_Bellenau  1.560858e-05
```



```
False
33                            Base Area_Conriston  1.034301e-05
False
24                           Base Area_Aberuthven  8.540059e-06
False
59                            Base Area_Willesden  4.491764e-06
False
86                                     Role_Army  1.418695e-06
False
46                            Base Area_Marnmouth  1.233061e-06
False
47                            Base Area_Mensfield  5.997069e-07
False
60                             Base Area_Woodpine  0.000000e+00
False
```