Report - Business Case Exercise

Maciej Romański

Table of Contents

Execution Summary

The purpose of the exercise was to create a predictive model for predicting the initial interest of the user in the long-term gym subscription. Input data contained structured socio-demographics data and unstructured data with interest groups for each user. The unstructured data was prepared to be used for model training by filling missing values and encoding categorical data into numeric values. For unstructured data, word embedding representation of interest groups of each user was calculated and added to the dataset. Two different machine learning classification models were tested: XGBoost and Neural Network. Models effectiveness was assessed using F1 score metric, which is suitable for classification problems with imbalanced classes. The model better performing on the validation set was XGBoost, so it was used to generate predictions on test data.

Input Data and Transformation

Input data provided for the exercise contained the structured tabular data in csv file, as well as the unstructured data in json file.

Structured Data

Data Exploration

As a first step, some basic data exploration and visualization was performed. Missing values in the dataset were visualized in Figure 1.

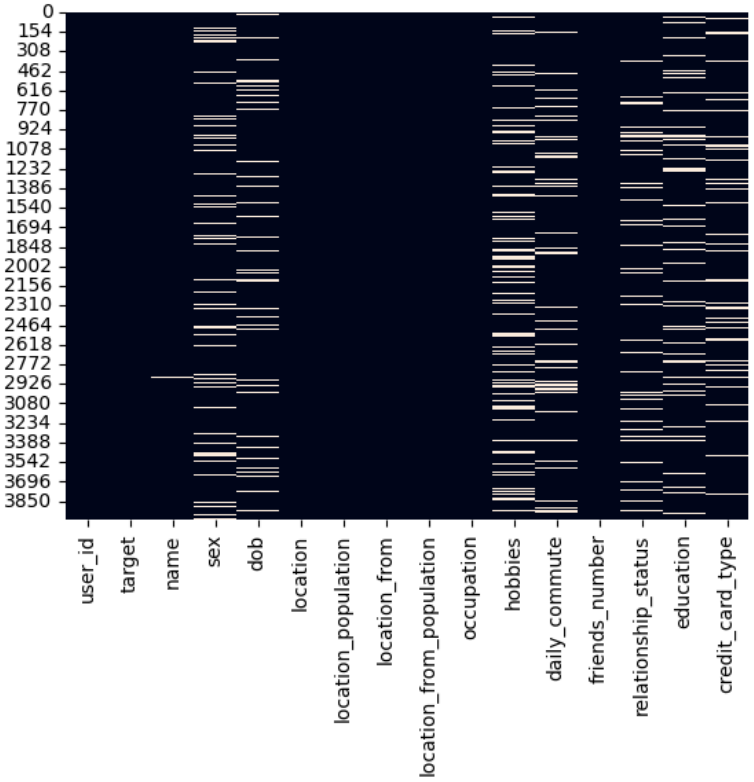

Figure 1. Heatmap of missing values in the training dataset

The cross-tab was created and plotted in Figure 2 to see the impact of the “sex” feature on the “target” feature.

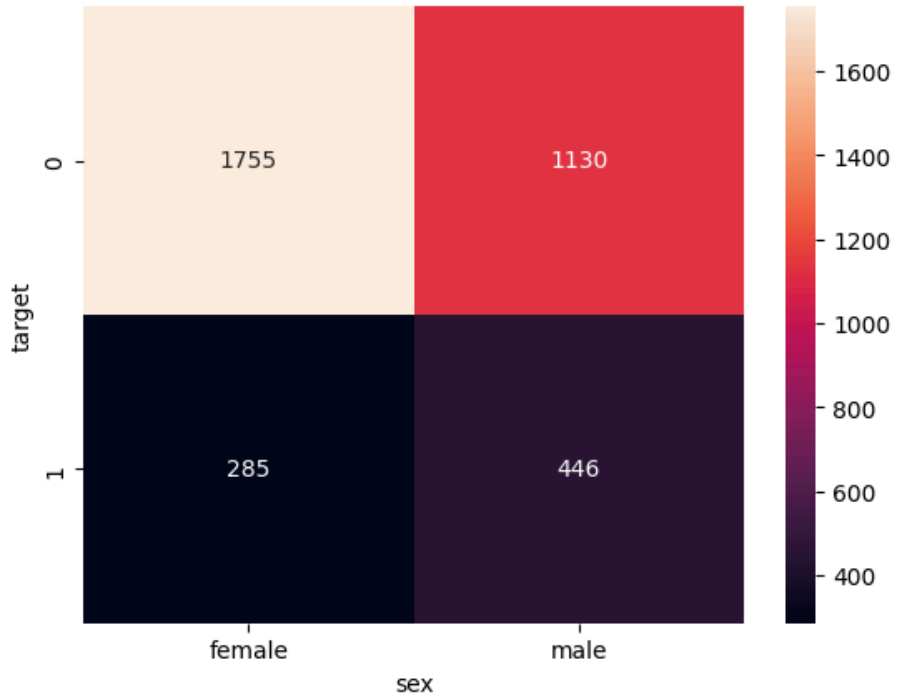

Figure 2. Sex vs Target cross-tab

It can be observed that men are more likely to buy the long-term gym subscription than women. Then, the connection between the relationship status and target was visualized using barplot in Figure 3.

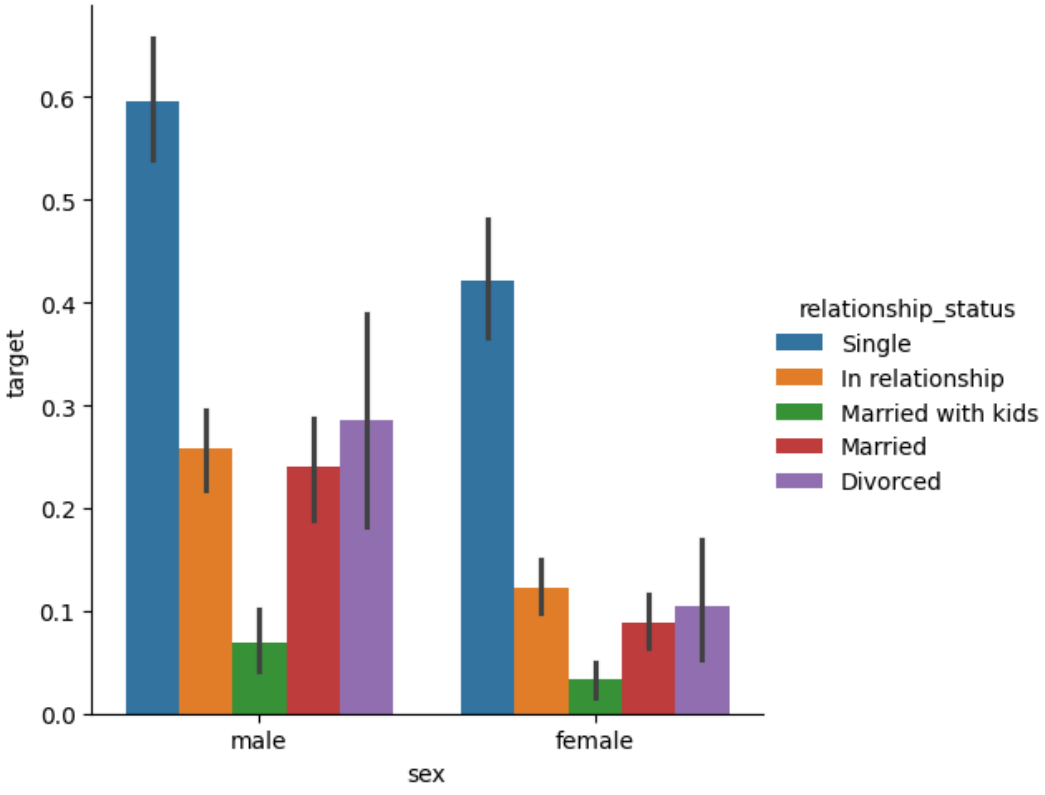

Figure 3. Percentage of target value 1 for each relationship status

As expected, the highest percentage of users interested in gym subscription is among the people who are single and lowest through married people with kids.

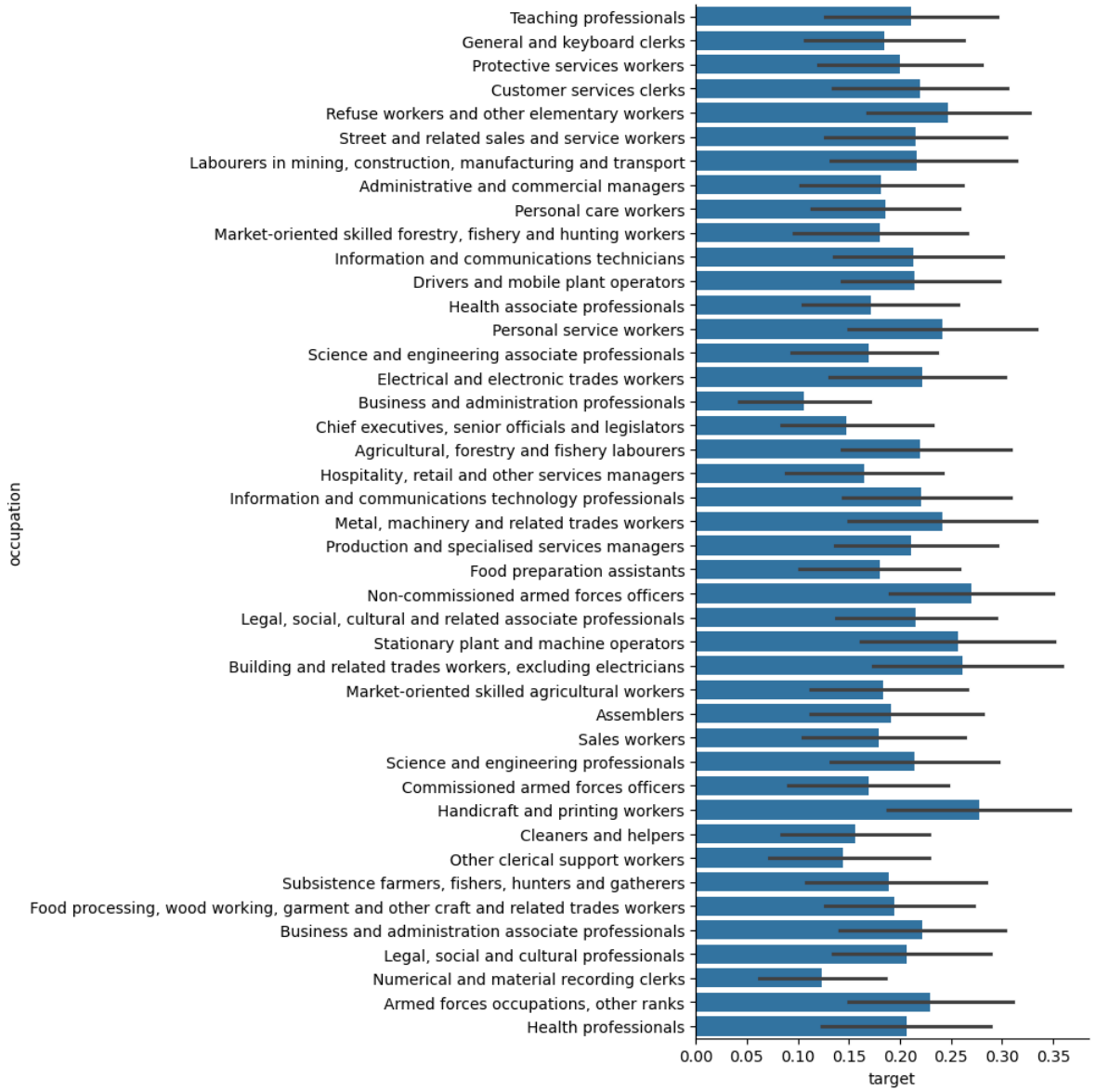

Figure 4. Percentage of target value 1 for occupation group

In Figure 4 it can be seen that, the lowest probability of buying gym subscription is through 'Business and administration professionals' and the highest is through 'Handicraft and printing workers' and 'Arm forces officers'.

Filling missing values

Sex

There were 384 missing sex values. In Polish language all (or almost all) female first names end with letter 'a'. Missing values of 'sex' feature can be filled based on this fact. To be sure this is true, the number of all female users was counted as well as the number of all females

whose name ends with letter 'a'. These numbers are the same, so the method can be applied.

Date of birth

There were 394 missing ‘dob’ values. To make a numerical feature from date of birth, it was replaced by age. Missing age values were replaced with mean. The distribution of the age variable was plotted in Figure 5.

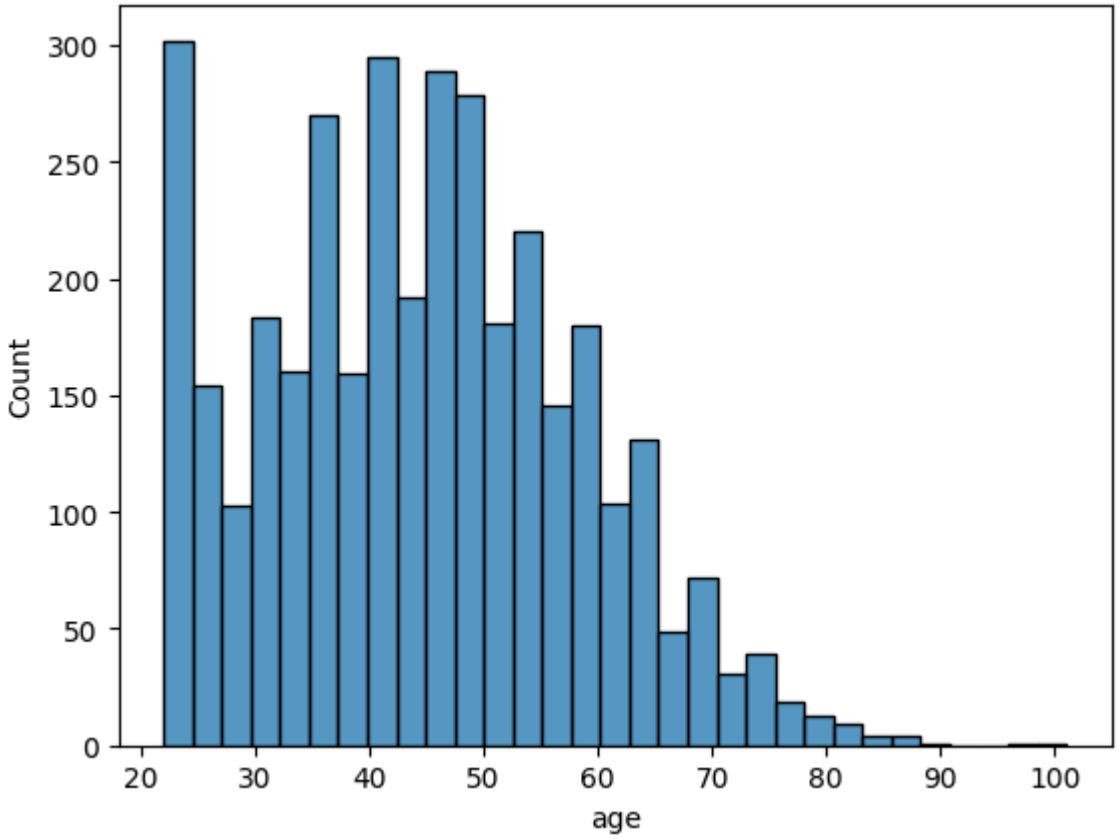

Figure 5. Distribution of age variable in the dataset

Daily commute

There were 405 missing daily commute values. They were replaced by mean.

Relationship status and credit card type

There were 393 missing relationship status values in the dataset and 428 credit card type. They were replaced with random values from the corresponding columns. This keeps the initial distribution of categories in the dataset.

Education

There were 408 missing education values. They were replaced with the most common value which is 4.

Hobbies

There were 678 missing hobbies values. They were replaced with a ‘no_hobbies’ string.

Encoding categorical data

As the next step of data preprocessing, categorical data was encoded to make it numerical.

Hobbies

Hobbies column contains for each user the string with a list of hobbies separated by the comma. Firstly each value was split into a list of strings, where each string is a 1 hobby. Then the MultiLabelBinarizer from scikit-learn package was used to encode the hobbies. As a result the long vector was obtained for each user with the binary 0/1 value for each hobby in the dataset. This vector was then concatenated with the initial dataframe.

Sex

LabelEncoder was used to replace male/female values with 0/1 values.

Occupation, Relationship Status, Credit Card Type

For these columns, the OneHotEncoder was used, to create a separate column for each class.

High Cardinality Columns

Variables with high cardinality, i.e. very high number of categorical values, like “name”, “location” and “location_from” were dropped from the dataset.

Unstructured Data

The json file provided contained the list of interest groups for each user. The approach chosen to extract the features from such data was to firstly merge the interest groups to one string for each user and then calculate word embeddings for this string using the BERT model.
For each user the average embedding across all tokens in the input text was computed. As a result, the vector of fixed size representing the string containing all interest groups names of the user was obtained. Such vectors were then added to the dataframe.

Model Selection and Training

Target Imbalance

The distribution of a target value was presented in Figure 6. It can be noticed that the target value is imbalanced. There are much more 0 values than 1 values in the dataset. This fact has to be addressed both during building a model and choosing appropriate validation metrics.

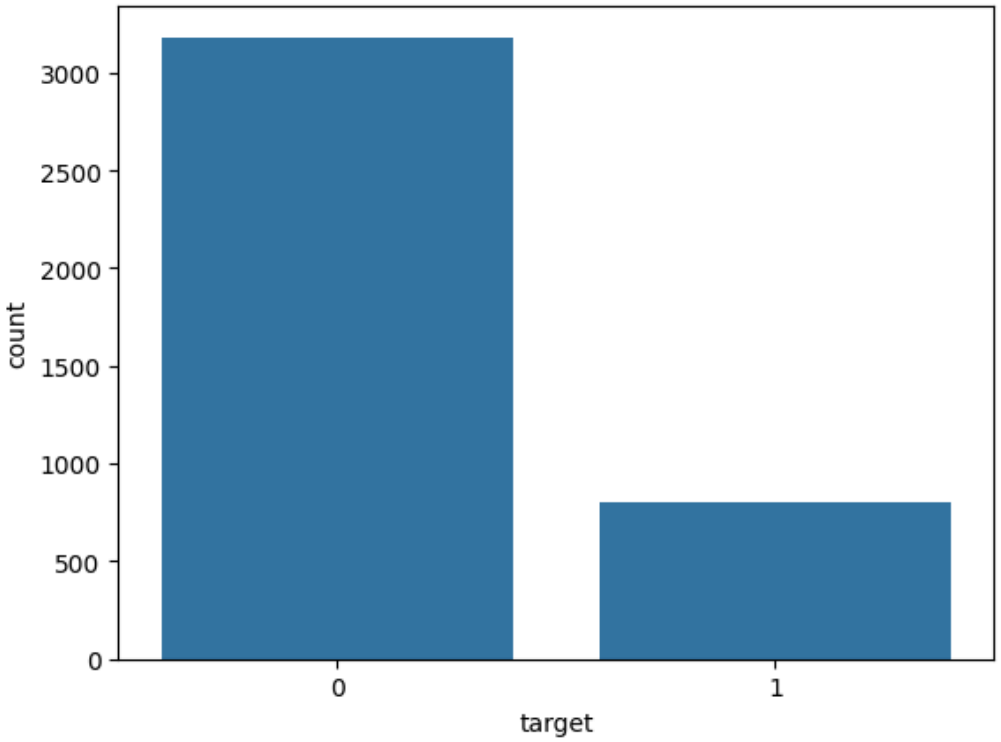

Figure 6. Number of records for each target value.

Splitting data to Train and Validation sets.

For the purpose of assessing model performance, the provided training data was split to training and validation sets in 80/20 ratio. Stratification based on target value was applied, to preserve the distribution of target value in both datasets.

Tested Models

Two models were trained and compared for two sets of data (with and without unstructured set).

XGBoost

First model that was considered was an XGBoost Classifier. To address the data imbalance, the “scale_pos_weight” parameter was used. Hyperparameters of the model were tuned using the Hyperopt package.

Neural Network

Second model was the neural network built in Pytorch. It was built of Linear layers with ReLu activations. Selected loss function was BCEWithLogitsLoss, which is intended for classification purposes. The network was trained using SDG optimizer.

Model Quality Assessment

Evaluation Metrics

The most commonly used metric for classification problems is Accuracy. It is the ratio of correct predictions to all predictions. But it is not a proper metric if we work with imbalanced data. For example, if we created a dummy model that always returned 0 as a result, it would have accuracy of 80%.

In this case, the suitable metric is F1 score, which is the harmonic mean of precision and recall.

Results without unstructured data

XGBoost

Trained on the data without including interest groups, the XGBoost model achieved 0.80 accuracy and 0.60 f1 score on validation set. The confusion matrix of the predictions was plotted in Figure 7.

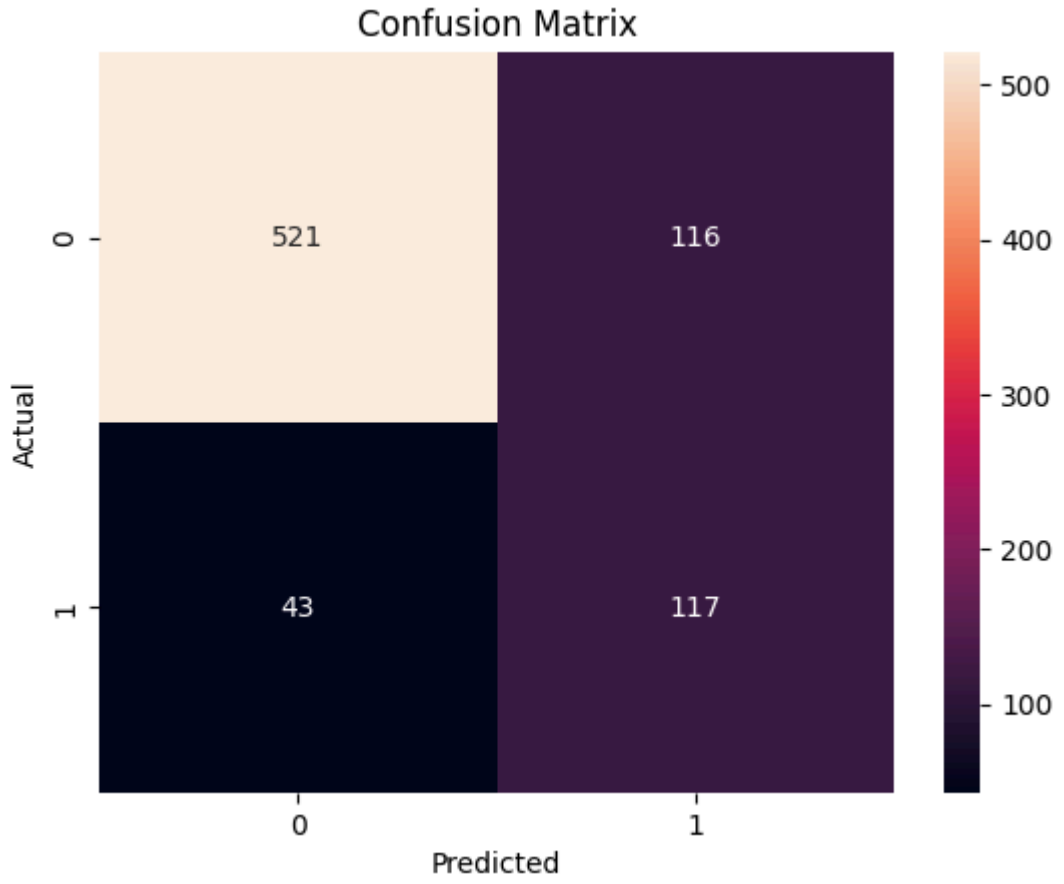

Figure 7. Confusion matrix for predictions on validation data - XGBoost model, without unstructured data

Neural Network

It achieved 0.78 accuracy and 0.54 F1 score. The confusion matrix was shown in Figure 8.

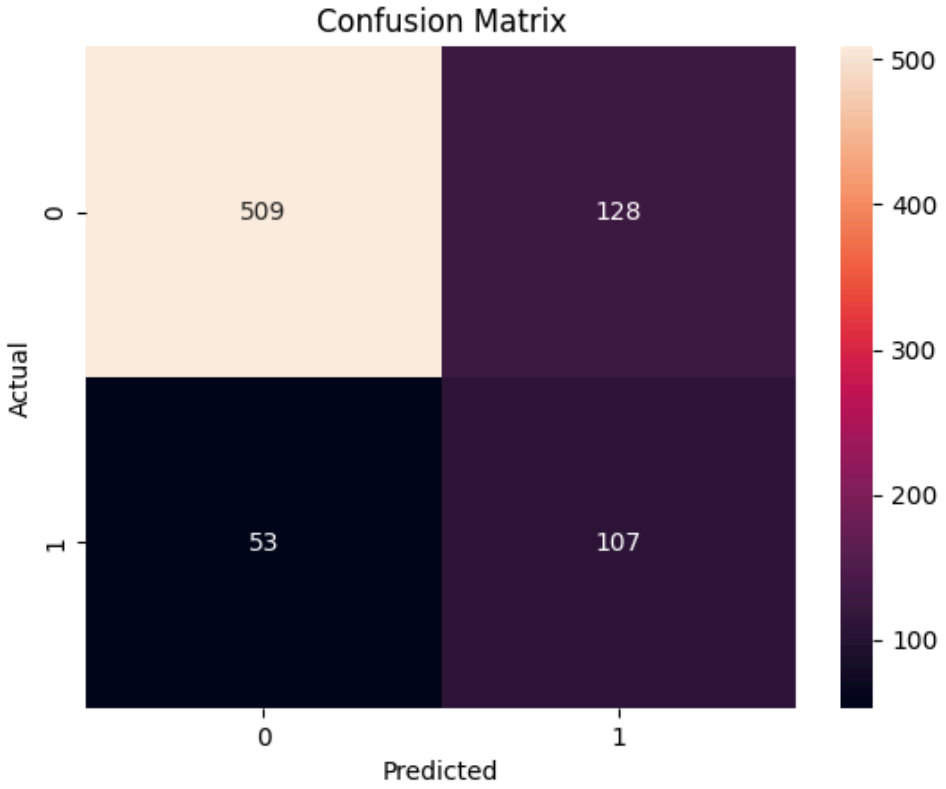

Figure 8. Confusion matrix for predictions on validation data - Neural Network model, without unstructured data

Both models perform poorly and return a lot of False Positive values.

Results after adding unstructured data

XGBoost

After adding interest groups word embeddings as a features to the dataset, the XGBoost Classifier achieved 0.91 accuracy and 0.77 F1 score, so the considerable improvement was observed. The confusion matrix of the predictions for a validation set was plotted in Figure 9.

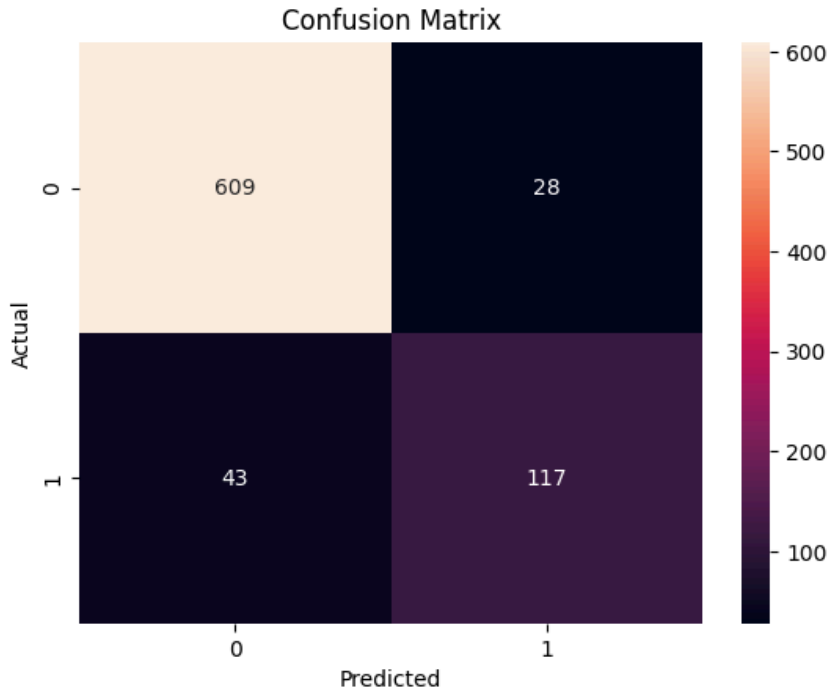

Figure 9. Confusion matrix for predictions on validation data - XGBoost model, after adding unstructured data

Neural Network

In this case neural network classifier achieved 0.82 accuracy and 0.67 F1. The improvement is also noticed, but the model performed worse than XGBoost. Confusion matrix was plotted in Figure 10.

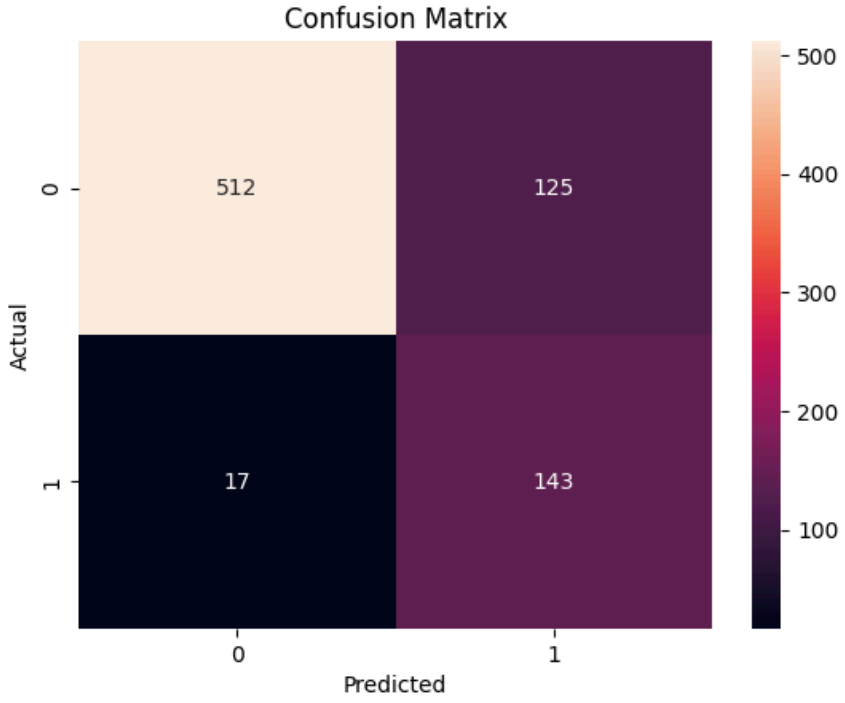

Figure 10. Confusion matrix for predictions on validation data - Neural Network model, after adding unstructured data

Findings

The conclusion of the experiments performed is that the use of word embeddings of unstructured data significantly improved models performance. The approach might be considered correct.

The XGBoost classifier turned out to be more effective, so it was used to generate predictions on test data.

The entire python code used for this exercise can be found in a jupyter notebook attached.

Limitations of the Approach

A lot of different experiments could be performed to improve the model effectiveness.

First of all, different approaches to filling missing data and encoding categorical values could be tested.

What is more, the more complex models could be tested tuned, having more time and computational resources.