Data Mining Techniques 1 - Group 29

Caterina Buranelli (2718588), Ignas Krikštaponis (2718072), and Charlotte Felius (2648506)

Vrije Universiteit Amsterdam
https://www.vu.nl/nl/index.aspx

1 Introduction

This assignment consists of three tasks.

For task 1, exploratory analysis was performed on an ODI dataset, after data was thoroughly cleaned; univariate analysis provided summaries on the single variables while bivariate analysis was meant to investigate interesting associations among variables. The association rule method was also used in this part of the report. For the second part of the task, a dataset describing Spotify songs was used to compare two regression models: Linear Regression and Random Forest.

Task 2 involved participating in a classification competition on Kaggle - *Titanic - Machine Learning from Disaster*. Four models were used and compared: Random Forest, Naive Bayes, Logistic Regression and K-NN.

Task 3 was theoretical; it involved an analysis of a winning solution to a Kaggle competition, a description of the difference between two performance measures (Mean Absolute Error and Mean Square Error) and an example of text mining.

2 Exploring Small Datasets

2.1 Exploration

Pre-processing The quality of the initial data was poor, especially for the questions that allowed answers as strings; the way these questions were processed is explained in this paragraph.

Answers to "date of birth" were kept only when they included day, month and year, then the variable was saved in the format "dd-mm-yyyy"; the first conversion was made with the help of the package `lubridate` and a second check was made by hand. Answers to "bed time" were treated similarly as the previous variable; the format used was "H M S". For the variables *"stress level"* and *"random number"*, answers that respected the boundaries (respectively, 1-100 and 1-10) were considered valid. It was checked if all the values of the variable *"gain"* were expressed in numbers and, when written as string, they were converted to float, by hand; the only boundary set was to be non-negative. It was checked if the number of neighbours was a non negative integer. Study program column was first converted to all lower case and then grouped. Biggest groups were then identified and remaining study program names were aligned to the biggest groups e.g. ”Fintech” → “Finance and Technology”. Some grammatical mistakes in the program names were also accounted for.

Answers that did not follow the above criteria, were set as `NA`. After deleting duplicated records, the final dataset contains 307 rows and 17 columns, including the timestamp.

Exploratory analysis Categorical and binary variables were investigated with pie graphs and percentages. For instance it was found that 29% of the units were following the AI Master programme; then another 30% was equally split between BA, CLS and Bioinformatics. For discrete variables as *“stress”* and *“rn”* (random number) first a univariate then a bivariate analysis was performed. Their distribution was investigated through histograms, that seemed to show a Uniform shape, but a Kolmogorov-Smirnov test discarded the idea (p-value < 0.01 in both cases). Secondly, they were analysed in function of categorical variables like *“gender”, “ML”, “Stats”, “IR”, “DB”, “chocolate”*; with the mean of boxplots, t-tests and ANOVAs. The only significant difference was found in the ANOVA for *“rn”* given *“gender”* and the respective boxplot is reported in Figure 1a. As the three distributions were not statistically Normal, a non parametric test was thought to be more reliable; the Kolmogorov-Smirnov test was performed and confirmed the significance of the ANOVA (p-value = 0.0054). A pairwise Wilcoxon test with Bonferroni correction showed that there is significant difference for the pair female-male (p-value =0.0072), while female-unknown and male-unknow were not significant (p-value > 0.05), probably due to the fact that “unknown” has 8 statistical unities only.

Correlations were explored between *“stress”*, *“gain”* and *“rn”*, however no significant linear relationships were identified.

Weekdays of birthdays were extract for analysis. It was found that Tuesday is the most common (19.40%) birthday while Sunday is the rarest (10.78%). By splitting the data according to gender we observed that among females Wednesday was the most common birthday, while for males it was Tuesday. Lastly, we saw that 45 people shared their birthday with at least one other person. Maximum of 3 people shared the same birthday.

Rule associations The variable `stress` was categorized in low (0-33), medium (34-66) and high (67-100), with respectively 92, 80 and 87 statistical units for each class; the dependence between the factorial stress and the attendance (or not) to four data-mining-related courses (Machine Learning (ML), Statistics (Stat), Database (DB) and Information retrieval (IR)) was investigated through the association rules method. Apriori algorithm was used to identified these rules - the minimal support was set at 0.01 and the minimal confidence at 0.5. In the graph in Figure 1b, the items are the labelled vertices while rules are represented as vertices connected to items using arrows. The LHS items are set as default (the attendance or not to the four courses) and are connected with arrows pointing to the vertex representing the rule; the RHS here is circumscribed to the value of the stress level and has an arrow pointing to the item. A higher size of the circles implies a higher confidence of the rule; the darker the color of the circle, the higher the lift. Overall, the graph is meant to answer the following question: *What combination of courses that the student did and did not attend results in a certain stress level.* [10]

From the graph we observe that for the rule `ML=no,IR=no,DB=no,Stat=no`, the confidence of having a low stress level is 0.75, and it drops to 0.6 when excluding IR from the rule. The rule `ML=no,IR=yes,DB=no,Stat=yes` results in a high stress level with confidence 0.6. The rule `ML=yes,DB=no,Stat=no` has confidence 0.54 for a high stress level. All the other rules can be extracted from the graph as touched upon above and their confidence is 0.5.

However, when performing a linear regression on the stress level, the four courses were not significant predictors.

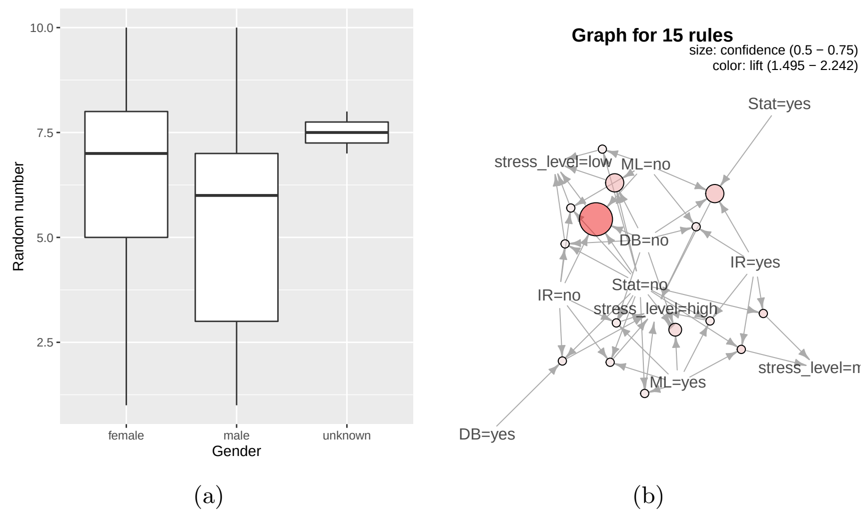

Fig. 1: (a) Boxplot of *"rn"* given *"gender"*. (b) Graph representing the rules for three levels of stress: low, medium, high.

2.2 Regression - Spotify songs

A Spotify data set [11] was selected for performing simple regressions. It is a suitable data set as it possesses a continues numerical target variable - *'track_popularity'* (range 0-100) - that allow us to apply regression algorithms in order to predict song's popularity based on its' characteristics: *'playlist_genre', 'playlist_subgenre', 'danceability', 'energy', 'key', 'loudness', 'mode', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'duration_ms'*. *'track_popularity'* is calculated by Spotify's algorithm and is mostly based on the amount of plays a song has received. The data posses 32833 rows.

Regressions Two regression models will be compared - Linear Regression (LR) and Random Forest Regression (RFR). LR algorithm linearly combines weighted explanatory variables to produce predictions. It uses the ordinary least squares (OLS) approach to optimise the weights. On the other hand, RFR uses multiple randomly generated decision trees to generate predictions based on average individual tree outcomes [9]. While LR works well only if there exists linear relationships between predictors and the target variable [15], RFR is applicable to non-linearly dependant variables [9].

Data pre-processing To allow the data to be applicable to both regression models we encoded the categorical variables with one-hot encoding. The data was divided into train-test sections with 67/33% split. Numerical data was standardised to account for range differences in the numerical attributes. This was done separately for training and validation data to prevent data leakage.

Model hyper-parameters Default hyper-parameter settings of `sklearn` [12] package were used for both models. For LR the hyper parameter of `fit_intercept` is set to `True` - as the OLS decides on the appropriate size of the intercept, forcing the intercept to 0 should only be done if required by design. RFR possesses > 10 hyper-parameters. `n_estimators` sets the amount of trees that a regression decision will be made on, it is set to 100. Another important parameter is `criterion` on which the tree chooses the split quality - the algorithm makes multiple splits and then chooses the split with the best criterion. By default it is set to `mse` (mean squared error).

Cross-validation 10-fold cross validation was performed to compare the performance of the models. The amount of folds gives us $32833 \times 0.67 \times 0.1 \approx 2200$ rows for each fold and will allow us to perform a statistical test between the two algorithms. It was found that average absolute mean error (AME) for LR model was 20.10 and for RFR it was 17.80. A t-test concluded that these results were significantly different, however RFR perform better than LR by only 11.44%.

Discussion Overall, both algorithms were able to produce predictions that were on average 20 popularity points off-the-target on the training set. However, upon further investigation of the data we can see that both algorithms completely failed to predict for songs that have popularity close to 0. Albeit the test data had around 9% of rows with popularity < 1, the predicted values only had 0.07%. Intuitively, to improve the model further we could include such predictors as the song artist and time period since the song release.

3 Predicting Titanic Survival

For the second task the Titanic data set provided by Kaggle [3] will be examined and classified by using Python3 and particularly the machine learning package `scikit-learn` [12].

3.1 Preparation

Description of the Data. The Titanic data set consists of 12 columns entailing PassengerId, Survived , Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin and Embarked and the data set consists of 891 and 428 rows for the training and test set respectively. PassengerID represents the ID of the passenger and thus corresponds to a unique Name. Pclass, Ticket, Cabin, and Embarked are all nominal values while Sex is a binary value. Age, Fare, SibSp and Parch are numerical and Survived is the target value; a binary value that indicates whether this particular passenger has survived (1) or passed away (0).

Missing Data and Distributions Remarkably, cabin has 687 missing values. In addition, in the training set "Age" and "Embarked" have 177 and 2 missing values respectively. The distribution of Age is rather normal, although if we zoom in on the age distribution of the people who survived we can conclude that very young people were more likely to survive. For people that did not survive, the age distribution has a peak around 28 years old. This indicates that children tend to survive and adults were more likely to decease, therefore we make a variable to indicate whether a person is a child, as in [2]. In this extent, people that traveled with siblings or parents encompass roughly 40% of the survivors. Moreover, women encompass 68% of all survivors while only 35% of the original training set is women and thus we expect that gender is a robust predictor for survival.

Data Cleaning, Feature Selection & Transformations To have an optimal outcome for the classification algorithm, not all variables are taken into account. 'PassengerID' and 'Name' are deleted, as they are unique for every passenger and therefore cannot be used for prediction. Following, Cabin and Ticket are omitted as they have either $> 75\%$ missing or unique values. Age too has a significant amount of missing values (177), and by analyzing these missing values independently it turns out that most of the corresponding passengers did not survive ($\sim 70\%$). Therefore, a new artificial feature `no_age` is created that indicates whether the passenger has a missing age. In addition, the missing values of Age are replaced by 28 which is the median of the age of people that deceased. The amount of missing values from `Embarked` are very low but this feature is deleted as it is not expected to give any indication of survival, but a new feature is constructed from `SibSp` and `Parch` to indicate whether a person was traveling with family. Furthermore, to enable the machine learning algorithms to properly learn from the training set and to classify the test set, we use one-hot encoding to transform nominal values to numerical values. `Pclass` and `Fare` are highly correlated, therefore these features are combined into a single feature consisting of three values: 2 (high-class and expensive), 1 (high-class or expensive) and 0 (low-class and cheap).

The correlation of the most influential remaining attributes with the target value is depicted in table 1. According to the correlations, we expect that particularly `Female` (Sex), `High_class` and `Child` have a significant influence on the prediction of survival.

Table 1: Correlations of features

	Survived	**no_age**	**Child**	**High_class**	**Female**	**Family**
Survived	1.000000	-0.092197	0.121485	0.344767	0.543351	0.016639
no_age	-0.092197	1.000000	-0.177031	-0.250068	-0.055215	-0.049043
Child	0.121485	-0.177031	1.000000	-0.097145	0.102403	0.398080
High_class	0.344767	-0.250068	-0.097145	1.000000	0.157615	-0.056129
Female	0.543351	-0.055215	0.102403	0.157615	1.000000	0.200988
Family	0.016639	-0.049043	0.398080	-0.056129	0.200988	1.000000

3.2 Classification & Evaluation

The training set includes the target variable, "Survived" and is provided on the Titanic competition page on Kaggle [4]. Since we need to predict the target value, we separate the target value from the rest of the data set to avoid that the ML algorithm takes this variable into account as a predictor. To evaluate how the machine learning (ML) algorithm performs, we use K-fold cross-validation which is provided by the Scikit-learn package in Python [12]. K-fold cross-validation splits the training set into K folds, trains the ML algorithm on K-1 folds and uses the remaining fold for testing. This process is repeated K times so that every fold is used as a test set once, which gives a more accurate estimate of the total (mean) accuracy of the performance of the ML algorithm. The amount of folds (K) we use is 10, and the different machine learning algorithms that we train on the training set are "Random Forest", "Naive Bayes", "Logistic Regression" and "K Nearest Neighbors".

After applying all preprocessing steps mentioned in section 3.1 and using K-fold cross validation, we obtain the accuracy scores depicted in Table 2.

Table 2: Accuracy and standard deviation of machine learning classifiers

	min	**max**	**mean**	**st. dev**
Random Forest	0.780899	0.848315	0.814795	0.023438
Naive Bayes	0.707865	0.825843	0.760988	0.043887
Logistic Regression	0.758427	0.843575	0.810288	0.028495
KNN	0.780899	0.842697	0.813684	0.019982

K-Nearest Neighbours and Random Forest perform almost equally good and do not appear to be significantly different using a t-test (p-value of 0.88). However, since the minimum, maximum and mean of Random Forest are slightly higher we use Random Forest to predict the values of the test set posted on Kaggle. Note that the preparatory steps performed on the training set in (3.1) are also deployed on the test set provided by Kaggle. Using our trained Random Forest classifier, we obtain a `final accuracy of` 78.5%. This is somewhat lower than the average accuracy on the training set (81%), but in line with our expectations since the test set on Kaggle is much larger (> 400) compared to the size of the test folds (~ 80) used in the training set.

4 Research and Theory

In this chapter we examine different sorts of data mining applications. Firstly, the winning solution of a finished datathon on Kaggle [3] is analyzed. Secondly, to go more deeper into the theory, the application of the MSE is compared to the MAE within data mining. Lastly, an alternative data set is analyzed.

4.1 3A WIDS 2020: Finished competition

Competition description. We elaborate upon the Women in Data Science Datathon [5], which lasted for two months and was due in February 2020. The Datathon stands out since it requires 50% of each team to be women and it focuses on the health of patients. Its focal goal is to predict the survival of a patient after their first 24 hours in the hospital Intensive Care Unit (ICU). The training set for the Datathon consist of 183 columns of particularly medical data. The data set is provided by MIT's GOSSIS community initiative [1] and is constructed by data from over 200 hospitals across the United States. The focal goal of this competition is to predict the survival of a patient after their first 24 hours in the hospital Intensive Care Unit (ICU), given $\sim$ 131000 rows where each row represent a visit of a patient and its corresponding medical or descriptive values. The predicted survival of the patient is depicted in the column `hospital_death`, where 1 corresponds to death and 0 corresponds to survival. Submissions are evaluated under the ROC (Receiver Operating Characteristic) curve between the predicted and observed deaths [5].

Solutions of the WIDS Datathon 2020. The competition was won by a team named "WomenPower" who achieved an accuracy of 0.91. They described their solution on [6], and stated that their pipeline consisted of: 1) Exploratory Data Analysis, 2) Imputation, 3) Extraction, 4) Selection, 5) Modeling and 6) Ensemble. In 1) they examined features that had a medical reason to be correlated with the target variable, "hospital_death". Subsequently, they imputed missing variables of features by using frequency distributions, medians, iterative imputations and they used machine learning for imputing some values. Following, they created a feature `mostly_dead` which indicates whether the patient 'should have been dead' according to the corresponding medical values of this patient. For 4) they used a combinations of different methods to smartly select features. These 5 methods entail examining collinearity, standard deviation, removing less important features by applying `Random Forest` (RF), `ELI5` (Explain Like I am 5) for removing non-important features and `Recursive Feature Elimination` (RFE) for the 15 best ranked features. In the modeling phase they used `Boosting` (XGboost, LightGBM, Catboost), `Scikit` (Random Forest) and `Pytorch/Tensorflow` for Deep Learning. Lastly, they used 3 different methods for model ensembles, from which they highlight a novel technique; `Test Time Augmentation` (TTA).

What made the winners stand out? The winners explicitly mentioned that one of the main challenges was dealing with the large amount of columns and their corresponding of missing, redundant or non-useful data. Therefore, we infer that this winning approach particularly stands out not because of the modeling part, but because the winners put an extreme amount of effort in the sophisticated feature engineering. The ensemble models most likely too contributed to their win, but the focus on imputation, extraction and selection combined with general medical industry knowledge that enabled the logical reasoning behind the construction of decisive (artificial) features such as `mostly_dead` and the elimination of other variables, was particularly what `WomenPower` made have stood out. Namely, with complicated and huge data sets like these, the way features are interpreted and handled can be more decisive then the actual ML model that is trained on the data.

4.2 3B MSE vs MAE

Given that $\epsilon_i = y_i - \widehat{y}_i$, the mean squared error (MSE) and mean absolute error (MAE) are calculated as follows:

$$MSE = \frac{\sum_{i=1}^{n} (\epsilon_i)^2}{n} = Var(\epsilon)$$
$$MAE = \frac{\sum_{i=1}^{n} |\epsilon_i|}{n}$$

The difference between these two error measures lays in the fact the MSE minimizes the mean squared error between predicted and observed values, while the MAE minimizes their median. The MSE can be also found as root mean squared error ($RMSE$), to provide the same dimensions of the predicted data [15].

The mean, unlike the median, is a consistent and unbiased estimator ($E(\hat{\theta}) = \theta$), and due to the Central Limit Theorem (CLT) [13], it can be approximated to a Normal distribution, which gives the possibility of building confidence intervals and quantify the error of the estimate (α, usually set to 5%). On the other hand, the mean is sensible to outliers and skewnesses, so when these features are present in the data, the median is a better tool, as it is a robust estimator: all errors are considered equally accordingly to their frequence [15]. Although these premises seem to make very clear whether to use one performance metric instead of the other, the debate on this topic is still vibrant. For instance, the MSE is the most commonly used one, on the other hand some studies considered it not to be a reliable error measure [14],[7], specially when dealing with time series [8].

MSE and MAE coincide when $\sum_{i=1}^{n} (\epsilon_i)^2 = \sum_{i=1}^{n} |\epsilon_i|$; this is possible only when the absolute values are the same for all the predictions, which is something that hardly ever occurs with real data.

Anthropometric data are often used for their good approximation to a Normal distribution. From the website https://www.kaggle.com/mustafaali96/weight-height, data of height (Y), gender (X_1) and weight (X_2) of 10000 statistical unities were taken to fit the following regression model:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \quad \text{with } \epsilon \sim N(0,1)$$

Despite median and mean of Y are statistically the same ($E[Y] = 66.37$ with $[58.82, 73.90]$ CI 95% and $M_Y = 66.31$), the errors are different, with $MSE = 2.06$ and $MAE = 1.14$. In this case MSE is the correct error measure, as data is Normally distributed and do not present outliers or asymmetries. [7]

4.3 3C Analyze a less obvious dataset

Data exploration The data provided in the `SmsCollection.csv` contains plain text messages along with classification whether the message is ham or spam. Since we have categorised data, classification is the best modelling technique to apply here. To gain a better idea of the data we analysed the most commonly used words in both categories. Data was cleaned by lower-casing the text and omitting punctuation along with numbers. The most common word (longer than 2 letters) for the spam group is "call" while for ham it is "you". For the ham group most common words are either pronouns or prepositions, on the other hand the top 20 words in spam category tend to create a sense of urgency with words such as "call", "txt", "claim". They also tend to lead to websites as "www" is among the top words. Among 5574 messages, 13.40% are label as "spam".

Data transformation Tokenization will be applied as a data transformer. This technique works by transforming the texts into a word frequency matrix. We used `CountVectorizer` from `sklearn` package to achieve this [12]. For this analysis we tried two different feature selection method - the first data set did not have any omitted words and the second data set had the most common English stop words and words that had a frequency of less than 0.1% removed. The full data set had 7822 features, while the reduced data set had 1301.

Model Naive Bayes model will be applied to these data sets. Naive Bayes model is a commonly applied model for spam classification and text analysis in general. The algorithm is based on the principle of conditional probabilities as laid out by Bayes Theorem. Multinomial Naive Bayes was selected here as it is the most suitable type of algorithm for data with frequency counts [9].

Performance analysis The data was divided into train-test sections with 67/33% split. 10-fold cross validation was performed on the two models. Average accuracy for the full data set was 97.54% while for the reduced data set it was 97.86%. A t-test confirmed that this difference is insignificant (p-value > 0.05). It is interesting to see that the two models managed to perform equally well even if they had significantly different dimensionality. This could be attributed to the fact that removing rarely occurring words does not results in significant information loss as those words are too specific to a particular message. Furthermore, the removal of stop words did not impact performance because these words were common between both spam and ham messages. The final accuracy score on the test set was 97.61% for full and 98.21% for reduced model. It is also useful to look at the confusion matrix (Table 3). We can see that the full model classified 5.71% of spam messages as ham, this increased to 7.35% for the reduced model.

Table 3: Confusion matrices

(a) Full model

		Predicted	
		Ham	Spam
Actual	Ham	1565	30
	Spam	14	231

(b) Reduced model

		Predicted	
		Ham	Spam
Actual	Ham	1580	15
	Spam	18	227

Improvements to the model The model could be further improved by creating extra features from the data, e.g. length of the message, however having continues variables would require a different model choice. More sophisticated transformations could also be applied here - e.g. lemmatization or stemming. This could further decrease the dimensionality of the data, while not losing important information, thus cutting down on the processing time of the model.

References

1. Global open source severity of illness score. https://gossis.mit.edu/
2. How to achieve more than 98 percent of accuracy on titanic dataset. https://medium.com/analytics-vidhya/how-to-achieve-more-than-98-of-accuracy-on-titanic-dataset-87241c18161a
3. Kaggle. https://www.kaggle.com
4. Titanic - machine learning from disaster. https://www.kaggle.com/c/titanic/
5. Wids datathon 2020. https://www.kaggle.com/c/widsdatathon2020
6. Wids datathon 2020 - solutions winning team. https://www.kaggle.com/c/widsdatathon2020/discussion/133189
7. Chai, T., Draxler, R.: Root mean square error (rmse) or mean absolute error (mae)? Geosci. Model Dev. **7** (01 2014). https://doi.org/10.5194/gmdd-7-1525-2014

8. Collopy, F., Armstrong, J.: Another error measure for selection of the best forecasting method: The unbiased absolute percentage error (10 2011)
9. Flach, P.: Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press (2012). https://doi.org/10.1017/CBO9780511973000
10. Hahsler, M., Grün, B., Hornik, K.: arules - a computational environment for mining association rules and frequent item sets. Journal of Statistical Software, Articles **14**(15), 1–25 (2005). https://doi.org/10.18637/jss.v014.i15, https://www.jstatsoft.org/v014/i15
11. Mock, T.: Spotify songs - tidy tuesday. bit.ly/32xZX5r (2020)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
13. Wanyonyi, S., Marangu, D.: Central limit theorem and its applications in determining shoe sizes of university students. Asian-European Journal of Mathematics **3**, 1–9 (02 2019). https://doi.org/10.9734/AJPAS/2019/v3i130082
14. Willmott, C., Robeson, S., Matsuura, K.: A refined index of model performance. International Journal of Climatology **32** (11 2012). https://doi.org/10.1002/joc.2419
15. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edn. (2011)