Data Mining Techniques
Assignment 1 (basic variant) - Group 116

Zhenyu Gao[1], Vincent Geschiere[2], and Krishnakanth Sasi[3]

[1] 2633314 zgo600 vu.gaozhy@gmail.com
[2] 2578811 vge900 vincent9@live.nl
[3] 11391952 ksi490 krishnakanthsasi@gmail.com

Introduction

In this assignment, we cover some important and basic topics for data mining. We begin by exploring a dataset and make some basic classification on a different dataset in section 2. Afterwards, we participate in a 'Kaggle' data mining competition on a relatively simple dataset called 'Titanic' in section 3. At the end, we gain some insights into research and more theoretical parts of data mining in section 4.

Task 1

A-Exploration

1. There are a total of 276 records with 17 attributes. There are different types of attributes, such as string, integer, time and Boolean. For data pre-possessing purposes, it is convenient to process the data of multiple-choice questions because the answers are constrained. However, for filling in the blanks, each person can have significantly different, and even wrong, answers which can lead to many complications while analyzing the data.
   We will start by plotting a graph showing the distribution of male and female students and a graph showing the distribution of majors within the class.

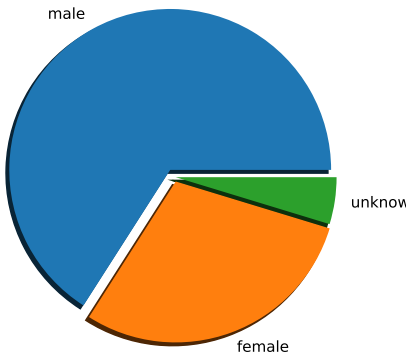

Fig. 1. Gender.

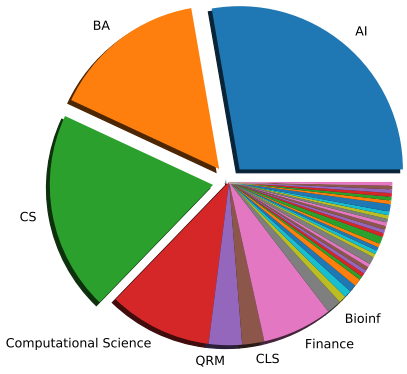

Fig. 2. Majors.

We can see that over 60% of the students are men (see Figure 1). It can also be seen that students from 'AI', 'BA', 'CS' and 'Computational Science' account for more than half of all the students (see Figure 2). We needed to change the data before we could analyze it, because the names of majors are written differently by different students. For example, some students wrote 'AI' while others wrote 'Artificial Intelligence'. In order to do this, we define a function called `merge` to merge the same majors together.

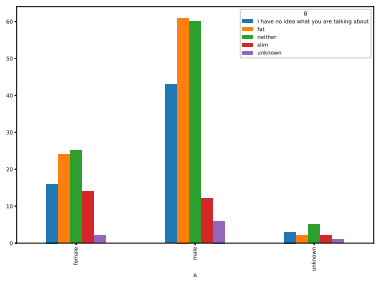

Fig. 3. Chocolate opinions by gender

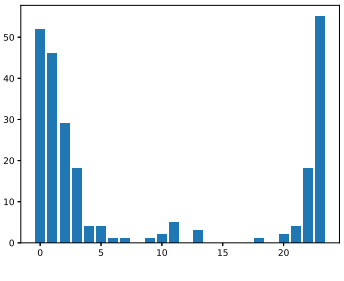

Fig. 4. Bed time statistics.

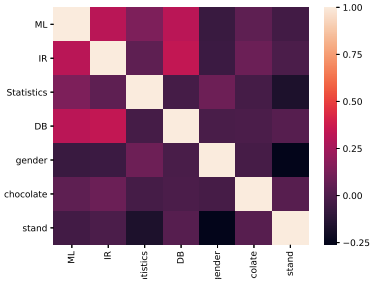

Fig. 5. correlation.

Afterwards, we plot the opinions of men and women towards chocolate (see Figure 3). As can be seen, more men think chocolate will lead to fatness than women.
We also plotted a barchart for the bedtime (see Figure 4). The data, however, is very unclean, which is why the values are so weird ('50' is not a bedtime). This is because students have answered the question in very different formats. Examples of different formats are: "01:30", "1", "I didn't" and "1 am". Therefore, this attribute requires to be cleaned in order to be able to analyze it correctly.
We also plotted a heat map to indicate the correlation of multiple attributes (see Figure 5). From the plot it can be concluded that there is strong correlation between the course 'Information Retrieval' and 'Database', while there is almost no correlation between 'whether someone is standing' and 'gender'.

Fig. 6. Word Cloud.

Finally, we made a word cloud to indicate the key words that make a good day for the students (see Figure 6). It is obvious that 'Sun', 'food' and 'sleep' can make a good day for many students.

2. It is really important to pre-process the data. For example, we should transform the data format to be unified, but that will take a long time, especially for open questions.

B-Basic classification/regression

1. We decided to use a different dataset than the ODI dataset. We chose to do this because the ODI set was very unclean with a lot of irregular values which would require a lot of work to clean. In order to find a new dataset, we decided to browse Kaggle to see if we could find a dataset covering an interesting topic and is useful for regression or classification.
We eventually found the "hearts.csv" dataset in the topic "What causes Heart Disease? Explaining the Model". Which, as the name suggests, is a dataset about heart diseases. The target value is either 1 or 0, implying whether the patient has a heart disease or not respectively. There were 13 parameters including "age", "Sex", "chol" (cholestoral measurement in mg/dl) and others.
We liked this dataset since we thought it would be interesting to see how machine learning can be used in the medical field. The dataset was also very useful for this assignment, because it was very clean (no pre-processing necessary!) and was a clear example of binary classification since the target value is either 1 or 0.

2. The first classification algorithm used is logistic regression. We used the default SKlearn solver, liblinear, and tried different amounts of folds for the cross validation to see what gave the best results. 2 folds gave an accuracy of 0.84 with a standard deviation of 0.03, 3 fold gave an accuracy of 0.83 with a standard deviation of 0.03 and 4 folds gave an accuracy of 0.82 with a standard deviation of 0.04.
   With these results we concluded that the higher the amount of folds, the lower the accuracy and the higher the standard deviation. We think this is due to the fact that the dataset is relatively small and therefore having many folds will result into a low amount of training data per fold and a low amount of training data results in a lower accuracy. If the dataset was bigger, the accuracy might have been higher with more folds.
3. The second algorithm we used was a Random Forest algorithm. We decided to set the amount of trees to 500 and we again tried different amounts of folds for the cross validation. Again, more folds resulted in a lower accuracy (0.83 for 2 folds and 0.81 for 3 folds).
4. We did notice that the accuracy for logistic regression is higher than that of the random forest algorithm. This surprised us, since random forest is a very strong algorithm. We think this is due to the small dataset. Decision trees can be weak in dataset with a lot of variation and with a small dataset there can be a lot of variation. Logistic regression is more 'resistant' against variation, so the small dataset might be the reason for logistic regression being more accurate than the random forest algorithm.

Task 2

A-Preparation

1. We first started by cleaning the data and afterwards we started to explore the data. We use the command `.info` to get the information of the data set `train` shown below.

RangeIndex: 891 entries, 0 to 890 **Data columns:** total 12 columns:

PassengerId	891 non-null int64
Survived	891 non-null int64
Pclass	891 non-null int64
Name	891 non-null object
Sex	891 non-null object
Age	714 non-null float64
SibSp	891 non-null int64
Parch	891 non-null int64
Ticket	891 non-null object
Fare	891 non-null float
Cabin	204 non-null object
Embarked	889 non-null object

dtypes: float64(2), int64(5), object(5) **memory usage:** 83.6+ KB

We plot the correlation between these attributes with a heat map(see Figure 7). Some relations are really strong. For example, the correlations between 'Deck' and 'Pclass', 'Title' and 'Survived' and 'Sex' and 'Survived' are really strong. While other correlation are very weak. 'Age' and 'Fare per person', for example, have a very weak correlation.
Afterwards, we plot the distributions of survivors for all attributes to see what attributes are significant to survival. These plots are shown below.
It can be seen that the distribution of 'Survived' is significantly different between the attributes 'Sex'(see Figure 8) , 'Deck'(see Figure 9), 'Familysize' (see Figure 10), 'Pclass'(see Figure 11) and 'Title'(see Figure 13). While 'Embarked'(see Figure 12), 'Age'(see Figure 14) and 'Fare per Person'(see Figure 15) seem to have little impact on 'Survived' (This can also be concluded from the correlation heat map Figure 7).

Fig. 7. correlation heat map.

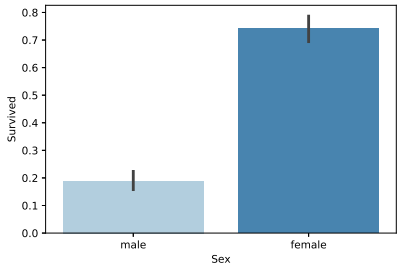

Fig. 8. Sex

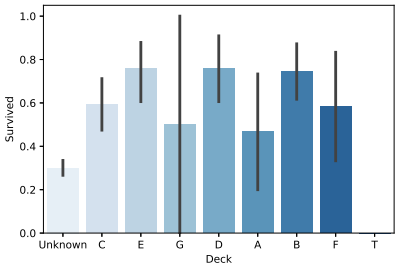

Fig. 9. Deck.

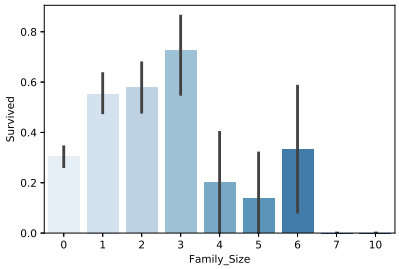

Fig. 10. FamilySize.

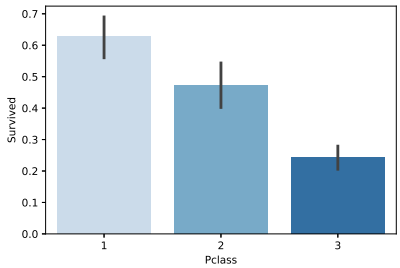

Fig. 11. Pclass

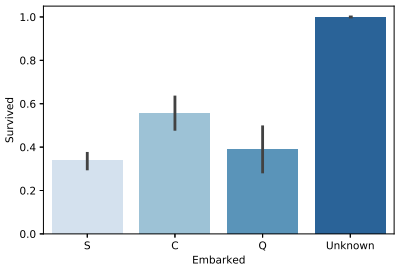

Fig. 12. Embarked

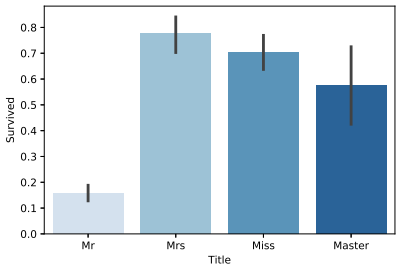

Fig. 13. Title.

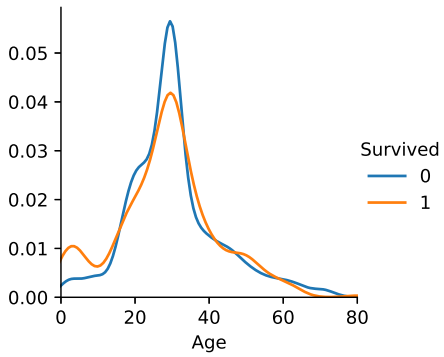

Fig. 14. Age

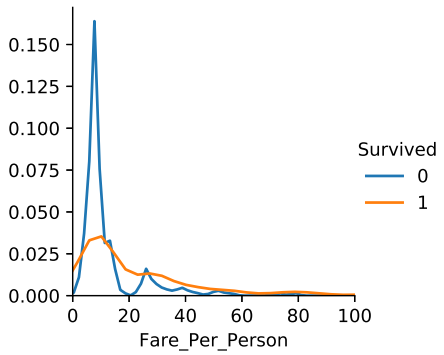

Fig. 15. Fare Per Person.

2. Finally, we decided to use the attributes sex, pclass, deck, the size of family and title as predictors for the classification because of the results shown above. But before we can use these predictors, we first needed to make some changes to the data.
   First of all, we change the value 'male' to 1 and 'female' to 0. In addition, we merge the attributes 'SibSp' and 'Parch' to get a new attribute: 'Familysize'. It is also more convenient to use 'Fare per person in a family' instead of 'Fare'.
   The attribute 'Cabin' is very unclean. Therefore, we extract the key information and create a new attribute called 'Deck'. We did the same for the names. We create the attribute 'Title' and passes on the value of 'Mr', 'Mrs', 'Miss' or 'Master' based on the name of the passenger.
   Finally, just like we did with sex, we change all string values to numerical values. For example, we set values 1, 2 and 3 for S, C and Q respectively for the attribute 'Embarked'.

B-Classification and evaluation

1. We used 3 methods to do the classification and the scores are 0.78468 for NN, 0.76555 for linear regression and 0.78947 for SVM. So the Support Vector Machine gave the best results, but only just. Even though Linear regression and SVM are actually regression algorithms, we could still use them as classification algorithms. The algorithms calculated the odds of survival. Therefore, we implemented that if the odds of surviving are higher than or equal to 50%, the passenger will survive. This way, we can still use regression algorithms for a classification problem.
2. The results are higher than we first expected, since we didn't use actual classification algorithms like random forest. However, there is still a lot of room for improvement for all three algorithms. We only used a few of the attributes, namely: sex, pclass, deck, familysize and title. Using different and/or more attributes can result in different and maybe even better results. But overall, the prediction of the algorithms are already relatively good.
3. Different classifiers have their own advantages and disadvantages. In principle, if the training set is small, high bias/low variance classifiers, for example Naive Bayes, have an advantage over low bias/high variance classifiers, for example NN, since the latter will most likely overfit. But low bias/high variance classifiers will be more precise as the training set grows because of a lower asymptotic error.
   Finally, SVM gets the best results in our model and has multiple advantages. It is highly accurate, contains nice theoretical guarantees regarding over fitting, and, with an appropriate kernel, can work well even if the data is not linearly separable in the base feature space.[1] However, SVM also has disadvantages. It can, for example, be very Memory-intensive and hard to interpret the model, so sometimes other algorithms like a Random Forest algorithm can be better.

Task 3

A-Research: State of the art solutions

1. The competition selected for this particular task was conducted as part of the Knowledge Discovery and Data Mining (KDD) Cup 2012, Track 1. The KDD Cup is an annual datamining/knowledge discovery competition organized by ACM. The goal of this particular competition was to predict user behavior on a Chinese social networking site called Tencent Weibo. There are two datasets spread out over 7 text files. They contain a sampled snapshot of users' preferences and various recommended items. The first two text files are information about whether a user follows a particular item or not when it is recommended. One of these files is used as the training file, and the other file is used as the test file. 4 of the remaining 5 text files give information about the profile and follow history or action sets of users in the recent past. The last text file contains information about the items. Natural Language is avoided throughout the data sets and user/item information is encoded numerically or uses single lettered strings. This is done to keep the privacy of the users intact and to avoid any unfair advantage to Chinese users.
   For the evaluation measure: assume m items in an ordered list are recommended to a user. Average precision at K is calculated for him/her by $ap@n = \frac{\sum P(k)}{\text{(total items clicked from the list)}}$. n is always taken to be 3, P(k) is the ratio of items clicked until the k^{th} item and is zero if the k^{th} is not clicked on. The average precision at 3 is then aggregated for all users in the test file, and this is taken as a final evaluation measurement for the participants.

2. Winner of the competition was a group from Shanghai Jiao Tong University consisting of nine participants. Extensive description of their approach along with participant names can be found here [3].
3. The winning team has combined feature-based factorization models with additive forest models.
4. The model uses feature based factorization to encode all the side information such user interaction, social network and taxonomical data. Additive forest model then takes in the sequential information available for the users. The two models complement each other well, and help in overcoming the most challenging parts of the problem ie; usage of heterogeneous data, and incorporating time evolution of the user behavior.[3]

B-Theory: MSE verse MAE

1. Two error measurements are used to compare the actual values to the predicted values, given by $A = (A_1, A_2, \ldots, A_n)$, and $P = (P_1, P_2, \ldots, P_n)$ respectively. MSE compares the values by averaging the squared difference between the values. MAE is the average of their absolute differences. They are formulated as follows:
$$MSE = \frac{\sum_1^n (A_i - P_i)^2}{n}$$
$$MAE = \frac{\sum_1^n \mid A_i - P_i \mid}{n}$$
2. In order to choose which measurement is preferred, one has to look at the task at hand and the type of error that one wants to minimize. If the intention is to reduce the outliers or large errors in prediction, MSE is preferred. If the error values are less than one or if reduction of outliers is not significant, MAE is preferred.
3. Mathematically, both error estimates would be equal when $\sum_1^n (A_i - P_i)^2 = \sum_1^n \mid A_i - P_i \mid$ .This happens in two cases. The first case is that the predicted and actual values are equal, i.e. zero. The second case is that if for each corresponding datapoint, the calculated error is 1 i.e; $A_i - P_i = 1 \, \forall i \in [1, n]$ Both of these conditions can occur by using logistic regression (the predicted variable is denoted as 1 or 0).
4. The types of regression methods used here to compare the error measures are linear regression, ridge regression, lasso regression and elasticnet regression. A dataset related to red variants of the Portuguese "Vinho Verde" wine is used for this purpose [2]. The dataset contains only physicochemical and sensory variables. All variables have numerical values. The objective of the regression method is to predict the quality of the wine from its physicochemical variables. The dataset is downloaded from Kaggle and is selected because it is easy to use and appropriate for this assignment. The pre-processed numerical values for all 12 attributes lets us directly delve into regression and error measure comparison.
The results of the comparison is tabulated below. As can be seen, the MLE is smaller than MAE for

Table 1. MSE vs MAE for various regression methods

Error measure	Linear regression	Ridge regression	Lasso regression	ElasticNet
MSE	0.45	0.51	0.59	0.57
MAE	0.52	0.55	0.63	0.63

all the regression methods, with linear regression containing the lowest value. This could be because of the lack of collinearity between the attributes. (Side note : The alpha value for Ridge, Lasso and ElasticNet is taken to be 1. This is to show the full extent of these methods, as setting alpha equal to 0 essentially reduces these models to that of a Linear regression model.)

C-Theory: Analyze a less obvious dataset

1. This is essentially a text classification problem. Bayesian text classification schemes are very useful for this end. We can try employing a multi-nominal Naive Bayes model for this specific case.

2. The data presented to us is in text format. In order for it to be parsed and encoded into a form accessible by our ML algorithm, we are going to use count vectorization. This method involves converting the text into a matrix of number tokens. This is done by making a single vector from whole text class consisting of all the unique words. The corresponding word counts for each instance is the transformed attribute value now.
3. The results of our model are displayed in Table 2. We use a 0.7/0.3 split for forming training and test data. This split ratio is taken arbitrarily. As can be seen the scores are high, deeming the model to be of high quality. The model could be improved further by finding the optimum test/train ratio for the given dataset.

Table 2. Classification report

	Precision	Recall	f1-score	Support
ham	0.99	0.99	0.99	1451
spam	0.93	0.94	0.93	221
average	0.98	0.98	0.98	1672

References

1. https://www.sciencedirect.com/topics/neuroscience/support-vector-machines
2. https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009
3. Combining factorization model and additive forest for collaborative followee recommendation, Chen, Tianqi and Tang, Linpeng and Liu, Qin and Yang, Diyi and Xie, Saining and Cao, Xuezhi and Wu, Chunyang and Yao, Enpeng and Liu, Zhengyang and Jiang, Zhansheng and others,KDD CUP,2012