Data Mining Techinques - Group 115

Julia Sudnik$^{1[2715115]}$, Meifang Li$^{1,1[2719570]}$, and Rafael Cárdenas-Heredia$^{1[2718909]}$

Vrije Universiteit Amsterdam

1 Explore a small dataset

1.1 Exploration and Data Preparation

For the first task the ODI data set was explored using python and its packages. Information from the data set was collected via a Google form during a Data Mining Techniques lecture. The data set consists of 313 records and 16 object types attributes (excluding timestamp). Due to high complexity of column names, several transformations were made and the following new column names were created:

```
['programme', 'ML', 'IR', 'ST', 'DB', 'gender','chocolate',
'birthday', 'neighbour', 'standup', 'stress', 'competition',
'random','bedtime','good1','good2']
```

In general, two kinds of questions from the previously mentioned form can be distinguished, i.e. open (with or without restrictions) and closed questions. While closed questions formed categorical instances, open questions (especially not restricted ones such as *birthday*) allowed answers of different types and relevancy. This resulted in the strong need for data cleaning processes.

Firstly, we decided to examine the *programme* feature, describing courses that DMT students are currently in. The data consists of various and at times unnecessary information. For this reason, a categorisation criteria was created based on key words. For instance, all entries including 'artificial' or 'AI', regardless the letter case, were categorised as *AI*. Similar process was conducted for courses *BA, CS, CLS, Econometrics, Finance, Bioinformatics, Information.* Less common courses were categorise together as *others*. Furthermore, 3 unrelated or ambiguous instances, e.g. 'python', were dropped.

Features *'ML', 'IR', 'ST', 'DB'* were of similar nature and described whether or not a student was following a certain course. In general, there were three answers possible in terms of meaning: 'yes', 'no', 'unknown'. Nonetheless, for some of the courses mentioned answers were represented by different kinds of words (e.g. 'mu' instead of 'yes'). Thus, a mapping was conducted connecting all instances meaning *Yes* with *1*, all instances meaning *No* with *0* and all instances meaning *Unknown* with *2*. Further, all entries were normalised.

Lastly, we found that 12 entries of *stress* that either exceeded the range of [0,100] or were text and thus will be omitted in the data visualisation.

1.2 Data visualization and discussion

The ODI data set describes students participating in the DMT course. The majority of students are male (65%). As seen in Figure 1, AI students make 28% of all participants and therefore, make the largest group followed by BA, CLS, Bioinformatics etc. This indicates that the data mining techniques can be applied in various fields, however is most popular among students with analytical backgrounds.

As shown in Figure 2 most students claim to be familiar with statistics and machine learning. Further, almost half (46%) of them followed a database course. However, only 29% of students have studied information retrieval. We may argue that some students, especially those without previous experience in statistics and machine learning might find the course more difficult then others. Mentioned results may also indicate that there is a similarity between courses DMT and IR. Thus, it is uncommon to follow both.

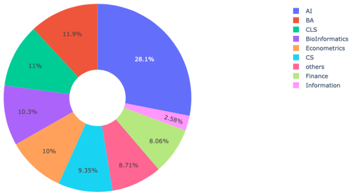

Fig. 1: Percentage of programme

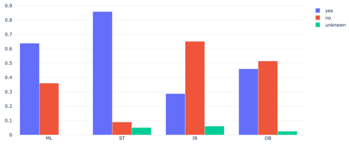

Fig. 2: Percentage of four courses

Surprisingly, only 11% of students claimed that chocolate can make you slim. The most common belief shared by 37% of students is that chocolate makes you fat. Further, we decided to see relation between *stress* and *chocolate*. For the purpose of this experiment we grouped stress values into: 'Low', 'Moderate' and 'High'. Surprisingly, we did find that the largest group that voted chocolate to makes you slim was the 'Low' stress group of student. Further, we found that the largest group that voted chocolate to makes you fat was the 'High' stress group of student. However, neither of the results was significant.

Interestingly, it was found that female students report higher levels of stress then males. As seen in Figure 5 females are a majority of students that reported stress higher then 35. This corresponds to general studies about gender differences in reporting stress [8]. Another interesting gender differences were found in the *competition* feature. In general males tended to choose more extreme values (closer to 0 or 100) , whilst females tended to report more middle values See Figure 4.

One of the questions in the form, explored what makes a good day for students. Due to a large variety of answers, we have decided to split each entry into single words and use wordcloud to analyze the *good1,good2* columns by obtain-

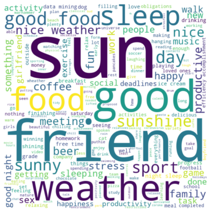

Fig. 3: Wordcloud about good day

ing the frequency for each word. By and large, as seen in Figure 3 most students report good weather, friend, sun, food and sleep to be indicators of a good day. Interestingly, stress is visible as a word with relatively high frequency, proving that this method might not take into account more complex phrases like 'no stress'.

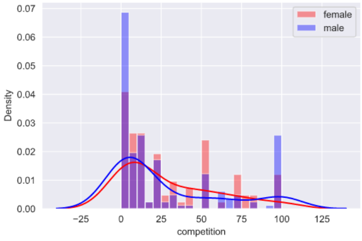

Fig. 4: Competition and gender

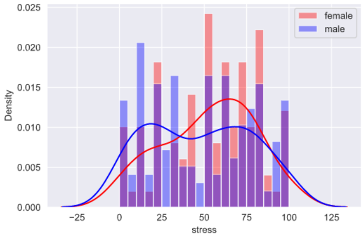

Fig. 5: Stress level and gender

Additionally, we decided to explore any potential correlations between features such as *ML, IR, ST, DB, gender, stress*. Nonetheless, no moderate nor high correlations were found. Examined correlation values are the following: 0.2376 between DB and IR; 0.1842 - between ML and IR; 0.1326 between ML and DB; 0.1056 - between gender and stress. Other pairs scored below 0.1.

To conclude, through the exploration of this raw data file, we found the data cleaning process to be largely time-consuming. We found it especially difficult to measure open questions like *birthday* and *bedtime* as many formats of this features exist. Moreover, even for closed questions such as the ones regarding previous experience in ML, IR etc., it is important to simplify and standardize answers of similar meaning. To sum up, we found great significance in restricting the format of answers during the data collecting process, as it can certainly facilitate data mining processes.

1.3 Basic Classification and regression

In terms of basic classification and regression, we are interested in the research of vowel recognition studies because this can be useful in speech recognition that develops methodologies and technologies both in computer science and computational linguistics. The vowel dataset we used was collected by Deterding [7], in which examples of the eleven steady state vowels of English were recorded when spoken by fifteen speakers for a speaker normalisation study. The reflection coefficients of the speech signals were used to calculate 10 log area parameters, giving a 10 dimensional input space. Each speaker yielded six frames of speech from eleven vowels. This resulted in 528 frames to fit the model (the training set) and 462 frames to evaluate to the predictive accuracy of the fitted model (the test set). Therefore, the input of data is ten dimensional and the output is to decide which one of the eleven vowels was spoken. Since this is a multi-class classification, we mainly studied two classification algorithms, Multinomial Logistic Regression and Support Vector Machine(SVM).

To fit the Multinomial Logistic Regression model with *sklearn*, we use the LogisticRegression module with *multi_class* set to "multinomial", *random_state* set to "0" to introduce randomization, *solver* set to "lbfgs" to handle multiclass problems and fit X and y to the training set. To fit the SVM model with *sklearn*, we use svm module with *decision_function_shape* set to "ovo" because one-vs-one ('ovo') decision function is always used as multi-class strategy and fit X and y to the training set.

Then for both algorithms, we can use the predict method to predict probabilities of testing dataset, as well as the score method to get the mean prediction accuracy. We also use *cross_val_score* to implement the cross validation by splitting the data automatically, fitting the models mentioned above and computing the score 5 consecutive times. The result of accuracy is 0.461 for LR and 0.602 for SVM. And The result of cross validation is 0.50 for LR and 0.64 for SVM. This shows us that for the vowel data, an SVM using the default radial basis function was more accurate. This is mainly because logistic regression is trying to maximize the posterior class probability, while the SVM is more geometrically motivated and trying to maximize the margin between the closest support vectors, meaning it is more robust to outliers than LR will normally tolerate.

2 Kaggle Competition

2.1 Preparation

Data visualisation and analysis The Kaggle Titanic training data-set consists of 890 entries and 11 columns. The attributes include:

```
['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age',
'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked']
```

Firstly, all columns were checked for missing values. Features 'Age', 'Cabin', 'Embarked' have respectively 177, 687 and 2 missing values. Secondly, all features were analysed individually.

Survival is the targeted categorical, although represented numerically, feature representing those who did ('Survived' = 1) and those who did not survive ('Survived' = 0) the Titanic. The overall Titanic survival rate is 61%.

Pclass There are three categories of class, although represented numerically. The vast majority of passengers of the Titanic were travellers from the 3rd class. Further, this feature seems to have one of the strongest effects on survival, i.e. passengers who acquired tickets with better class had in general higher chances of living (see Figure 6).

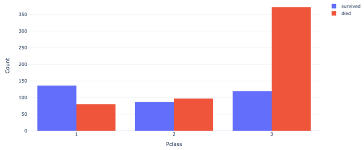

Fig. 6: Pclass in Titanic

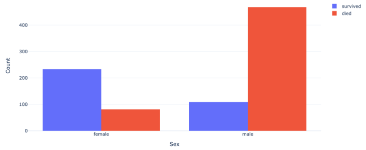

Fig. 7: Sex in Titanic

Name According to many *Name* is the most underrated feature of this dataset [6,11]. By extraction of titles and surnames it is possible to gain information about sex, age, status, and estimated number of family members.

Sex The majority of passengers were male. This seems to be, alongside *Pclass* one of the strongest predictors of *Survival*. Females ($P(Survival = 1) = 0.74$) had a significantly higher survival rate then males($P(Survival = 1) = 0.19$). Interestingly, despite females being outnumbered by males almost twice, still overall more women survived the titanic then men (see Figure 7).

Age The majority of passengers were young adults with 50% of all of them being between 20 and 38 years old. There are some gender related differences. Firstly, age distribution seems more normal in females whilst in males there is a clear majority of young adults. Secondly, a higher peak of non survival can be observed in adult males between 25 and 35 years old that is not so high in females. Lastly, in both gender groups infants have a visibly higher survival rate.

Missing values of *Age* have been replaced by the average ages depending on a person's title [12]. This method seems to be more accurate than using only averaged ages from the whole sample as some titles distinguish a minor from an adult (e.g. Ms versus Miss).

Sibsp, Parch The vast majority of passengers traveled without any family members. Further, chances of survival were higher for people travelling with less family members for both *SibSp, Parch*(See Figures 10,11).

Ticket 681 out 891 tickets are unique numbers. Overall, this feature is complex and does not bring significant information about survival, therefore is not considered for the model.

Fare Ticket prices resemble a logarithmic distribution. 75% of passengers paid under 31, while the maximum fare was 512. This feature shows similar

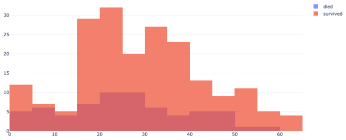

Fig. 8: Age of female passengers

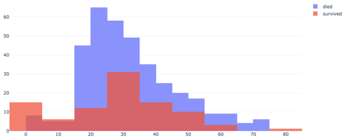

Fig. 9: Age of male passengers

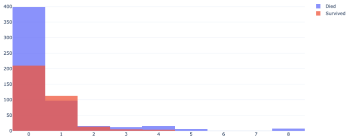

Fig. 10: Number of Siblings and Spouses

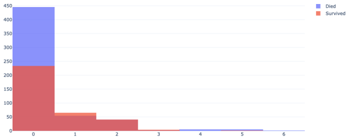

Fig. 11: Number of Parents and Children

information to *Pclass*, as it costs more to buy a ticket with a higher class. Due to high complexity and similarity to a simpler feature, *Fare* is not considered in the model.

Cabin The majority of *Cabin* data is lost. The remaining 204 are considered to show similar information to *Fare* [1], and therefore to *Pclass*. For those reasons *Cabin* is not considered in the model.

*Embarked*The majority of passengers entered the Titanic in Southampton. This is also were most of the 3rd class passengers boarded, which explains why the survival rate is the lowest for this port. The highest survival rate was observed for Cherbourg, which we assume comes from the fact that most people that embarked in this port bought tickets with 1st and 2nd class. We assume that in general the connection between *Embarked* and *Survived* can be explained by the ticket class of passengers boarding in individual ports. For this reason this feature seems redundant and is not considered for the model.

Feature Engineering and Data Transformation In this process several new features were considered.

Title As previously mentioned, feature *Name* can give plenty of insight. Thus, we have decided to extract titles from names and group them into 5 following types [11]: 'Mr', 'Mrs', 'Master', 'Miss', 'Officer', 'Royalty'. The following feature, very well distinguishes gender between passengers, and therefore makes the feature *Sex* redundant. For this reason *Sex* is no longer considered for the model.

FamSize Since attributes *SibSp* and *Parch* strongly relate to similar feature, i.e. family size, it has been assumed that the features might gain from combining them together. A new feature *FamSize* was created with categories 'Alone', 'Small' and 'Big' [12].

MissingAge An exploration of importance of missing age values was conducted. Nonetheless, does not really seem to have a strong effect on survival. Therefore in the end this feature is not used.

IsInfant When exploring the *Age* feature it was clear that children aged 0-5 had a higher chance of survival. We found that *IsInfant* had a stronger correlation with *Survival* then *Age*. Spearman 0.15 versus 0.07.

MissingCabin Not having a value for cabin seems to be an indicator for survival. Many 3rd class passengers did not have a cabin at all on Titanic. Might be a similar indicator to the *Pclass* feature, however because it is simpler, i.e. has only two possible values instead of three, we decided to further explore it.

After having examined all the features in the Titanic data-set and their relevance, the following were chosen as predictors for the classifiers: 'IsInfant', 'FamSizeCat', 'MissingCabin', 'Title', 'Pclass'. For some of the features additional preparation was necessary. *FamSizeCat* and *Title* were changed into numerical variables. Further, all values were normalised for all features. None of the features had missing values in the data-set. Lastly, the same transformations mention in this chapter and the Feature Engineering chapter were used on the training data-set.

2.2 Classification and Evaluation

In order to evaluate the classifiers, the Titanic training data-set from Kaggle was further split into training (70%) and testing (30%) sets. Three different classifiers from *sklearn* package were then applied, i.e. a Decision Tree Classifier, a Random Forest Classifier, and a K-Nearest Neighbors Classifier. An evaluation was conducted by comparing training and test accuracy of all classifiers. We found the Decision Tree Classifier to perform best.

Table 1: Accuracy results

Classifier	Training Accuracy	Test Accuracy
DecisionTree	0.857143	0.813433
RandomForest	0.850722	0.798507
KNeighbors	0.836276	0.753731

2.3 Kaggle Result Comment

Our score on Kaggle was 0.77990, which did not come as a big surprise as we were expecting the score to be a bit lower then our accuracy score. Overall, our team ranked 7757 out of 30689, which put us in the top 26%.

3 Research and theory

3.1 Research - State of the art solutions

For this assignment we have chosen an "Airbnb New User Bookings" competition held 5 years ago on Kaggle [2]. [1] In the competition participants had to predict a country which a new user will want to choose as a destination for their next scheduling through AirBnB. For this purpose, all the competitors were provided with both training and test data set consisting of a statistical summary of the destination countries, a statistical summary of the age ranges, gender categories and destination countries associated with previous users, the records of the web sessions of previous users (with information about their devices, nature of their interactions and time spent in them). The evaluation metric corresponded to the Normalized Discounted Cumulative Gain (NDCG) with k = 5 where:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2 (i+1)}, \qquad nDCG_k = \frac{DCG_k}{IDCG_k},$$

Being $IDCG_k$ the ideal DCG for a set of queries, and where rel_i is the relevance of the result at position i. The result expected for the evaluation of the performance of the competitor was a dataset composed of two columns (id, country) where, for each new user in the test set, the possible preferred countries are displayed in order or probability of preference.

While the approach of the first winner was not disclosed nor explained in the forum, the method used by the second winner (who achieved the score of 0.88682), Keiichi Kuroyanagi, was. According to the winner, she owes her success to thorough preprocessing [10]. Keiichi Kuroyanahi created 1,312 features from the given data set. Moreover, she found signifance in features other competitors did not put attention to such as the out-of-fold CV predictions of categorized lag.

Next for the machine learning model, 18 out-of-fold cross-validation predictions were calculated by means of 18 models, using stacked generalization. This was done with the following assignation: four *XGBoost* models [3] for *country/destination features*, three *XGBoost* models and two *Generalized Linear Models (GLM)* [5] for the *age features*, four *XGBoost* models and two *Generalized Linear Models (GLM)* for the *lag features*, and three *XGBoost* models for the set comprising *Age* and other features. These two previously mentioned processes constituted the first layer. From the set constituted by the base features and the out-of-fold cross-validation predictions, random sets of features with a size of 90% were repeatedly selected and used for the iterative application of single *XGBoost* models. From these, the models with the best performance (*XGBoost model (5 fold-CV: 0.833714)*) was selected to constitute the third layer.

[1] It is important to mention at this point, that our group struggled with finding a competition at a relatively beginner level (i.e. using techniques learned in the course so far). For this reason we make our best to present the winning strategy, nonetheless our presentation might lack a good explanation for why this strategy stands out as the methods used were significantly beyond our knowledge.

3.2 Theory - MSE versus MAE

The function of a regression model is to predict or estimate unknown numerical values from the behavior of given data. A simple definition of the error of a regression model is the difference between the point data that you have and the trend line generated by the algorithm. This error can be determined in multiple ways, and each methodology conveys different information about the implemented analysis. Two of these measures are the MSE and the MAE.

The MSE is an error metric that seeks to measure the average squared error of the predictions, by executing the calculation of the square difference between each prediction and each data point, which it then averages. As a result it provides information on the accuracy of the model with the desirable value being 0. This metric, however, can be misleading in situations in which the data contains high amounts of noise, since squared the differences will provide a pronounced negative result even in models with high accuracy. To counteract this accentuated interpretation of the variation, this model is used as a component of the RMSE, which uses a square root to reduce the sum of the squared-weighted values.

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2 \qquad RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2}$$

MAE is a metric that is calculated by taking the average of the absolute difference between the given data points and the predicted values. The linearity of its behavior leads, then, to the error penalty not being as accentuated as in MSE, since the nuances provided by noise or variation will never be violently emphasized. The major advantage of this metric is that it is more tolerant towards outliers, which makes it more adaptable to datasets with a certain degree of stochasticity. This difference is due to the non-linearity of the first, contrasted with the linearity of the second metric.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|(y_j - \hat{y_j})\,|$$

A rare example situation in which the accentuated difference between MSE and MAE would not exist, is that of a regression in which the error each data point is either 0, 1 or -1, since, for all values involved in the sum, the squared value and the absolute value would be equal.

In order to evaluate how would the MAE and MSE error metrics differ between regression models, a dataset of several weather metrics taken in Szeged, Hungary between the years 2006 and 2016 was obtained from Kaggle [4], which was further split into training (70%) and testing (30%) sets. For the purposes of the comparison, three regression models were implemented in order: Ordinary Least Squares Linear Regression, Polynomial Linear Regression (degree 20), and Decision Tree Regression. These were implemented with three independent variables (Temperature, Humidity and Pressure) and one dependent variable (Visibility). Both MAE and MSE were generated for each case.

Table 2: MAE and MSE errors for different regression methods

Regression	MAE	MSE
OLSRLR	3.10292	14.34899
Polynomial	2.66300	10.68879
Decision Tree	0.01742	0.05307

As can be seen, the best performance was that reached by means of the Decision Tree Regression, yielding substantially more accurate results than those obtained through the other methods. This doesn't come as a surprise, given the differential level of detail and sophistication of the latter model with respect to the other two.

3.3 Theory - Analyze a less obvious dataset

The data collection contains 5574 text messages, in the form of strings that include alphanumerical and non-alphanumerical characters. Parting from this data, a testing set has to be sorted in the form of Ham or Spam, in order to predict if the messages in question are unwanted messages or not. When it comes to text information the texts have to be converted to numerical information, and the higher the amount of data that is obtained by the wide diversity of transformations applicable, the more accurate the prediction will be (given that one uses a model that can take advantage of this information). For this reason, models that can manage dense feature sets are preferable.

For the purposes of this activity the data set was separated into a training set comprising 4000 data entries and a testing set with 1576 data entries. Lenght (full extent), word count, average word length, count of numeric characters and count of words containing upper case letters were obtained as features from the texts in the testing dataset [9], as well as a matrix composed of the count for every single word comprised in these texts, weighted by means of a tf-idf transformer; these approaches, together, yielded 7330 features.

Following this, two models, one corresponding to a multinomial Naive Bayes classifier and the other being a Random Forest classifier were trained with the previously obtained features and the Ham/Spam labels from the training set. After this, the same transformations were applied to the texts in the testing set (taking in consideration that the creation of the matrix must correspond to the words present in the training texts, not the testing texts), and then these features, along with the labels in the testing set, were fed to the previously trained models. The accuracy obtained by the Naive Bayes classifier was that of a 90.3%, while the Random Forest classifier obtained a score of 97.7%. Additional actions could have been taken towards the improvement of the results, such as counting stopwords or non-alphanumerical characters to create more features, or subjecting the texts to pre-processing such as the removal of punctuation marks, upper-case letters, common words, tokenization, or finding the roots of the words by means of stemming or lemmatization. Code can be provided upon request.

References

1. https://medium.com/analytics-vidhya/random-forest-on-titanic-dataset-88327a014b4d
2. Airbnb new user bookings, https://www.kaggle.com/c/airbnb-recruiting-new-user-bookings
3. Brownlee, J.: (2016), https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
4. Budincsevity, N.: (2016), https://www.kaggle.com/budincsevity/szeged-weather
5. Burkov, A.: The hundred page machine learning book (1956)
6. Cdeotte: Titanic using name only [0.81818] (Oct 2018), https://www.kaggle.com/cdeotte/titanic-using-name-only-0-81818
7. Deterding, D.H.: Speaker normalisation for automatic speech recognition. Ph.D. thesis, University of Cambridge (1990)
8. Harutyunyan, A., Musheghyan, L., Hayrumyan, V.: Gender differences in perceived stress level among undergraduate students in armenia. European Journal of Public Health **30**(Supplement_5), ckaa166–1028 (2020)
9. Jain, S.: (2018), https://www.analyticsvidhya.com/blog/2018/02/the-different-methods-deal-text-data-predictive-python/
10. Kaggle: (2016), http://blog.kaggle.com/2016/03/17/airbnb-new-user-bookings-winners-interview-2nd-place-keiichi-kuroyanagi-keiku/
11. Sgneves: Titanic: Eda, models benchmarking and tuning (Apr 2021), https://www.kaggle.com/sgneves/titanic-eda-models-benchmarking-and-tuning
12. Zlatankr: Titanic random forest: 82.78% (Feb 2017), https://www.kaggle.com/zlatankr/titanic-random-forest-82-78Age