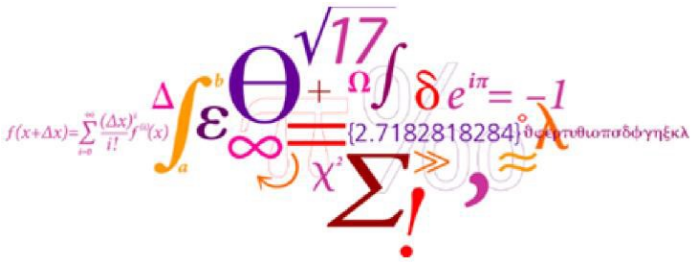
02450 Introduction to Machine Learning and Data Mining

Supervised learning: Classification and regression

Report 2

Name	StudentID	Contribution
Longfei Lin	s185882	Classification, Discussion
Sai Chaitanya Annadatha	s181739	Regression part b
Iraklis Chrysikopoulos	s182995	Regression part a



09/04/2019

1. Regression PART A

1.1 Description of attribute we used

In this project, the data that is used is based on many attributes in order to achieve the prediction of heart disease to patients. Specifically, the attribute that is used are ''oldpeak. This ''oldpeak'' attribute is continuous, so with linear regression we will visualize some results. Also, there is no need to make feature transformation in our attributes such as one-of-K coding. From project 1, the data matrix X of our data are transformed in a way that mean is 0 and standard deviation 1, so a better visualization of our data is succeed. In the code the file ''Heart_2.mat'' contains all the data for the regression. In the matrix X, there are 12 attributes and 302 observations for each of them. The matrix Y contains continuous data of the ''oldpeak'' attribute.

Attribute	**Oldpeak**
Count	303
Mean	1.039604
Std	1.161075
Min	0
25%	0
50%	0.8
75%	1.6
Max	6.2

Table 1: Predicted attribute 'Oldpeak'

1.2 Introduction of regularization parameter

For this question a parameter λ is introduced in order to make regularization in our data and with linear regression to predict what will happen. First, we make a selection of our data, some of them will be TRAINING data and the rest TESTING data. We separate our data in blocks. For example, the first blocks i.e. 75% of our data will be training data and the 25% of the rest will be testing data. With the method of K=20-fold cross-validation, we can complete a whole examination of all the blocks and select which block should be training data and which will be testing data. It summarizes the results at the end, as cross validation uses them all and presents the best model. As it is shown in the lectures algorithm 5, some values of the parameter λ will show the amount of generalization error.

Trying different λ values, the optimal value will be selected in order to get the best model for our predictions.

We can also solve for the optimal weights to get:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} E_\lambda(\boldsymbol{w}) = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}(\boldsymbol{X}^\top\boldsymbol{y}) = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})\backslash(\boldsymbol{X}^\top\boldsymbol{y}).$$

The Ridge and Lasso Regression are used together in order to get the best model and best fitting line. Overfitting will be done, as with Ridge regression maybe we get low bias and high variance and with Lasso there is high bias and low variance.

In this way, some results will be extracted also for the generalization error of our potential models.

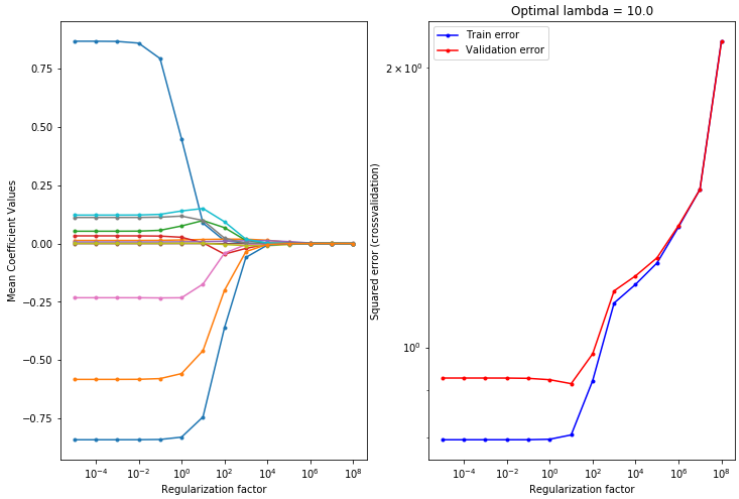


Fig: 1 Optimal lambda in our regression model

1.3 The results of predicting on our data set

There are the results for our training and testing data for the errors.

Linear regression without feature selection:

- Training error: 0.7775218435536388

- Test error: 0.8548581351366968

- R^2 train: 0.4206286197938574

- R^2 test: 0.351242487008715

With selection feature.

Regularized Linear regression:

- Training error: 0.784719590722744

- Test error: 0.8629690201700881

- R^2 train: 0.41526520943270323

- R^2 test: 0.34508708251978215

In the previous graphs, we can see our attributes how they are contributing to our model in

terms of training and testing data. Our code computes and generates the best case for parameter λ and generalization error.

When λ is small, the weights are large indicating high variance but low bias. When λ is larger, the weights become smaller indicating lower variance. [1]

The optimal lambda is 0.1 and in the second figure we can see that the validation error is near the train error. Also, λ=0.1, λ=1 and λ=10 is tested as it can be seen in the python file.

2. Regression PART B

2.1 Baseline, linear regression and ANN

Two level cross validation is implemented using Algorithm 6. and baseline model with no feature values are implemented in Two level cross validation and predicted Y on the test data. ANN model is also implemented in using two level cross validation.

For Baseline model, linear regression model with no features is implemented. This computes the mean square error on the train data and test data. The interpreted values are as follows

Linear regression without feature selection:
- Training error: 0.7812252532076115
- Test error: 0.8548330798642233
- R^2 train: 0.4184089511032533
- R^2 test: 0.3484378681795293

The training error and the test error is computed with the following formula:

$$E(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \tilde{\boldsymbol{x}}_i^\top \boldsymbol{w}\right)^2.$$

On analysing the results the test error is slightly higher than the training error in the baseline model.

For Regularized Linear Regression we computed 2 level cross validation and the generalization error we got is shown in the figure below.

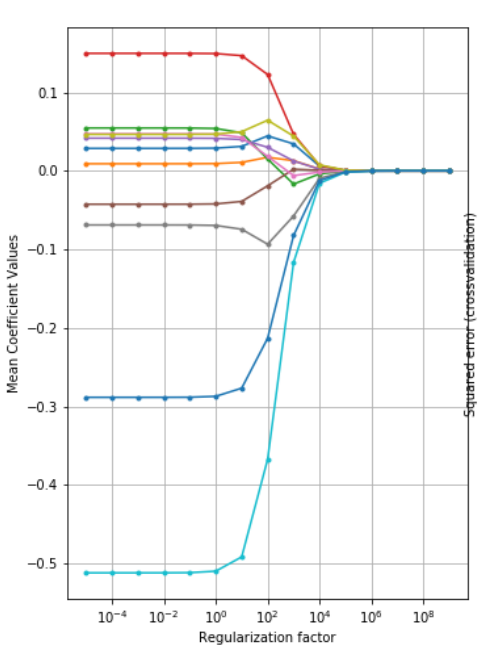


Fig: 2(a)

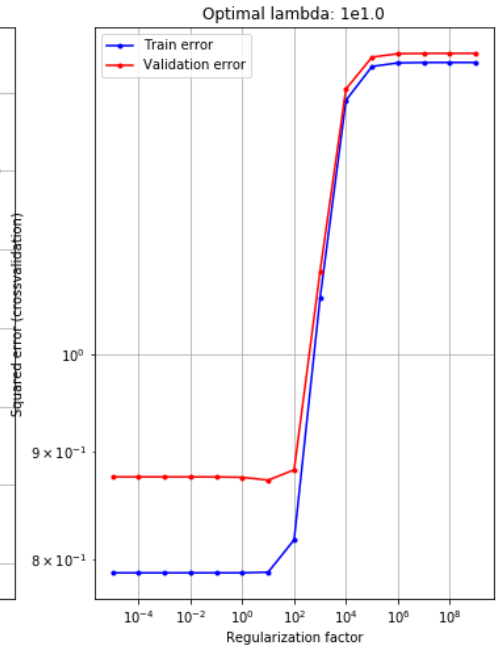


Fig: 2(b)

For the Regularized linear regression model the λ values are chosen between 10^{-5} to 10^{9}. One can observe that, as λ increases the mean coefficient values decreases. When the there is less λ the model is having high variance and low bias. When there is more λ the model is having low variance and high bias. It can be observed from the above Figure 2(b) that the minimum test error can be found around $\lambda = 10^{0}$. After the $\lambda = 10^{0}$ the error starts increasing.

Regularized linear regression:

- Training error: 2.424352245067184
- Test error: 2.418912805464008
- R^2 train: -0.8052231108473349
- R^2 test: -0.8781909105713602

The training error and the test error doesn't have much difference.

For **ANN** we computed two-level cross validation. The below figure shows how the input data is predicted based on the Neural Networks. The neural network consists of 3 units. Input unit, output unit and the hidden unit. The red and green lines in the below figure are the weighted connections. Here to save the time we reduced the number of iterations to 1000. The Average error for the ANN model is 0.861.

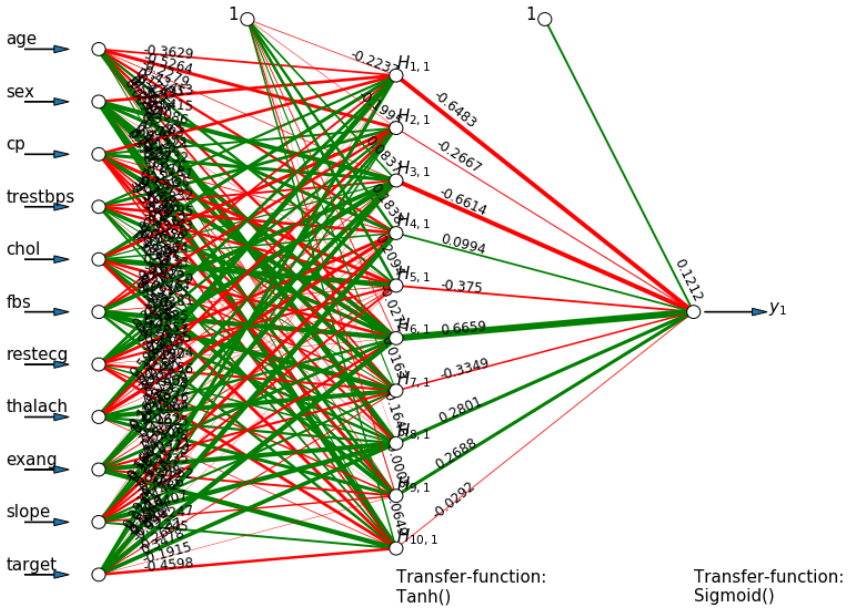


Fig: 3 simple ANN

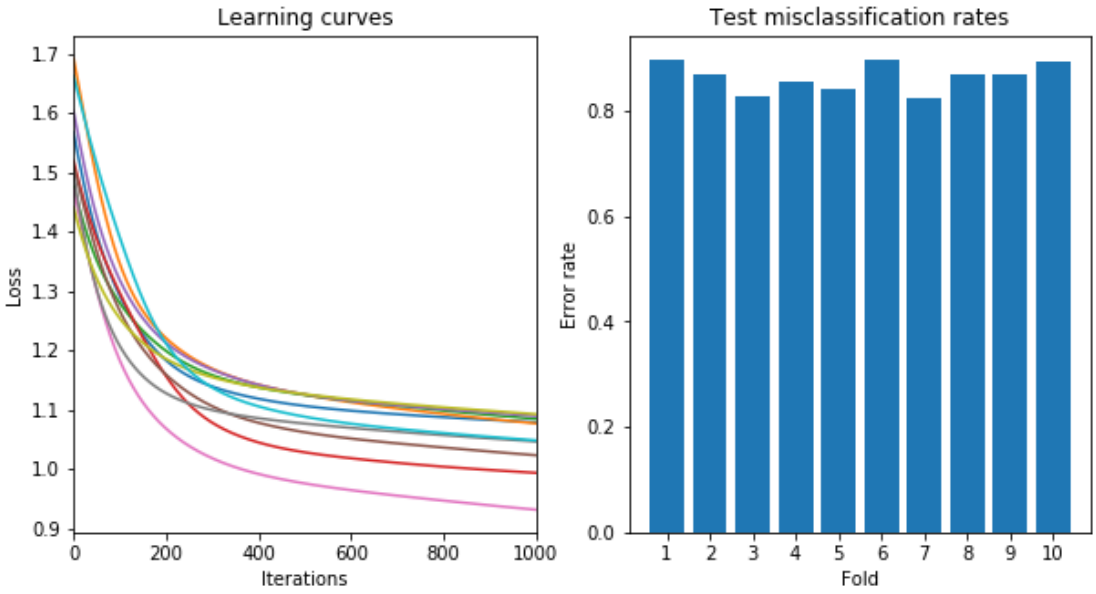


Fig: 4 Learning curves & misclassification rate

From the above figure we can clearly see that this model needs some more iterations to check the learning curves. But we can see that as the more the iterations the lesser the losses. We can clearly interpret that 3rd Fold has low error rate compared to the all folds.

2.2 Model selection by two-level cross-validation

K1-Outer fold i	ANN		Linear Regression		Baseline
	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	7	1.5052	9.5e-02	2.78226	1.01122
2	5	0.27472	2.6e+01	1.68387	0.63916
3	7	1.5052	1e+02	2.29645	1.24533
4	5	1.68172	1e-05	2.28033	0.571679
5	6	0.774791	1e-05	1.72667	0.795373
6	3	1.1569	5e+01	1.9	0.636937
7	5	0.27472	1.3e+01	1.99333	0.751642
8	1	0.578793	2.2e+02	3.857	0.97545
9	1	0.578793	1e-05	4.14633	1.4054
10	5	0.27472	2.68e+01	1.595	0.581457

Table 2: model selection by two-level cross-validation

If we compare the linear regression and the Baseline model, it can be concluded that the Base line model without any features is the best model. It has low errors. This is because our observations are very low and this data is classification based. When comparing our data with ANN model, we can observe that the average of test errors is 1.065. whereas for baseline the average is 0.08206. So on comparing the ANN is better than the Linear Regression. But on comparing with the Baseline ANN is having large error. The number of iterations used here are 1000, if we increase our iterations the model will give better results than the Baseline Model.

2.3 Statistical evaluation

In this part we need to compare three models with each other, for that we use credibility interval method. we can use this interval to determine whether one model is better that the other. We can calculate the difference and generate a 95% credibility interval based on this difference.

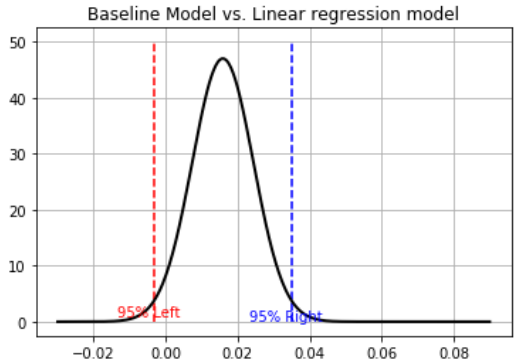


Fig: 5 Credibility intervals between baseline and regularized linear regression

The above figure is the comparison between Baseline Model and Linear regression Model.

Hear we can observe that the two models is in between -0.3 and 0.03. where this has negative lower bound and positive higher bound which tells that there is no big difference between the two models. We can also interpret that by looking at the mean value. We can see that the mean value is almost '0', representing not much difference between two models.

Model	***K***	***V***	***Mean***	***Sigma***	θ_L	θ_H
Baseline vs Linear Regression	10	9	0.0158	0.008	-0.0033	0.035
Baseline vs ANN	10	9	-0.0008	0.1858	-0.421	0.492
ANN vs Linear Regression	10	9	-1.4688	0.2367	-2.004	-0.933

Table 3: statistic evaluation for three models

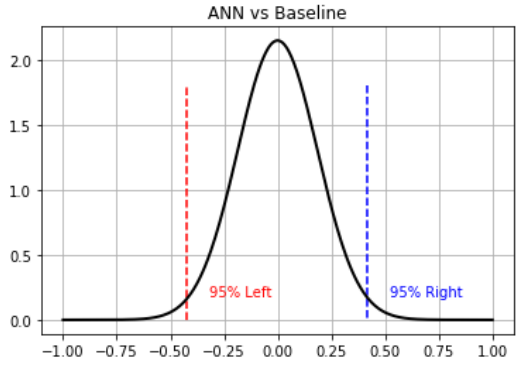


Fig: 6 Credibility intervals between baseline and ANN

The above figure is the comparison between ANN vs Baseline. Hear we can observe that the two models are in between -0.4 and 0.4. where this has negative lower bound and positive higher bound which tells that that there is no big difference between the two models. We can also interpret that by looking at the mean value. We can see that the mean value is '0', representing not much difference between two models.

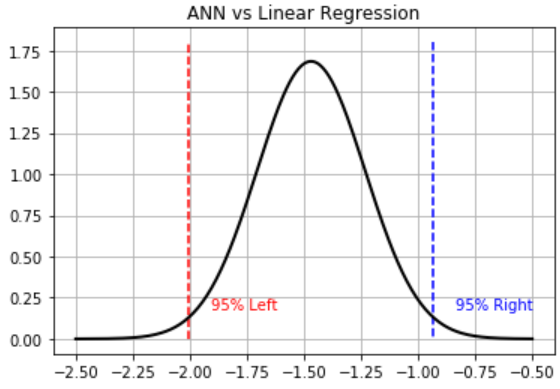


Fig: 7 Credibility intervals between regularized linear regression and ANN

The above figure is the comparison between ANN vs Linear Regression. Hear we can observe that the two models are in between -2.0 and -0.9. where this has negative lower bound and

negative higher bound which tells that that there is significantly big difference between the two models. We can also interpret that by looking at the mean value. We can see that the mean value not close to '0', representing the difference between two models.

3. Classification

3.1 Classification problem

We use a data set which classified if a person has heart disease or not. We try to use this data set to create different classification models which can predict if a person has heart disease or not. So, it is obvious that our data set is a binary classification problem.

Fig: 8 Aim of Classification

3.2 Classifiers for our data set

3.2.1 Baseline Model

In classification, we use a basic logistic regression model with a bias term and no features as our baseline model, this baseline model can be used to predict whichever label occur more frequently in our data set.

We used our data set to fit a baseline model, and the results as follows:

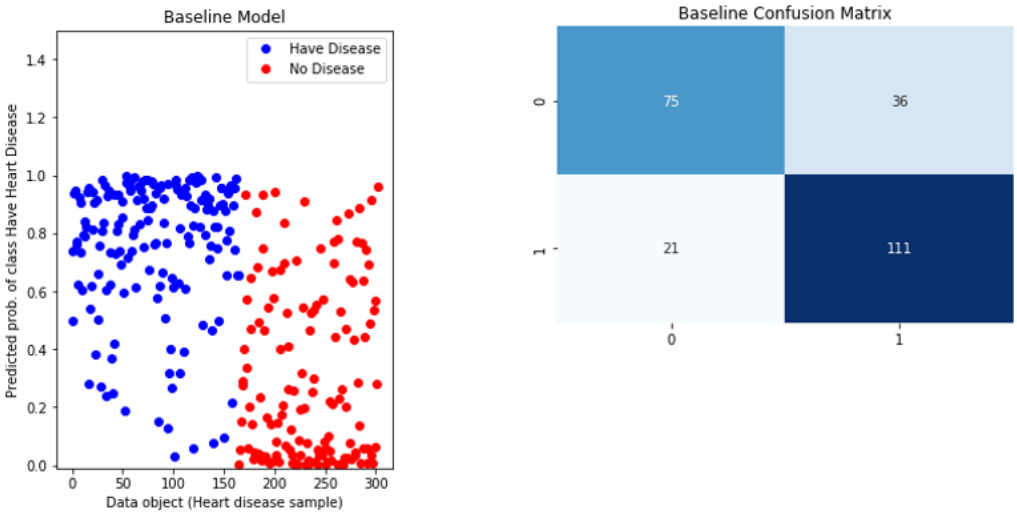


Fig. 9 Performance of baseline model

From the left-hand pane, we get the probabilities of whether a person have heart disease from baseline model, and based on these probabilities, we can calculate the predict value of our test data and determine which class the data belong to. In the right-hand pane, we generate a confusion matrix of test data to show how often the classifier is right. We can also calculate the accuracy and error rate based on following expressions:

$$Accuracy = \frac{TN + TP}{N}, \quad Error\ Rate = \frac{FP + FN}{N} = 1 - Accuracy$$

So, the results will be:

- Test Error: 23.45679012345679%
- Accuracy: 76.5432098765432%

3.2.2 Logistic Regression with regularization term

In this section, we use regularization term λ in logistic regression model to manage model complexity. Based on the principle of regularization term in textbook [1], we know that when λ is large, the models have low variance but high bias; when λ is large, the models have high variance but low bias. Adding regularization term does not improve the performance on the data set that the algorithm used to learn the model parameters. However, it can improve the performance on new data, which is exactly what we want.

So, in our codes, we designed a for-loop to find the optimal λ from 10^-5 to 10^3. We use

```
train_test_split(X, y, test_size=.50, random_state=0, stratify=y)
```

to fairly split the data into two part: train data and test data, and then we selected the optimal λ by comparing their test error. The optimal λ and the Confusion matrix of Logistic

regression as follows:

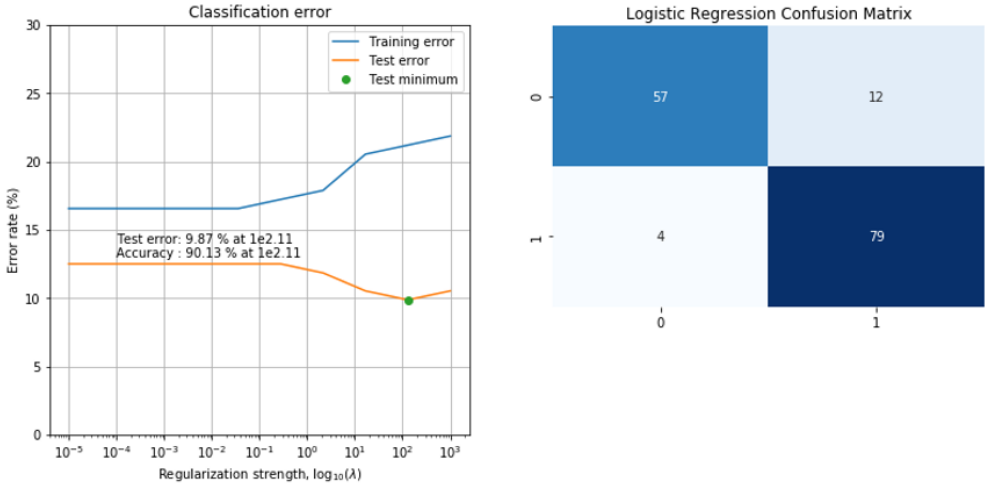


Fig. 10 Performance of Logistic Regression

In the left-hand pane, we can find the optimal $\lambda = 10^{2.11}$, we also get the minimum test error when λ equal to this value. In the right-hand pane, we generate a confusion matrix from test data using logistic regression, the accuracy and error values as follows:

- Test Error: 9.868421052631579%
- Accuracy: 90.13157894736842%

3.2.3 Decision Tree with pruning

Decision tree builds classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. In our codes, we use a technique called Pruning in order to control the tree complexity.

We designed a loop with different maximum tree depth and implement this parameter by adding it into function such as

```
DecisionTreeClassifier(criterion='entropy', max_depth=t)
```

Then we can select the best tree depth based on their test errors. We also use 'Entropy' as our criterion because it performs better than 'Gini'. The results of decision tree as follows:

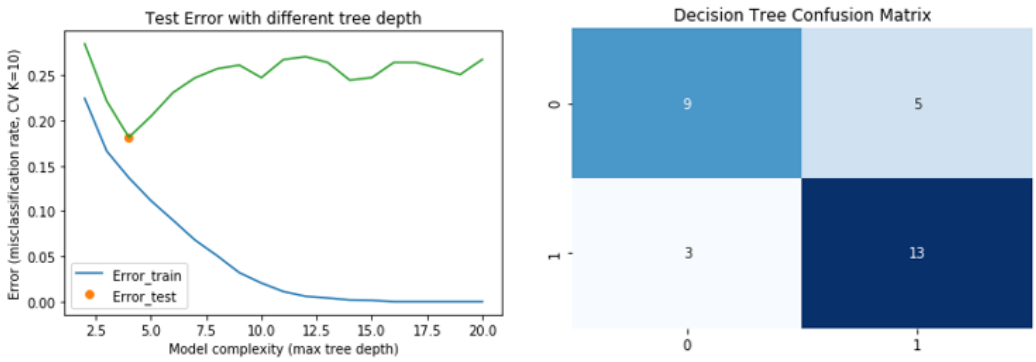


Fig. 11 Performance of Decision tree

In the left-hand pane, we draw a picture to show the test error with different tree depth, it's obvious that we get the best tree depth which equal to 4, and in the right-hand pane we generate the confusion matrix based on this depth.

We can also get the accuracy and test error value:

- Test Error: 18.150537634408603%
- Accuracy: 81.8494623655914%

3.3 Model selection by two-level cross-validation

In this section, we created a table and compared the logistic regression, decision tree, and baseline. We use 2-level cross-validation to calculate select the model and training the model with its optimal parameter. In inner loop, we selected the parameter for Logistic Regression and Decision Tree, and in outer loop, we trained these two models with the optimal complex-controlling parameters, and finally we get 10 optimal parameter sin 10 folds and 10 test errors.

The results of the parameter and test error in each fold as follows:

Outer fold	Logistic Regression		Decision Tree		Baseline
i	λ	E_i^{test}	t	E_i^{test}	E_i^{test}
1	1e-8	16.12903226	2	15.72146177	15.84147334
2	27.8256	12.90322581	4	12.21001221	16.78978429
3	1	16.12903226	11	15.75091575	16.43365893
4	1e-8	6.4516129	2	11.46616541	17.28327228
5	1e-8	12.90322591	2	13.8053467	17.68009768
6	1e-8	13.3333333	6	13.64522417	16.54456654
7	774.264	13.82478632	4	13.34318703	16.00936101
8	1e-8	13.17867318	2	18.05903648	16.51200651
9	59.9484	17.24137931	2	15.74770259	16.02055352

10	2. 15443	10. 34482759	3	10. 27568922	17. 16117216

Table 4: Two-level cross-validation table used to compare the three models

From this table, we can say that the test errors of baseline are relatively stable among all folds, we think the reason is that the baseline model doesn't add any parameter so that it will not be affected too much in different folds. The logistic regression model and decision tree model have their own complex-controlling parameter, and this parameter will be strongly affected in different folds because they have different train and test data in different folds. So, the parameter λ and t will be a little different in each fold.

3.4 Statistical evaluation

When we use cross-validation to compare three models, we need to know how to compare them, and is there a true difference in different models. Based on Bayes' theory, we know there is a symmetric interval which named credibility interval, we can use this interval to determine whether one model is better that the other. We can calculate the difference $E_A^{gen} - E_B^{gen} = \frac{1}{K}\sum_{K=1}^{K} z_k$, and generate a 95% credibility interval based on this difference.

The result of the comparisons of different 3 models as follows:

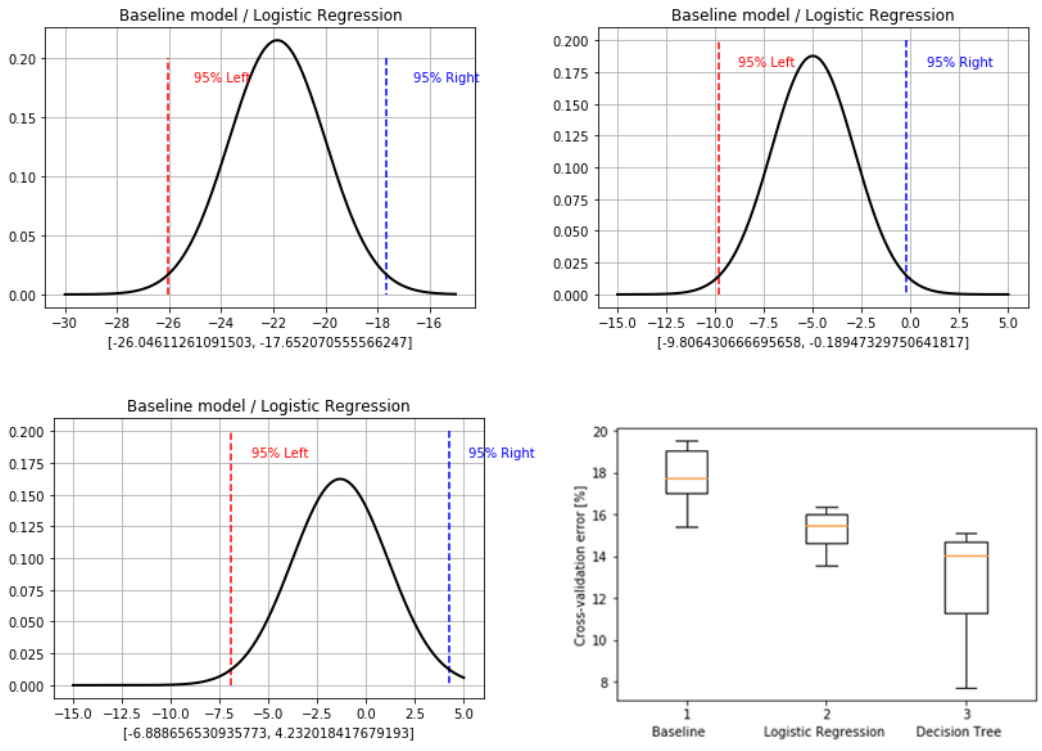


Fig: 12 95% credibility interval for comparing three models

It's clear that there are significantly different between baseline and decision tree, and also between logistic regression and decision tree, but there is almost not significantly different between baseline and logistic regression.

Based on the previous discussion, we can say that in our data set, decision tree is better than logistic regression and baseline, and logistic regression is a little better than baseline. And the results of error evaluation for different models can be found in the bottom right-hand pane.

The tables of comparisons as follows:

	K	v	$\bar{z}$	$\tilde{\sigma}$	θ_L	θ_H
Baseline *Decision Tree*	10	9	-21.84909	1.855318	-26.04611	-17.65207
Logistic Regression *Decision Tree*	10	9	-4.997951	2.125616	-9.806430	-0.189473
Baseline *Logistic Regression*	10	9	-1.328319	2.457980	-6.888656	4.232018

Table 5: Credibility interval for comparing three models

3.5 Logistic regression using regularization term λ

In this part, we train a logistic regression model with a suitable regularization term.

We also did a simple feature selection using logistic regression model that we trained before to identify which feature is more important is our data set. We use KFold() and Forward Selection to implement this function.

We generate a picture of the features which were selected in each fold, the results as follows:

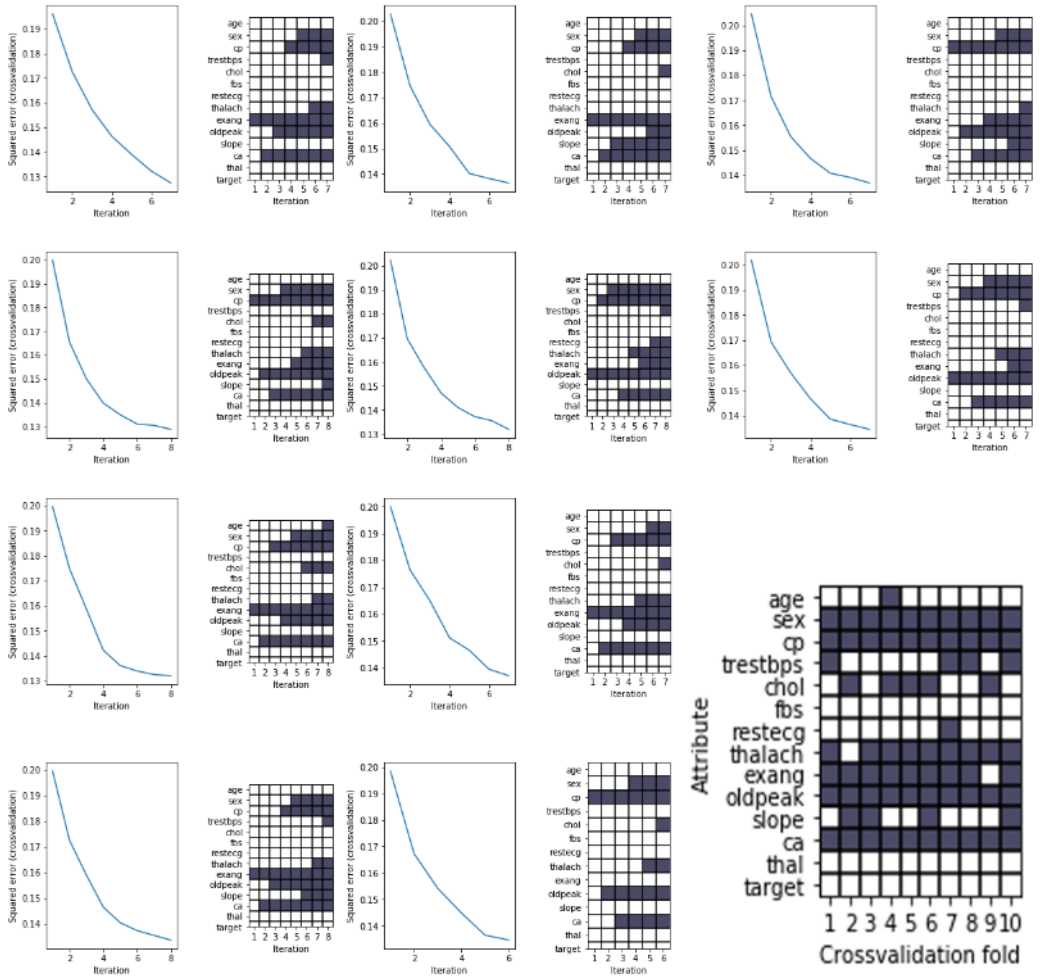


Fig: 13 Feature selection

From this figure, we can say that ‘sex’, ‘cp’, ‘thalach’, ‘exang’, ‘oldpeak’ and ‘ca’ are essential in our data set and we can also get a better performance in training by selecting these features.

Comparing with the features in regression part, these features are not relevant as for the regression part because we use different attribute as our target in classification, and the interdependency between attributes are totally different.

4. Discussion

Based on this project, we learned the theories and principles of regression and classification in supervised learning and learned how to implement the application of supervised learning on real-world data.

In regression part, we have a lot of models such as linear regression and ANN, and the model we used most is linear regression model. In linear regression, we know that linear regression is a basic and commonly used type of predictive analysis, it is used to determine whether there is a linear relationship between a dependent variable and one or more independent attributes. We can say that the core idea is to obtain a line that best fits the data, and the best fit is the one for which total prediction error (all data points) are as small as possible.

In the classification part, we also have a lot of models such as logistic regression, decision tree, ANN and KNN which can be used to classify discrete attributes. In this project, we focus on the logistic regression model and decision tree. Logistic regression is very similar to linear regression, the core difference is that in logistic regression, it uses a function named 'sigmoid' to re-parameterize the Bernoulli distribution so that this model can classify discrete attribute based on probability. The decision tree analyses a data set in order to construct a set of rules, or questions, which are used to predict a class, these rules can be built up to create a model that can classify complex situations. In our project, the creation of these rules is usually governed by an algorithm learning which questions to ask by analyzing the entire data set. We also use some complex-controlling parameters in our model to avoid overfitting the data.

References

[1] Introduction to Machine Learning and Data Mining, Technical University of Denmark. Tue Herlau, Mikkel N. Schmidt and Morten Morup

[2] https://archive.ics.uci.edu/ml/datasets/Heart+Disease