Data Mining and Text Mining Course Project

Team: Cash me outside

We used Python 3 with commonly used Data Mining libraries like: numpy, pandas, matplotlib, scikit-learn, imblearn, seaborn and xgboost. For data exploration and comparison we also used KNIME and Orange 3.

First part of our work included preprocessing and exploration of given datasets. By inspecting the dataset we noticed that there are a lot of missing values, string values and categorical values.

We filled missing values of the dataset with most frequent one in categorical features and in numerical features we choose median over mean to fill missing data since median is more resistant to outliers. Next, we noticed that BIRTH_DATE attribute was not so useful in current date format, so we computed the age of the client. In the dataset there are three categorical features: SEX, EDUCATION and MARRIAGE. So we used *LabelEncoder* to translate them to numerical values first and after that we applied *OneHotEncoding* technique in order to get more useful representation of them. We also removed CUSTOMER_ID column from dataset since it is not relevant for the model. By looking at correlation heatmap we noticed that there are high correlation between BILL_AMOUNT attributes as well as between PAY_AMOUNT attributes. We decided to use just the mean values of BILL_AMOUNT and mean values of PAY_AMOUNT features. After that we normalize dataset using *StandardScaler*.

Since after visualizing the input target values, we noticed that output is highly unbalanced, so we needed to do oversampling of minority class. For this purpose we used *RandomOverSampler*.

Second part was more about model creation. We tested multiple models among which the best performance was given by XGBoost, LogisticRegression and MLP. For evaluation we used stratified holdout, because we oversampled the train dataset with minority class and CV was not a practical because oversampling in that case need to be applied in each fold. The best F1 - measure score had the BaggingClassifier with XGBoost as base classifier. Parameters were tuned using GridSearch (most important were gamma, subsample, learning_rate, n_estimators, max_depth for XGBoost and max_samples and max_features for BaggingClassifier). We achieved 0.5536 f1 score on 25% holdout test data.