Datamining: Assignment 1 basic

Jier Nzuanzu (2529760), Steven Raaijmakers (2655645), and
Lennart Mettrop (2528910)

VU Amsterdam

1 Introduction

In this report we discuss experiments on three data sets. The first data set consist of answers from students on a quiz, given during a lecture. We try to predict which study program the students are enrolled in based on some courses they have followed. The second data set contains information of 1309 passengers boarded on the Titanic. Using this data set we try to predict whether a passenger has survived the sinking by using known variables as their age and gender.

For the third data set a comparison is made between different entries of a competition. Here contestants must predict from which country a recipe comes from. Here we are interested in how the best approach differs from a more general approach.

2 Exploration of a small dataset

In this section we study a small data set containing answers for a quiz given to 276 students during a lecture. This quiz consists of 17 questions, seven of them being multiple choice while the remaining ten questions are open questions.

Some questions required the student to insert a certain number, however we see some answers not containing digits. For example, a student answered the question "How many neighbours are sitting next to you?" with "Many".

We want to see the distribution of the students programs using the data set and compare this to how many have enrolled to the course. Therefore we make a piechart of the distribution of the different study programs.

In the quiz multiple questions concern the courses followed by the student. We expect a correlation between the courses a student has followed and the students study program. Thus we will try to predict whether a student is studying Artificial Intelligence based on whether she/he has the courses machine learning, information retrieval, statistics and databases.

2.1 Methodology

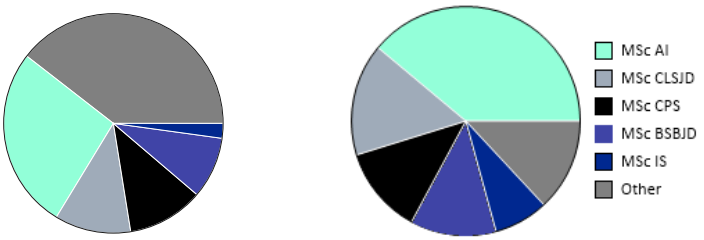
For making the plots we immediately stumble upon the problem that the students answered very inconsistent. The questions about the followed courses were all multiple choice, so these didn't give many problems. However, the question about the study program was an open question. For example, the course Artificial Intelligence was sometimes written with only lowercase letters, the abbreviation AI was used and some students made spelling mistakes. We first map all different writings s.t. all different names were categorised as the same. Study programs which only a few students followed are categorised as 'other'.

As we have prepared the data, we can make plots and predict if a student follows the 'AI' program. We do a regression experiment using three different algorithms. The first is C-support vector classification (SVC) where we have taken the linear kernel and a penalty factor of 1. The second is a decision tree classifier using Gini impurity as measure of a split and the best split is chosen. The third is Label Propagation where we took 'rbf' as kernel.

We have used cross validation which means we first divide the training set into k folds. For every fold we use the other $k-1$ folds to train and test it with the selected fold. The resulting scores are averaged. We tested with two different scoring functions. The first is the accuracy of the predictions and the other function is the weighted F1-score which is the weighted average of precision and recall.

2.2 Results

Fig. 1a shows how many students attended the lecture from each study program. Fig. 1b shows how many students registered for the course from each study program which is found from the courses page on the site Datanose.nl [1]. We see a lot more AI students registered for the course compared to how many showed up at the lecture. Surprisingly the proportion of student following a program categorised as other has increased. Apparently these students had a higher attendance at the lecture.



(a) Pie chart of people that showed up at the lecture.

(b) Pie chart of people that showed up at the lecture.

Fig. 1. Pie charts of how many people are registered from multiple study programs.

Table 1. Scores for the regression experiment for the small dataset collected by students. Scores are calculated using multiple regression algorithms and different scoring functions.

Regression algorithm	weighted f_1 score	accuracy
SVC	40.4%	46.9%
Decision tree	42.5%	48.3%
Label propagation	43.3%	48.6%

[1] https://datanose.nl/#course[69270]

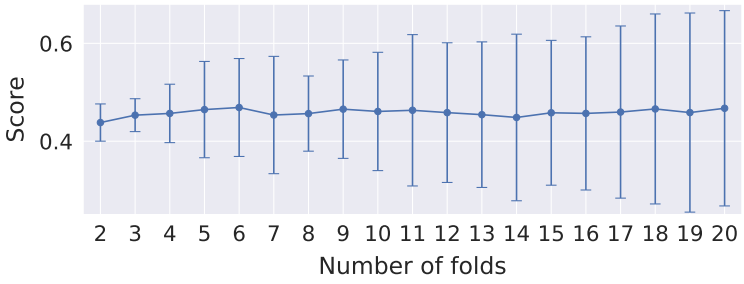


Fig. 2. Score using SVC regression and the accuracy as score function for multiple number of folds for the cross validation. The score does not get influenced much, however the standard deviation rapidly increases.

For predicting whether a student is in the AI program we use three different regression algorithms of which the scores are listed in table 1. These scores give the percentage of right predictions among the test set in case of the accuracy function. Also cross validation was used.

We see the accuracy score function gives a better score. For both score functions, the Label Propagation algorithm gives the highest score, thus Label Propagation being the best algorithm in this case. However in all experiments, the score does not become very high. This can be caused by choosing too many variables for the regression, or the variables may not have been totally independent.

We also studied the effect of the number of folds in the cross validation process. Figure 2 shows the accuracy score using the SVC regression algorithm for different number of folds. We see the accuracy does not get influenced that much. When the number of folds increases, each fold contains a smaller train set or test set, so the standard deviation rapidly increases.

3 Titanic

The Titanic data set contains information about 1309 passengers who boarded the titanic. The training set consists of 891 passengers, with 12 features, 7 containing numeric values (PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare) and 5 categorical values (Name, Sex, Ticket, Cabin, Embarked). The 418 passengers in the test set have the same features except for Survived.

While we pass through the data set we see most columns containing values in all rows, except for the Age, Cabin, Embarked and Fare features. Fare and Embarked are missing only a few values, while Age is missing in about one-third of the cases. The Cabin-column only has a non-null value for 295 passengers. We notice a relation between the passenger owning a cabin and its survival change. It seems that passengers who have a value for Cabin have a higher change of surviving, namely an average of 0.67 against 0.30 for passengers without a cabin. Another interesting fact we notice is the average survival change of women being 0.74, while men have an average change of 0.19 of surviving.

We are going to make a classifier which predicts whether a passenger has survived or not.

3.1 Methodology

First we must modify the data to make it suitable for a classifier. The process consists of determining which features we want to use and filling in empty values.

Filling in the empty values for Fare and Embarked are trivial since there are only 3 missing values. Thus we fill in the missing values with the most common value. Hereafter we modify the Cabin feature s.t. it returns true when a passenger has an entry in the Cabin column and false otherwise, and rename it to Has_Cabin.

Finally, we fill in the missing values for Age. We choose not to fill them with the mean age of 29.88 since this would create a large group op people being the exact same age. Thus for each missing Age value we draw a random integer within the range of the mean age minus the standard deviation and the mean plus the standard deviation. After this we categorize Age into 5 age range groups. We pick each range s.t. all groups have similar sizes.

Inspired by notebooks found online [2] [3] [4] we add new features to our data. We add the feature Name_Len containing an integer representing the length of a passengers name. Also we add a Title feature which is extracted from the Name feature. Hereafter we create the feature Is_Alone which is a boolean value set to true if the product of Parch and SibSp is zero.

After this we make a selection of features we are going to use: Pclass, Sex, Embarked, Has_Cabin, Is_Alone, Age_Cat, Title, Name_Len, Fare. We have dropped features that seem irrelevant such as PassengerId, Age and Ticket.

Hereafter we map all categorical values to numerical values in order for it to be valid input to the classifier.

The ground truth of survival for the passengers in the test set is not available without interacting with Kaggle. Since Kaggle only allows 10 uploads per day, we need to do local experiments. We do so by splitting up the training set into a smaller train and test set with size ratio 2 : 1. This allows for testing and determining the accuracy of our classifiers locally. Finally, we use the classifier with the best performance to predict the survival on the original test set and upload the result to Kaggle [5] to retrieve its accuracy.

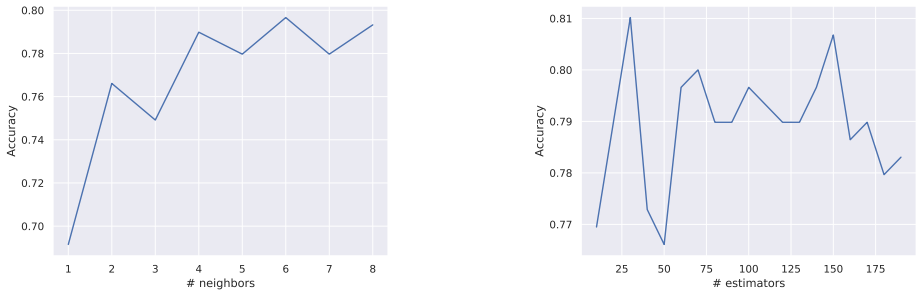
For the local testing we use 2 classifiers. We start with a Random Forest classifier [1] (RFC) using different numbers of estimators. We also use K nearest neighbors [2] (KNN) with different values for k. Finally we will validate our best performing classifier.

3.2 Results

In Fig. 3 we can see the accuracy versus k for KNN and the accuracy for RFC with different number of estimators. This shows us that the RFC with 30 estimators has the highest accuracy.

In Fig. 4 we see the importance of each feature for the RFC with the 30 estimators. In hindsight this plot shows us that the addition of features like Is_Alone and Has_Cabin are somewhat idle since their low importance score.

[2] https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8
[3] https://www.kaggle.com/sinakhorami/titanic-best-working-classifier/notebook
[4] https://www.kaggle.com/francksylla/titanic-machine-learning-from-disaster
[5] https://www.kaggle.com/



(a) Accuracy for different k of KNN.

(b) Accuracy of different # of estimators for RFC.

Fig. 3. Pie charts of how many people are registered from multiple study programs.

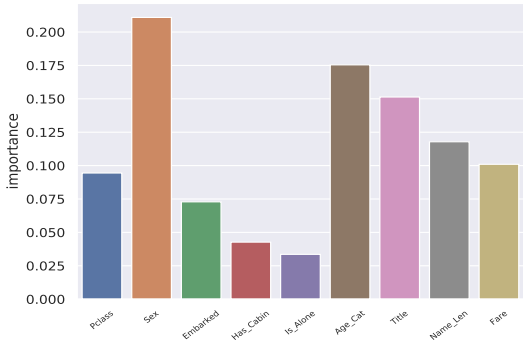


Fig. 4. Feature importance in the RFC (with 30 estimators).

Finally, we train the RFC with 30 estimators on our original train set, to predict the survival of the passengers in the test set and upload the result to Kaggle. This returns an accuracy of 0.7895 which grants us the 3080th position on the leaderboards. This accuracy is slightly lower than the accuracy shown in Fig. 3b, which is due to the train and test being different in both cases.

4 Research and theory

4.1 State of the Art solutions for a data mining problem

The competition discussed here is an online competition [6] whose goal it is to reward participants with knowledge or a large sum of money. The competition we have chosen sets a challenge where each participant should match an ingredient list to a geographical region in the world. This competition was held from 22/06/2018 to 24/09/2018 and its purpose was to gather knowledge. The dataset provided in

[6] https://www.kaggle.com/c/whats-cooking-kernels-only/rules

this competition is composed of geographical cuisine entries in text, for example "Indian" with the accompanying ingredients entries. The challenge is to match the ingredient with the appropriate cuisine, which is a classification challenge.

For this competition contestants are ranked by the algorithm score but are also grouped by popularity displayed by the amount of up-votes given by the community. We decided to select two contestants using the same methods. The winner by popularity obtained a 0.82119 [7] accuracy on the task where the contestant grouped with the highest score arrived at 0.82803 [8] accuracy. The winner according to the site leader board had an accuracy of 0.82783 [9] but did not provide any source code in contrast to the other two contestants.

Methodology In both source codes we discovered the two contestants using approximately the same algorithms. They used a Support vector machine (SVM) using different parameters, Term Frequency inverse document frequency (TF-IDF), One versus the Rest algorithms (OvR) and lemmatization. The most notable difference in approach is that one participant pre-processes the text before classification, while the other participant just classifies.

First, we briefly explain each mentioned technique. The purpose of a SVM is to classify labelled data in n homogeneous fields with a maximum margin between these fields, with n being dependent on the labels found in the data. The difference in accuracy for both contestants is due to their values for the C and γ parameters. The highest obtained accuracy was set on a high C which translate to a harder margin between classes. This is the result of imposing correct classification. A higher γ influences how close other data points will be in a homogeneous field. These parameters together could explain the difference in accuracy. However, after both contestants ranked the input data in terms of frequency of word in the whole data input using TF-IDF, only one contestant applied lemmatisation on the data. This pre-processing step where each erratum, misspelling of the same words, non-alphabetic character was corrected proved to be beneficiary. Taking such measures to pre-process the data and optimise the parameters for this classification task has shown to be essential for the contestants with the highest accuracy.

4.2 Mean Square Error vs Mean absolute error

There are different evaluation metrics that show the aptitude of a model. In this section we discuss Mean Square Error (MSE), Mean absolute Error (MAE) and R-squared error R^2. These metrics are formally defined as the following:

$$MSE = \frac{1}{N}\sum_{i}^{N}\left(y_i - \hat{y_i}\right)^2, \tag{1}$$

$$MAE = \frac{1}{N}\sum_{i}^{N}\left|y_i - \hat{y_i}\right|, \tag{2}$$

[7] https://www.kaggle.com/shivamb/tf-idf-with-ovr-svm-what-s-cooking/code
[8] https://www.kaggle.com/oracool/natty-svc-better-score-than-the-first-place
[9] https://www.kaggle.com/c/whats-cooking-kernels-only/leaderboard

with y_i denoting the actual expected value and $\hat{y_i}$ the computed value,

$$R^2 = 1 - \frac{\frac{1}{N}\sum_i^N (y_i - \hat{y_i})^2}{\frac{1}{N}\sum_i^N (y_i - \bar{y})^2}, \tag{3}$$

where the numerator denote the MSE of the model and the denominator variance of the model.

In the MSE error we see that each value is squared and thus cannot be negative. Moreover, this method is susceptible for high or low values since the squaring may contribute to the over or underestimation of the model. In other words a bad prediction may have a tremendous impact on the model output. However this disadvantage is also an advantage when the goal to evaluate the data is to handle outliers rigorously. On the other spectrum of evaluation we have the MAE where every data point is evaluated equally. This method still penalises big errors but less strictly than the MSE. It is robust for outliers due to taking each value equally probable [10].

In a dataset where output values are treated equally probable, say a finance sheet, a MAE is easier to fit the data without penalising outliers tremendously. Conversely, when MSE or MAE are used as loss function to evaluate the data in a model, the gradient of MSE is continuous where MAE is discontinuous around zero. This is less favourable when the goal is to apply a gradient descent algorithm on the data. Thus, if the goal is to compare models MAE is in the interpretation sense a favourable choice and not for a loss function. In the case of R^2 metric which is mostly used in the exploratory phase of the data, the aim is to explained how well selected independent variables explain the variability in the dependent variables. The value range of this metric is in theory from $-\infty$ to 1, where values closer to 1 indicates a model close to zero error, a perfect fit and values closer to zero or beyond the opposite. Table 2 shows that for the same values of the data the MSE is still higher than the MAE and that the R^2 indicates how well the prediction model is. This means that MSE ouput is emphasising more on outliers, thus producing larger results. Squaring the MSE results will still yield higher values than the MAE. If we apply gradient descent on this data, we will likely overfit the training data by the inherent property of the MSE for treating outliers. However, in the case of more perturbed values the MAE will not be completely different from this result as this metric treats every data input equally probable. The data discussed here is retrieved from the python scientific package, sci-kit [11].

Table 2. A comparison of the metric MSE, MAE and R2-Score applied on the boston housing price

	MSE	MAE	R2
Train	31.78	4.03	0.63
Test	26.39	3.79	0.66

[10] https://www.quora.com/How-would-a-model-change-if-we-minimised-absolute-error-instead-of-squared-error-What-about-the-other-way-around

[11] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.htm

4.3 Less obvious dataset

For this section we explored and analysed a SMS data set [12]. The goal is to discern spam from non-spam text messages. It is imperative to apply text processing since we have a large quantity of text data. Mainly, we have explored the possibilities of using a bag of words with tf-idf, stemming of words and the length of words as technique for data transformation. For a dataset where prediction of two binary classes is needed, we have used the Support vector machine (SVC) and Multinomial Naive Bayes modelling (NM) techniques. For the SVC we have chosen for the sigmoid kernel as we are interested in just two cases, spam or no spam. The γ parameter equals 1 since we want a good fit independent of the default value where it is set to the ratio number of features. In our case we explore the features with data transformation. The Multinomial Bayes classifier has not been optimised for this data with a gridsearch and α was chosen to equal 0.2 which translate to a classifier with less smoothing than the default value 1.0. We think lowering α imposes the classifier to be more strict on the data to classify. As our goal is to create a model with classification of spam or no spam, we value a model whose prediction power of no-spam to be more precise. Therefore we stratify our data when splitting the train and test set. This translates to the two score metrics: accuracy score and harmonic mean where the emphasis is on the precision of the model rather than the recall. Table 3 shows the first two columns where minimal data transformation is applied, i.e. no stemming nor considering the length of the words has been applied yet. The accuracy of SVC is lower than the harmonic mean but higher with the harmonic mean. However, with more data transformation not all predicted values are found during the prediction phase on the test data. Remarkably the Multinomial Bayes classifier does not suffer from this. Still the Multinomial Bayes classifier decreases in accuracy and harmonic mean when more data transformation is applied. This leaves us to believe that a gridsearch is necessary in order to obtain or at least maintain a high score.

Table 3. Measurement of the sms data set with different operations on the text messages with the label as class. The none values are due to the difference in length of predicted and expected values. We suspect this is caused by taking into account the length of the words and/ or lemmatisation of it. Acc denotes the accuracy and F_Beta denotes the harmonic mean. Len and stem are transformation of data, including length of text and/or the stem of each word respectively.

	Acc	F_beta	Acc stem	F_beta stem	Acc len	F_beta len	Acc len stem	F_beta len stem
SVC	0.974	0.942	0.979	0.961	0.867	None	0.866	None
NM	0.982	0.939	0.986	0.965	0.981	0.938	0.977	0.936

[12] https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

References

[1] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[2] James M Keller, Michael R Gray, and James A Givens. "A fuzzy k-nearest neighbor algorithm". In: *IEEE transactions on systems, man, and cybernetics* 4 (1985), pp. 580–585.