STAGE 3 PROJECT MACHINE LEARNING – MATHIEU COWAN, ADRIEN DE MAILLY NESLE, ALARIC DE BASTARD

Previous methods and limitations

In the precedent stages, we explored various approaches to be able to make accurate predictions of the revenue in our online retail dataset, for example by using the product descriptions with many machine learning and ensemble learning models. We noticed that the overfitting was still present because our dataset couldn't allow us to make the best prediction possible. We had improvements such as the object of the embeddings for the description of the products and the clustering and categorization of the behavior of the clients. All of this laid a path for further exploration and amelioration in this stage.

Improvements assumptions

For this third stage, we are convinced that we will find more accurate predictions of the Revenue based on the embeddings of the products. As well we believe that we will be able to classify correctly the revenue with appropriate classes with a thorough analysis of the data.

Before we start our predictions, we did the same pre-processing we did during the previous steps. Also, we filtered our data on the Germany Country, one of the countries with the most transactions during the 2010 year. We believe that Germany represents the patterns of the clients in general very well and that it is judicious to use the Germany country for such an assignment.

Part 1 : Prediction of the Revenue with Product Descriptions for the 2010 Germany Transactions

Problem formalization

The goal of the first prediction is to make an accurate prediction of the revenue from transactions in Germany during the whole calendar year of 2010. Here, we aim to have better performance, especially by extending the hyperparameter combinations we use and to use a new model which is **MLPRegressor**.

This model is a neural network-based model which is supposed to capture complex relationship between features and the target variable. We believe that this model will approach us to the goal of this project, which is to make the most accurate possible prediction of the revenue, here based on the product description so that we understand the patterns of the clients which serve better the industry that represents our dataset. Here we will use a parameters grid which will supposedly help to identify the best hyperparameters grid for such a prediction. The hyperparameters chosen were aimed at balancing the complexity of the model (number of layers and neurons) with the need to **prevent overfitting** (regularization through alpha and learning rate adjustments).

Method

We used a 80-20 split of the data for the model validation. The embedding of the products were done by the all-MiniLM-L12-v2 version of SentenceTransformer to be able to encode in the best way possible the descriptions.

When it comes to the choosing of the hyperparameters, we had the following logic :

- **Hidden_layer_sizes :** this hyperparameter defines the architecture of the neural network and indicates the number of neurons in each layer. We did a search over three configurations to be able to detect which depth and width were the best.
- **Activation :** it determines the activation function used for the neurons, we used 'relu' and 'tanh' for their ability to model relationships which aren't necessarily linear.
- **Solver :** this is for the optimization algorithm. We compared here the 'adam' and 'sgd' solver even if 'adam' is the most used
- **Alpha :** this is a parameter that helps to reduce overfitting. We tested multiple values such as 0.0001, 0.001, 0.01.
- **Batch_size :** it controls the learning rate adjustement strateguy during testing
- **Max_iter :** maximum number of iterations during training.

Metrics and results

When after a long running of the many possible configuration of the algorithms, we then ordered it by RMSE_test to find the most efficient algorithms. We then have the best results for **MLPRegressor** which are:

- **RMSE Train:** 28.386649
- **RMSE Test:** 36.465481
- **MAE Train:** 12.653684
- **MAE Test:** 15.215993

These results indicate that the model performs well in both the training and test sets, with some small overfitting observed due to the difference in RMSE between the two sets compared to the overfitting we observed during the previous stages.

The overfitting here is way more controlled than for previous stages and previous model used. We can also see that with the relatively small gap between RMSE and MAE. We believe that it is because this model can identify better a complex relationship between the embedding of the description of the product and the revenue of each of those transactions.

Second Prediction: Classification of the Revenue of Each Transaction in Germany in 2010 Using Clustering Prior to it

Problem Formalization

In this second prediction task, the goal is to classify the revenue of each transaction in Germany for the year 2010 into two categories: **low** and **high** revenue. Classifying the revenue helps later to perceive our capacity to understand the state of the global revenue in our industry and where we can make improvements, and also make assessment regarding revenue requirements we wish to have, and the improvements we need to make in the business.

Method

The first step in this task was to apply **clustering** to the dataset in order to segment the revenue values into different groups, which were then used as labels for classification. We compared the performance of multiple classification models, including **MLPClassifier**, **LogisticRegression**, **RandomForestClassifier**, and **SVC**.

First to understand the data better we showed the distribution of the logrevenue (application of the logarithmic function on the revenue) to understand it better. We added 1 to the revenue in the log to avoid log(0).

For clustering, we initially used a manual split based on the **quantile 0.5** of the log-transformed revenue, which allowed us to divide the dataset into two bins: **low** and **high** revenue.

We believe that it is better to use the quartiles to do a classification, we believe that mathematically it is the most judicious thing to do. At the beginning, we made tests of the models with four categories with the quantiles, but we saw that the results were better with 3 categories and even better with two, so we have here two categories which are split by q[0.5] (represented by the red line above].

- **KDE Plot**: We visualized the density of log-transformed revenue using Kernel Density Estimation (KDE) to assess the distribution of the revenue.
- **Quantile Split**: We then used this split to assign labels to the dataset: Low for low revenue and High for high revenue.

For the **clustering**, we initially applied **DBSCAN** and **Mean Shift** clustering models, with DBSCAN providing the best results for separating the data into two categories based on the outliers' handling.

Classification Models and Hyperparameter selection

For the classification task, the following models were selected, mainly because we wanted to compare the efficiency of the MLPClassifier, each with carefully chosen hyperparameters:

- **MLPClassifier**: We chose the MLP classifier due to its ability to capture complex patterns.
- **LogisticRegression**: This model was chosen as a simple baseline for classification, to make a good comparaison with all of the models, and also because it is suitable for linearly separable problems.
- **RandomForestClassifier**: Chosen for its robustness and ability to handle complex datasets.
- **SVC**: A powerful model for classification tasks, particularly when the data is not linearly separable.

The hyperparameters for each model were put in a hyperparameters grid :

- For **LogisticRegression**, we tested regularization parameters C, different solvers (liblinear, saga, and lbfgs), and class weight options.
- For **MLPClassifier**, we adjusted the number of hidden layers, activation functions, and solver types.
- For **RandomForestClassifier**, we experimented with the number of estimators, tree depth, and minimum samples for splitting and leaf nodes.
- For **SVC**, we explored different values of C, kernel types, and class weight adjustments.

Metrics and Results

The performance of each model was evaluated using several metrics:

- **Precision**: The proportion of true positive predictions among all positive predictions.
- **Recall**: The proportion of true positive predictions among all actual positive instances.
- **F1-Score**: The harmonic means of precision and recall, which is useful for imbalanced datasets.

For this first approached we obtained the following best results:

The MLPClassifier had the best results (about 0.82 or 0.88 for each train or test metric), same goes for Random Forest Classifier results, even if the MLP classifier had slightly better results. The SVC results were a little bit lower, and they were above 0.8 but the logistic regression model was the model with the worst results with less than 0.8 for each metric. Again, the results were selected by sorting these results by f1 score of tests, then by Precision test, then by recall test. This was our strategy to select the best choice of hyperparameters possible within the combinations we tested.

We have here a globally very satisfying classification with the manual clustering we did. The MLP Classifier model seemed to be the best model here for this case.

Second approach using models for clustering

We then did a second approach by using models such as DBSCAN and Mean Shift to determine better the adapted clusters to our dataset. Unfortunately, as seen on the dataset, the distribution was not equilibrated (9002-9 between the class 0 and 1, we took out the outliers) which led to not good enough clustering and obviously biased results for classification.

The results for such a classification were all between 0.99 and 1 for the best results for each metric whether it is for logistic regression of MLP Classifier. This confirms our assessment unfortunately. We tried to make the same work for other countries which a lot of rows in the dataset such as France, Spain or Irland (“EIRE” in the dataset, it is another name for this country). We also tried to make the same work with the UK part of our dataset (which represents most of the transactions) using different periods of the year for filtering, but unfortunately the observation on the distribution of the dataset leading to this issue repeated itself every time.

Conclusion and discussion

For this project stage, we had important improvements to our revenue prediction and on the task of classification (even if the results for the second part were different than expected). We had better control on the overfitting, we expanded the optimization of the hyper parameters, and we improved the performance of the models. The embeddings of the product description and MLP-based models helped us do that and helped us understand better the complex relationship between transactions and revenue. Also, we explored multiple possibilities to identify a good way of clustering and classifying the revenue in the low and the high categories. This can allow to do good strategic decision-making in the management of the products and in the pricing, which responds to the business scope of this project

However, we had a big challenge in this project which was to find a way to make robust predictions despite the imbalance in the revenue distribution when applied to specific parts of the dataset. We explored a lot of machine learning models including ensemble learning and Neural network-based models to try to achieve this task, and we had relatively good results when it comes to predictions concerning directly the revenue in this stage, which means understanding better the patterns of the clients. We had meaningful progress, but using more complex neural-network models could theoretically be a good idea to go even deeper in the analysis even if the results were globally satisfying in this stage and in the logical progression throughout the stages.

References which inspired us to the idea of using MLP-based models :

For MLPClassifier : https://link.springer.com/chapter/10.1007/978-3-540-79474-5_6

For MLPRegressor : https://www.sciencedirect.com/science/article/pii/S1877050923000868