Danmarks Tekniske Universitet

DTU

02450 Introduction to Machine Learning and Data Mining Report 2

Authors

Paula Gambus i Moreno- s233219
Fernando Augusto Marina Urriola- s233144
Nerea Suarez Caballero - s233132

December 2, 2023

Contents

List of Figures

List of Tables

1 Contributions

	Regression part a	Regression part b	Classification	Exam quest
Paula Gambus - s233219	60%	30%	10%	33.33%
Nerea Suarez - s233132	10%	60%	30%	33.33%
Fernando A - s233144	30%	10%	60%	33.33%

Table 1: Project contribution.

2 Regression

2.1 Part a

After the results obtained in the previous report, we discovered that our data set is suitable for a regression task. The objective of our regression is to predict patients age taking into account all the other attributes. This information can be useful to healthcare professionals since age is a critical factor when determining a patient's risk of heart-related issues. Knowing the ages can help to asses their risk more accurately and also influence the choice of treatment.

Therefore, in this subsection, regularization parameters λ were introduced in order to estimate the generalization error for different values of λ. These parameters control the complexity and simplicity of the model, avoiding over-fitting and allowing feature selection in some cases. That is why it is important to find appropriate values to have efficient and generalized models.

First of all, the explanatory attributes were standardized and our "Age" variable was centered. Afterwards, in this case, three different types of linear regressions were performed (Ridge, Lasso and Elastic Net) [1] to assure the best value of λ and, with each of them, a linear regression was applied obtaining the generalization error. This process was repeated for different values of λ between 0 and 5 to determine the optimal λ^* from the best technique. After observing the results from Figure 1, we decided to keep the Lasso model when using $\lambda = 0.30$. Therefore, using the Equation 1 for the regularized linear regression, we obtain the results displayed in Table 2.

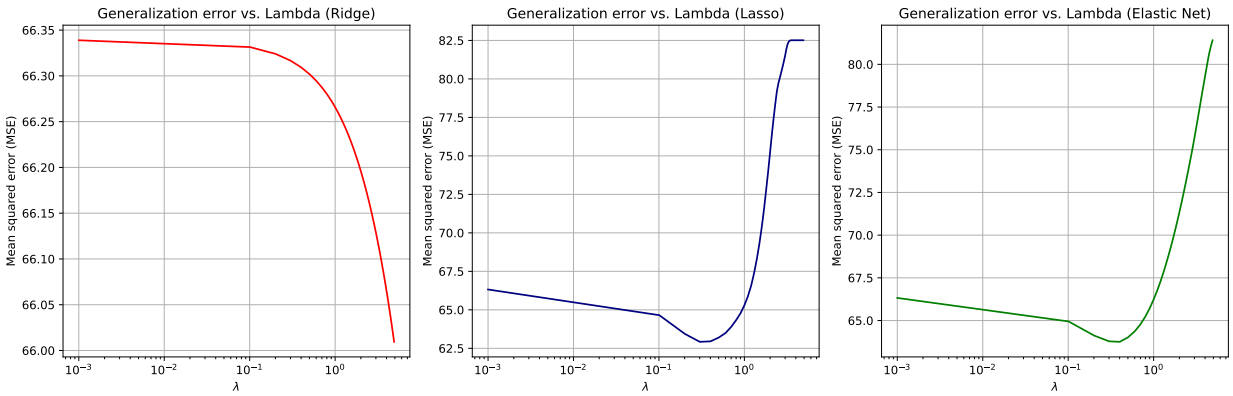

Figure 1: Generalization error (y-axis) as a function of λ (x-axis) according to the different logistic regression techniques.

$$\omega = (X^T X + \lambda I)^{-1}(X^T y) \tag{1}$$

The ω from the Equation 1 explain the contribution of each attribute to the prediction of the wanted variable. In a model of linear regression, the relationship between the characteristics and the objective variable is represented in Equation 2 where ω_0 is the intercept value and the other ω_n are the coefficients which multiply the predictive variables. In order to evaluate the change of these parameters when increasing λ, the graph of Figure 2 was generated.

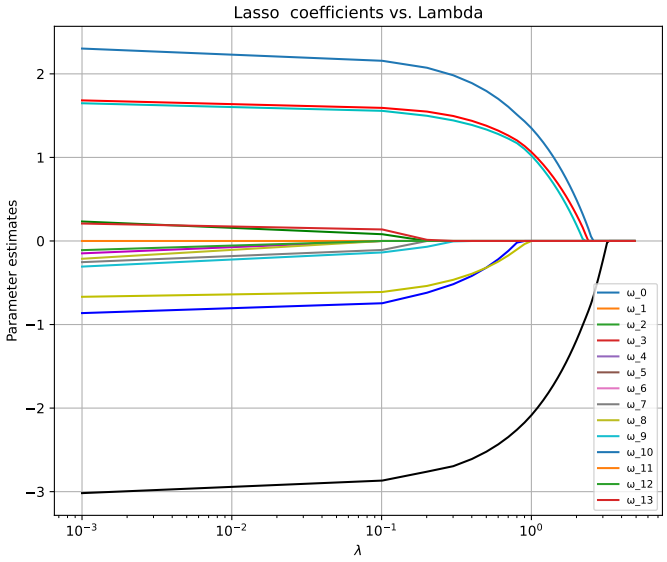

Figure 2: Graph of the attribute weigths (y-axis) estimated via Regression as a function of λ (x-axis)

$$y = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_p x_p + \varepsilon \tag{2}$$

For the sake of completeness, the different parameters estimated by the regularized regression when using $\lambda = 0.30$ were calculated after using the Equation 1. They can be seen in the Table 2 where the attributes with weights different to zero are shown. We can observe that the attribute that has more effect to the target value is the "thalachh" since its magnitude is high. However, since it is a negative value, a higher resting heart rate will be associated with a lower predicted "Age". In contrast, the "chol" feature has a weaker influence since the value is lower but its increase means an increase of the age predicted value. Therefore, a positive coefficient indicates an increase in the feature value that will tend to increase the probability if the outcome being 1, while a negative coefficient suggests that an increase in the feature value will tend to decrease the probability being 0.

Feature	**Intercept**	**Sex**	**trtbps**	**chol**	**restecg**	**thalachh**	**slp**	**caa**
weight	ω_0	ω_1	ω_3	ω_4	ω_6	ω_7	ω_{10}	ω_{11}
value	54.457	-0.518	1.496	1.443	-0.465	-2.696	-0.006	1.984

Table 2: Table with the weights of each attribute.

2.2 Part b

In this subsection we will make the same task as in part a. We will try to predict the "Age" attribute based on all the other attributes, but this time we will use a 2-level cross validation with $K_1 = K_2 = 10$ ($K_1 = K_2 = 5$ for the ANN) and we will also evaluate two additional methods, Artificial Neural Network and baseline.

An ANN is a simply designed artificial neural network with a flexible number of hidden units called $\boldsymbol{h}$. The input layer is systematically composed of as many neurons as there are attributes to work with (14 in our case), each of which corresponds to one of the attributes of the data set. The variability in the number of neurons in the hidden layer is dictated by $\boldsymbol{h}$ parameter, which controls the complexity. In contrast, the output layer consist of a single neuron whose ideal function is to assign the **"Age"** attribute.

The baseline method involves a straightforward linear regression model with no features, essentially predicting the target variable y (age) on the test data by computing the mean of the output y from the training data. The baseline model is simple because it does not use any extra information and just predicts based on the average of the target variable. This basic approach helps us figure out if the complicated models such as the ANN do much better than the simple ones. It is a key point to see how well the overall modeling setup performs.

So, for both methods, the inner loop of the 2-level cross validation was used to determine an optimal complexity controlling parameter. In the ANN's case, it is the number of hidden units, while for the regression, we introduced a regularization parameter λ.

The results of our evaluation can be seen in Table 3.

Outerfold	**ANN**		**Linear Regression**		**Baseline**
i	$\mathbf{h}_i$	$\mathbf{E}^{\mathbf{test}}$	λ_i	$\mathbf{E}^{\mathbf{test}}$	$\mathbf{E}^{\mathbf{test}}$
1	8	114.03	0.56	60.505	81.986
2	9	115.6	0.47	60.209	83.416
3	7	157.9	0.58	59.157	80.462
4	8	106.5	0.40	58.590	82.126
5	7	141.3	0.31	59.157	80.366
6			0.39	58.498	80.438
7			0.64	56.922	78.727
8			0.34	62.540	85.243
9			0.45	40.083	84.561
10			0.37	59.836	82.295

Table 3: Optimal regularization parameter and test error for each outer fold i in regression.

As we can see at first glance, it was decided to make fewer loops in the ANN due to the high computational cost that $K1 = K2 = 10$ entailed

As illustrated in Table 3, we have compiled a summary detailing the optimal number of hidden units and the best λ for each outer fold. In our data set, during the training of the Artificial Neural Network (ANN), we observed significantly improved performance with a higher number of hidden units compared to a lower value. This observation is likely attributed to the ample amount of data available, accompanied by the amount of features we have. The efficacy of employing a larger number of hidden units tends to be more pronounced when dealing with extensive training data sets. Conversely, in cases where the data set is relatively small, a simpler model with fewer hidden units may yield better generalization results.

When it comes to the λ values, we know that this parameter control the complexity and simplicity of the model. The test errors for this linear regression range from approximately 40.083 to 62.540. Interestingly, certain λ values seem to correspond to lower test errors compared to others. For instance, λ values around 0.45 and 0.64 result in notably lower test errors compared to the rest, what suggests that these values strike a better balance between bias and variance. They are penalizing the model to some extent (avoiding over-fitting) but not excessively, allowing the model resulting in lower test errors compared to the other values.

The Baseline model provides a benchmark for comparing the performance of the ANN and the Liner Regression models, showcasing the relative effectiveness of these models in the given context

2.3 Correlated t-test for cross-validation

This method uses a simple transformation of the average difference of the generalization error to get a t-distribution when identical performance is assumed. Based on this distributional assumption, p-values and confidence intervals can then be calculated. In Table 4 the results using the mentioned approach can be observed:

tested method	***p-value***	***CI lower bound***	***CI upper bound***
ANN vs Linear regression	0.00044	91.93	156.7
Linear regression vs Baseline	1.77e-11	18.7204	20.9473
ANN vs Baseline	0.001	-135.22	-69.2

Table 4: Results of statistical comparison (regression).

The p-value is a measure that tells us how likely it is to obtain results as extreme as those observed if we assume that there is no real difference between the models compared.

A small p-value suggests that there is a significant difference bewteen the two models.

The confidence interval provides a range within which the true value of the parameter is expected to lie. In our case, in the table we can see that all the p-values are extremely low, very near to 0, what means that all the models present significant differences between them.

On other hand, the CI shows us that ANN seems to outperform Linear Regression but it surprisingly performs worse than the Baseline, as indicated by the negative range in the CI for the ANN vs Baseline comparison. This means that further research into the specific features of the reference model and why it outperforms ANN in this context is needed.

3 Classification

In agreement with our assessment from the previous report, our classification problem consists of predicting the classes heart attack and no heart attack. Hence, our problem can be considered a binary classification. Nowadays, predicting and preventing heart attacks is crucial for early intervention and improving patient outcomes, since heart diseases continue to be leading causes of death globally, presenting a significant challenge to global health.

Mirroring what we did in the regression subsection, we again compare three models: a baseline, a logistic regression, and an elective model. As our elective model, we decided to choose the k-nearest neighbor (KNN) classifier. The used methods are shortly described in the following:

- The **k nearest neighbour (KNN)** [2] approach makes a class-prediction for a new data-point by assessing the k-nearest neighbours within the train data set. It then predicts the data-point to be part of majority class of these k assessed data-points. If k is even and there is no majority class, we fall back to the base rate to make a prediction.
- As a **baseline**, similar to the previous subsection, we use again a predictor that uses the larger class in the train data set as a prediction regardless of its shape.
- **Linear Regression**, specifically Lasso (Least Absolute Shrinkage and Selection Operator), is a regression technique used for predicting a continuous numeric outcome variable based on one or more input features. Lasso extends linear regression by adding a penalty term that encourages sparsity in the model. It performs feature selection by shrinking some of the feature coefficients to exactly zero, effectively removing them from the model.

As in Regression part b, we use a 2-level cross-validation with $K1 = K2 = 10$, where an optimal complexity parameter is chosen within the inner loop and the test-error is computed in the outer loop. Based on conducted test runs, a range of 1 to 8 was compared for k and 0.1 to 5 for λ in the course of each inner loop. The obtained results are presented in the following table.

As can be seen in the Figure 3, after having done the double cross validation, the best value for $K=7$ is shown, so this is the value of K that will be used from now on.

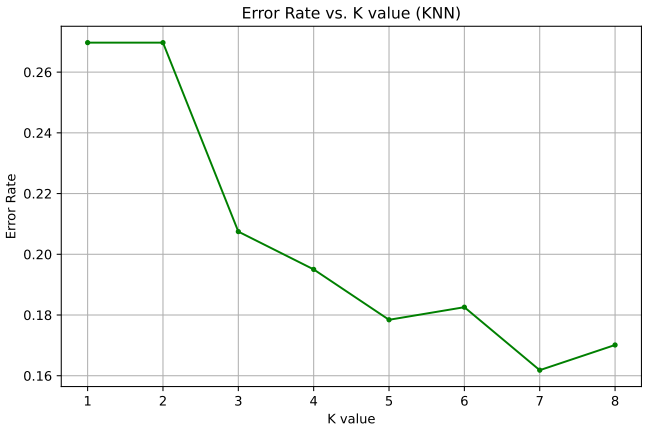

Figure 3: Best K value for KNN in double-cross validation.

Outerfold	**KNN**		**Linear Regression**		**Baseline**
i	$\mathbf{k}_i^*$	$\mathbf{E^{test}}$	λ_i^*	$\mathbf{E^{test}}$	$\mathbf{E^{test}}$
1	1	0.203	0.1	0.165	0.457
2	3	0.202	0.1	0.162	0.454
3	3	0.194	0.1	0.158	0.460
4	2	0.203	0.1	0.161	0.467
5	1	0.207	0.1	0.160	0.441
6	1	0.199	0.1	0.165	0.467
7	1	0.212	0.1	0.152	0.449
8	1	0.191	0.1	0.157	0.463
9	2	0.211	0.1	0.169	0.456
10	1	0.199	0.1	0.155	0.456

Table 5: Optimal regularization parameter and test error for each outer fold i.

In the same way we perform in regression part b, in Table 5 we can see which of the k_i perform better in the KNN model for each outer loop, in our case $k = 1$.

The results table suggest that, in general, Linear Regression model performs better than the other two models, the KNN and the baseline. However, this does not imply a universal superiority of Linear Regression over KNN or the baseline model. There are some outer folds, like outer fold 4 and 9 where KNN outperforms the Linear regression model. This variability might be due to the nature of the data, the way the folds were constructed or the specific parameters of each fold.

3.1 Mc Nemar's test

The McNemar test was applied to compare the performance of Linear Regression, K-nearest neighbors (KNN), and a Baseline model on a binary classification task. Surprisingly, while Linear Regression and KNN did not show a significant difference in performance, both of them displayed a highly significant difference when compared to the Baseline model, indicating that both methods outperform the Baseline significantly in this specific task.

The p-value of 0.44 suggests that there is no significant difference in the performance of Linear Regression and KNN on the binary classification task. The confidence interval (CI) further confirms this, as it contains values that are close to zero, indicating that the methods are statistically similar.

The extremely low p-value (7.85e-17) suggests a highly significant difference between Linear Regression and the Baseline model. The confidence interval also indicates that the CI lower bound is far from zero, reinforcing the significant difference in their performance.

Similarly, KNN and the Baseline model exhibit a very low p-value (1.32e-14), indicating a highly significant difference in their performance. The confidence interval also confirms this, as the CI lower bound is significantly above zero.

tested method	***p-value***	***CI lower bound***	***CI upper bound***
Linear regression vs KNN	0.44	-0.0170	0.0501
Linear regression vs Baseline	7.85e-17	18.7204	0.3618
KNNvs Baseline	1.32e-14	0.2143	0.3471

Table 6: Results of statistical comparison (classification).

3.2 Logistic regression model with suitable value of λ

In order to make this part, we have used a value of $\lambda = 1.9$, this value has been obtained thanks to the different tests that have been carried out previously.

The coefficients and the intercept have been calculated for this model and the results show that, it's evident that each feature's coefficient influences the model's predictions as it has been explained in subsection 2.1. For example, the coefficient for "trtbps" (Resting Blood Pressure) is 0.7328, which is positive. This implies that an increase in the resting blood pressure tends to increase the probability of heart disease [3].

The error rate of 0.1357 is a measure of the model's performance, suggesting that the model accurately predicts the presence or absence of heart disease approximately 86.43% of the time.

4 Conclusion

After the analysis of the results obtained in this report, we were able to conclude a variety of interesting things. We applied the requested methods to our data set. The objective of this report was to find how the different attributes extracted from parameters of patients with heart condition influence the accuracy of predicting the age of the patients (regression) and whether or not they will have a heart attack (classification).

In the first two parts, regression part a and regression part b, we have seen how different λ values affect to the model that was being trained, finally deciding that $\lambda = 30$ was the best option for the Lasso model, the one we were focusing on since it was the one that gave us the least error. Afterwards, we interpreted the relation between the output obtained given an specific input and the λ value used. In this case, we saw how each attribute with its respective weight were affecting the output.

For regression part b, we compute several cross-validation with different models and different parameters such as the λ value for the Linear Regression and the hidden units in the case of the ANN. In the case of the Network, after an extensive training with a high computational cost, we saw that the best hidden unit number to work with was 8, since it was the one which low more the error. At the end of this part, we conclude that more investigation of the baseline parameters was needed to really see which of the models perform better, so we believe that this investigation is a good future line to follow for the project.

In our extensive classification part, Lasso Regression emerges as the standout performer in predicting age across multiple folds, showcasing a remarkable consistency with a mean inner error rate ranging from 0.151 to 0.168. This stability is not only commendable but also validated by the McNemar's test, which underlines a significant difference when compared to the baseline. The resilience of Lasso against different folds and its superiority over the baseline make it a compelling choice for regression tasks in our dataset.

While Linear Regression exhibits variability in its performance across folds, the average baseline error rate of approximately 0.457 suggests a less robust predictive ability compared to Lasso. On the other hand, KNN, with its competitive performance, doesn't showcase a significant difference from Lasso in the McNemar's test, indicating a neck-and-neck competition between the two models. The nuanced decision between Lasso and KNN may rely on factors beyond the presented metrics, considering their comparable performance.

In the article [4] conducted a study comparing several categorization algorithms used for heart disease prediction. The authors used the Heart Disease dataset from the University of California, Irvine (UCI) 3 and the Heart Disease dataset of Cleveland 4 to train and evaluate the classifiers. The authors reported that the Random Forest classifier had the best performance in terms of accuracy, sensitivity, and specificity.

5 Exam Questions

5.1 Question 1. Spring 2019 question 13.

Answer: Option C.

A and B can be excluded since as we see from the graph the ROC curve moves from (1, 1) to (1, 0.75) and since we have 8 elements and 4 items per class itâs impossible to see that kind of configuration. To prove that between C and D the right answer is C we can compute some confusion matrix.

		Actual	
		T	***F***
Predicted	***T***	1	3
	F	2	2

Table 7: Confusion Matrix to ROC curve at point d for Prediction C.

		Actual	
		T	***F***
Predicted	***T***	2	2
	F	1	3

Table 8: Confusion Matrix to ROC curve at point d for Prediction D.

From Table 7 and Table 8 one can computer True Positive Rate (TPR) and False Positive Rate (FPR) as follow:

$$TPRC = TP/TP + FN = 1/4 \tag{3}$$

$$FPRC = FP/FP + TN = 1/2 \tag{4}$$

$$TPRD = TP/TP + FN = 1/2 \tag{5}$$

$$FPRC = FP/FP + TN = 1/4 \tag{6}$$

As we can see only prediction C has as output a point in the ROC curve.

5.2 Question 2. Spring 2019 question 15.

Answer: Option C.

In this question is asked to compute the impurity gain of split $x7 = 2$ After the split we will have two branches. For $x7 = 2$ left branches will contain only one element of class y

$= 2$ while the right branch contains 37 elements of class 1, 30 of class 2, 33 of class 3 and 34 of class 4. The exercise forces us to use the classification error to compute the purity level at each point of the tree:

$$ClassError = 1 - max[\frac{37}{135}, \frac{31}{135}, \frac{33}{135}, \frac{34}{135}] = \frac{98}{135} \tag{7}$$

$$ClassError_v1 = 1 - max[\frac{0}{1}, \frac{1}{0}, \frac{0}{1}, \frac{0}{1}] = 0 \tag{8}$$

$$ClassError_v2 = 1 - max[\frac{37}{134}, \frac{30}{134}, \frac{33}{134}, \frac{34}{134}] = \frac{97}{134} \tag{9}$$

At this point one can compute the impurity level as:

$$\Delta = ClassError(r) - \sum_{k=1}^{2} \frac{N(v_k)}{N(r)} ClassError(v_k) = 0.0074 \tag{10}$$

5.3 Question 3. Spring 2019 question 18.

Answer: Option A.

We know that there are i = 7 nodes as input, h = 10 hidden level and o = 4 nodes on output. The number of parameters that the network contains is easily computed by consider:

- Connection between first and second level= i * h
- Connection between second and third level= h * o
- Bias of second layer = h
- Bias of third layer = o

Therefore we can compute the result through this formula:

$$N = (i \cdot h + h \cdot o) + h + o = 124 \tag{11}$$

5.4 Question 4. Spring 2019 question 20.

Answer: Option D.

The question can easily been resolved trying to apply every set of rules at the decision tree and classification boundary. Doing this one find out that the only set of splitting rules that fits the decision tree is letter D.

References

[1] J. Ranstam and J. Cook, "Lasso regression," *Journal of British Surgery*, vol. 105, no. 10, pp. 1348–1348, 2018.

[2] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pp. 986–996, Springer, 2003.

[3] S. Pocock, A. Shaper, D. Ashby, H. Delves, and B. Clayton, "The relationship between blood lead, blood pressure, stroke, and heart attacks in middle-aged british men.," *Environmental health perspectives*, vol. 78, pp. 23–30, 1988.

[4] R. Thomas and G. Princy, "Supervised machine learning-based cardiovascular disease prediction: A comparative study," *Mathematical Problems in Engineering*, vol. 2021, p. 1792201, 2021.