Group 29 - Nina Dobša, Ghazal Arzanian, Joachim Biberger April, 2024.

Assignment 1

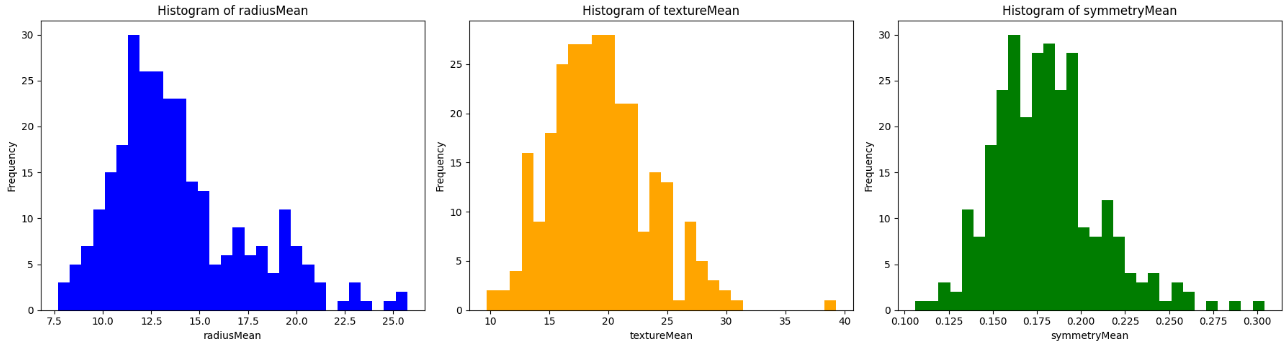
Introduction

In this report we will compare performances of three different classification algorithms on four data sets. The whole machine learning process will be explained starting from description of data, preprocessing, making predictions, evaluating and comparing results. Also, we will discuss how characteristics of a data set can affect performance and run time (such as size of a data set, target value distribution, types of variables, …). Data sets used in this exercise are Loan default prediction, Breast cancer, Predict students' dropout and academic success and Estimation of obesity levels based on eating habits and physical condition.

Data set description and preprocessing steps

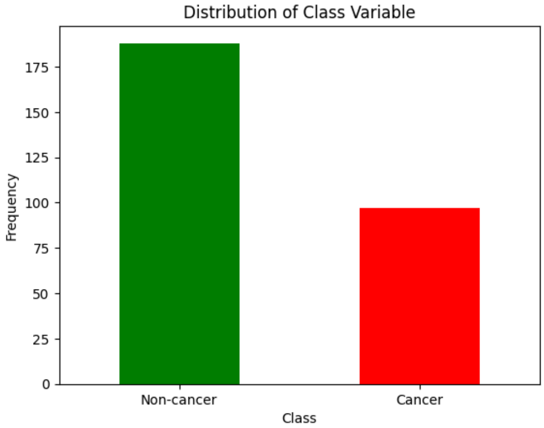
Breast Cancer Dataset

Description of dataset:
Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. The records are classified in true / false classes which represent malignant / benign cancer. Breast cancer is a very small data set consisting of 569 instances and 31 attributes. Data type of each attribute is float except for the target variable column class which is boolean. In the data set there are no missing values, which makes the preprocessing part easier. Below, you can find the distribution of radiusMean, textureMean and symmetryMean.



Picture 1.1 - radiusMean, textureMean and symmetryMean distribution histograms

From the distribution of class variables, we can conclude that the dataset is not balanced. We will have to keep this in mind when picking metrics for comparing models.



Picture 1.2 - target attribute distribution

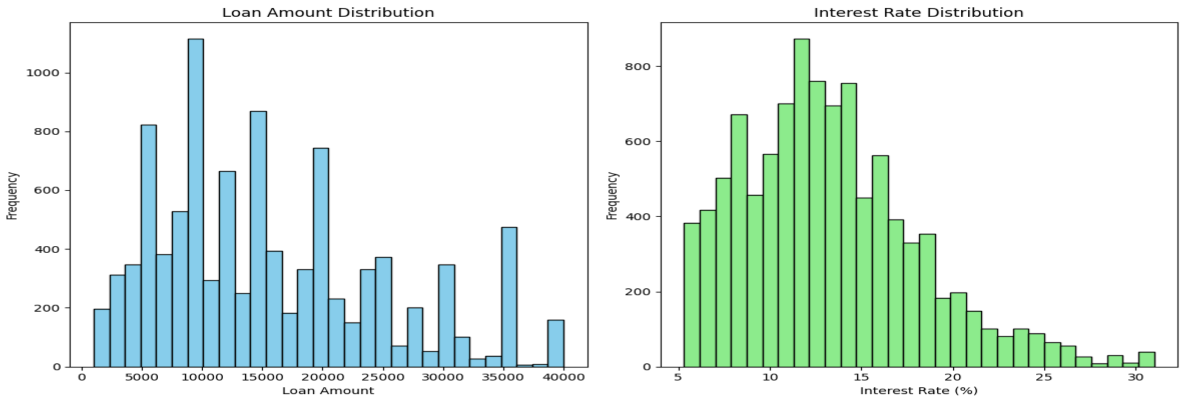
Preprocessing steps:

The Breast Cancer dataset didn't require much preprocessing work. As mentioned before, there were no missing values present. All attributes, except for the target variable, were numerical, and the target variable was encoded into binary form (0s and 1s). The 'ID' column seemed irrelevant for modeling purposes so we removed it. Additionally, we searched for outliers (identified by having an absolute z-score exceeding 3) but it turned out that 38 rows out of 286 should be removed, so we decided that it is better to keep the data as it is. Considering the context of breast cancer dataset, outliers may hold critical information and their removal might not be the most prudent choice. The last step of preprocessing for this dataset was standardization of variables.

Loan Dataset

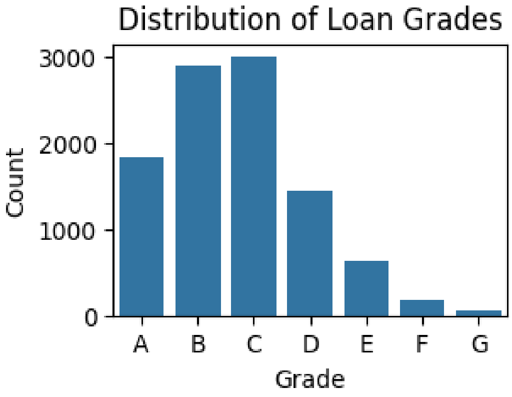
Description of dataset:

The Loan Default Prediction dataset contains information about borrowers and their credit profiles, aiming to predict the likelihood of loan default based on various features. The records are classified according to the variable "Grade", which encompasses seven distinct values: A - lowest risk of default, B - low risk of default, C - moderate risk of default, D - moderate to high risk of default, E - high risk of default, F - highest risk of default. The data set contains 10000 observations in total with 92 variables - 78 numerical (69 float, 9 integer) and 14 of type object. There are no missing values in this dataset. Below you can find distribution of values for Loan Amount and Interest Rate.



picture 2.1 - Loan Amount and Interest Rate distribution histograms

From the distribution of class variables, we can conclude (just like we did in the breast cancer dataset) that the dataset is not balanced.



Picture 2.2 - target attribute distribution

Preprocessing steps:

Preprocessing steps were organized into a pipeline which enabled chaining together multiple preprocessing steps and applying them sequentially to the data. We constructed separate pipelines (with sklearn library) for numeric and categorical features. One-Hot Encoding was done for categorical and standardization for numerical features. As mentioned before, there were no missing values in the dataset so we didn’t have to deal with that. At the end, we also did feature selection to get better results for some classifiers with the function SelectKBest. This transformer selects the top k (in our case 10) features based on a specified scoring function. For numerical features we picked the top 10 features with the highest ANOVA F-values and for categorical features we used the chi-squared scoring function.

Predict Students’ Dropout and Academic Success Dataset

Description of dataset:

The dataset includes information about each student at the time of his/hers enrollment – academic path, demographics, and social-economic factors in order to estimate students academic success. The problem is formulated as a three category classification task (dropout, enrolled, and graduate) at the end of the normal duration of the course. Dataset contains 4424 instances and 35 attributes. All attributes are integer or float but not all of them are numeric. Most of them are actually categorical (nominal and ordinal) attributes already encoded to integers to make the dataset usable for all ML methods (for e.g. marital status, education level, nationality, occupation, …). You can find more information about this dataset in the assignment 0.

Preprocessing steps:
For this dataset, all variables were numerical. Therefore, for preprocessing, we only used the standardization (StandardScaler from the sklearn library) module. We also used make_pipeline from the sklearn module to create a pipeline that performs standardization for the models. The reason for using pipeline is that it encapsulates multiple preprocessing steps and model fitting into a single object. This simplifies the code, ensures consistency in preprocessing across different models, and helps prevent data leakage by automatically applying the same transformations to both the training and test data. Additionally, pipelines facilitate code reusability and make it easier to experiment with different preprocessing techniques and model architectures.

Estimation of obesity levels based on eating habits and physical condition Dataset

Description of dataset:
This dataset includes data for the estimation of obesity levels for individuals from Mexico, Peru and Colombia, based on their eating habits and physical condition. The records are classified according to the variable "NObesity" (Obesity Level), which encompasses seven distinct values: Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II, and Obesity Type III. Dataset contains 2111 instances and 16 attributes. There are exactly 8 numeric and 8 categorical attributes. You can find more information about this dataset in our assignment 0.

Preprocessing steps:
This dataset comprises both numerical and categorical variables. Therefore, we used both standardization and one-hot encoding techniques. For the efficiency of these preprocessing steps, we used a pipeline approach as well.

Models

Our task was to select three classifiers that differ in their characteristics. After careful consideration, we decided to go with Naive Bayes, Support Vector Machines (SVM), and Random Forest classifiers. First and foremost, we required classifiers capable of handling diverse data types, including numerical and categorical variables. Each of these classifiers possesses mechanisms to handle various data characteristics, such as outliers, noise, and missing values. This versatility ensures their effectiveness across different datasets and under varying conditions.Lets explain briefly how each of classifier works and what are the main differences between them;

1. Naive Bayes classifier is based on Bayes' theorem which calculates the probability of a class given a set of features using conditional probability.
   One of their key advantages is their simplicity and computational efficiency, making them efficient for large datasets with high-dimensional features. However, their main limitation lies in the assumption of feature independence, which may not hold true in all datasets.

2. Support Vector Machines (SVM) classifier is a powerful classification algorithm that aims to find the optimal hyperplane that best separates data points belonging to different classes. It works well for both linearly separable and non-linearly separable data, thanks to the use of kernel functions that map input data into higher-dimensional spaces. SVMs are effective in high-dimensional spaces and are relatively robust against overfitting, making them suitable for a wide range of classification tasks. However, SVMs can be computationally expensive, especially when dealing with large datasets.

3. Random Forest: Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputting the mode of the classes (classification) of the individual trees. Random Forests are known for their robustness against overfitting, ability to handle noisy data and missing values, and capability to provide insights into feature importance. They are particularly useful when dealing with high-dimensional datasets and can capture complex relationships between features and target variables.

Metrics selection

The selection of metrics for evaluating a model depends on the specific objectives and the nature of datasets, particularly the balance between classes and the cost of different types of errors. As our datasets are mostly imbalanced and cover different topics like medicine and banking, it is important that our metrics cover needs from each dataset. We decided to go with accuracy, precision, recall and F1 score. Here is an explanation of each metric;

1. Accuracy: Representing the proportion of correctly classified instances out of the total instances.This metric is useful when the classes are balanced.

2. Precision: Quantifies the proportion of true positive predictions among all instances predicted as positive. High precision indicates fewer false positives, which is crucial in scenarios where the cost of false positives is high, such as medical diagnoses.

3. Recall: It calculates the proportion of true positive predictions among all actual positive instances. High recall indicates fewer false negatives, which is essential when missing positive instances could lead to severe consequences.

4. F1 score: Metric commonly used to evaluate the performance of a classification model. It combines precision and recall into a single value, providing a balance

between the two. A high F1 score indicates both high precision and recall, making it ideal for scenarios where false positives and false negatives are equally important.

Combination of these metrics will give us a rounded view on our model's performance in terms of both error types (false positives and false negatives) and overall effectiveness. At the end in our tables we used only F1 score and accuracy so it would be easier to read.

Performance

In our research, we tested performance of models on both standardized and original data. We also evaluated model performance with both tuned and default parameters to assess the impact of parameter optimization on predictive accuracy. For finding the best parameters, we used grid search with a custom grid. Furthermore, we compared the performance of models trained using the hold-out method with those trained using 5-fold cross-validation. We didn’t test all possible combinations of preprocessing techniques, parameter settings, and evaluation methods but we ensured to have enough results to compare the influence for each of the components. We expect to get better results with tuned parameters compared to default parameters and with standardized data compared to not standardized data. We also expect that the results may vary for different evaluation methods, such as cross-validation and hold-out, due to the randomization of data splitting but comparing the running time the cross-validation should be more expensive. While all three classifiers possess notable capabilities, considering the broader context, we think that Random Forest and SVM classifiers may outperform Naive Bayes.

Breast Cancer Dataset

Breast cancer is a small data set so we expect the shortest run time compared
to others. Preprocessing was done with dropping columns and standardization of variables. We tested the performance of the models with 5-fold cross validation and hold-out method on standardized and unstandardized data and with default and best parameters. For finding the best parameters, we used grid search with a custom grid. Run times for each of the variante was low, under 3s in most of the cases. Run times including grid search, as expected, took a bit longer. The longest run was for Random Forest grid search (32 sec). You can see performance of models for this data set in the table below.

	tuned cross validation stand	default cross validation stand	tuned hold-out stand	default hold-out stand	default hold-out not stand	default cross validation not stand
NB	A = 0.944 F1 = 0.917	A = 0.930 F1 =0.892	A = 0.982 F1 =0.979	A =0.983 F1 = 0.979	A = 0.983 F1 = 0.979	A = 0.939 F1 = 0.904
SVM	A = 0.986 F1 = 0.978	A = 0.974 F1 = 0.958	A = 1.000 F1 =1.000	A = 1.000 F1 =1.000	A = 0.930 F1 = 0.909	A = 0.886 F1 = 0.793
RF	A = 0.947 F1 = 0.945	A = 0.947 F1 = 0.922	A = 0.982 F1 = 0.979	A = 0.983 F1 = 0.979	A = 0.983 F1 = 0.979	A = 0.952 F1 = 0.900

Table 1 - Breast Cancer data set results

For this specific problem, the recall might be the most critical metric because missing a positive case could be life-threatening. Precision is also very useful to minimize unnecessary treatment caused by false positives. Therefore, the F1 score, which balances both recall and precision, takes precedence over accuracy in evaluating model performance. All classifiers performed really well with accuracy, recall, precision and F1 score all exceeding 0.85 for every combination. The results suggest that the SVM classifier with standardized input data yields the best performance, achieving perfect scores across all metrics in the hold-out method (Accuracy, Recall, Precision, F1 Score) for both default parameters and parameters from grid search. As expected, the results with tuned parameters are better or the same in all cases as well as for standardized data. Although, we can see that parametrization affected much on the results for SVM, it didn't affect much on Naive Bayes and Random Forest models. Also we noticed that results with the hold-out method were better in comparison with the results with cross-validation. Below you can find default and tuned parameters used in evaluating the model.

NAIVE BAYES

```
default parameters: { 'var_smoothing': 1e-09}
parameters with grid search: {'var_smoothing': 0.1}
```

SVM

```
default parameters: {'C': 1.0,  'gamma': 'scale',  'kernel': 'rbf'},
parameters with grid search: {'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'}
```

RANDOM FOREST

```
default parameters:
{ 'max_depth': None,  'min_samples_split': 2,  'n_estimators': 100}
parameters with grid search:
{'max_depth': 10, 'min_samples_split': 10, 'n_estimators': 10}
```

Loan Dataset

Loan default training data set has 10000 observations and 92 attributes so we expect that creating and tuning models will result in longer running time compared to smaller datasets like breast cancer. First we made models without standardization testing both 5-fold cross-validation and hold-out methods. Then we repeated the process with standardized variables and in the end with tuned parameters of classifiers. For finding the best parameters, we used grid search with a custom grid. Run times for models were max 10 seconds for hold-out and max 32 seconds for cross-validation. You can find the performance of models in the table below.

	tuned cross validation stand	default cross validation stand	tuned hold-out stand	default hold-out stand	default hold-out not stand	default cross validation not stand
NB	A = 0.482 F1 =0.418	A = 0.056 F1 =0.05	A = 0.41 F1 =0.48	A = 0.052 F1 = 0.07	A = 0.042 F1 =0.051	A = 0.040 F1 = 0.048
SVM	A = 0.850 F1 =0.847	A = 0.797 F1 =0.76	A =0.85 F1 =0.86	A = 0.790 F1 = 0.779	A = 0.366 F1 = 0.177	A = 0.386 F1 = 0.187
RF	A = 0.799 F1 =0.776	A = 0.810 F1 =0.79	A =0.86 F1 =0.78	A = 0.798 F1 = 0.777	A = 0.308 F1 =0.183	A = 0.329 F1 = 0.204

Table 2 - Loan default prediction data set results

Just like in the breast-cancer dataset, the F1 score will be of critical value in the context of the data. Initially, when working with non-standardized data, the model results were quite bad, particularly for the Naive Bayes classifier. However, after standardizing the data, significant improvements were observed for both the SVM and Random Forest classifiers, while the performance of the Naive Bayes classifier remained relatively unchanged. With tuning parameters, we got good results with the best performance for the SVM classifier achieving scores of 85% accuracy and 84.7% F1 score. Naive Bayes' best performance resulted in 48% accuracy. We wanted to improve so we explored feature selection techniques(Selected the top 10 best features based on their chi-squared statistic using SelectKBest for categorical features and selected the top 10 best features based on their ANOVA F-value using SelectKBest for numerical features), resulting in a significant boost in performance for the Naive Bayes classifier, achieving 76% accuracy and an F1 score of 76%. Also we noticed that results with the cross-validation got better than results with the hold-out method. Below you can find default and tuned parameters used in evaluating the model.

NAIVE BAYES

```
default parameters: { 'var_smoothing': 1e-09}
tuned parameters: {'var_smoothing': 0.03511191734215131}
```

SVM

```
default parameters: {'C': 1.0,  'gamma': 'scale'}
tuned parameters: {'C': 10, 'gamma': 'auto'}
```

RANDOM FOREST

```
default parameters: { 'max_depth': None, 'n_estimators': 100}
tuned parameters: {'max_depth': None, 'n_estimators': 200}
```

Estimation of obesity levels based on eating habits and physical condition Dataset

We tested the performance of the models with 5-fold cross validation and hold-out method on standardized and unstandardized data and with default and best parameters. For finding the best parameters, we used grid search with a custom grid. Running times for each model were under 2 seconds with cross-validation method resulting in longer times compared to hold-out method.. You can see performance of models for this data set in the table below.

	tuned cross validation stand	default cross validation stand	tuned hold-out stand	default hold-out stand	default hold-out not stand	default cross validation not stand
NB	A = 0.585 F1 =	A = 0.518 F1 = 0.462	A = 0.598 F1 =0.572	A = 0.515 F1 = 0.462	A = 0461 F1 = 0.371	A = 0.471 F1 = 0.403
SVM	A = 0.956 F1 = 0.952	A = 0.909 F1 =0.909	A = 0.957 F1 =0.957	A = 0.931 F1 = 0.931	A = 0.619 F1 = 0.603	A = 0.593 F1 = 0.574
RF	A = 0.940 F1 = 0.94	A = 0.935 F1 =0.936	A = 0.939 F1 =0.939	A = 0.931 F1 = 0.932	A = 0.610 F1 = 0.592	A = 0.591 F1 = 0.574

Table 3 - Estimation of obesity levels based on eating habits and physical condition data set results

Again, we can see a big influence of standardization on SVM models with accuracy of 59% for non standardized data and 93% for standardized data and no significant improvement on Naive Bayes and Random Forest models. With tuning parameters, we got much better results for all models with the best performance for the SVM classifier achieving 95.6% accuracy and 95.2% F1 score. Naive Bayes models didn't perform very well on this data set, this we could approve with feature selection or dropping some columns that may be correlated with others. Below you can find tuned and default parameters for each model.

NAIVE BAYES

```
default parameters: { 'var_smoothing': 1e-09}
parameters with grid search: {'model__var_smoothing': 0.08111308307896872}
```

SVM

```
default parameters: {'C': 1.0,  'gamma': 'scale',  'kernel': 'rbf'},
parameters with grid search: {'model__C': 100, 'model__gamma': 'auto'}
```

RANDOM FOREST

```
default parameters:
{ 'max_depth': None,  'min_samples_split': 2,  'n_estimators': 100}
parameters with grid search: {'model__max_depth': None, 'model__n_estimators': 200}
```

Predict Students’ Dropout and Academic Success Dataset

With around 5000 features, we expected some reasonable time of running the models. We tested the performance of the models with 5-fold cross validation and hold-out method on standardized and unstandardized data and with default and best parameters. For finding the best parameters, we used grid search with a custom grid. Running times for each model were under 11 seconds with the longest running time for cross-validation with random forest classifiers. Models with Naive Bayes classifier were the fastest with under 1 second result for both hold-out and cross-validation methods. You can see performance of models for this data set in the table below.

	tuned cross validation stand	default cross validation stand	tuned hold-out stand	default hold-out stand	default hold-out not stand	default cross validation not stand
NB	A = 0.688 F1 = 0.594	A = 0.686 F1 = 0.596	A = 0.690 F1 = 0.599	A = 0.693 F1 = 0.603	A = 0.699 F1 =0.611	A = 0.689 F1 =0.600
SVM	A = 0.771 F1 = 0.700	A = 0.761 F1 = 0.678	A = 0.763 F1 = 0.686	A = 0.758 F1 = 0.672	A = 0.472 F1 = 0.214	A = 0.501 F1 = 0.229
RF	A = 0.775 F1 = 0.689	A = 0.770 F1 = 0.688	A = 0.766 F1 =0.670	A = 0.762 F1 =0.673	A = 0.760 F1 =0.670	A = 0.770 F1 = 0.687

Table 4 - Predict students dropout of academic success data set results

As in the data sets before, we can see that standardization affected a lot of models with SVM classifiers while it didn’t affect Naive Bayes and Random Forest. With tuned parameters we got a bit better result than with default ones. The results indicate that the Random Forest classifier achieves the highest accuracy, reaching 77.5%. However, the SVM classifier performs almost as well in terms of accuracy, while also demonstrating a better F1 score of 70.0%. Given the imbalanced distribution of the target variable, where certain classes may be underrepresented, the F1 score provides a more comprehensive assessment of model performance. This is why we would choose a model with an SVM classifier for the best performing one. You can find default and tuned parameters used in training the model.

NAIVE BAYES

```
default parameters: { 'var_smoothing': 1e-09}
parameters with grid search: {'gaussiannb__var_smoothing': 0.3511191734215131}
```

SVM

```
default parameters: {'C': 1.0,  'gamma': 'scale',  'kernel': 'rbf'},
parameters with grid search: {'svc__C': 10, 'svc__gamma': 0.01}
```

RANDOM FOREST

```
default parameters:
{ 'max_depth': None,  'min_samples_split': 2,  'n_estimators': 100}
parameters with grid search: {'randomforestclassifier__max_depth':
10,'randomforestclassifier__n_estimators': 100}
```

Conclusion

Standardization typically improves model performance, particularly for SVM classifiers, which rely on geometric margins and distance calculations. In contrast, Naive Bayes classifiers, which assume feature independence, may see less impact from scaling. However, the effect of standardization may be minimal if the data naturally clusters well into classes or if features already have similar scales for any model.

As expected, models with tuned parameters consistently outperformed or matched models with default parameters. Parameter tuning involves systematically searching for the best hyperparameter combinations, optimizing the model for the dataset's specific characteristics so it makes sense that the results can only get better.

The choice between cross-validation and the hold-out method depends on factors like dataset size and model stability. While cross-validation provides a more reliable estimate of performance, it can be computationally expensive, resulting in longer running times.. On the other hand, the hold-out method is computationally efficient but may yield higher variance in performance estimates and less efficient data usage. This was confirmed by longer running times for cross-validation compared to the hold-out method.

Interestingly, the best results were achieved with the smallest dataset, specifically the breast cancer dataset, which attained an accuracy of 98.6% using the SVM model with cross-validation and 100% with the hold-out method. Overall, SVM classifiers and Random forest performed the best on almost all datasets. Naive Bayes didn’t work well for all datasets but we are sure that results can get better with more preprocessing steps.