Data Mining Techniques - Assignment 1 - Gr185

A.J. Hazenberg[2649192], D. Pop[2618006], and T.J. Siebring[2683240]

Vrije Universiteit Amsterdam, 1081 HV Amsterdam, Netherlands

This report presents the group's proposed solutions to the first assignment of the course Data Mining Techniques. In the first part (Task 1), an Own Dataset Initiative (ODI) data set will be explored, after which a new downloaded data set on student results will be used in two regression models. In the second part (Task 2), a data set consisting of information regarding the persons on board of the Titanic is prepared such that it can be used to predict the survival chances using a machine learning model. The last part of this report (Task 3) elaborates on state-of-the-art data mining approaches as well as the more theoretical aspects of data mining. The different tasks were solved using the programming language Python in Jupyter notebooks.

TASK 1: EXPLORE A SMALL DATASET

TASK 1A: EXPLORATION

The ODI data set that was loaded into the Jupyter notebook, consists of data of questions that were asked in a questionnaire during the the first lecture of the course Data Mining Techniques (DMT). The questionnaire contains questions on, among other things, the study of the student, the courses that he or she follows and the gender of the student. All the data is gathered in one data set named ODI.

In total 304 students filled in the questionnaire (number of records) consisting of 17 questions in total (number of attributes). A few things can be noted when loading the data set:

- First of all, the data is not clean. When the question states 'Give an answer between 1-100', some students filled in 200, etc., which makes the data not trustworthy.
- Secondly, there are a lot of different ways of entering a birth date. Some students use 'dd.mm.yyyy', someone else uses 'dd-mm-yyyy', another student uses 'dd/mm/yyyy' and even more versions are filled in in the questionnaire.
- A last thing that can be noted is that the results of the questionnaire show 123 different studies in total (entered by the students). The reason for this is that some studies are entered in different ways. Take for example the study Artificial Intelligence. Some students enter 'AI' as study and others enter 'Artificial Intelligence'. There are multiple ways to refer to a study program and it is important for further exploration of the data, that those studies are grouped together.

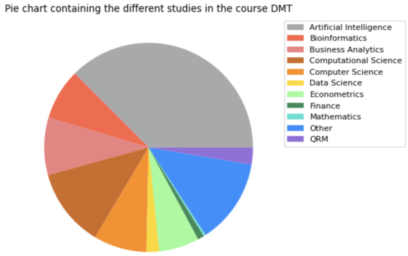


Fig. 1: Pie chart containing the different studies in the course DMT

To investigate the different studies present in the lecture hall, the questionnaire contained the question 'What programme are you in?'. As stated before, the results were very diverse and in total 123 different study programs were detected. After investigations, study programs that are similar, such as 'AI' and 'Artificial Intelligence', were grouped together and eventually 11 categories (studies) were formed. Among these 11 categories, one is the category 'Other', which consists of programs that were undetectable or studies that only contained one match. The results of grouping the studies can be found in the pie chart in Figure 1.

The pie chart shows that mostly Artificial Intelligence students filled out the questionnaire. This however does not say anything about the deviation of the studies in the total course. In total around 600 students are following the course Data Mining Techniques and only 304 students filled out the survey. Because we are missing almost half of the students in the results of the questionnaire, no conclusions can be made about the total group of students following the course.

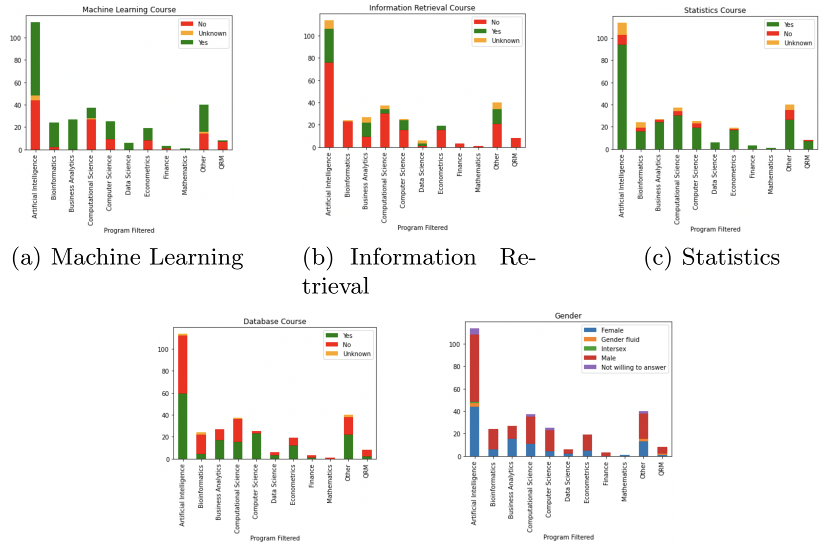
As stated before, the questionnaire consisted of questions regarding the study and courses that the student follows, but also 'random' questions like 'How many neighbours are sitting next to you?' and 'Chocolate makes you ....?'. It was decided that only the questions relevant to the study were investigated and the results can be found in Figure 2.

It can be noticed from the plots above that mostly male filled out the questionnaire. Besides that, almost all students that filled in the questionnaire followed a course in statistics, while only little followed a course on Information Retrieval or Databases.

Even though both the pie chart and the bar plots give very nice insights into the background of the students in the course Data Mining Techniques, it is very important to keep in mind that only half of the students filled in the questionnaire. If the other students would also fill in the questionnaire, different results could come out and thus the plots are not very representable for the course.

TASK 1B: BASIC CLASSIFICATION/REGRESSION

CORRELATION In this section two regression algorithms are investigated to predict the mathematics results of students, based on different features among which their gender, race, parental level of education and other scores. It was decided to use this new data set regarding student results because the previously mentioned data set (ODI) was not representable for the course, contained several 'not interesting' features and because we are students ourselves and therefore



(a) Machine Learning (b) Information Retrieval (c) Statistics

(d) Database (e) Gender

Fig. 2: Course insights per study

found it interesting to investigate this particular data set. This specific data set is suited for regression models, because the model is trying to predict a continuous value, in this case the mathematics score, based on other values. Regression analysis is a reliable method to identify which variables have an impact on a topic of interest. After the data was one-hot encoded, it could be used to set up a correlation matrix, which can be found in Figure 3.

The colours in the correlation matrix show the intensity of the correlation between two variables. The whiter/more black the colour, the intenser the correlation. It can be noticed from the matrix, that the target variable math score, has a very high correlation with the variables reading score and writing score. The negative correlations (-1, black) that can be seen in the correlation matrix, make sense because the student is/ has either one of both. So, if he is a male, he is definitely not a female, etc. Out of the ethnicity groups, group E has the highest positive correlation with the feature math score, which means that students from group E are often better in math than students from other ethnicity groups. Another interesting point is that a standard lunch has a positive influence on the math score, while a free or reduced lunch has a negative impact on the math results.

MODELS It was decided to run both a linear regression model and a decision tree model on the student results data set, using K-fold (with K=10) cross validation. The linear regression model was chosen because it takes features and predicts a continuous output, which seemed very suitable for the current data

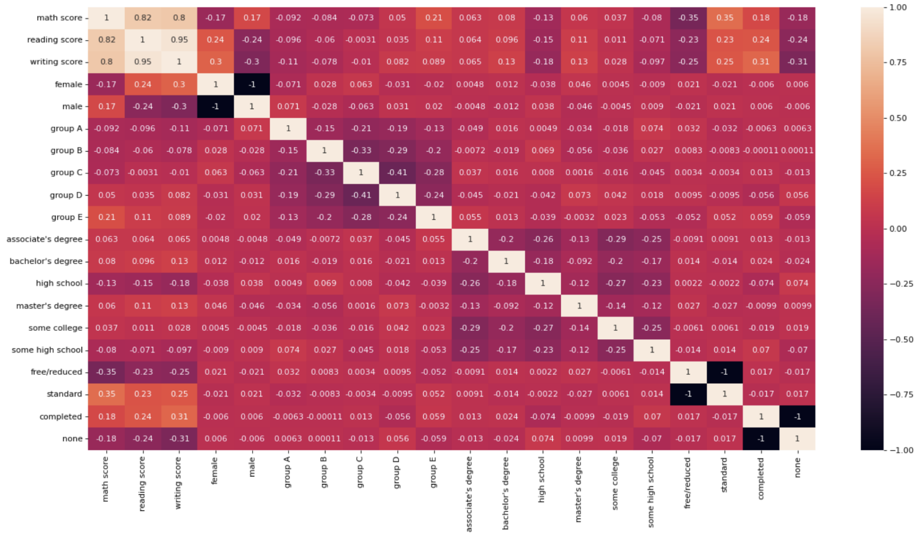


Fig. 3: Correlation Matrix student result data

set. A decision tree algorithm was chosen because it can be used to solve both regression and classification problems. With a decision tree, an inverted tree is framed which is branched off from a homogeneous probability distributed root node, to highly heterogeneous leaf nodes, for deriving the output. Regression trees are used for dependent variables with continuous values.

As stated before, K-fold cross validation is used in training the models. This means that the part of the data that is used for training (lets say 80%) is split up into k parts (as can be seen in Figure 4 below) [2]. K-1 parts are used to train the data on, and one part is used to validate the results (validation set). The performance measure that is reported by K-fold cross-validation is the average of the values that are computed. It is a very extensive method, but is very useful when dealing with small data sets, since the validation set differs every round.

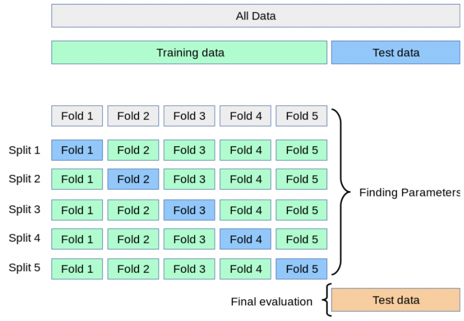


Fig. 4: K-Fold cross validation

RESULTS The results that follow by using both a linear regression model and a decision tree model can be found in Table 1.

	r^2	MAE	RMSE
Linear Regression	0.87	4.32	5.41
Decision Tree	0.70	6.65	8.24

Table 1: Results of both regression algorithms

The r^2 score is the coefficient of determination and shows the proportion of variation in the dependent variable, which is predicted from the independent variable(s). The r^2 is a well known metric used in regression problems. The RMSE is the root of the mean square of errors and the MAE is the mean of absolute value of errors. Errors are the differences between the predicted values and the actual values of a feature. More details on the MAE and MSE are explained in task 3. The r^2 shows how good the model is and thus needs to be as high as possible but since the MAE and RMSE are based on errors, the goal is to reduce those values as much as possible. As can be seen from the results in table 1, the linear regression model has the highest r^2 and the lowest MAE and RMSE. The linear regression model is better in predicting the continues target variable in this data set.

TASK 2: COMPETE IN A KAGGLE COMPETITION TO PREDICT TITANIC SURVIVAL

TASK 2A: PREPARATION

After importing the train data, we noticed it consists of 891 passengers of Titanic each having 12 features. These are enumerated next to their type as follows: "PassengerId" (int64), "Survived" (int64), "Pclass" (int64), "Name" (object), "Sex" (object), "Age" (float64), "SibSp" (int64), "Parch" (int64), "Ticket" (object), "Fare" (float64), "Cabin" (object), and "Embarked" (object). Most of their names are self-explanatory, however, there is "Pclass" which is the ticket class, "SibSp" which indicates the number of siblings/spouses aboard the boat, and "Parch" which specifies the number of parents/children of a passenger.

Concerning the types of the features, one could interpret the "object" type as a string of characters. Accordingly, we noticed that the "Sex" feature could be mapped to a binary, int64 feature, instead of keeping "male"/"female" entries, in order to prepare the variable for a Machine Learning model. Additionally, the "Age" is surprisingly of float type, despite one's expectations of being an integer. Consulting the documentation [6], it was found that the age of a person is fractional when it is less than 1 or when it is estimated. Other than these, the other features have the anticipated types.

In figure 5, the distributions of the numeric features are visible. Starting with the "Age", we see that the distribution is different from an anticipated normal distribution, due to the lack of teenage passengers. Next, we observe the class imbalance in ”Pclass”, ”Sex”, and ”Survived”, with the majority of the people of the boat in the third class, male (encoded 0), and did not survived. Finally, the variables ”Fare”, ”Parch”, and ”SibSp”, roughly resemble an exponential distribution, with the counts decreasing drastically as the value of the respective feature increases incrementally.

Accordingly, it is fair to anticipate the members of the two groups to correlate against the other members in the group. Moreover, we also expected to see the pairs ”Sex”-”Survived” and ”Age”-”Survived” highly correlated since it is well known that women and children were prioritised for lifeboats, as well as the ”Fare” correlated with the ”Pclass”. Looking at correlation heatmap (Fig. 6), we can see that the first and last pairs are indeed correlated, scoring a coefficient of 0.54 and -0,69, respectively, while the other pair, ”Age”-”Survived”, is not (-0,05). Furthermore, the variables ”Fare”, ”Parch”, and ”SibSp” are correlated with each other, achieving a correlation coefficient of over 0,4.

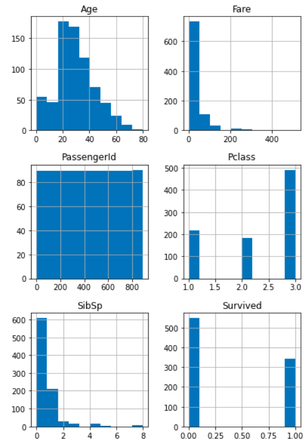


Fig. 5: Numeric features distribution

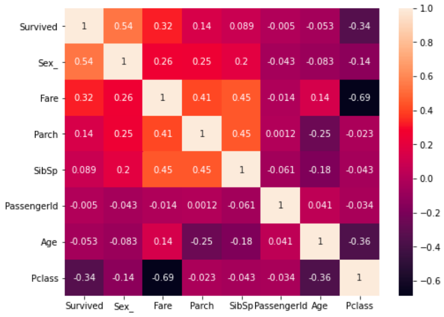


Fig. 6: Correlation heatmap

As previously stated, it is well known that the children were prioritised for lifeboats, however, the age does not correlate with survival. For this reason, a new binary variable was derived based on the age, to represent whether the passenger was a minor. This was done with the aim of helping the model reach a higher prediction performance. Next, we noticed that each person aboard has a title in their name, such as ”Mr”, ”Miss”, or ”Master” and we extracted this in a separate categorical feature because we thought some important persons might have been sheltered. Another feature was derived from the fare of the ticket, bucketing together free ($0), cheap($0-$50), medium($50-$100), and expensive ticket (over $100) holders. Finally, two more variables were constructed from ”Cabin”, namely the letter and the digit of the cabin. Since certain groups of cabins were closer to the lifeboats, it is fair to assume that their ticket holders had a better chance of survival.

Prior to modeling, the missing entries were inspected and filled with "-999" because this value is suitable for all the data types present in the data. Then, the features which were deemed to have small power over survival prediction and an unsuitable type for a learning algorithm, such as "Name" and "Ticket", were removed. Finally, the other categorical features were one-hot encoded to represent the data in a way that allows the algorithm to learn from it.

TASK 2B: CLASSIFICATION AND EVALUATION

Before designing any model, 20% of the train data was held out for testing purposes, while the rest was used for training. Regarding the fitting procedure, two different strategies were employed, one where the default settings were used and a model was fit on the whole data, and another where the hyperparameters were tuned using Grid Search with 5-fold Cross-Validation.

When it comes to the selected classifiers, we decided a gradient boosting tree model is suitable for this task and likely to have a high prediction power, reflected in a high position in the ranking. Among these types of models, CatBoost [1] was picked because it is generally agreed that it has an advantage in terms of performance against other boosting algorithms [4, 7]. In order to assess the performance and benchmark the model, two simpler tree models were designed as the baseline, namely a Decision Tree and a Random Forest.

Subsequently, the accuracy of the three models computed on the test set, for both fitting strategies can be found in table 2. As expected, CatBoost performs the best, followed by Random Forest, and then by Decision Tree, and the tuning helps every model reach better accuracy.

Model	Default setting	Hyperparameter tuned
Decision Tree	0.77	0.80
Random Forest	0.81	0.83
CatBoost	0.83	0.84

Table 2: Benchmark tree models

As the CatBoost model with tuned hyperparameters reached the highest performance on the held-out set, it was selected as the final model for participating in the competition. Before predicting the target feature for the test set given by Kaggle, the CatBoost model with the best hyperparameters resulted after Grid Search was fitted another time, this time on the whole train data, including the held-out set. This gives the model an increased chance of learning the patterns in the data, especially given the limited size of the train data of only 891 records.

Lastly, the predictions on the test set were submitted on Kaggle and they scored an accuracy of 79,4%, suggesting a slight overfit on the train data as the score is larger than the one achieved during the tryout. The score places the team in the top 5% of the leaderboard, which was anticipated because the prediction power of a gradient boosting model is stronger than simpler algorithms, commonly used in a "Getting Started Prediction Competition", but lower than state-of-the-art Deep Learning Frameworks. Besides, since the competition is largely popular, the true values of the test data were discovered, hence the numerous submissions at the top of the ranking reaching a perfect accuracy.

TASK 3: RESEARCH AND THEORY

TASK 3A: RESEARCH - STATE OF THE ART SOLUTIONS

We investigate the Kaggle competition *Personalize Expedia Hotel Searches - ICDM 2013*, on which the second assignment is based. The data consists of information about Expedia search queries from users to find a hotel, properties of the shown hotels, and for the training data whether the user clicked on it and booked. The competition is a ranking task and the evaluation metric is called Normalized Discounted Cumulative Gain (NDCG). The winner of the competition is called Owen Zhang, who gave a short presentation on his solution which is used as the basis for this research [5]. In the following paragraphs we elaborate on its content.

An important technique used is extensive feature engineering. Although other teams do this as well, the extent to which this is done matters. However, as this does not stand out, we won't dive into the details. To train the model, an ensemble of gradient boosting machines (GBMs) is taken using the NDCG loss function. Two models are considered, one with all engineered features and one with part of it. In total, 26 separate modes were trained, each taking 20-30 hours on machine with 256GB RAM and 200GB swap space. Next, a weighted average is taken as the final result. Most participants try out many different techniques, whereas Owen has put significant effort and training time into one particular technique, GBMs, and optimized for this. The decision for GBMs makes sense as they have proven to be effective for ranking tasks. Aside from good preprocessing and feature engineering, we believe the focus on one particular suitable technique in combination with impressive hardware and training time was decisive to win.

TASK 3B: THEORY - MSE VERSUS MAE

The formulae for MSE and MAE are as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \qquad (1) \qquad\qquad MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \qquad (2)$$

The clearest difference between the two is the fact that the MSE puts significantly more weight on outliers than the MAE. Namely, each difference is squared in (1), whereas in (2) the square root is taken before summation, since $|y_i - \hat{y}| = \sqrt{(y_i - \hat{y})^2}$. As a consequence, the predicted values are pulled towards the outliers more in MSE. If the goal is to have a small error for most estimations, MAE is better as it minimizes the sum of all absolute errors. If the goal is to reduce large errors, MSE is more suited.

Since the MSE and MAE ensure the average of all predictions is close to the mean and median of the predicted value, respectively, identical results are expected when the data is distributed symmetrically. Namely, in that case the mean is equal to the median. This is because (large) under estimations are cancelled out by (large) over estimations.

In case of a skewed distribution different results are expected. To see this clearly expressed in a difference between the MSE and MAE, we need to estimate a numeric variable. Therefore, we look at the distribution of house prices [3]. House price is a continuous numeric variable which is known to be skewed. Its thick positive tail generally causes the mean to be significantly higher than the median. Removing house price outliers, that is below percentile(25)-1.5*IQR and above percentile(75)+1.5*IQR, is therefore expected to delete points especially on the upper percentiles.

We train a simple linear regression and random forest (with 2 estimators) on 70 percent of the data, with *Above ground living area (square feet)* to predict the sales price of a house. The MSE and MAE are calculated on the remaining 30 percent. Subsequently, we remove the outliers from the train data, rebuild the models and calculate the MSE and MAE again. The result: for the linear regression a 14% drop is observed for the MSE and 5% drop for the MAE. For the random forest a 21% and 3% drop are observed, respectively. As expected, after outlier removal, the relative drop in MSE is much larger than that of MAE. We do note that these result might change for a different number of estimators and/or train-test split.

TASK 3C: THEORY - ANALYZE A LESS OBVIOUS DATASET

Many techniques can be applied on text - the best performing ones involving deep learning. Over the past 10 years this field has improved rapidly, where techniques such as RNNs, LSTMs and Transformers have shown strong results. However, for the purpose of this assignment these techniques are not considered, as they generally require a large set of training data and significant effort to build. Our data set consists of 5572 data points, so we stick with conventional machine learning techniques. The average length of a text in the data is 80 characters, the equivalent of one long sentence. Little information is available about the macro structure of the text, such as how sentences/paragraphs relate. Therefore we rule out modeling techniques based on this, so each text can be considered as one bag of words.

Regardless of the model, each statistical technique requires quantified data, i.e. scalars, vectors, etc. A common technique applied in NLP is to vectorize words. One-hot-encoding or embedding are techniques to achieve this. Drawbacks of the former are that each word is independent of all other words, and we end up with a very large n-dimensional space, where the the curse of dimensionality can cause difficulties. The latter creates a latent space with a fixed number of dimensions d, where d is generally 1 or multiple orders of magnitude smaller than n. Each word corresponds to a vector in this space, called the word embedding. A popular technique to achieve this is called *word2vec*, which uses either the continuous bag-of-words or skip-gram model [8]. However, embedding techniques tend to require a large set of training data. Given the limited size of our data, we focus our effort on the former.

To not solely focus on single words, a little information about consecutive word occurrences can be added in the form of (word) bigrams. A bigram assigns a one-hot-encoding for each combination of two words following each other. However, as the number of possible combinations quickly explodes using this technique, usually only the most frequent combinations are included in the one-hot-encoding. In our case, the 5000 most frequently occurring words/bigrams are taken as features. Lastly, to add extra information into the one-hot-encoded we apply TF-IDF, which stands for Term Frequency Inverse Document Frequency. It adds information on the originality of a word, in our case, in a specific text message. This is done by multiplying the number of occurrences of the word in that text message (term frequency) by some inverse of the number of occurrences of that word in all other text messages in the corpus (inverse document frequency) [9].

Before discussing the technique applied on the resulting vectors, we still need to discuss how to transform the raw text into refined text (bag of words) fed into the one-hot-encoding. This is usually referred to as NLP preprocessing, and encompasses several steps. The first step is tokenization, which splits up the sentence into a list of words. A common second step is lemmatization, which aims to reduce multiple different forms of one word such as am, was, is, into one word: be. An alternative to lemmatization is stemming. The main goal of this step is to reduce n and improve the quality of the relations observed between words. Given the limited number of data points, we omit this step. Lastly, we remove stopwords such as *a, an, its*, in a text occur frequently and contain little information. Therefore, these are removed from the list of words. A common step following this is Part Of Speech tagging, which adds e.g. the label *noun* to the word *book*. Given the limited structure and accuracy of words in text messages, we omit this step.

The technique we apply on the resulting vectors is called Naive Bayes. It assumes that occurrences of words in sentences are uncorrelated to other words. Although this is most likely not the case in reality, a big advantage is that (multiple) suspicious words can strongly impact the probability for a spam classification, whereas non-indicative words are expected to not impact the probability by much. The resulting accuracy of our model 97.85%, with confusion matrix: 1433 TPs, 34 FP, 2 FN, 203 TN.

Although these results are quite promising, there's still a strong bias towards False Positives in comparison to False Negatives. Balancing these two will help in improving accuracy. This could be achieved by adding small extra steps, for example by using better techniques to do the tokenization, implementing the lemmatization step, perhaps involving trigrams in the analysis, Part of Speech tagging, word embedding instead of one-hot-encoding, and further feature engineering such as length of a text. Of course, different techniques such as random forests or deep learning models might yield better results as well.

References

1. Catboost package. https://catboost.ai/, accessed: 21-04-2022
2. Cross-validation module. https://scikit-learn.org/stable/modules/cross_validation.html, accessed: 21-04-2022
3. House prices advanced regression techniques - data. https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data, accessed: 21-04-2022
4. How to choose between different boosting algorithms: Adaboost, gradient boosting, xgboost, light gbm, catboost. https://towardsdatascience.com/how-to-select-between-boosting-algorithm-e8d1b15924f7, accessed: 21-04-2022
5. Personalized expedia hotel searches – 1st place. ICDM 2013 – Owen Zhang
6. Titanic - machine learning from disaster — data description. https://www.kaggle.com/competitions/titanic/data, accessed: 21-04-2022
7. When to choose catboost over xgboost or lightgbm. https://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm, accessed: 21-04-2022
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013). https://doi.org/10.48550/ARXIV.1301.3781, https://arxiv.org/abs/1301.3781
9. Rajaraman, A., Ullman, J.D.: Data Mining, p. 1–17. Cambridge University Press (2011). https://doi.org/10.1017/CBO9781139058452.002