Machine Learning - Winter 2023 - Mini Project I

Alex Montoya Franco
alex.montoyafranco@abo.fi
EDISS Master's Programme - Åbo Akademi University, Turku, Finland

1. Introduction

Marketing might be one of the most important things a business can do and with more and more competition every day, companies are in a constant effort not only to attract new customers, but to keep the ones already established. One could argue that a company's purpose in today's saturated markets is to stay relevant. But how to stay relevant when customers' behavior seems to change faster than businesses can adapt? - We claim that a successful business not only offers services that meet people's needs, but also runs the appropriate marketing campaigns so that customers are aware of these offerings. We believe a successful marketing campaign is targeted, well timed and relevant.

This research project aims at predicting whether a client would respond positive or negative to direct marketing campaigns from a banking institution using a Neural network model.

This type of project is specially relevant for the banking industry, where traditional banks and Fintechs are releasing financial services constantly and where marketing campaigns play a significant role on customers' loyalty and growth. For instance, JPMorgan Chase - the largest commercial bank in the United States by revenue - invested 3.04 billion U.S. dollars in marketing activities in 2021 [1].

2. Data Processing

The data is related with direct marketing campaigns of a banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

The dataset has 20 features related to client's data, bank communications with the client, and social and economic context attributes. 41.188 entries are considered.

Initial data exploration showed that no column had missing values, so handling was not needed there. However, further understanding of the data showed that multiple columns, specifically, those with a ['yes', 'no'] answer included 'unknown' as one of its values. We consider this needs further exploration and if consider as a missing value, then appropriate techniques should be used in this case.

We continued our exploration by analyzing the values of our target column ['y'], which represents whether the client subscribed a term deposit (binary: 'yes','no'). This revealed, as it can be seen in **Figure 1**, that our data is highly unbalanced.

Imbalanced data refers to those types of datasets where the target class has an uneven distribution of observations, i.e one class label has a very high number of observations and the other has a very low number of observations [2].

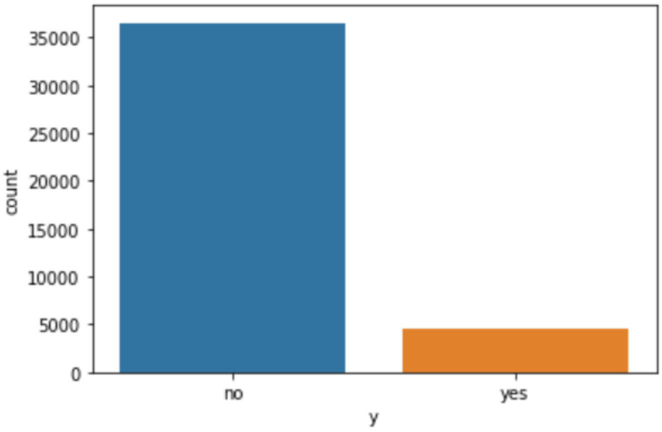

Figura 1: **Values distribution of target variable y**

This behavior is probably a good reflection of the business. Given that in most marketing campaigns, the number of clients that end up acquiring a service is small compared to the number of people targeted. One could argue that a "good"model will be able to predict most of the instances correctly and just fail in some specific cases. However, the problem that we face here, is that a model that is always predicting "no", will probably get a very good performance, given that this is the dominant class in our dataset.

An initial approach to address this issue is to show different evaluation metrics besides the accuracy score, for instance, precision and recall. Here is vital to identify the minority class correctly.

An additional possible step to address this issue would be to apply resampling (oversampling and undersampling). However, we considered this to be outside of the scope of this project. Future work could potentially include this to compare performance.

The data consists of numerical and categorical features. We will separate these to apply the appropriate methods depending on our data types. For instance, to apply One-hot encoding for our categorical features and to Normalize our numerical values.

[1] https://www.statista.com/statistics/497485/financial-services-ad-spend-usa/

[2] https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/

2.1. Numerical Features

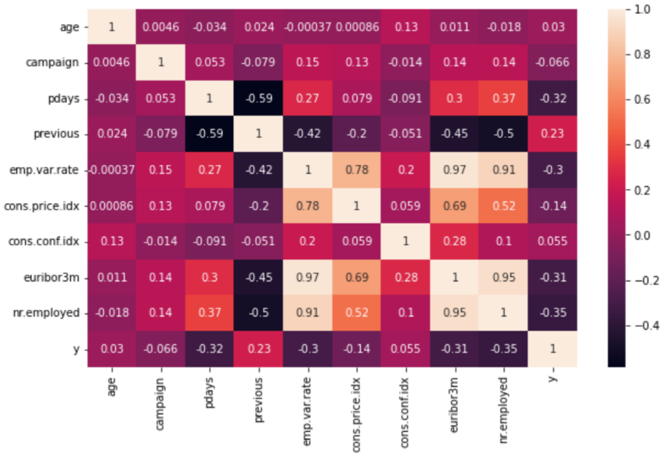

Figura 2: **Correlation matrix of numerical features**

The 'duration' feature was removed as indicated by the data description of the column. After this, we ran a confusion matrix, as shown in **Figure 2**.

Here we can see that there is no single variable that is highly correlated to our target variable (y). However, we do have some highly correlated features among our predictors. Let's get rid of some of those to avoid redundancy. We removed the column ['euribor3m']. This column had more than 95 % correlation to other two columns.

We continued by running a Variance Threshold on our data in order to remove columns with low variance i.e., columns with small variations in its values which would probably not help with our predictions. The variance function showed that 'y' and 'previous' had low variance, 'y' being our target variable while 'previous' being the number of contacts performed before this campaign for a specific client. We first considered removing this column. However, the fact that our target variable is highly unbalanced could actually have a relation to this. For instance, 'no' customers might have not be contacted before this campaign while the few 'yes' customers might have been.

We now proceed to normalize our numeric features using the StandardScaler from sklearn. Binning was consider for the Age column, but it was not included given the normalization strategy chosen.

2.2. Categorical Features

Initial exploration of our categorical features showed that most of them seem to have certain balance. For example, the type of job, although clearly skewed towards the admin, blue-collar and technician roles, it also has substantial entries for other roles.

One of the only columns that we considered unbalanced in here would be the 'loan' feature showing that very few clients posses one of these. This could actually be an interesting predictor since we could perhaps expect a person with a loan to also be willing to take a term deposit.

We proceed to apply One-hot encoding to all our categorical columns. Research into the topic [3] and types of encoding showed that categorical data without meaningful order and with cardinality less than 15 is suitable for this type of encoding.

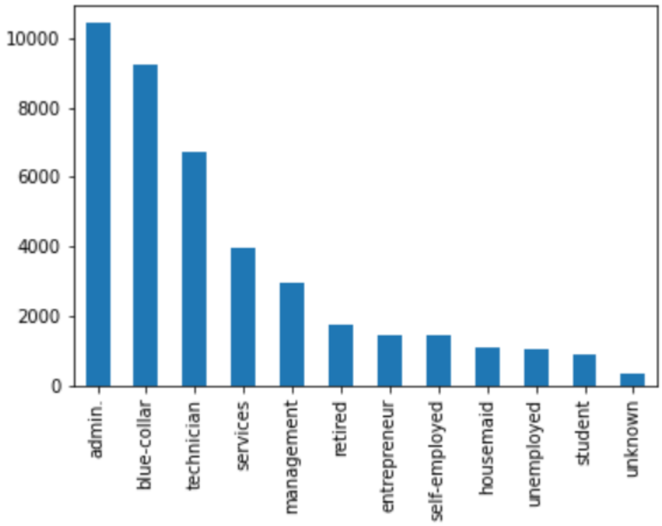

Figura 3: **Type of job customer has**

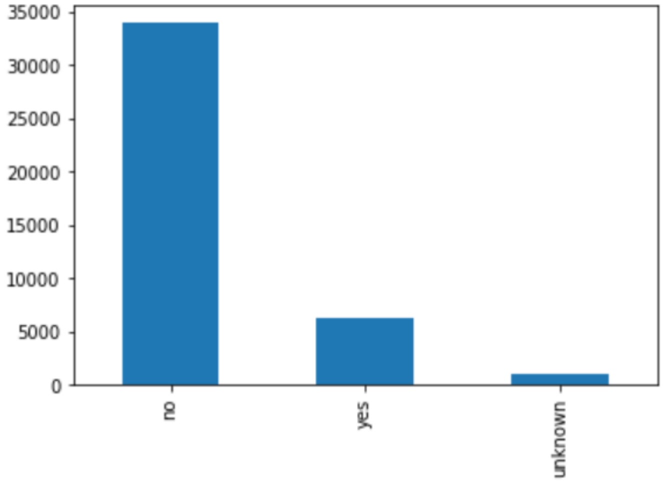

Figura 4: **Does the customer has a personal loan?**

We considered encoding time by grouping values together into some number of sets, and use the set as a categorical attribute. For example, grouping month by quarters of seasons or grouping the days of the week into weekday and weekend. These ideas were scrapped in order to maintain consistency across all categorical features encoded.

3. Modelling

We use Neural Network as the model to fit our data and predict whether a client would respond positive or negative to the campaign.

We divide our data into training and testing first. With an 80/20 division. Train set therefore consists of 32.950 rows

[3] https://www.kdnuggets.com/2021/05/deal-with-categorical-data-machine-learning.html

while Test set has 8.238 values. Data is also shuffled to provide diverse entries to both training and testing steps.

We use Keras to build our model with 3 layers (input layer, hidden later and output layer). We used binary cross entropy as the loss function for our binary classification model. The binary_crossentropy function computes the cross-entropy loss between true labels and predicted labels [4].

To train our network we used 10 epochs getting the accuracy history shown in **Figure 5** and the loss history shown in **Figure 6**.

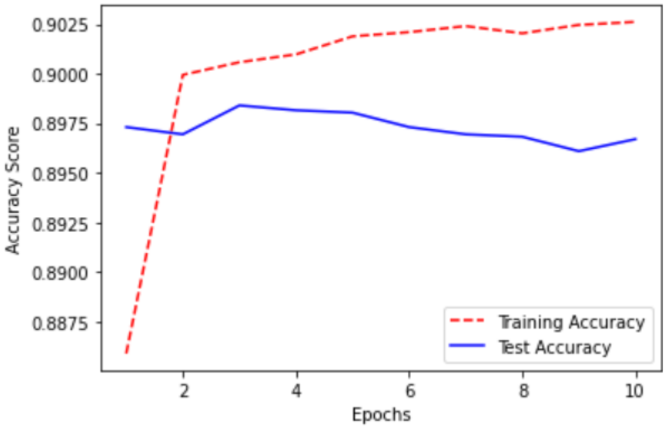

Figura 5: **Training and Testing Accuracy**

In terms of Accuracy, our training process showed good results, starting on 0.8859 and finishing on 0.9026. Validation accuracy was a bit lower but still around the same value, this time 0.8967. As a percentage this sits around 89 %, 90 % accuracy.

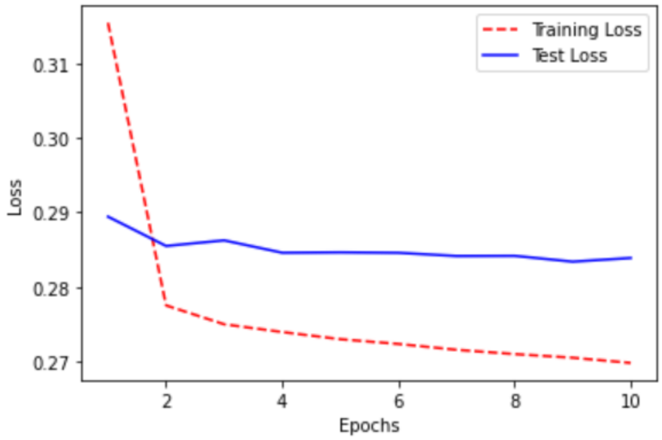

Figura 6: **Training and Testing Loss**

In terms of Loss, the training process started with a higher loss of 0.3154 and went down to settle around 0.27 for most of the epochs and finally reaching 0.2698. Validation loss started and remained around 0.28.

We added **Early Stopping** to our network with a patience of 2 and it showed pretty similar results. Only difference is that it stopped before the last epoch run since two previous runs had the same loss.

We finally display the network architecture below.

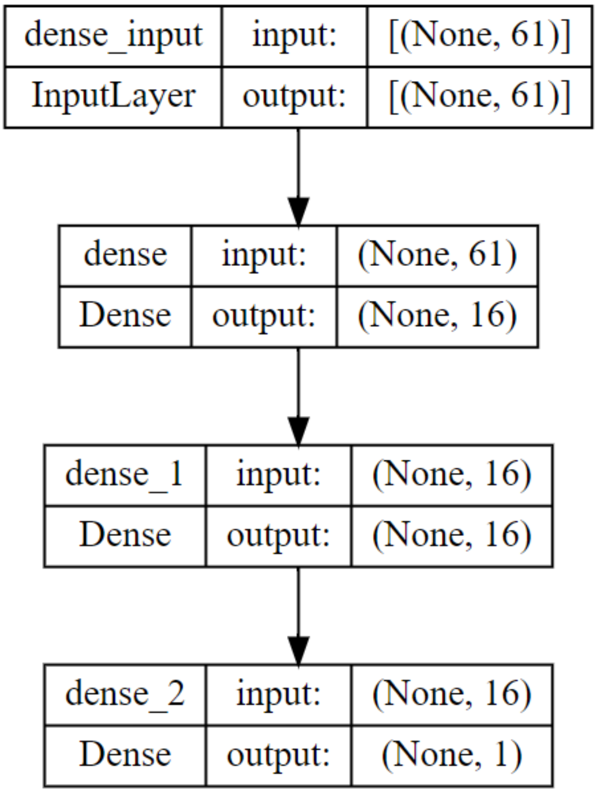

Figura 7: **Network Architecture**

4. Conclusion

This project presented itself as a learning opportunity at many levels. First of all, improving upon the correct handling of both categorical and numerical features from a single dataset.

Understanding the role of normalization as a way to have a common scale across features with different ranges, while learning different types of encoding algorithms for our categorical features.

One specific learning outcome that we consider valuable to highlight is the fact that encoding a categorical column as a set of ordinal values could potentially confuse our machine learning models. This might create the illusion that certain order exists across our categorical values. Thus, One-Hot encoding is used.

These new learning experiences also come with challenges. For instance, should encoded variables be normalized together with our original numerical values? - We have found contradicting opinions about this question so further research is required.

In addition to that, we still find challenging how to process an unbalanced dataset since resampling techniques seem interesting, but might not reflect the real behavior of our data.

[4] https://vitalflux.com/keras-categorical-cross-entropy-loss-function/

Finally, neural network was a new topic and therefore, there is clearly room for improvement with the current implementation and hyperparameters. Although good accuracy has been reached, we will further explore in future work, how to determine the right number of units, layers, epochs and the right activation function, among other relevant factors.