Danmarks Tekniske Universitet

DTU

02450 Introduction to Machine Learning and Data Mining

Group 31 - Project 2

Adikrishna Murali Mohan - s212940
Spyridon Safikos - s223409
Mohamad Ashmar - s176492

Student ID	Regression A	Regression B	Classification	Exam Questions
s212940	30%	40%	30%	33.33%
s223409	40%	30%	30%	33.33%
s176492	30%	30%	40%	33.33%

Date: 18th April 2023

Contents

1 Regression A

For the linear regression analysis of the South African heart disease dataset , we had to decide on a continuous variable from the attributes in the dataset. From our previous data visualisation result the heat map showed the correlation between different attributes among which adiposity and obesity had a high correlation of 0.72, which suggested they are highly correlated with each other. With linear regression, we hope to find the best-fit values of the slope and intercept that makes the line come close to the data. Thus we chose adiposity as the independent variable to predict obesity.

Since we will use regularization, we consider one-of-K coding as a feature transformation technique to convert categorical variables into numerical values that can be used in our statistical model. In one-of-K coding, each category is represented by a binary vector of length equal to the number of categories, where all elements are 0 except for the element corresponding to the category, which is 1.

In our dataframe we apply standardization which involves subtracting the mean of each column from the column and then dividing by the standard deviation of the column. This transformation ensures that all the columns have the same scale and allows us to compare the importance of different variables in the model.

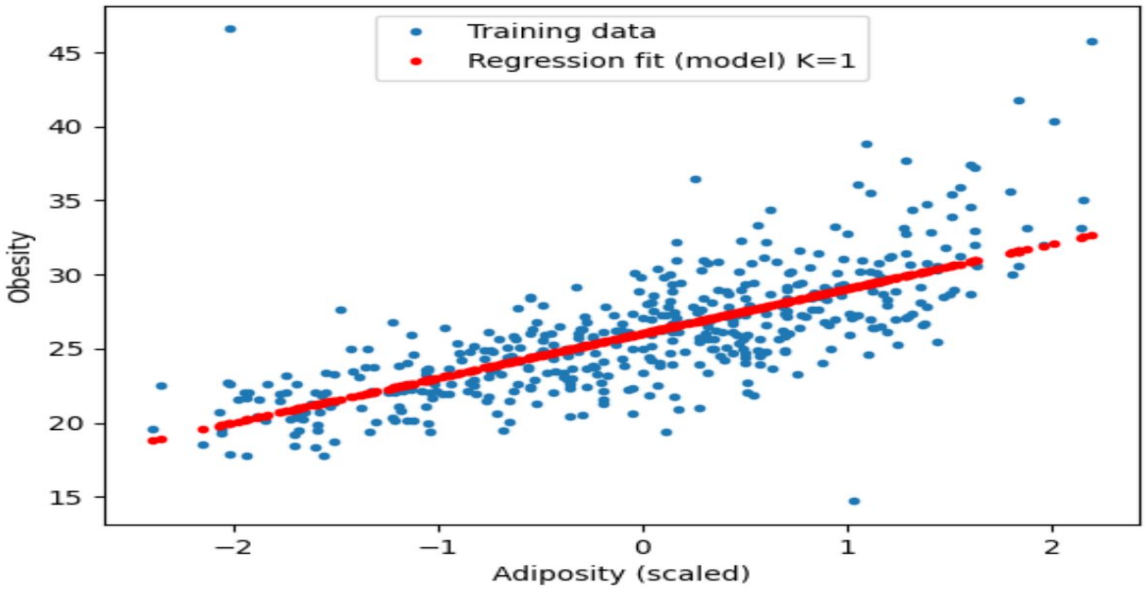

Figure 1: Plot showing the linear regression between adiposity and obesity

The regularization parameter lambda controls the strength of the penalty term. Larger values of λ leading to smaller parameter values and simpler models.
To estimate the generalization error for different values of λ, we can use K-fold cross-validation. It involves splitting the data into K folds, and then training the model K times, each time using a different subset as the validation set and the remaining subsets as the

training set. The generalization error is then estimated as the average error across the K validation sets.
We vary the learning rate over a range of values and compute the average mean squared error across the K validation sets for each value of the learning rate. We then plot the estimated generalization error as a function of the learning rate.
To choose a range of values of λ , we can start with a small value of λ and gradually increase it until the model starts to overfit the data. We can then choose a value of λ just before the overfitting occurs.

For each value of λ we set K = 10 fold cross-validation to estimate the generalization error. The resulting plot shows the graph of generalisation error for each value of λ. The generalization error first decreases as the λ increases, reaches a minimum, and then increases as the λ continues to increase. This suggests that there is an optimal λ that minimizes the generalization error.

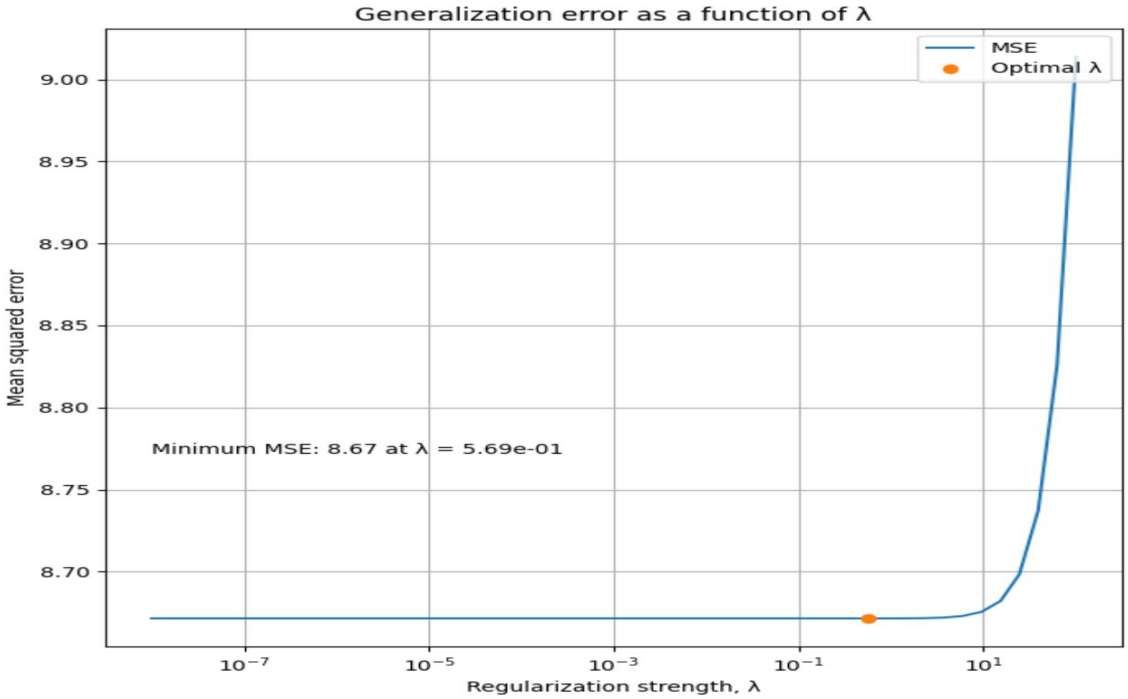

Figure 2: generalization error for different values of λ

From the regularisation factor graph below we can see that the generalisation error changes after lambda = 10^2. This indicates that the model starts to be affected by the regularization strength at that point, and it may begin to experience either overfitting or underfitting problems.

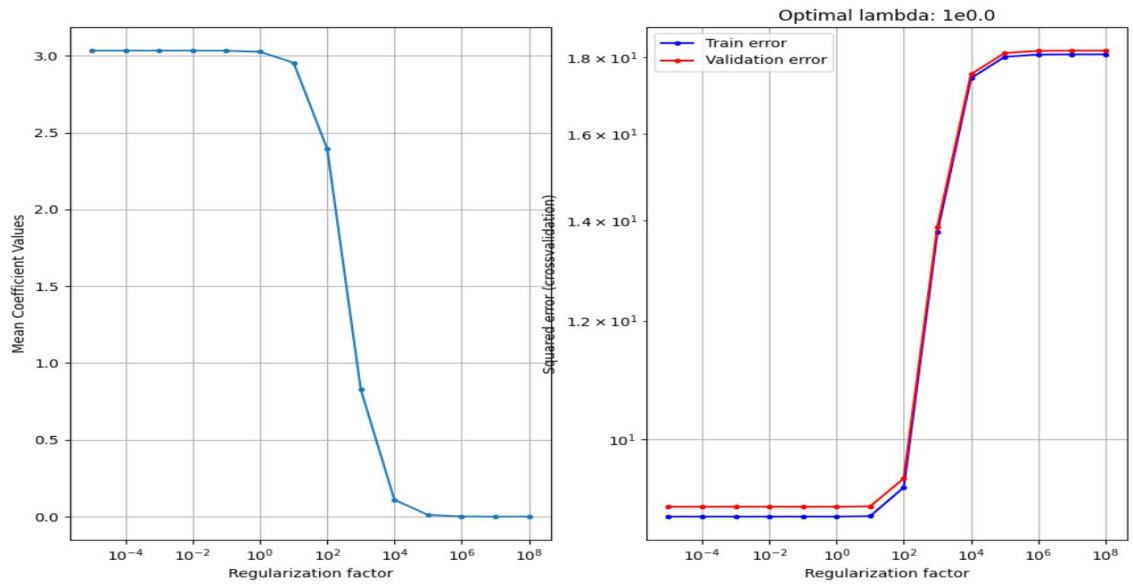

Figure 3: Regularization factor

To predict a new data observation using the linear regression model with the lowest generalization error, we first use the same feature transformation on the new observation as we did on the training data. That is, we center and scale the features so that each column has mean 0 and standard deviation 13. We then use the coefficients of the linear model to compute the predicted target variable.

2 Regression B

2.1 Compare Regression and ANN model using cross-validation

In this section we compared the performance of a linear regression model with an ANN model using two-level cross-validation with 10 folds for each level. We continue comparing the performance of different regression models for predicting obesity based on the feature adiposity. We use an artificial neural network (ANN) model with two-level cross-validation and 10 folds for each level. We used different numbers of hidden units h and the regularization parameter λ to find the best optimal model for the ANN model.

For the ANN model, we selected the number of hidden units h from 1 to 20 and the regularization parameter λ from $10^{-5} to 10^{2}$.

In each fold of the outer loop in two-level cross-validation we randomly split the data into training and test sets. The training set was continued to be split into 10 folds in the inner loop, where we trained the models in the inner loop and used to calculate Mean Squared Error (MSE) of the predictions using the test set on the trained model.

After the inner loop is completed we select h and λ that have the lowest average of MSE during running the inner folds. Where those selections consider the best model for the outer fold. Then we trained the outer training set with the best parameters and evaluated the performance on the outer test set. The final step was to compare the different models and choose the optimal model.

We can conclude that the simple linear regression model performed poorly compared to the ANN model. The average MSE of the linear regression model was 24.5179, while the average ANN was 672.5282. This indicates that the ANN model performed significantly better than the simple linear regression model. We also know that the best ANN model has 2 hidden units and a regularization parameter λ of 0.00215. These values were selected based on the lowest average MSE across all inner folds for that particular outer fold.

2.2 Table of optimal values of each fold

id	Best h	Best λ	Baseline Error	Best ANN Error
0	10	100.000000	22.930121	728.698557
1	20	100.000000	11.557204	696.582861
2	2	2.782559	5.463190	661.715623
3	20	100.000000	4.875189	645.083560
4	1	100.000000	11.216089	693.967698
5	5	0.000359	6.680161	658.948975
6	1	0.464159	6.352755	653.861820
7	5	0.464159	4.036484	682.313379
8	20	0.077426	6.261912	640.655117
9	20	0.464159	7.209111	654.255880

Table 1: Shows the values of hidden units h, regularization λ, and estimated generalization errors for each fold.

We can see that the optimal values of h and λ within the folds, with the best h ranging from 1 to 20, and the best λ ranging from 0.000359 to 100. The baseline errors range from 4.036484 to 22.930121, while the best ANN errors range from 640.655117 to 728.698557. In general, the baseline model has lesser errors than ANN, which indicates that the predictions are close from the actual values, which shows that the baseline model is a better predictor of obesity based on adiposity.

The value of λ In this table above is not the same as the value found in the previous section, which was 0.00215, because we randomly split the data into training and test sets in the two-level cross-validation, which can lead to different optimal values of the hyperparameters for each fold.

2.3 Statistical test

To conclude based on p-values and confidence intervals, we check if the p-value is less than a significance level e.g., 0.05) and if the confidence interval does not contain zero, which is enough evidence to reject the null hypothesis, which means there is strong evidence that those models are different from each other.

In our case, we can see from the table below that the ANN and Linear Regression models have a true difference in errors that lies within (618.92, 1065.15). Higher positive errors in this range indicate a worse performance for the ANN model compared to the Linear Regression model. On the other hand, the Linear Regression vs Baseline comparison has lower errors, which suggests better performance.

There is a difference in the errors in all model pairs because all of the p-values are less than 0.05. The difference in errors is calculated as (Linear Regression errors - Baseline errors). A negative confidence interval indicates that the linear regression errors are smaller than the baseline errors. Finally, the performance of the Linear Regression model is obviously better than the ANN model and the Baseline model.

Comparison	**Result**
ANN vs Linear Regression	Confidence interval: (618.9158995001742, 1065.1464151787093) p-value: 1.3120354154448714e-05
ANN vs Baseline	Confidence interval: (610.6730468456259, 1055.0686004229283) p-value: 1.386470926010315e-05
Linear Regression vs Baseline	Confidence interval: (-11.939874404930247, -6.380793005399369) p-value: 3.870749042494397e-05

Table 2: Comparison results

3 Classification

We have decided to predict the presence of coronary heart disease (CHD) in patients based on the number of risk variables for the categorization problem. The outcome variable's have two possible values—presentation (1) or absence (0) of CHD—this is a binary classification problem. We can determine the most crucial CHD risk factors and how they interact by using a classification model. High-risk patients can be identified using the categorization model, and the proper measures can then be put in place to lower their chance of developing CHD.

3.1 Comparing LR, KNN, and Baseline by using cross-validation

We have implemented logistic regression, K-Nearest Neighbors (Method 2), and a baseline model to predict class based on test data. We used λ=0.1 as a parameter for logistic regression, and for K-Nearest Neighbors we used n_neighbors=5 as a parameter, λ and n_neighbors can be adjusted based on a trial run. Finally, a baseline model which predicts the majority class in the training data of test data.

After running and testing the models we calculated the accuracy scores where we found that logistic regression model has accuracy of 0.763, while K-Nearest Neighbors has accuracy of 0.612, lastly baseline model has 0.645 accuracy. So based on those results we can conclude that the logistic regression model performs better than other models in predicting CHD in heart diseases detection based on this dataset.

3.2 Table of optimal values of each fold

Perform two-level cross-validation to compare logistic regression, k-nearest neighbors method 2, and a baseline model using the error rate as an error measure. The results are displayed below in a table.

id	Best k	Etest_knn	Best λ	Etest_logistic	Etest_baseline
0	4	0.234043	1000.00000	0.234043	0.297872
1	2	0.319149	100000.00000	0.234043	0.234043
2	3	0.500000	100.00000	0.326087	0.478261
3	2	0.347826	0.00001	0.217391	0.369565
4	5	0.326087	100.00000	0.260870	0.304348
5	4	0.456522	10000.00000	0.347826	0.413043
6	2	0.347826	10000.00000	0.304348	0.326087
7	1	0.326087	0.00001	0.282609	0.434783
8	2	0.304348	0.00001	0.152174	0.304348
9	4	0.326087	0.00001	0.152174	0.304348

Table 3: Shows the values of k and Etest_knn, regularization λ and Etest_logistic, and Etest_baseline for each fold.

The **k-nearest neighbors** method (Method 2) has different performance on different outer validation splits, with error rates ranging from 0.234043 to 0.500000. The best k value chosen in each fold is also different, which means that the model's performance is unstable to the choice of k where applying this model to new data seems to be sensitive.
The **logistic regression** model has a more stable performance on different outer validation splits, with error rates ranging from 0.152174 to 0.347826. The best λ value changed for each fold, which shows that the choice of the regularization parameter impacts the model's performance. However, the overall performance seems to be more stable compared to the k-nearest neighbors method, where the differences between the error rates in each fold of logistic regression are really small.

The **baseline** model's performance changed on different outer validation splits, with error rates ranging from 0.234043 to 0.478261. This change in performance is because of the differences in the majority class in each fold.

In summary, we can see that the logistic regression seems to be the most stable model and performs better than the k-nearest neighbors method and the baseline model. This is a good sign of logistic regression to make almost correct predictions on new/unseen data.

3.3 Statistical test

We performed a statistical evaluation of three models: KNN, Logistic Regression, and Baseline Model. We used McNemar's test to compare the models pairwise, as shown in the table below:

Comparison	theta_hat	CI	p-value
KNN vs Logistic Regression	0.0065	(-0.0369, 0.0499)	0.8454
KNN vs Baseline Model	-0.0606	(-0.1026, -0.0185)	0.0066
Logistic Regression vs Baseline Model	-0.0671	(-0.1100, -0.0241)	0.0032

Table 4: Comparison Results

Small theta_hat value (0.0065), having 0 in the CI range, and obtaining a p-value of 0.845 for KNN vs Logistic Regression all indicate that there is no significant difference in performance between these two models. However, for KNN vs Baseline Model, the theta_hat value is -0.0606 (a negative value indicating that KNN has higher accuracy than the Baseline), 0 is not included in the CI, and the p-value is 0.0066, which demonstrates that the KNN model performs significantly better than the Baseline model. Lastly, the values for Logistic Regression vs Baseline Model also show that the Logistic Regression model performs better than the Baseline Model.

In conclusion, both the KNN and Logistic Regression models perform significantly better than the Baseline Model. However, there is not a significant difference in performance between the KNN and Logistic Regression models. However, by trying different hyperparameters for models could improve the performance.

3.4 Features Relevant in Classification and Regression

Logistic regression makes a prediction by computing the probability of an observation belonging to a specific class. In the case of CHD binary classification, logistic regression calculates the probability of an observation having a disease. This probability is applied to a sigmoid function with a combination of features and model's weights. If the probability is greater than a threshold (could be 0.5), then the observation is assigned to class 1, otherwise to class 0.

After running the code, we got a minimum test error of 33.49%, an optimal lambda value of 1e 1.18, and model weights for the optimal lambda [0.08548744, -0.07651973, 0.14756655, 0.15879283, 0.15699256, -0.19304003, -0.01512293, 0.14665036, 0.20728808]. The model weights represent the contribution of each feature to the prediction. The higher the absolute value of a weight, the more relevant the current feature is to the model. In this case, we can see that feature-6 (-0.19304003) and feature-9 (0.20728808) have the most impact on the predictions.

The feature-6 (famhist) and feature-9 (age) are more relevant in the logistic regression model as compared to 'adiposity' (feature-5). However, 'adiposity' was the only feature used in the linear regression problem which is the most relevant feature. In the logistic regression model, 'adiposity' still has a significant weight (0.15699256), indicating its relevance, but not as strong as 'famhist' and 'age'.

4 Discussion

From the linear regression we could see that optimal λ value is an important parameter in regression as it controls the amount of regularization applied to the model. We should use cross-validation to select the value of lambda that results in the lowest validation error. We used the linear regression to predict obesity from adiposity. However, including both attributes adiposity and obesity in a predictive model could lead to multicollinearity issues since they have a similar effect on CHD, causing both variables to capture almost the same information. Therefore one possible use of linear regression could be to predict the systolic blood pressure (sbp) of an individual based on other attributes.

In our case, we could see that the ANN and Linear Regression models have a true difference in errors. Higher positive errors indicate a worse performance for the ANN model compared to the Linear Regression model. On the other hand, the Linear Regression vs Baseline comparison has lower errors, which suggests better performance. From the classification model, both the KNN and Logistic Regression models perform significantly better than the Baseline. However, there is not a significant difference in performance between the KNN and Logistic Regression models. Therefore we would like to try using different hyperparameters for models to see any improvement in the performance.

5 Exam questions

Problem 1: Answer: option

we can look for the points in the ROC curve in Figure 1 that correspond to the black circles and red crosses in Figure 2. Specifically, the points in the ROC curve correspond to the true positive rate (TPR) and false positive rate (FPR) at different threshold values for the classifier.
Problem 2: Answer: option B

Impurity error for parent node = 1 - max((33+28+30+29)/124, (4+2+3+5)/124) = 0.3935
left child node (x = 0 or 2) = 1 - max((33 + 28 + 30 + 29)/97 , 0) = 0.3093
right child node (x = 1) = 1 - max(0, (4+2+3+5)/27) = 0.2963
impurity gain of the split x = 2 = I(parent) - 97/124*I(left) - 27/124*I(right) = 0.0178

Problem 3: Answer: option A

Total parameters = hidden parameters + output parameters
For the hidden layer: 7 input features (x1 to x7) and 10 hidden units. Therefore, the number of parameters in the hidden layer = (7×10)+10 = 80
For the output layer: 4 possible classes (y=1, y=2, y=3, y=4) and 10 hidden units. Therefore, the number of parameters in the output layer = (10×4)+4 = 44
Total number of parameters = 80 + 44 = 124 parameters

Problem 4: Answer: option C

Option C = A: $b2 \geq 0.03$, B: $b1 \geq -0.76$, C: $b2 \geq 0.01$, D: $b1 \geq -0.16$
Given: structure of the decision tree and the predicted label assignments for 135 observations.
To find the correct rule assignment to the nodes in the decision tree we find the splitting rules (values of bi and z) that correspond to each node in the tree. We can do this by looking at the predicted label assignments for the observations that fall into each node and determining the threshold values that best separate those observations into their correct classes.
Node 1: $b2 \geq 0.03$
Node 2: $b1 \geq -0.76$
Node 3: $b2 \geq 0.01$
Node 4: $b1 \geq -0.16$

Problem 5: Answer: option A

Given:
Training time for a single neural network model = 20 ms,testing time = 5 ms
Training time for a single logistic regression model = 8 ms, testing time = 1 ms

Time taken to train and test a single neural network model for each of the 40 combinations of hyperparameters is 20 + 5 = 25 ms.
Therefore the total time for each outer fold is 40 x 25 = 1000 ms
Time taken to train and test a single logistic regression model for each of the 40 combinations of hyperparameters = 8 + 1 = 9 ms. Therefore the total time taken for each outer fold = 40 x 9 = 360 ms.

Since there are 5 outer folds, the total time taken to compose the table = 5 x (1000 + 360) = 6800 ms

Problem 6: Answer: option B

To determine which observation will be assigned to class y=4, we compute the $\hat{yn}$ for each observation using the given weights.

First, let's calculate $\hat{yn}$ for each observation and class:

Observation B:

$$\hat{y}1, B = [1, -0.6, -1.6] \cdot w_1 = 1.2 - 2.1(-0.6) + 3.2(-1.6) \approx -2.66$$
$$\hat{y}2, B = [1, -0.6, -1.6] \cdot w_2 = 1.2 - 1.7(-0.6) + 2.9(-1.6) \approx -2.42$$
$$\hat{y}3, B = [1, -0.6, -1.6] \cdot w_3 = 1.3 - 1.1(-0.6) + 2.2(-1.6) \approx -1.56$$

We do the same calculation as B for A, C, and D

For observation A, C and D are none of the $\hat{yn}$ close to 0, so it won't be assigned to class y=4.