LLSMF2014 - Data Analytics applied in Business - Final Report

Colin Leonetti, Margaux Mouyard, Manon Schots and Jades Souillard,
Group 9, UCLouvain

Professor—Marco Saerens
Assistants—Sylvain Courtain and Pierre Leleux

Abstract—This Final report aims to break down the classification models performed on a training dataset and analyse their accuracy in order to assign new individuals to their correct segmentation.

1 Introduction

This final report outlines step by step the construction of different classification models within the framework of the LLSMF2014 Machine Learning group project. The pursued objective of our work is to obtain an efficient model with a satisfying accuracy rate of prediction for assigning new observations to the correct category of the dependent variable, *Segmentation*. This classification model will be based on tests previously performed on a training set.

Firstly, the analyses that were performed on the entire database will be synthesized. This exploratory data analysis will allow for a better understanding and simplification of the dataset. Then, feature selection and feature extraction techniques will be performed. Next, the supervised classification models will be described before being applied to the newly constructed databases. The performances of these classification models will be compared, so that ultimately, the best model will be chosen to assign correct segmentation to a new sample of individuals whose exact *Segmentation* category is unknown. The effective final accuracy will be revealed by the academic team in charge of this project in due time.

The term *function* will be used several times in this report. It refers to functions of the programming language "R".

2 Exploratory data analysis

The objective of the exploratory data analysis is to get a first global insight of the dataset and to summarize its main characteristics such as the relationship between the variables. In addition, this analysis also enables the database to be cleaned and prepared for modelling.

2.1 Univariate descriptive statistics

2.1.1 Non-relevant variables removal

The initial database consisted of 14 features and 7131 observations. To be able to work with this data properly, the variables that were not relevant have been removed.

Therefore, the attribute *License Plate* was immediately removed from the database as its information was clearly useless for any classification of individuals, each having a unique license plate.

Then, looking further into the attributes, it was quickly noticed that the information contained in the *Child* attribute was essentially part of the information already stored in the *Family Size* attribute. The *Child* variable was therefore chosen to be deleted as the feature *Family Size* conveys the same information.

2.1.2 Imputation of missing values

By analysing the database, it was noticed that there was a non-negligible number of missing values in the observations. The *vis_miss* function showed that a total of 1.8% of the required values were missing.

In order to process these missing values, the *missMDA* package was used. This package uses a model that is based on Principal Component Analysis (PCA) for the continuous variables and on Multiple Correspondence Analysis (MCA) for the categorical variables in order to predict appropriate values for the missing elements. It thus takes into account the similarities between the variables and between the observations to predict new values as accurately as possible. This imputation method enabled us to start the bivariate statistics with a fully defined dataset and without losing any existing observations.

2.2 Bivariate statistics

2.2.1 Correlation between features

In order to examine the correlation between the variables, the matrix of correlations between the quantitative variables was first computed with the *cor* function, as well as its associated plot (Fig. 1). What could be noticed first was that the attributes *Car* and *Age* convey similar information, their correlation being at a level of 97%.

The correlation circles of the PCA lead to the same conclusion. Moreover, they showed that the variables *Work Experience* and *Family Size* were not correlated at all (Fig. 2).

The Cramer's V metric based on Pearson's chi-squared statistic allowed to measure the relationship between categorical variables. This indicator was computed for each pair of categorical features and it appeared that none of them were strongly correlated.

Fig. 1: Correlation between numerical variables

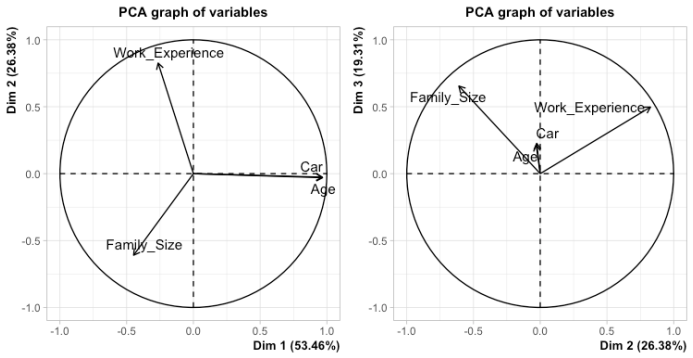

Fig. 2: Correlation circles of PCA

2.2.2 Independence from the target variable

In order to observe the relationship between the explanatory variables and the dependent variable, visual representations were analysed by computing plots and contingency tables.

These plots can already give indications on the classification trends. Indeed, Fig. 3 revealed that the majority of the observations that were falling under the *Category D* of the *Segmentation* variable included young people of around 30 years old on average.

On top of these visual representations, hypothesis tests were performed. The ANOVA test was used for the numerical variables and the Pearson's Chi-squared test was used for the categorical variables.

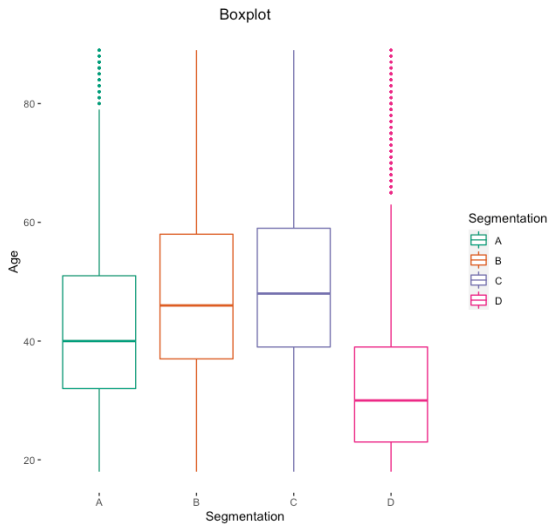

Fig. 3: Boxplots of variable *Age* for each category of *Segmentation*

Variable	Test type	p-value
Age	ANOVA	$< 2.2 \times 10^{-16}$
Car	ANOVA	$< 2.2 \times 10^{-16}$
Work_Experience	ANOVA	$< 2.2 \times 10^{-16}$
Family_Size	ANOVA	$< 2.2 \times 10^{-16}$
Gender	Chi-squared	0.1059
Ever_Married	Chi-squared	$< 2.2 \times 10^{-16}$
Graduated	Chi-squared	$< 2.2 \times 10^{-16}$
Profession	Chi-squared	$< 2.2 \times 10^{-16}$
Spending_Score	Chi-squared	$< 2.2 \times 10^{-16}$
Credit_Owner	Chi-squared	$< 2.2 \times 10^{-16}$
Var_1	Chi-squared	$< 2.2 \times 10^{-16}$

As shown on the table above, for all but one of the tests, the p-value allowed for a rejection of the null hypothesis, meaning that, on average, the value of the feature is significantly different from one *Segmentation* category to another, and that the explanatory variables and the target variable are not independent. The feature *Gender* was the exception. With a p-value exceeding 5%, the null hypothesis could not be rejected, meaning that *Gender* is independent of *Segmentation*. However, as the p-value is not that high above the rejection level, it is not clear whether *Gender* is useless or not for our predictions. Tests will therefore be done with and without this variable in the feature selection section in order to draw a conclusion.

3 Feature selection and feature extraction

In the continuity of the previous analyses, feature selection and feature extraction techniques were performed. Their objective was to simplify the database by reducing its dimensionality, while keeping as much relevant information as possible to potentially improve the accuracy rate of prediction of the different classification models.

3.1 Feature Selection

For the feature selection process, two variants of the original database were created. These databases were:

1) A database without *Car* – because of its correlation with *Age* (see section 2.2.1).
2) A database without *Car* and without *Gender* – to test if the independence highlighted by the Pearson's chi-squared test is significant (see section 2.2.2).

As a reminder, the *Licence-Plate* and *Child* variables had already been removed. The feature selection process thus outputs 2 databases on top of the initial one, made of 11 and 10 features respectively.

3.2 Feature Extraction

For the feature extraction, PCA and MCA were performed with the objective to construct three additional types of databases:

1) A database with a few principal components and all the categorical variables.
2) A database with some dimensions of MCA and all the numerical variables.
3) A database with a few principal components and some dimensions of MCA.

The challenge was to select the relevant number of dimensions to keep for both PCA and MCA. This selection is detailed in the following subsections.

3.2.1 PCA

The percentage of variance of the initial variables represented in each principal component (cumulatively equal to 1) is shown in Fig. 4 and in the table below.

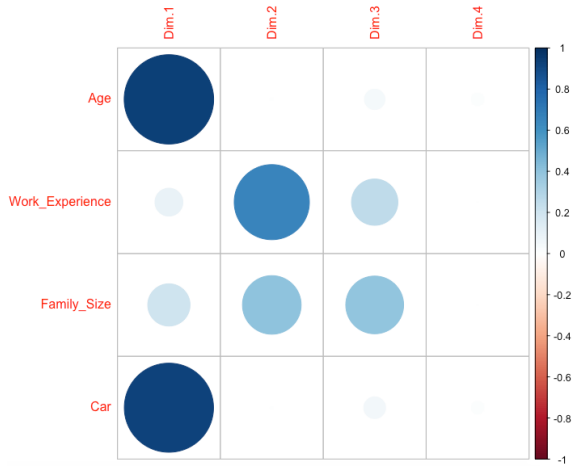

Fig. 4: Contribution of each variable to each dimension of the PCA

	Variance	Cumulative
Comp. 1	0.5403	0.5403
Comp. 2	0.2655	0.8057
Comp. 3	0.1857	0.9914
Comp. 4	0.0086	1

Considering these values, a dilemma occured between keeping the first 2 or 3 dimensions. Keeping 2 dimensions would better reduce the dimensionality while keeping 80% of variance representability. Keeping 3 dimensions would enable for an even smaller loss of information but would not conduct to a large simplification of the data. Consequently, both options were considered by building two new databases.

3.2.2 MCA

In order to choose how many dimensions would be selected amongst the 20 dimensions (each categorical variable being one-hot encoded), three different criteria were considered.

The first criterion was the *"elbow criterion"* which is a visual method based on the histogram of eigenvalues with the percentage of inertia on the ordinate axis (Inzenman, 2008). This method indicates to select only the dimensions represented before the decrease in percentage of inertia becomes steady. According to this criterion, 6 dimensions were selected which represent together 44.81% of the total inertia (see Fig. 5).

The second one was the *"Kaiser criterion"* which states that only the axes for which the inertia is greater than the average inertia should be kept (Inzenman, 2008). The mean inertia being 0.143, only the first 9 dimensions were retained which together account for 60.25% of the total inertia.

The last criterion suggested keeping an 80% rate of inertia to avoid too much information loss (Inzenman, 2008). This led to keeping the first 13 dimensions.

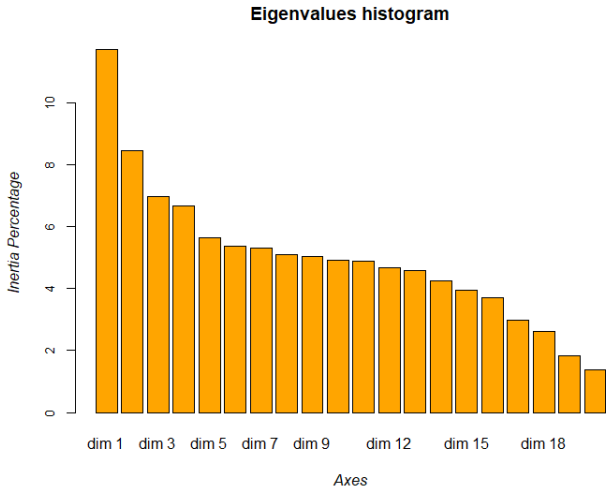

Fig. 5: Dimensions obtained by the MCA

3.2.3 Dimensions testing

The five databases created through feature extraction (i.e. two based on PCA and three on MCA) were tested on each model described in the next section. Relatively good results regarding the accuracy were obtained when using the 2-dimensions database built through PCA. However, the three databases constructed through MCA produced poor results. Besides, when merging the reduced datasets built through PCA and MCA, results were even poorer.

As a consequence, the final choice for feature extraction was to keep only the 2-dimensions PCA dataset of which results will be presented in section 5 and which would be confronted to the performance of the initial database and the two datasets created through feature selection.

3.3 Conclusion on Feature Selection and Extraction

As a result of the previously performed analyses, the classification models will be performed on 4 distinct datasets:

1) **Train**, made of the 12 initial variables
2) **Train2**, the database "Train" without the variable *Car*, made of 11 variables
3) **Train3**, the database "Train" without variables *Car* and *Gender*, made of 10 variables
4) **TrainPCA**, the first 2 dimensions of the PCA added to the 8 categorical variables.

4 Supervised classification models

The following section will detail the theory behind each constructed classification model.

4.1 Multinomial Logistic Regression

The multinomial logistic regression is a classification model that aims to predict a categorical variable with at least three discrete outcomes. This prediction is obtained using a set of quantitative and/or qualitative explanatory variables that are likely to influence the target categorical variable.

The particularity of this regression is that it directly estimates the a posteriori probabilities, i.e. the conditional probabilities of belonging to a class given features which are the explanatory variables. In addition, the data is separated by hyperplanes.

The mathematical form of this regression is exponential in order to make sure to have only positive values and that the a posteriori probabilities respect the axioms of Kolmogorov. This expression is the following :

$$P(\omega_k|\mathbf{x}) \approx \hat{y}_k(\mathbf{x}) = \frac{e^{\mathbf{w}_k^T \mathbf{x}'}}{\sum_{j=1}^{q} e^{\mathbf{w}_j^T \mathbf{x}'}} \quad (1)$$

$$where : 0 \leq \hat{y}_k(\mathbf{x}) \leq 1 \; and \sum_{k=1}^{q} \hat{y}_k(\mathbf{x}) = 1 \quad (2)$$

where $\mathbf{x}' = [1, x_1, x_2, ..., x_p]^T$ is the feature vector augmented by 1 to add a bias term, $\hat{y}_k(\mathbf{x})$ the membership values distributed according to a multinomial logistic distribution, q the number of classes, and $w_k = [w_{0k}, w_{1k}, w_{2k}, ..., w_{pk}]^T$ the vector of parameters for each class k.

For each class, this last vector is approximated by maximum log-likelihood and its parameters are chosen to maximize the likelihood of the data set under the assumption that the observations are independent. The log-likelihood of the data is expressed as follows:

$$logL = \sum_{i=1}^{n} \sum_{k=1}^{q} y_k(\mathbf{x}_i) ln(\hat{y}_k(\mathbf{x}_i)) \quad (3)$$

To find the global maximum, the gradient ascent based algorithm, which is an iterative algorithm, can be used (Saerens & Decaestecker, 2021).

An extension of standard logistic regression is the *stepwise regression*, which is an automatic procedure for making a step-by-step model selection by selecting the most relevant features to predict the class of membership. The stepwise method combines the forward and the backward methods and therefore, at each step, an explanatory variable will be added or subtracted from the set of explanatory variables on the basis of statistical criteria such as statistical significance (Kuhn, 2013).

4.2 Naive Bayes

The Naive Bayes algorithm is a basic classifier builder, based on the probabilities for an observation to be part of a specific category given the probabilities for each of its modalities to be part of that specific category. The Naive Bayes algorithm computes the probabilities of being in the *k* class and assigns each observation to the class with the highest probability. This is known as

the *Maximum A Posteriori* (MAP) decision rule (Rish, 2001). Thus The Naive Bayes classifier assigns each feature x to class k through

$$\arg\max_k \; p(K = k) \times p(X = x|K = k) \qquad (4)$$

The weakness of a Naive Bayes classification is the assumption made that every observed variable is independent and identically distributed (iid) (Hand, 2001). Given the previous observation of highly correlated variables in this context, it could be assumed that the Naive Bayes algorithm would work best once the correlated variables are removed from the model (Saerens & Decaestecker, 2021).

Other improvements to the Naive Bayes classifier can be performed by tuning its hyperparameters. One of these is the Kernel density estimation replacing the assumption of Gaussian distribution. Given the fact that, in this context, every variable already presents a positive number of observations in each class, another classic hyperparameter called the Laplace smoothing wouldn't be useful here Saerens & Decaestecker, 2021).

4.3 Decision Trees & Random Forest

This section details the theoretical ideas behind the random forest. Given the fact that Random Forest is an improvement of the decision tree algorithm, here are the basics on which this algorithm is built.

The decision tree algorithm is used to construct a tree of rules based on a training set. This algorithm discriminates between the different classes in order to classify the data in the most accurate possible way (Saerens & Decaestecker, 2021). A significant advantage of decision trees is that they are compliant with continuous, discrete and categorical variables and that they perform automatic feature selection.

To construct the tree, a greedy algorithm is used for splitting the observations, based on a feature test. The homogeneity of the observations' distribution within classes will determine which type of split should be used. Quite intuitively, nodes with a high level of homogeneity of class distribution are preferred as they lead to easier classification and thereby a higher level of accuracy in the predictions. In this context, the algorithm used the GINI index to decide which variable to branch on.

The GINI measure is based on a rough estimate of the proportion of observations of each class falling in the leaf (end node) t ($P(w_j|t)$). Its equation can be written as follows:

$$GINI(t) = 1 - \sum_j [P(w_j|t)]^2 \qquad (5)$$

The Random Forest model is a machine learning algorithm that consists of a large number of individual decision trees operating as an ensemble. This process is quite similar to a bagging process applied to decision trees (Yiu, 2019). The key improvement of a Random forest compared to the decision tree is that, at each split of the tree, rather than considering all features for the split, a subset of these features is sampled and only these few variables are considered as candidates to branch onto. Adding this randomness leads to a collection of trees that are further non-correlated from one another. The outputs are several trees and different predictions for each observation. The final estimation for a categorical classification is obtained by the most frequent predicted class (Yiu, 2019).

4.4 Support Vector Machines

The Support Vector Machines (SVM) is a model meant to separate data samples into classes with the help of hyperplanes. In order to do so, it is based on two main ideas : maximizing the margin and using the Kernel Trick (Saerens & Decaestecker, 2021). The margin is the distance between a threshold defined by the algorithm and the closest observation of a class. By maximizing this distance, the SVM is able to put a hyperplane of coefficients $[w, w_0]^T$ as far as possible of each "extreme" observation (Kuhn, 2013). That is, for a vector of observation x

$$x = x_p + d(x)\frac{w}{\|w\|} \qquad (6)$$

where x_p is the orthogonal projection of x on the hyperplane, the distance $d(x)$ is maximized from

$$d(x) = \frac{y(w^T x + \omega_0)}{\|w\|} \qquad (7)$$

The weakness of such a method appears when some classes contain outliers, that is, observations far from all the others belonging to the same class. This problem is solved by allowing misclassification in the SVM by tuning the hyperparameter C.

The Kernel trick allows the Support Vector Machine to be applied in non-linear classifying cases. Its idea is to transform the scalar of the training vectors $x_i^T x_j$ through different kernel functions. Two of those equations are the polynomial Kernel of degree p (Hardle & Simar, 2003):

$$K(x_i, x_j) = (x_i^T x_j + 1)^p \tag{8}$$

and the Radial Basis Function Kernel (Vert, 2004):

$$K(x_i, x_j) = exp(-\frac{\|x_i - x_j\|}{2\alpha^2}) \tag{9}$$

where α is a free parameter.

The utility of this trick is to compute only the coordinates of each transformed observation, to maximize afterwards the margin and get the best hyperplane separations.

5 Results and Discussion

In this section, the results of the 4 supervised classification models previously described are presented and discussed. The k-fold cross-validation method was used to compute the final prediction accuracy rate of each model as it is more robust to assess accuracy than a simple holdout and prevents overfitting. This method consists of dividing the train set into 10 folds with 1 fold at a time becoming the validation set. The final accuracy was computed as the average of the accuracy of each fold (Saerens & Decaestecker, 2021).

5.1 Logistic Regression (normal and stepwise)

The results of the normal, i.e. with all variables, multinomial logistic regression and the stepwise logistic regression are displayed in the table below. The Train2 model with all variables except *Car* performs best in both cases but it can also be noticed that models derived from feature selection and feature extraction gave better accuracy than with the initial Train database for the normal logistic regression.

	Accuracy (%)			
	Train 12var	**Train2 11var**	**Train3 10var**	**TrainPCA 10var**
Normal	47.231	**47.497**	47.245	47.259
Stepwise	47.413	**47.497**	47.161	47.259

5.2 Naive Bayes

In the Naive Bayes classification model, a hyperparameter was tuned in order to replace Gaussian distribution by the Kernel density estimation which gave better accuracy in any case.

Accuracy (%)			
Train 12var	**Train2 11var**	**Train3 10var**	**TrainPCA 10var**
46.389	46.810	**46.852**	46.460

As the Naive Bayes classifier is based on the assumption that all features are independent, a better accuracy was reached with the models from which the highly correlated variables had been removed, i.e. Train2 and Train3.

5.3 Random Forest

As mentioned in the previous section, the Random Forest algorithm computes a large number of trees, all independent and different from one another. By default, the *randomForest* function creates 500 decision trees. Moreover, the default number of features available for branching at each split is equal to the square root of the number of features, in our case $\sqrt{11}, \sqrt{10}, \sqrt{9}$ and $\sqrt{9}$ for the four databases.

When using the default features, the *randomForest* algorithm performs with a maximum accuracy of 0.47455 on the Train2 dataset. In order to try to obtain better results, the two parameters mentioned above can be tuned. Quite intuitively, increasing the number of decision trees (*ntree*) built would increase the chances of a better final accuracy rate. However, this only

occurs until a certain point above which it can no longer be increased. In our case, using more than the default value of 500 trees did not improve the performance of the model, meaning that maximum performance had already been reached in this regard.

Concerning the number of available variables to choose at each split (*mtyr*), the default value approximates 3 for all 4 dataset. The value was increased to observe changes in the accuracy rate of prediction but without success. On the other hand, decreasing this value may lead to overfitting of the model and should therefore be avoided (Hackerearth, 2020).

As Random forest is an improvement of the decision tree algorithm, only the results of Random forest models are presented in the table below. Again, it can be noticed that feature selection and feature extraction improved the accuracy systematically.

Accuracy (%)			
Train 12var	**Train2 11var**	**Train3 10var**	**TrainPCA 10var**
46.838	**47.455**	47.146	46.936

5.4 Support Vector Machines

To perform the SVM classification model, multiple Kernel functions were tested. The only two with acceptable accuracy were the Radial basis function Kernel (RBF) and Polynomial Kernel degree 2 explained in section 4.4.

Then, for each of the two functions, the hyperparameter C was tuned in order to obtain better results. The C parameter adds a penalty for each misclassified data point. For a large value of C, the model will choose a smaller-margin hyperplane if it is better in getting all the training points classified correctly because of the high penalty while, a small value of C will lead the model to choose a larger-margin separating hyperplane, even if that hyperplane misclassfies more observations (but leaving more room for the value of new observations of the test set).

The gamma hyperparameter (for non linear hyperplanes only) controls the distance of influence of a single training point. High values indicate that the points need to be very close to each other in order to be considered as part of the same class, therefore the higher the gamma value, the more value it tries out to fit the training dataset exactly (Hardle & Simar, 2003).

As too high value of C and gamma could lead to overfitting, the value of these hyperparameters should be chosen carefully. After trying multiple combinations, the conclusion was that the best results were obtained with C=10 for RBF and C=100 with the polynomial kernel degree 2, and with a gamma value of $\frac{1}{data\ dimension}$ (default value in *svm* fonction). Results are shown in the table below.

	Accuracy (%)			
Kernel function	**Train 12var**	**Train2 11var**	**Train3 10var**	**TrainPCA 10var**
Radial $C = 10^1$	47.610	47.708	**48.114**	47.660
Polyn. $C = 10^2$	47.708	47.862	47.960	47.960

As it can be seen, the SVM classification method gave quite good accuracy compared to other models. Moreover, this was the first time that accuracy higher than 0.48 has been obtained.

5.5 Discussion

What emerged from each of the models tested and was further emphasized in the analysis of recall, accuracy and F1 (see Section 6), was that the models have more difficulty in classifying class B observations. Therefore, an alternative approach was tested.

The method consisted in first transforming the variable to be predicted into a one-vs-all binary variable, i.e. level "B" vs. level "Other", and training a first model to later predict membership of class B or Other. Then a second model was trained to predict class A, C or D membership. This means that two different classification models could be used.

The transformation of the target variable *Segmentation* into a one-vs-all variable resulted in an imbalanced dataset as the observations of classes A, C and D were pooled. Consequently, to deal with the imbalanced data, oversam-

pling was performed i.e. randomly replicating instances in the minority class (B).

Several combinations of classification models were tried but none of them gave an outstanding accuracy. Therefore, this alternative method is not retained for the final evaluation of predictions of the "TestStudent" dataset.

6 Comparison & Final Model Selection

By analysing and comparing the performance based on the accuracy rate, the best result for each model was selected:

1) **Support Vector Machines RBF (C=10 & γ =0.1)**, with Train3: 48.114%
2) **Logistic Regression**, with Train2: 47.497%
3) **Random Forest**, with Train2: 47.455%

The Naive Bayes classification model was not considered in the final ranking as it performed very poorly on any of the databases.

It can be already pointed that the Feature Selection was indeed useful to lead models to better classifications as datasets where features were left out (Train2 and Train3) were here chosen.

According to the accuracy rate, the SVM model associated with the Train3 database seemed to be the most efficient for classification. However, before choosing the final model that will be used to predict the segmentation of the new sample of individuals, other performance criteria were considered to compare these three models.

Indeed, while the accuracy rate indicates the global proportion of correctly classified individuals, the precision and the recall can also be calculated from the confusion matrix. Precision is the percentage of correct predictions for a certain class, while recall is the percentage of instances of a class that have been correctly predicted. These ratios are therefore per-class metrics and are expressed as follows:

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \tag{10}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \tag{11}$$

where TP_k (True Positive) is the number of individuals correctly predicted to be in the class k, FP_k (False Positive) the number of individuals predicted to be in the class k while they are not, and FN_k (False Negative) the number of individuals predicted to be out of the class k while they are (Juba & Le, 2019).

In order to have a measure that balances precision and recall, the F1 score was also computed per class using the following formula:

$$F1_k = 2 \times \frac{Precision_k \times Recall_k}{Precision_k + Recall_k} \tag{12}$$

These three per-class metrics were calculated for each of our top three models:

SVM Train3			
	Precision	**Recall**	**F1**
A	0.4014	**0.5129**	0.4504
B	0.3950	**0.2691**	0.3201
C	0.5081	0.4960	0.5020
D	0.6016	0.6076	0.6046
Accuracy : 48.114 %			

Logistic Regression Train2			
	Precision	**Recall**	**F1**
A	0.3945	**0.4871**	0.4359
B	0.3902	**0.1649**	0.2319
C	0.4624	0.5602	0.5066
D	0.6018	0.6405	0.6206
Accuracy : 47.497 %			

Random Forest Train2			
	Precision	**Recall**	**F1**
A	0.4041	**0.4386**	0.4206
B	0.3675	**0.2898**	0.3241
C	0.5049	0.5082	0.5065
D	0.5805	0.6281	0.6034
Accuracy : 47.455 %			

After the analysis of our different criteria to assess the performance of our models, the logistic regression model with Train2 was directly rejected. Indeed, in addition to not having the highest accuracy rate, the recall rate for class B is very low (16.5%), which reduces the F1 score. Consequently, our attention was focused on the two other models for which the F1, recall and precision scores were better balanced between the classes.

Then, comparing these scores between the Random Forest model with Train2 and the SVM model with Train3, the only significant difference concerned the recall percentage of class A which is much higher in the SVM model (51.29%) than in the Random Forest model (43.86%).

Finally, since the SVM model had the highest accuracy rate and the highest and most balanced F1, recall and precision scores between classes, this model is selected as the final model to classify the new sample of individuals of the TestStudent dataset.

7 Limits & Conclusion

As a disclosure for this report, the observations retrieved from these supervised classifications models revealed recurrent difficulties to correctly classify individuals into the associated segmentation. Indeed, none of the classifying models predicted the accurate classes beyond a symbolic 50% milestone.

It was pointed out however that some models were more adapted to categorize the observations. In particular, Support Vector Machines and Random Forest models, once applied to a more adapted dataset, produced satisfying accuracy rates in addition to balanced precision and recall results.

Nevertheless, a few limits could be raised concerning this report, as it is the work of students which only followed an introduction course on the broad field of Machine Learning. In particular, the main limits encountered were related to the tuning of hyperparameters for which dealing with such a large quantity of possibilities wasn't easy to tackle for non-experts. Choices to make were often produced by trials and errors, which often resulted in disappointments when a new accuracy rate ultimately dropped once its parameters were modified or in fear of overfitting the models.

Coming to the last word, at the light of the best accuracy, recall and F1 score obtained, the final preference goes to the Support Vector Machine classifying model applied to a dataset reduced of its features *Child, Car, Gender* and *License Plate*. This algorithm will thus be applied to the new "TestStudent" dataset, with a prior transformation of the data through the same computations directed in the exploratory analysis: removal of the same four variables mentioned above, setting *Credit Owner* as a factor and ordering *Spending Score*.

The suggested classification will be submitted to the academic team of the LLSMF2014 course in order to confront its accuracy to other concurrent classifications. The quality of the accuracy estimated in this paper will be revealed in due time.

References

[1] Hackerearth. (2020). *Practical tutorial on Random Forest and parameter tuning in R.* https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/tutorial-random-forest-parameter-tuning-r/tutorial/

[2] Hand, D. J., & Yu, K. (2001). *Idiot's Bayes—not so stupid after all?* International statistical review, 69(3), 385-398.

[3] Hardle & Simar (2003), *"Applied multivariate statistical analysis"*. Springer-Verlag

[4] Izenman, A. J. (2008). *Modern multivariate statistical techniques. Regression, classification and manifold learning*, 10, 978-0.

[5] Juba, B., & Le, H. S. (2019, July). Precision-recall versus accuracy and the role of large data sets. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 4039-4048). https://doi.org/10.1609/aaai.v33i01.33014039

[6] Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling (Vol. 26).* New York: Springer.

[7] Rish, I. (2001, August). *An empirical study of the naive Bayes classifier.* In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

[8] Saerens & Decaestecker. (2021). *Lecture 01: General Introduction: Supervised Classification.* [PowerPoint Slides]. Louvain School of Management, UCLouvain, Louvain-la-Neuve.

[9] Saerens & Decaestecker. (2021). *Lecture 02: Supervised Classification Models.* [PowerPoint Slides]. Louvain School of Management, UCLouvain, Louvain-la-Neuve.

[10] Saerens & Decaestecker. (2021). *Lecture 03: Supervised classification: models and assessment.* [PowerPoint Slides]. Louvain School of Management, UCLouvain, Louvain-la-Neuve.

[11] Saerens & Decaestecker. (2021). *Lecture 05: Artificial Neural Networks and Support Vector Machines* [PowerPoint Slides]. Louvain School of Management, UCLouvain, Louvain-la-Neuve.

[12] Yiu, T. (nd). *Understanding random forest.* https://towardsdatascience.com/understanding-random-forest-58381e0602d2, 2019.

[13] Vert, J. P., Tsuda, K., & Schölkopf, B. (2004). *A primer on kernel methods. Kernel methods in computational biology, 47,* 35-70.