Data Mining Techniques - Assignment 1 - Basic

Group 190 - Caroline Hallmann, 2640914, Aneesh Makala 2730226, Nader Sobhi 2702248

Vrije Universiteit Amsterdam

1 Introduction

In this report, we perform various tasks to gather an introductory understanding of data mining techniques. In the the first task, we perform exploratory data analysis on a small custom-built dataset, which is an essential prerequisite when working with data. Then, we move on to work with a real world dataset from the Titanic ship. We explore different classification models and test against the public leaderboard on Kaggle. Finally, we dive a little deeper into theoretical aspects by studying state of the art solutions, evaluation metrics, and techniques for text mining.

2 Task 1: Exploring a small dataset

2.1 Exploration

The ODI-2022.csv dataset consisted of the collected student survey responses during the first 2022 DMT lecture. To explore the data it was imported and run via RapidMiner. After importing the file, the properties of the raw dataset were investigated and cleaned. The dataset comprised of 304 total records, 303 real records, 17 variables and 16 attributes. The data was distributed over 16 questions. The data was cleaned by using the RapidMiner attribute function. The first row "Tijdstempel" was insignificant and removed. The attributes of the cleaned dataset are shown in Table 1. During the cleaning process, the open-questions were difficult to filter and analyse. Stress level generated for both numeric and nominal answers. When filtering the data, outliers were detected for float types (Stress level and euro question) like the stress level answer "over 900". Other answers had no relevance to the given question and were removed, producing few missing data points. The cleaned data was then analysed with statistical models and descriptive visual representations. The study programme attribute was filtered and outliers were removed. A pie chart was created to identify the programme distribution of students enrolled in the class. Some answers were written differently but had the same meaning (i.e., Artificial Intelligence/AI or Computer Science/CS). Therefore, values of the same meaning were grouped into one category accordingly. Due to the large number of enrolled students in the course, a diverse set of study programs were found (Fig. 1). The pie chart shows that majority of students are enrolled in the study programmes Artificial Intelligence, Computational Science, Computer Science and Business Analytics.

Column	Question	Type
1	Study programme	enum
2-5	Courses(ML,IR,STAT,DB)	bool
6	Gender	enum
7	Chocolate influence	enum
8	Birthdate	date
9	Neighbors	int
10	Stand up	bool
11	Stress level	float
12	Euro question	float
13	Random number	float
14	Time to bed	time
15-16	Good day	string

Table 1: Given attributes of ODI-2022 dataset.

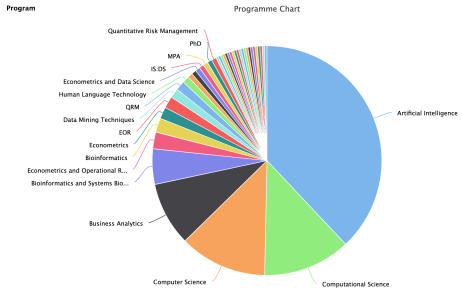
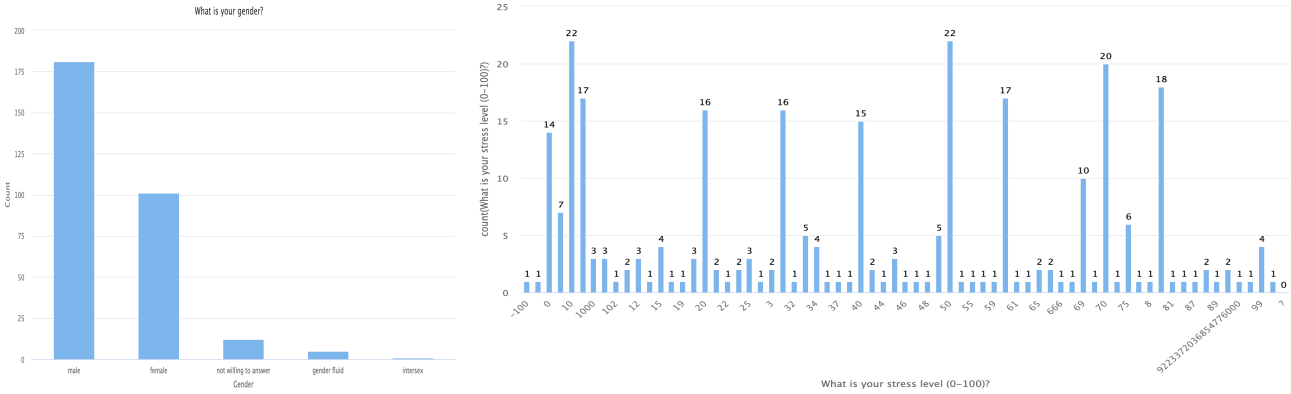


Fig. 1: Programme Chart

Gender distribution The bar-chart shows the gender distribution of students enrolled in the DMT course (Fig.2(a)). It shows that the majority (181 count) of students enrolled in the class are male. This was followed by 101 female students. The gender 'intersex' only accounted for 1 student taking the DMT course. There is a downward trend from the most dominant population; male, to the least dominant one in the course; intersex.



(a) (b)

Fig. 2: (a) Gender distribution (b) Stress level distribution of students enrolled in DMT course.

Stress level distribution The stress level amongst students in the DMT course is not equally distributed (Fig.2(b)). (regardless of the out-of-range outliers that the data shows)

(a) (b)

Fig. 3: ML and Database course distributions

2.2 Stress level and courses

Stress level and Machine Learning (ML) course To analyse if there is a correlation between student stress levels and prior courses taken, a z-test was conducted. This test is appropriate as the dataset has a large sample size. The mean score for stress-level (0-100) for people who took machine learning was 44.53 and for people who didn't, mean 53.87. Next, a 2-sided z-test was conducted with a resulting p-value of 0.014. Thus, the null hypothesis (H0) is to be rejected that the means are the same with 95 percent CI. The results show higher stress levels amongst students who did not take a ML course. Thus, taking a ML course has a positive significant effect on students stress level.

Stress level and Database course To see if the pattern is evident for the database course, the same procedure was followed. The z-test generated a 43.88 mean score for students who took a database course and a mean score of 52.66 who did not. The p-value was 0.019, rejecting H0, showing that students who took a database course had reduced stress levels than those who did not take a database course. Comparing this result to the ML course result, the same positive correlation is found but with ML having a slightly more significance as shown by the p-value.

2.3 Basic Classification

Basic classification models were used on the ODI-2022 dataset. First, the data was parsed into training and test datasets. Two main classifiers were used: 1) KNeighbors Classifier (KNN) and 2)DecisionTree Classifier. Cross-validation was applied with RapidMiner for selecting the best parameters. An association rule to Male, Female, ML, IR, DB was applied. The target feature used were "programme" and the the chosen attributes were "ML, IR, DB, male, female, stress level".

KNN Classification Model For the KNN model, the default KFold cross-validator was applied. The model returned a 0.30 accuracy rate with a 0.07 standard deviation (SD). The predicted accuracy score was 0.298. To test the quality of the KNN model, misclassifaction error (MSE) was calculated. The MSE cross validation scores had the same accuracy score (0.298) as the KFold model. However, the MSE prediction score resulted in 0.35 accuracy with a SD of 0.07.

Decision Tree Classification Model A decision tree classifier was built from the training set.The scores were cross validated and returned 0.27 accuracy with 0.04 SD. The predicted accuracy is 0.25. Comparing both methods, decision tree is better as it takes into account noise and incomplete data. It is suitable for both numeric and nominal values whereas KNN mostly numeric values.

3 Task 2: Predicting Titanic Survivors

3.1 Preparation

Preparation of the data was done by loading the CSV into a pandas dataframe which allowed us to immediately get a sense of the columns and types of the data. The attributes are PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked. Which were described on the Kaggle data page for the Titanic competition[12].

Distributions Distributions are calculated only from the training data, so as to not take into account the test data. This way getting more accurate idea of our accuracy. Each column that had values that represented categories or had a very low variance was plotted in pie charts. One that showed the distribution of the categories and values, and one that showed the distribution of the categories in the survivors. Doing so allowed us to see if there was an effect of the various categories that disproportionately affected the chances of survival. Worth noting that sex, class, siblings/spouses, parents/children, and name don't have missing values but the rest of the attributes do. Since there was a lot of missing data in for the cabin attribute it was not used. The fare was not used due to the fact that it was incorporated more appropriately in the class attribute and the exact fare likely had no affect on survival chances. The ticket data was also not used since it does not inform anything more specific in terms of properties that might assist in predicting survival chances.

Survival: The overall survival rate across the board without distinguishing passengers, is 38.384% (342 out of 891). **Sex**: The distribution of the two sexes included on the voyage is 64.759% male and 35.241% female. On the survivor side however, we can see that males had a 31.871% share of the survivors, whereas females had a 68.129% share. This indicates that the sex of a passenger plays a big role on whether they survived or not.

Class: The distribution of the classes is 491 passengers in 3rd class (55.107%), 184 passengers in 2nd class (20.651%), and 216 passengers in 1st class (24.242%). First class passengers made up 39.766% of survivors, second class passengers

made up 25.439%, and third class passengers made up 34.795% of survivors. Showing that the class a passenger belonged to greatly affected survival chances.

Age: Age is highly variable per row of data and is not a categorical so the first approach for analysis was to make a histogram plots that show the total distribution of the ages of the passengers. Followed by a histogram that compared the distribution of the ages of the survivors compared to the dead. This gives the general idea that more passengers died than survived at first glance. Additionally it can be noticed that children generally survived, where as between the age range of 18-30 there were a lot of casualties. To abstract the value of age to allow for a bit of leniency to the classifiers where there is an uneven number of deaths at a particular age we also made an attribute age group. This allows a better picture of the effect of age on survival, the ranges for this bucketing were obtained from here[10]. From the histogram in Figure 4 we can see that between the ages of 18 to 30 the chances of survival are less than that of the ages before and after.

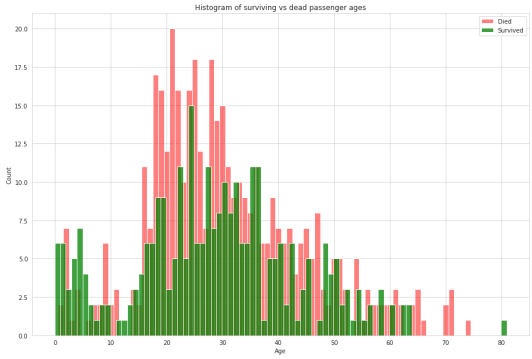
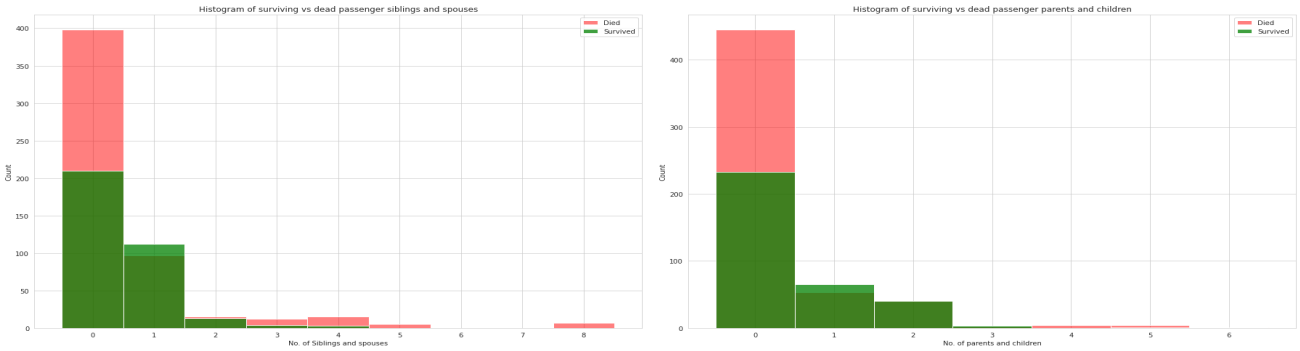


Fig. 4: A histogram of the surviving passengers ages overlaid with the perished passengers survival rates.

Siblings and spouses: Although siblings and spouses is strictly speaking not a categorical attribute, it can be seen as such due to a very low variance. From the distribution of all passengers that most people had 0 siblings or spouses, about 68.238%, and 23.457%, having 1 sibling or spouse. The survival rate of passengers who had 1 sibling or spouse is 32.749% of the survivors. People with 0 siblings and spouses made up 61.404% of the survivors. The percentage of those having 0 siblings or spouses is 68.2% for all passengers and 61% amongst survivors. For those having 1 sibling or spouses, the percentage is 23.45% for all passengers and 32.49% amongst survivors. This indicates that there might be a slight effect of having at least a single sibling or spouse on the ship, as it may be helpful to have an extra pair of hands on the sinking ship (Figure 5(a)). This could probably also explain how women had a higher survival rate, since their husbands probably helped them escape.



(a) (b)

Fig. 5: A histogram of the surviving passengers overlaid with the perished passengers survival rates. for (a) no. of siblings and spouses, (b) no of parents and children

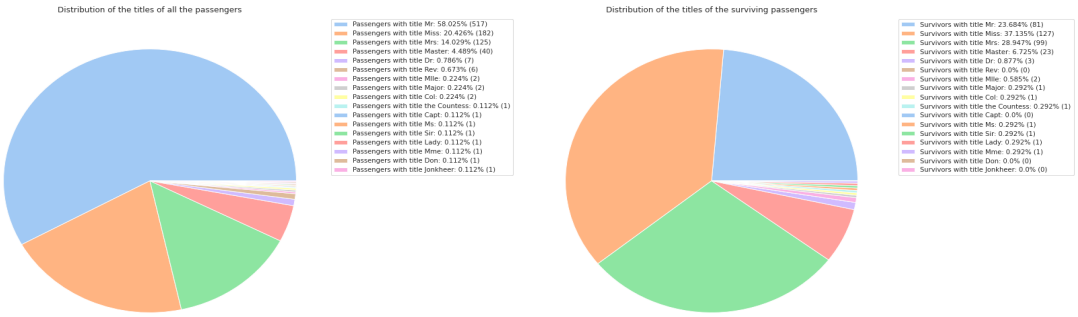
Parents and children: We can see that most passengers (76.094%) had no children and/or parents on the ship. With the majority of the remainder having 1 or 2 parents and/or children (22.223%). From the distribution of the survivors we can see that people with no parents or children have dropped to 68.129% and people with 1 or 2 parents or children are around 30.702%. This reinforces the fact that children and infants had a much higher survival rate seeing as how they are the majority of passengers with 1 or 2 parents (Figure 5(b)).

Port of embarkation: Most passengers boarded the ship in Southampton where the ship departed, followed by Cherbourg, and then Queenstown. The survival rates are slightly different with Cherbourg making the biggest gains, this could be coincidental or potentially could be loosely linked to where passengers that boarded later would stay during the voyage.

Title: The title was also extracted as previously mentioned, this extracted attribute not only informs the sex of a passenger but also the social status of the passenger. As seen in Figure 6, we see that the distribution of titles amongst survivors is quite different as compared to the distribution of all passengers, indicating that this is indeed an important feature.

3.2 Classification and Evaluation

We then take the data and replace all the strings with integers that represent the categories. The training data was also split into training and test, 80% and 20% respectively. We decided to try out various classifiers: Decision Trees, K Nearest Neighbour, Random Forest, Multi-layer Perceptron (MLP), and AdaBoost. We see that the best score attained for our various created classifiers were consistently the K Nearest Neighbour, Random Forest, and MLP classifiers. We then ran the models on the test data provided as part of the Kaggle dataset, and



(a) (b)

Fig. 6: The distribution of the titles of all the passengers (a) and of the survivors (b)

No.	Method	Local Scores %	Kaggle Scores %	Difference
1	Decision Trees	0.79888	0.70095	0.09793
2	K Nearest Neighbour	0.81564	0.74401	0.07163
3	Random forest	0.81006	0.75837	0.05169
4	MLP	0.82123	0.78229	0.03894
5	AdaBoost	0.79888	0.76315	0.03573

Table 2: The results of our created classifiers locally and on the Kaggle competition

then uploaded the results as a submission to the Kaggle competition. From the uploaded results, we found that our best classifier was the MLP classifier, which is a fully connected feedforward artificial neural network based classifier. We placed 1073rd (as of writing) under the Team "Nader Sobhi", in the competition leaderboard. The score we obtained only slightly dipped compare to our locally obtained score. So in that regard we closely matched our expected score.

4 Task 3: Research and Theory

4.1 State of the Art Solutions

Description For this task, we pick a competition and study the winning solution. The competition that we've chosen is that of Web Traffic Time Series Forecasting [1]. The competition involves forecasting future values of web traffic of 145,000 wikipedia articles. What is particularly interesting about this competition is that the testing process includes a stage where actual future events are predicted, (as opposed to merely predicted historical data)

Winning Technique The winner of this competition was Arthur Suilin. He used a seq2seq model along with engineering new features to incorporate quarter-to-quarter and year-to-year seasonality in data.

Feature Engineering As described in the discussion post [2], a combination of local features and global features were used for prediction. Local features refers to analyzing the recent trend, identifying a spike (and correcting accordingly), and accounting for holidays. The global features, on the other hand, refer to broader seasonality in data, for example, year-to-year, and quarter-to-quarter correlations. For example, quarterly correlations may be useful in the financial domain where public companies publish financial statements every quarter and yearly correlations in competitions that are played annually, such as Wimbledon.

Apart from these, features like day of the week, agent, country, site, page popularity, etc were also used.

Model: A GRU(Gated Recurrent Units)-based Encoder-Decoder model was used. GRUs are a type of RNN (Recurrent Neural Network) which are typically used to train sequential data. GRUs overcome the problem of short-term memory in RNNs by containing hidden states and internal mechanisms called gates which regulate the flow of (past) information from the sequence. Encoder-decoder models are a type of model that involve two parts - the encoder and decoder. These are models which given a sequence of inputs, can generate as output a sequence of predictions. They are typically used in applications such as language translation. Since the web traffic forecasting involves a series of inputs, and a series of outputs, the author used an encoder-decoder model with GRUs for both the encoder and the decoder units. Refer figure from [11] fore more details.

Validation and Hyper-Parameter Tuning: For the process of validation and tuning, the usual process of splitting a dataset three-way (train, validate, test) has a problem in that the latest data points would not be used for training the model, which would be detrimental given that the model was going to be tested on future data. Therefore, the final model that was submitted was trained in blind mode, without any evaluation and tuning.

How it was different?: Quite a few solutions involved the use of moving average models, and simpler median models. There were a few other solutions that used RNNs(particularly LSTMs). In our opinion, the winning solution stood out because of two aspects: feature engineering, and blind-mode training on latest time series data, which would allow for better capturing the latest trends.

4.2 MSE versus MAE

Mean Squared Error (MSE) and Mean Absolute Error(MAE) are two examples of evaluation metrics used for regression models.

MAE is the mean of all the absolute errors in the predicted dataset. The absolute error is the absolute of the difference between the predicted value and the actual value. ($\sum_{i=1}^{D} |x_i - y_i|$)

MSE, on the other hand is a mean of the squares of the errors in the predicted dataset. ($\sum_{i=1}^{D}(x_i - y_i)^2$)

Table 3 describes when it would be useful to use which.

MAE	MSE
Same units as that of the predicted value. Therefore it is more useful when the impact is proportional to the actual increase in error (for example, in the financial industry)	Square of the predicted value unit
Does not penalize large errors. Therefore, less sensitive to outliers. In situations where it is known that there are outliers in data, one might choose to minimize MAE instead of MSE.	Penalizes large errors because of the square operation. Therefore it is more sensitive to outliers.
-	It is a differentiable function that makes it easy to perform mathematical operations. Therefore, in many models, RMSE (square root of MSE) is used as a default metric despite being harder to interpret than MAE.

Table 3: MAE vs MSE

Identical Results: A simple scenario where the MSE and MAE would yield identical results is when all the prediction errors are either $+/-$ 1 or when all are 0, as in the case of 0, 1, $x = x^2$

Experiment on Datasets: We chose the insurance dataset [4] to experiment with as it is conducive for regression models. It has the following features - age, sex, BMI, children, smoker, region. The prediction column is the charges that they have claimed with the insurance company.

The first step was to perform some preprocessing. The categorical variables were converted to ordinal (numeric) so as to be able to a fit a regression model. After that, we experimented with two models - Linear regression and Polynomial regression. We estimated a non-linear relationship between the features and the prediction variable. Therefore, we chose these two models to help validate that hypothesis. For linear regression, the MAE= 4464.79495619427 MAE= 2767.4696942350747. For Polynomial Regression, MSE= 41764386.59725513 MSE= 20070414.751107506.

Based on these results, we see that

- MSE >> MAE. This is expected as the MSE involves a square operation, which penalizes larger errors.
- With degree=2, the polynomial regression is able to fit the data better as compared to linear regression, which validates our hypothesis about the existence of a non-linear relationship.
- Increasing the degree beyond 2 increases the error as well. Therefore, 2 seems to be the optimal degree value for this model and this dataset.

4.3 Analysis of a less obvious dataset

As the sms dataset is purely textual data, text mining techniques would be suitable. Particularly, Natural Language Processing techniques such as lemmatization, TFIDF, word embeddings would come in very handy.

Data Transformations The first task was to clean the data in the file before reading. There were html tags, and emojis present in the text which made the separation of columns using the ";" delimiter non-trivial. With the help of some built-in python functions and regex, we were able to clean the data

The second task was to extract important features from this dataset. The following were extracted: word count, uppercase count, symbol count. The uppercase count counts the number of upper case characters in the text, and the symbol count counts the number of symbols(nonalphanumeric) characters in text. This is particularly useful in the context of spam because upon visual inspection, it was clear that most spam messages were high in these counts.

The final task was to clean the text and vectorize the text. This is a necessary step as strings needs to be converted to numbers to be interpretable by a model. We tried out two methods of vectorization: (1) TFIDF(term frequency–inverse document frequency) which demonstrates the importance of a word to a document in a corpus. (2) Word2Vec word embeddings (with a mean calculation to come up with one single vector for the entire sentence)

Model, quality and further improvements We experimented with two variants of models: a logistic regression classifier with tf-idf vectorization, and a neural network with word embeddings vectorization. The rationale behind these choices was to pick a relatively simple model for tf-idf, and since word embeddings are high-dimensional numeric data, neural networks would be the better choice.

We used the AUC (area under curve) metric. This is because the dataset is biased towards ham label. Therefore a more comprehensive measure of area under the ROC curve(which would help evaluate if the model was able to achieve false positive and true positive rates which are significantly better than a random chance).

The AUC for the Neural net (0.9934) is marginally better than the Logistic regression (0.9903)

Areas of improvement Firstly, the emoji/emoticons present in the text are an interesting piece of information, and therefore can be extracted and represented as a feature in some form. Secondly, using word embeddings that are trained on text messages would give a better semantic representation of the words as compared to the ones currently used, which are trained on generic text.

References

1. https://www.kaggle.com/competitions/web-traffic-time-series-forecasting/overview

2. https://www.kaggle.com/c/web-traffic-time-series-forecasting/discussion/43795
3. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21
4. https://www.kaggle.com/datasets/mirichoi0218/insurance
5. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
6. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
7. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
8. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
9. LNCS Homepage, bluehttp://www.springer.com/lncs. Last accessed 4 Oct 2017
10. MyAgeCalculator, bluehttps://www.myagecalculator.com/blog/life-stages-and-age-groups-40. Last accessed 20 Apr 2022
11. https://github.com/Arturus/kaggle-web-traffic/blob/master/images/encoder-decoder.png?raw=true
12. Kaggle Titanic Data, bluehttps://www.kaggle.com/competitions/titanic/data. Last accessed 16 Apr 2022