Data Mining Techniques - Assignment 1 - Basic

Dylan Xinyu Fu 2656355 and Gongze Cao: 2656348

VU-DM-2020-118

1 Explore the dataset

The dataset we consider there comes from the answer from 280 students to a designed questionnaire. This dataset has 16 entries of different types, one is date, another is time, the rest of them are either categorical or numerical/string. Categorical variables are simply represented by integers to facilitate the later analysis. There are 8 entries having missing values, determined by either the "unknown" answer or an invalid answer. The missing value and specs of each attribute is shown in the table below:

Table 1. All attributes presents and the number of missing values.

Attribute	Data Type	NumMissedVal	Attribute	Data Type	NumMissedVal
Program	Str	0	Gender	Int	4
ML	Cate	3	IR	Cate	13
Stats	Cate	12	DB	Int	6
Neighbors	Int	49	StressLevel	Int	1
DM Compete	Int	61	Choco	Cate	33
Birthday	Str	49	rand	Int	16
gd1	Str	0	gd2	Str	0
bed time	Str	0	stand up	Cate	9

1.1 Preprocessing

We select several attributes to further analysis, consisting of "Gender", "ML", "IR", "Stat", "DB", "Neighbors", "Stress level", "Program", "DM", as they are prune to contains information about the person and our further prediction task.

Categorical Categorical data, such as "standup", "ML", "IR", are stored using integer as format, as most of them come from multiple choices, they are marked as missing value provided only specific option is chosen, such as "unknown". We ignore the missing value temporarily and assign them with the maximum option afterward when analysis.

Numerical There are a few of numerical attributes either stored in Integer or Float way, for those attributes who has a specific range, such as "StressLevel" and "DM", we mark all entries falling out of this range to be the closest range margin before reading it. After read all entries regardless of the missing values, we assigned the missed value with the median. Specifically, for later analysis of the classification, we mark all entries with "StressLevel" greater than 50 to be label 1, and all entries with "StressLevel" smaller than 50 to be label 0, per intuition of human nature.

String "Program" consists of many different strings inputted by the students, and we roughly classified all program to 10 classes, that is: "Artificial Intelligence", "Business", "Computational Science", "Computer Science", "Engineer", "Finance", "Information Science", "Others", "Physics", "QRM". After this we store this attribute in a one-hot way.

General We drop the duplicate, normalize the data per attribute level after all the pervious processing is done. The final feature of all entries in all is (278, 17).

1.2 Feature properties

Percentage The first thing we care about is the percentage of each property for an categorical variable, so as a warming up we plot a bar graph for the programs of all enrolled student:

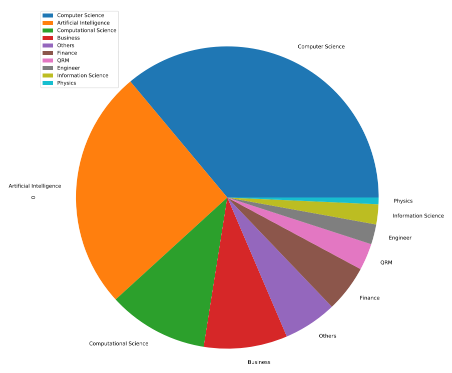

Fig. 1. The bar plot of the program distribution among all enrolled students. The top 3 most program is "Artificial Intelligence", "Computer Science" and "Computational science".

Description of statistics of all attributes We also show some of the most commonly used statistics of all attributes in this dataset, see3:

It is interesting to see that the median of stress level is around 0.40, meaning the major part of the students is not burnt, as it's not yet the final test. But meanwhile there is a quarter of students having "StressLevel" more than 0.65.

Correlations We selected several attributes that shows relatively strong correlation and plotted their correlations and corresponding p-values into a plot as below:

We found that the "Machine Learning", "Information retrieval" and "Database" attributes have p-values smaller than 0.05 and are strongly correlated. While other attributes generally have p-values larger than 0.05 with every other attributes, so they are independent.

Specifically we are considering the sex ratio in every program involved, the following is a bar chart of showing the sex ratio of all program listed above:

Table 2. The maximum validation accuracy of all classifiers through experiments.

Classifier	LR	KNN	GBDT	RF	SVM	GNB	DT
Accuracy	0.546	0.546	0.615	0.600	0.579	0.557	0.593

Table 3. The common data statistics upon all attributes.

	gender	ML	IR	Stat	DB	DM	stress level
mean	0.4307	0.6400	0.2981	0.8985	0.5294	0.2937	0.4207
std	0.4960	0.4808	0.4582	0.3025	0.5004	0.3632	0.2773
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00
max	1.00	1.00	1.00	1.00	1.00	1.00	1.00
0.25 quantile	0.00	0.00	0.00	1.00	0.00	0.01	0.20
0.5 quantile	0.00	1.00	0.00	1.00	1.00	0.10	0.40
0.75 quantile	1.00	1.00	1.00	1.00	1.00	0.50	0.65

We found that in almost all program the male students are more than the female students. But for the largest program involved, Computer Science, the male students and female ones are of almost equal number.

1.3 Classification

The task we are interested is whether we could infer the stresslevel of the student using other features. The feature set we choose is:"program" "gender","machine learning","information retrieval","statistics","databases","DM". The label comes from the way stated in 1.1. We try with several different classifier and tune the hyper-parameters based on the cross-validation accuracy, and we choose the fold to be 5 and fixed for all classifiers.
The results are shown below:

– For Linear Regression(LR), among penalty=["l1", "l2"], C=[0.1, 1, 10], we choose penalty="l1" and C="10". Since the feature might be redundant, so we tend to choose a larger C at the beginning and the results seem to align with the observation. We also hypothesize that l1 penalty might lead to more generalized model, but the experiments shows l1 and l2 penalty differs little.
– From the experience of LR, we use C=100 when training SVM and it gives boosted performance in terms of Cross-validation accuracy.

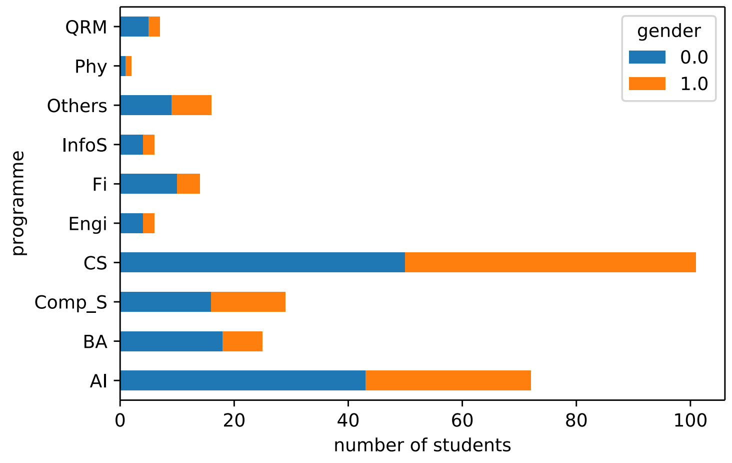

Fig. 2. The bar chart of all genders ratio in every program, where 0.0 stands for male students and 1.0 stands for the female students.

- For KNN classifier, among k=[1,2,3,4,5,6] we empirically choose k=6 as it gives better cross-validation accuracy. We think the larger k might suffer from over-smoothing.
- For decision tree, among max-depth=[1,2,3,4,5,6], we empirically find smaller depth lead to better generalization performance, when depth=2 the train-valid tradeoff is optimized.
- For Gradiant-boosting-decision-tree(GBDT) and Random Forest(RF), from the experience of decision tree above, we choose smaller max-depth again and it leads to good performance. For GBDT we used 20 base estimators and 2 as max-depth, for RF we used 200 base estimators and 4 as max-depth.
- For Gaussian Naive classifier we used the default setup as a baseline.

In general, the GBDT works the best and used much less parameters than its opponent RF. It might indicate the sturcture of GBDT fits better for this kind problem than any other classifier.

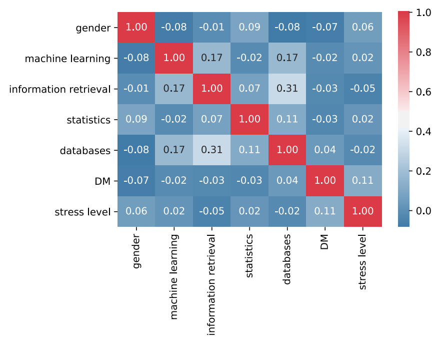

Fig. 3. Correlation coefficient and p-values for 7 features. If p-value is larger than 0.05 we can think the two variables are independent.

2 Kaggle: Titanic Survivals Prediction

2.1 Data Preparation

Data at a glance At first, we merged the train and test data (1309 rows in total) and demonstrated the information of the full data in the table below.

Table 4. Explanation of train and test data

Attributes	Name	Sex	Ticket	Cabin	Embarked	PassengerId	Pclass	SibSp	Parch	Age	Fare	Survived
Data type	Object	Object	Object	Object	Object	Int	Int	Int	Int	Float	Float	Float
#Missing values	0	0	0	1014	2	0	0	0	0	263	1	0

Potential correlation between features and labels And we found the following correlations:

- Gender and Survived: the female has (far more) higher survival rate than males, as shown in Fig.4.
- Pclass and Survived: passengers in a better class have a higher survival rate, as shown in Fig.4.
- Age and Survived: passengers whose age is less than 10 have a higher survival rate, as shown in Fig.5.
- Fare and Survived: passenger paying Fare less than 20 has a lower survival rate, as shown in Fig.5.

Besides, we grouped SibSp and Parch to familysize and familynum features. We have also grouped passengers having the same ticket number.

Before we move on, we found that the distribution of Fare is skewed to the left, as shown in left Fig.4. Thus, we logarithmic the "Fare" data to prevent it from having uneven weights for later classification, as shown in right Fig.4.

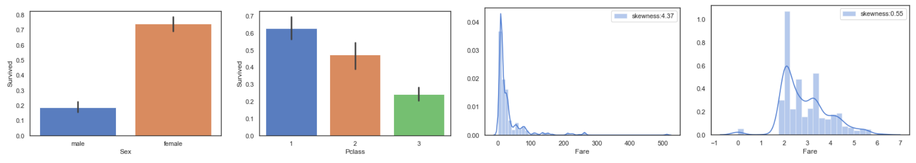

Fig. 4. Left figure shows females have (far more) higher survival rate than males. 2nd figure shows passengers in better classes have higher survival rate.3rd and 4th figure shows the original and logarithmic Fare distribution.

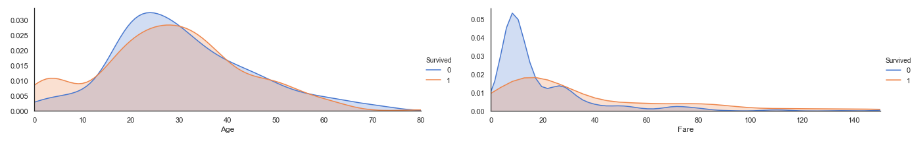

Fig. 5. Left figure shows passengers with age less than 10 has higher survival rate. Right figure shows passengers paying fare less than 20 has lower survival rate.

Handle missing values There are four types of data having missing values, 1014 values in Cabin, 263 values in Age, 2 values in Embarked, and 1 in Fare. Therefore, we employed the following methods to fill in the missing values.

– We filled in 1014 missing values of the Cabin with U, meaning "unknown" since they are irregular categorical values.
– For the 2 missing Embarked data, we set them both to "S", Due to that people embarked from "S" (Southampton) takes up nearly 70% total population rather than 21% and 9% for people embarked from "C" and "Q" respectively
– To fill 1 missing Fare, we fill it with the mean of Fare values from the 3rd class passengers embarked from Cherbourg since we know the passenger having missing Fare is one of them.
– To fill in 263 missing ages, we conducted a random forest algorithm (with 500 trees) to predict the Age regarding the other attributes. The cross-validation score of the RFT model is 0.5870 .

Feature engineering In this subsection, Here, we conducted feature engineering to find the underlying features having impacts to target.

We suppose different titles of passengers may have an impact on the survival rate since titles can reflect 1. the gender sometimes (Mr and Mrs) 2. social class of passengers.

As shown in the left of Fig.6, passengers titled as Mr has the least survival rate and follows the passenger titled as officers. Passenger titled with Royalty, Mrs and Miss has a generally higher survival rate

Besides, we did the same finding for passengers in a different cabin. But since 70% of data in the Cabin column are unknown, we suppose the result is not influential to the final classification.

Dimensionality reduction To reduce the dimensionality of data, we artificially dropped part of features and kept the following attributes: 1. Survived, age, Embarked, Fare, Parch, Pclass, Sex, SibSp, Title, familyNum, familySize, Deck, TickCot, TickGroup.

To find the correlation between features and target, we plotted the heat map, as shown in the right Fig.6. And we decided to drop features of familyNum, SibSp, TickCot, Parch since there are less correlated to target (Survived).

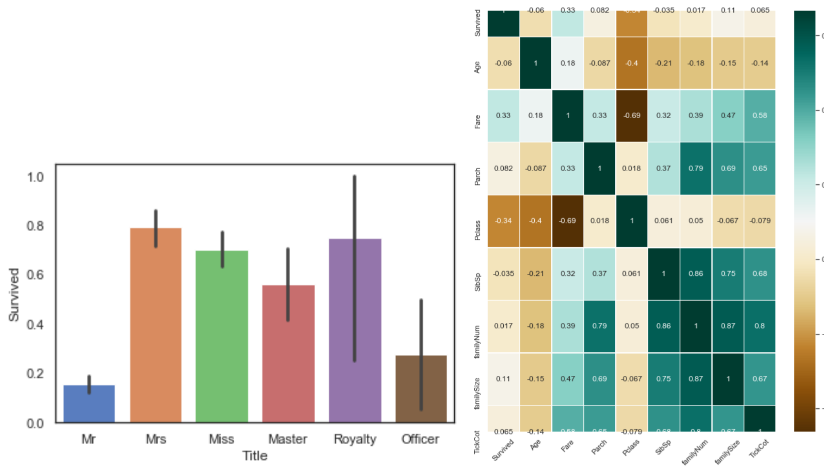

Fig. 6. Survival distribution among titles (left). Heat map of correlations among available attributes (right)

2.2 Model setup and evaluation

Model comparison and selection We quickly set up some common classifiers provided in sklearn to compare their performance by cross-validation scores. We choose the following classifiers: SCV, Decision Tree, Extra Trees, Gradient Boosting, Random Forest, KNN, Logistic Regression.

As shown in Fig.7. Gradient Boosting (GBC) and Logistic regression (LR) have slightly better CV mean as 0.831 and second highest CV mean as 0.829. So we decided to tune the GBC and LR models further.

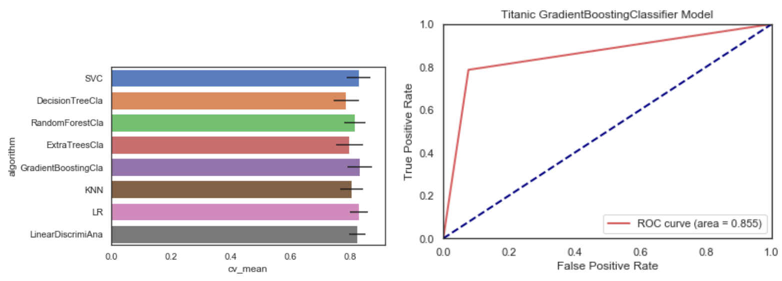

Fig. 7. Cross validation score for tested common classifiers (left). ROC curve for GBC.

Model tuning: GBC and LR The main idea for Logistic regression is that, we construct a linear relation ship between the features and log-odds of event (such as "survival") and then predict the probability the event is triggered. And the idea for GBC is combining weak learners to a strong learner. Also it optimised the learner iteratively by using differentiable loss function. To find the best estimator for above models, we use Grid Search CV provided by Sklearn to search for the optimum model. The parameters grid for both models is set as the following:

- For GBC, we used the following parameter combination for grid search: deviance loss function, $n_{estimators} = [100, 200, 300]$ $learningRate = [0.1, 0.05, 0.01]$ $depth_{max} = [4, 8]$ $minSamplesLeaf = [100, 150]$ $features_{max} = [0.3, 0.1]$
- For LR, we set the inverse of regularisation strength as $C = [1, 2, 3]$ with penalty being either l1 or l2 regularisation method.

For the two obtained estimators, we performed cross-validation to compare their accuracy with Kfold where $k = 10$. And eventually, we earned the CV score for GBC is 0.842 and 0.832 for LR.

2.3 Result and discussion

Next, to find the best model from selected GBC and LR, we checked the confusion table and corresponding ROC (Receiver operating characteristic) curve as well as AUC (the area under the ROC curve).

With the confusion table shown in Tab.5, we conclude GBC is better than LR classcifier in this case for the following reasons:

1. GBC has good Sensitivity as $TPR_{GBC} = 0.9234 > TPR_{LR} = 0.8743$
2. GBC has good false positive rate as $FPR_{GBC} = 0.1228 < FPR_{LR} = 0.2017$
3. GBC has better AUC since $AUC_{GBC} = 0.824 > AUC_{LR} = 0.810$. A ROC figure for GBC could be seen in the right Fig.7

In conclusion, we decided to use GBC for the final prediction. Thus, we submit predicted result by selected GBC and obtained the Kaggle score as 0.79904 which is publicly ranked at 1907 out of 18319, nearly top 10.4%.

	True (1)	False (0)
GBC Positive	507	42
GBC Negative	73	269

	True(1)	False(0)
LR Positive	480	69
LR Negative	80	262

Table 5. Confusion matrix for the chosen GBC and LR estimators

3 Research and theory

3.1 TASK 3A: State of the art solutions

In this section, we aim to demonstrate a winner solution for a Kaggle compmetition: IEEE-CIS Fraud Detection, which is hosted 7month ago (roughtly Sept of

2019). Most of the literature comes from the following Kaggle forums (clickable): Summary, Part 1, Part 2

The goal of fraud detection is to predict the fraudulent client given the data of Vesta's real-world e-commerce transactions and identity of user.The evaluation used is the accuracy determining the $isFraud$ for each transactionsID.

The 1st place (winner) is Chris Deotte who assembled 3 high scoring models: CatBoost LGBM and XGB).The main idea is that rather than find the fraudulent transaction, he tried to find the fraudulent clients by identity information and link all his/her transaction as fraudulent.

There some magics making him stand out.

- He performed in-depth feature engineering during EDA process. A trick is time consistency check. He firstly train a model with a few features and test the AUC. He found some features are hurting the model. Namely, the prediction derived from the first couple months are different from the prediction for the last month.
- He also employed various model validations. Such as he used kfold to split the data monthly. And test the CV score by: train 4 month, skip a month, predict last month.
- He ensembles three high-scoring models (CatBoost LGBM and XGB) to obtain a good result.

3.2 TASK 3B: THEORY - MSE VERSUS MAE

Mean squared error is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

where there is a vector of n predictions from a sample of n data points. Y is a vector of observed values, $\hat{Y}$ is the a vector of predicted values.

While mean absolute error is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|$$

For MSE, the optimal prediction will be mean target value. And MSE is sensitive to outliers. Thus, we would use MSE if we believe the targets are normally distributed around a mean value and we want it be sensitive to outliers.

For MAE, the optimal prediction will be median target value. Thus, we would use MAE if the distribution is multimodels or we dont want prediction to be sensitive to outliers.

It may happen the $MSE = MAE$.

Assuming residual $r = |Y_i - \hat{Y}_i|$, given $MSE = MAE$ we have

$$\vec{1}^T r = r^T r$$

By rearranging the formula, we obtained

$$(\overrightarrow{1}^T - r^T)r = 0$$

Thus we conclude any residual vector satisfying above formula would result in $MSE = MAE$. Two example solutions are $r = \overrightarrow{0}$ and $r = \overrightarrow{1}$

As shown in Tab.3.2 below, we compared Linear regression and Lasso applied to two artificial data with outlier and without outlier. We found MSE is more sensitive to outlier for both models. And MSE is more appropriate metric in this case, since the data are spread around the mean target.

	MAE	MSE	MSE/MAE
Linear Regression	0.312	0.799	2.57
Lasso	0.627	1.144	1.82

	MAE	MSE	MSE/MAE
Linear Regression	0.242	0.083	0.344
Lasso	0.544	0.426	0.783

Table 6. Regarding Linear regression and Lasso, the left figure reveals the MSE and MAE for dataset having outliers. And the right figure reveals the MSE and MAE for dataset having outliers removed.

3.3 Theory: Analyze a less obvious dataset

– Firstly do some cleaning on the given dataset. We extract the labels of the dataset by convert the first word from ”ham” to 1 and ”spam” to 0. Then we use the text after the label as the raw feature. After filtering out all punctuation in the raw feature, we used TF-IDF to transform the raw feature to a numerical vector of size (9534,). The final shape of the feature matrix and label are (5574, 9534) and (5574,) respectably.
– We then trained 6 classifiers as did in section **??**, the cross validation fold is set again as 5 and results are shown in the table below:

Table 7. The cross validation accuracy of all classifiers through experiments on SmsCollection.

Classifier	LR	KNN	GBDT	RF	SVM	DT
Accuracy	0.956	0.927	0.966	0.970	0.866	0.959

All the classifiers use the default parameters provided by the sklearn. We did not perform further tuning as the performance is already very impressive. Among all the classifiers the decision tree based classifiers perform the best, GBDT, RF and DT all achieve validation accuracy above 95 percents. To further improve the performance we advice not only further tuning with the base estimators number and max depth in DT-based classifiers, but also include more linear classifiers and create a boosting mega classifier.

References