INTRODUCTION TO MACHINE LEARNING AND DATA MINING
REPORT 2

Group 314	Regression, part a	Regression, part b	Classification	Discussion	Exam questions
Andrea Arieti s225570	30%	40%	20%	40%	33.33%
Andrea Tessarin s221771	45%	30%	20%	40%	33.33%
Stefan Slavkovsky s221931	25%	30%	60%	20%	33.33%

Regression, part a

Introduction

Project 2 has been done using the dataset used in the Project 1, the 'Adult' dataset from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Adult).

Starting from this dataset, also known as 1994 US Census Data, we have performed a basic linear regression. We used the data prepared in Project 1 with removed redundant columns, removed rows with missing values or outliers, grouped several categories and one-of-K encoding. For regression and classification problems, we also need to normalize each attribute so we can use regularization. We also decided to use train data, since we will be using two-level cross-validation, nevertheless. For the regression problem we decided to select age as our target variable, the quantitative response; it is a continuous numerical feature. All the other features will be considered as predictor variables.

Our main goal is to predict the age of a specified individual using different techniques. We used the cross-validation with 10 folds in order to estimate the generalization error and optimal regularization parameter.

Problem description

We introduced a regularization parameter λ. We used ridge regression adding the L2 penalty term to our error function and we have assigned different values to λ.

First, we performed the training for a large range ($\lambda \in [10^{1,}10^{8}]$). The ideal lambda will not be larger than 10^{4} as we can see in Figure 1. We can perform the training with a smaller but more precise range to find the optimal lambda ($\lambda \in [10^{1,}10^{4}]$). We can list the values of ideal lambda for each fold: 1258, 1258, 1995, 1258, 10, 10, 10, 630, 10, 10. We can see that for half of the folds, the best lambda is at the lower bound of the range, but the error does not change for lambda under 10 as can be seen in Figure 1. Therefore, the best lambda can be any number from range $\lambda \in [0,10]$ for these folds.

Results

Finding the best lambda for mean of all folds' results in $\lambda = 100$. This can be explained by folds with small λ averaging out with folds with bigger λ. The generalization error is 106.71.

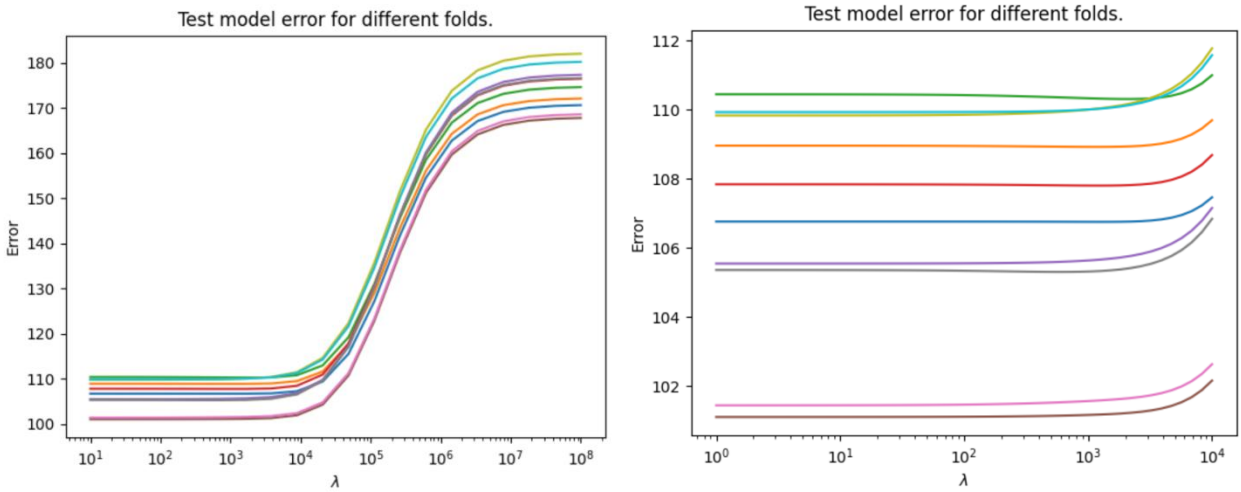


Figure 1, Error for wide range of regularization constants(left) and error for smaller range(right).

Linear regression tries to minimize the error function and use weights vector for making predictions from data. Each weight in vector corresponds to one input attribute. The bigger the weight, the

bigger contribution attribute has. Since our data is normalized, we can compare weights directly to determine, for example, 10 most contributing attributes (both positive and negative contributions in order):

1. (+3.35) Marital status: widowed
2. (-2.87) Marital status: never married
3. (-2.64) Relationship: own child
4. (+2.07) Marital status: divorced
5. (+2.01) Relationship: married
6. (+1.02) Earning over 50,000$
7. (-0.92) Work-class: private
8. (+0.90) Work-class: self-employed not inc.
9. (-0.76) Hours per week
10. (-0.69) Education number

Most attributes make sense (e.g., there are not so many divorced young people, but there are many never married youngsters). But attributes as work-class are harder to explain at first sight.

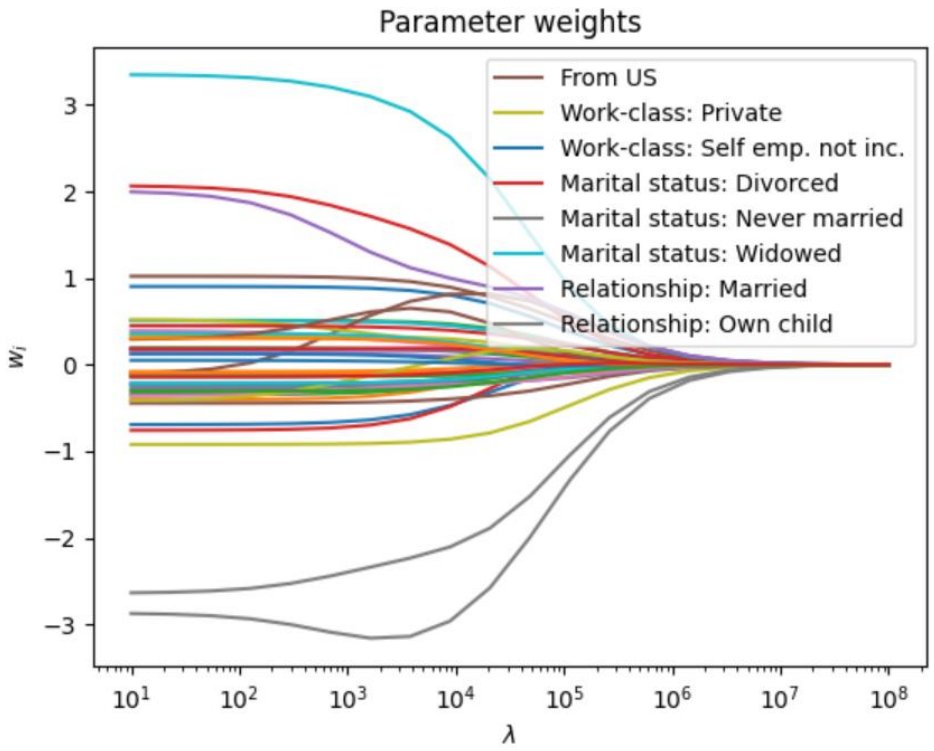


Figure 2, Effect of regularization constant on weights.

Regression, part b

Description

In this part we compared three different models:

- Regularized linear regression model
- Artificial neural network model
- Baseline model

We have implemented a two-level cross-validation to compare the models with ten folds in each level. The baseline model is a linear regression model with no features. For the ANN, we will use number of hidden units (h) as complexity-controlling parameter. Based on test runs and result of 'Regression part a', we have chosen as suitable a range of values for regularization constant and number of hidden units as follows:

$$\lambda \in [10^1, 10^3] \text{ with multiplicative step } 1.58$$
$$h \in (1, 2, 4, 6, 8, 10, 12, 15)$$

Results

In this part we have created a table that shows the best values for the different models:

- For the Linear model we have the best λ regarding the regularization strength and the respective generalization error
- For the ANN we have the optimal value of the number of hidden units and the respective generalization error
- For the Baseline model we have the test error

Outer fold	Linear regression		ANN		Baseline
i	λ_i^*	E_i^{test}	h_i^*	E_i^{test}	E_i^{test}
1	63.1	106.8	8	101.0	170.7
2	63.1	108.9	6	101.9	172.2
3	39.8	110.4	12	100.8	174.7
4	63.1	107.8	6	102.5	176.6
5	63.1	105.5	6	101.3	177.4
6	63.1	101.1	8	93.8	167.9
7	63.1	101.4	10	93.3	168.7
8	39.8	105.3	6	97.6	176.7
9	63.1	109.8	10	100.8	182.1
10	63.1	109.9	10	100.9	180.3

Figures 3,4 show that we found suitable ranges for parameters as all folds contain the minimum error within the range.

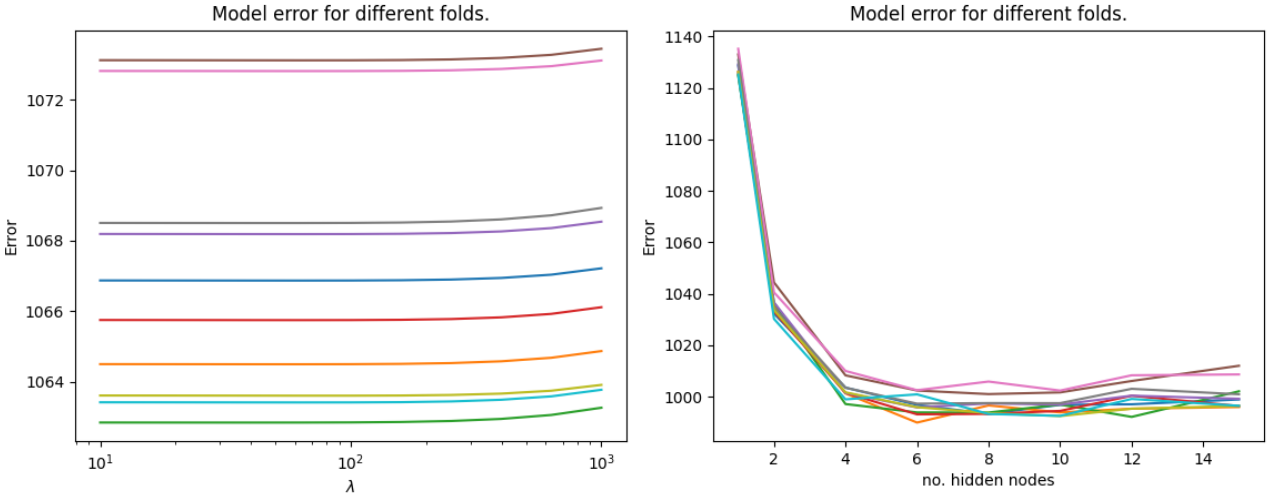


Figure 3, Test error for 10 outer folds.

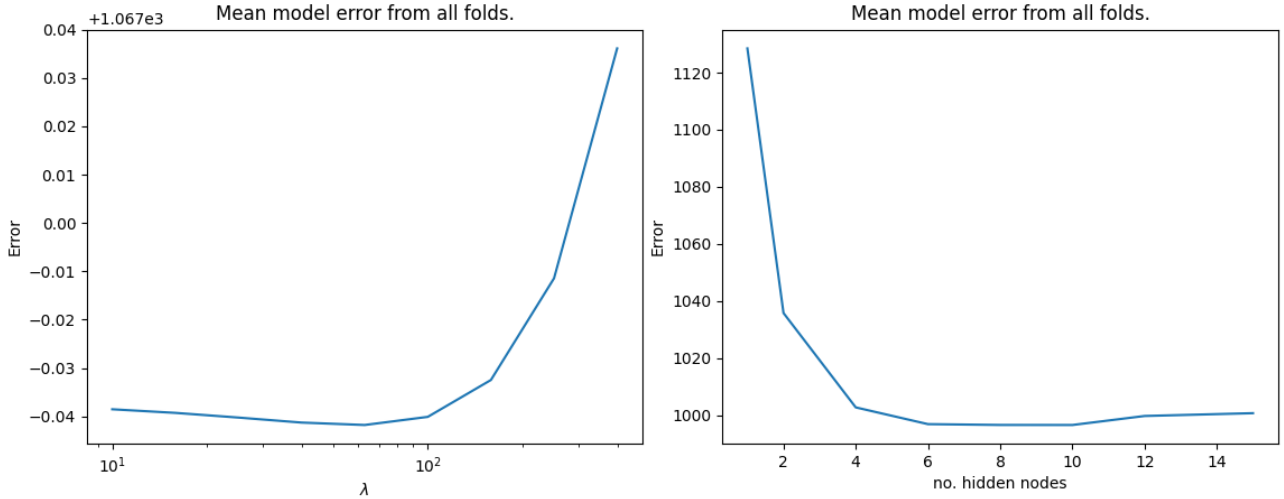


Figure 4, Mean test error of all outer folds.

Statistical evaluation

The minimum error is reached using ANN, although it is only slightly better than linear regression model. In contrast for the Baseline model, we can see that the error is high compared to the other two and that was what we expected from the results. In addition, we can see that the values of lambda are same as final lambda from previous section.

Here we have analysed the 95% confidence intervals for pairwise comparisons and p-value for null hypothesis:

$$CI_{baseline,\,linear} = [66.3,\ 69.7]\ p = 0$$
$$CI_{baseline,\,ANN} = [73.5,\ 77.2]\ p = 0$$
$$CI_{linear,\,ANN} = [6.6,\ 8.0]\ p = 4 \cdot 10^{-97}$$

Classification

Description

We intend to classify people into two categories; earning less than 50,000$ a year and earning more than 50,000$ a year. Therefore, we have a binary classification problem. We will be training and comparing two classifier models to baseline model.

Baseline model finds largest category in training data and predicts this category for all observations in test set. The baseline model does not use training data, only categories in training data.

We will use logistic regression with regularization to prevent any possibility of overfitting. Our regularization term will be $\lambda|\omega|^2$; where ω are weights of logistic regressor. For finding optimal value of regularization constant, we will use two-level cross-validation. We will be finding λ in range $[0.001,\ 10]$. The range was decided on from trial run on wider range.

As the next model, we will use k-nearest neighbours classifier. In this case, we will be finding optimal value of parameter k . We will be using k from list:

$$(1,\ 5,\ 10,\ 16,\ 19,\ 22,\ 25,\ 28,\ 31,\ 34,\ 38,\ 42,\ 50,\ 100)$$

This list is based on previous experiments that showed optimal k in vicinity of 25.

Training

We will be again using two-level cross-validation to evaluate our models with 10 folds in each level.

Outer fold	Logistic regression		KNN classification		Baseline
i	λ_i^*	E_i^{test}	k_i^*	E_i^{test}	E_i^{test}
1	0.464	0.2413	25	0.1644	0.2482
2	0.001	0.2420	42	0.1538	0.2340
3	0.001	0.2627	22	0.1627	0.2442
4	0.464	0.2423	25	0.1578	0.2443
5	0.001	0.2405	28	0.1538	0.2487
6	0.001	0.2394	22	0.1556	0.2474
7	0.464	0.2478	25	0.1736	0.2501
8	1.166	0.2376	42	0.1578	0.2407
9	0.001	0.2498	25	0.1618	0.2396
10	0.001	0.2498	34	0.1587	0.2429

Table 2.

In figure 5, we can see that for each fold we found optimal lambda within searched range, or the optimal lambda found was 0.001 in which case we can consider ideal lambda to be in range $[0,\ 0.001]$ as the test error does not change in this range. We also have good range for k , because we found local minima for error.

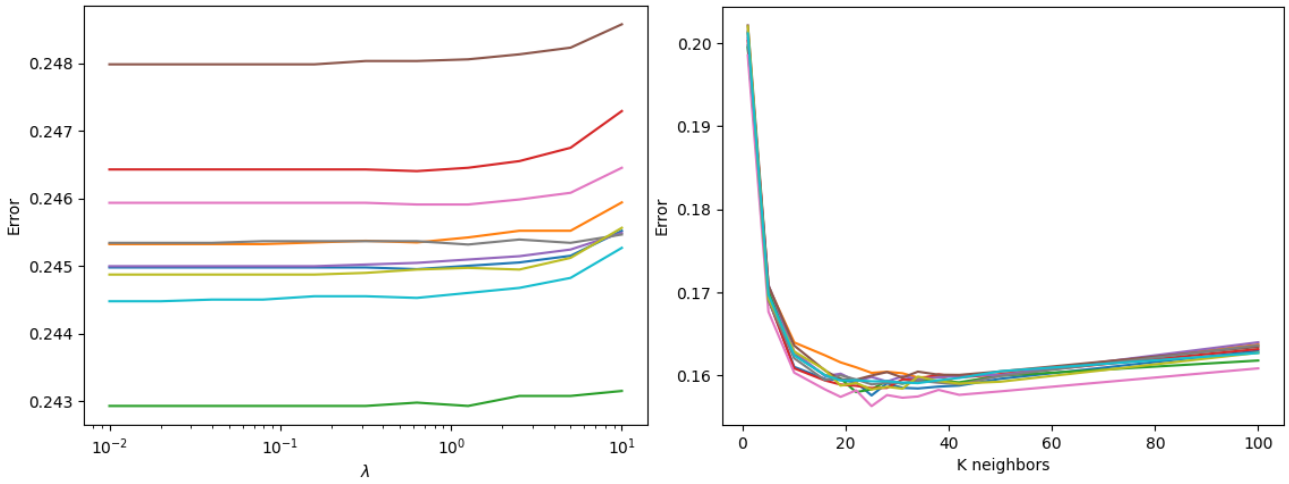


Figure 5, Classification error for all outer folds for linear regression(left) and KNN classifier(right).

Statistical evaluation

We can now evaluate pairwise performance of all three models in setup I, using McNemar's test. Thanks to large number of observations, we have high power level and can easily recognise even slight difference in classification error. We define difference in two models' accuracies:

$$\theta = \frac{n_{12} - n_{21}}{N}$$

We also state null hypothesis that two models have same accuracy:

$$H_0\colon \theta = 0$$

Where n_{12} is number of observations classified correctly by first model and incorrectly by second, n_{21} vice versa.

We start by comparing both models to the baseline model. 95% confidence interval for baseline model compared to logistic regression is $CI_{0.95} = [-0.005,\ 0.007]$ with p-value for null hypothesis 0.67 which means that we can't reject null hypothesis. Logistic regression seems to have similar accuracy as baseline model.

Comparing baseline model to KNN regression gives us 95% confidence interval $CI_{0.95} = [-0.088, -0.080]$ with p-value for null hypothesis 0 with respect to accuracy of floating-point numbers. This means that we can reject null hypothesis and say that KNN classifier performs better on our dataset.

Finally, we can compare two trained models against each other, but it will come as no surprise that KNN classifier will outperform logistic regression. $CI_{0.95} = [-0.090,\ -0.081]$ and p-value $4 \cdot 10^{-309}$.

From results of statistical evaluation, we can safely say that KNN classifier performs better on our dataset than logistic regressor.

Logistic regressor workings

Logistic regression works similarly to linear regression by minimizing $E(\omega)$; where E is error function and ω is the weight vector used for making predictions from data. Each weight in vector corresponds to one input attribute. The bigger the weight, the bigger contribution attribute has. Since our data is normalized, we can compare weights directly to determine, for example, 10 most contributing attributes (both positive and negative contributions in order):

1. (+1.50) Capital gain
2. (+0.48) Education number
3. (+0.31) Relationship: married

4. (+0.21) Age
5. (+0.20) Occupation: executive/managerial
6. (+0.19) Hours per week
7. (-0.18) Relationship: not in family
8. (+0.15) Marital status: married civ. Spouse
9. (+0.14) Occupation: prof. Specialty
10. (-0.13) Occupation: farming/fishing

Comparing these weights to weights in regression part, we can see that earnings are significant contribution for age prediction and vice versa. In both cases, the correlation is positive. Only three other attributes in top 10 are shared by both tasks, of which two have different sign depending on task, what is counterintuitive. Other than that, we can see that age is more dependent on marital status and relationship, and earning are more dependent on occupation.

Discussion

Discussion of results and what we have learned

Let us now have a look at the results of the different parts of the project and what we learnt from them.

First, in the 'Regression part a' we can say that most attributes make sense, and we can have a deeper analysis of them. Looking at the Figure 2 we understand for example that the widowed are not young people and of course this makes sense if we refer everything to the general statistics of an average population. The same could be said for the Divorced and Married people, they tend to be middle aged (surely not youngsters, at least in the country of analysis). On the opposite side we can analyze and understand that generally the ones that are never married tend to be young people and this is in line with what we expected. Same results could be obtained by those who have their own child.

We can look at table 2 to analyze results of the classification models we have tried in order to predict if a candidate earns more than 50 thousand dollars as annual income.

In the table are reported the test errors of each fold for each method used that have been computed through cross validation. To evaluate how accurately the different algorithms, we have used are able to predict outcome values for previously unseen data we are interested in calculating the generalization error E_i^{Gen} for each method i .

We can simply calculate these errors by computing the arithmetic mean of the test errors over the 10 folds, obtaining the following results: $E_{LR}^{Gen} = 0.245$ for logistic regression, $E_{KNN}^{Gen} = 0.160$ for K nearest neighbours and $E_{BL}^{Gen} = 0.244$ for the baseline.

Looking at the parameters we have just calculated we can see that logistic regression has a generalization error comparable to the one obtained with baseline, instead KNN method performs clearly better.

Although we achieved a lower generalization error for KNN we cannot say that our models performed particularly well. This might be as a result of the features we worked with. Furthermore, since there is a very high variance within the data it is not advised to use PCA for dimensionality reduction as it would only select features with very high variance which doesn't necessary implies that those features are related to our aim.

From the analysis carried on in the previous report we stated that, since previous attempts by others achieved error rate as low as 15%, We could aim for error under 20%, so in conclusion we can say that our objective has been fulfilled.

Previous study analysis

In the paper: "A Beginners Guide to Data Analysis & Machine Learning with python — Adult Salary Dataset" by Anirudh Raj (https://towardsdatascience.com/a-beginners-guide-to-data-analysis-machine-learning-with-python-adult-salary-dataset-e5fc028b6f0a), logistic regression is applied to the Adult dataset to predict if someone is earning more or less than 50 thousand dollars a year based on all the other attributes. Although data preparation in this case is quite different from what we have done, since much less work has been done in this case and they have just limited to drop observations with missing attributes, it's still a valid benchmark for our results.

To perform logistic regression the dataset is divided into a dependant feature that is income and independent features that are chosen to be relationship, race, occupation, gender, marital status, and work class. Y axis is therefore the dependent variable, and for the X axis relationship, education, race and occupation columns are concatenated using numpy.c_ provided by the numpy library.

Then it is arbitrarily chosen to split the dataset into a train set and a test set of size 77% and 33% respectively.

After training, the model is evaluated using accuracy evaluation metric provided by sklearn library of Python.

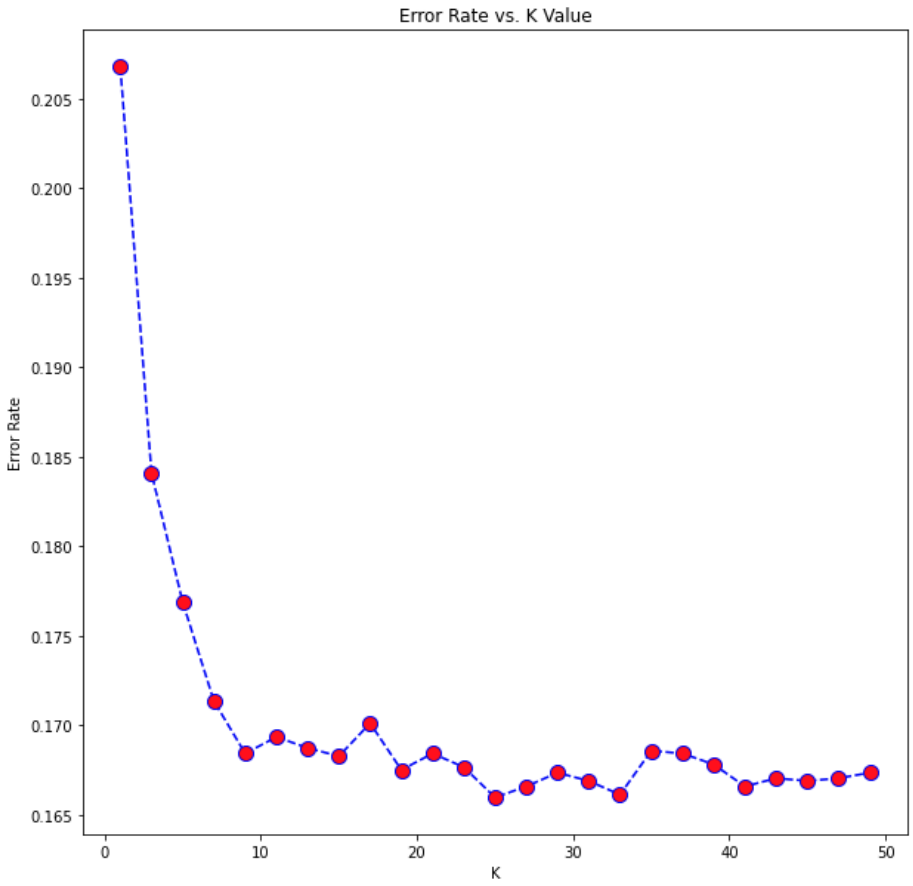
The accuracy value reportedly achieved is 76% and it is consistent with what we have obtained performing logistic regression, this value is pretty small, especially if we take into account the imbalanced distribution of our dataset, that implies that since 75% of data are people who earn under 50k, we can get 25% error rate by simply always predicting lower earnings.

In the paper "UCL Adult Data for Earning Potential of people" by Aniket Mishra (https://aniket-mishra.github.io/projects/Adult-UCI-Dataset-Analysis.html), various classification models are compared, again with the aim of finding the best model for predicting the annual income based on the other attributes in the dataset.

First, logistic regression is applied, in this case we can see that the author has chosen to split the dataset into a train set of size 80% and a test set of size 20%. The target is again the income attribute column but in this case for the X axis all the other features of the database are accounted, including the final weight column "fnlwgt" that we chose to drop in our data preparation process.

After training the model the obtained accuracy results are considerably higher than in the previous analyzed paper, with an accuracy score of 82.8%.

For KNN classifier model it is found that the optimum value of K is 25 as we can see from the graph reported below.



Accuracy achieved with this model is reportedly equal to 83.1% c.a., so we can see that KNN did a little better than Logistic regression. This can be because of the number of features we have in the dataset. We can see that the accuracy found is consistent with our performance in terms of error obtained by applying KNN.

We can also notice that the best performing algorithm is found to be KNN and that is another time consistent with our results.

Exam problem solutions

Question 1:

Question 2: C

We need to calculate the impurity gain with the formula:

$$\Delta = I(r) - \sum_{k=1}^{K} \frac{N(v_k)}{N(r)} I(v_k).$$

We can see that N(r), corresponding to the dimension of the root is 135 and can be calculated as the total of the observation:
N(r) = 33+4+28+2+1+30+3+29+5
Since we need to calculate the impurity gain for just one split, we have that k=2 and that N(v1)=1 and N(v2)=134. This is because the left branch will contain by definition all the observations having x7=2, in this case just one, and the other will contain the rest of the observations, having chosen a two-way split by definition.
We need now to calculate the impurity functions I(r), I(v1) and I(v2) using classification error method with the formula:

$$\text{ClassError}(v) = 1 - \max_c p(c|v).$$

We have that maxp(c|v1)=1, this is trivial since N(v1)=1, and therefore I(v1)=0.
For the root we have that maxp(c|v) = 37/135 and therefor I(r)=1-37/135
For the impurity function of the right branch v2 we have that maxp(c|v2)=37/134 and therefore I(v2)=1-37/134.
Now we can compute the value of delta:

$$\Delta = 1 - \frac{37}{135} - 0 - \left(\frac{134}{135}\left(\frac{134-37}{34}\right)\right) = \frac{1}{135} = 0{,}0074$$

Question 3: A

i = number of neurons in input layer = 7
nh = number of neurons in hidden layer = 10
o = number of neurons in output layer = 4

Number of connections between input layer and hidden layer = 7 * 10 = 70
Number of connections between hidden layer and output layer = 10*4 = 40
Number of connections between bias of input layer and hidden layer = 10
Number of connections between bias of hidden layer and and output layer = 4

Number of parameters to be trained = 70+40+10+4=124

Question 4: D
We can see from figure 3 that the answer is D since the outcome of node C creates a leaf node indicating if a particular observation has congestion level 4 or not, since we see from figure 4 that

this decision corresponds to say that an observation fall into the right part of the PCA plot, precisely where b1>=-0.016 we can exclude all the other answers.

Question 5: A
Time for ANN = 3125 ms
Time for logistic regression = 3675 ms
Total time = 6800 ms.

Question 6: B
For each observation we compute the values of $\mathbf{y_1}$, $\mathbf{y_2}$, $\mathbf{y_3}$ and subsequently calculate the per-class probability using the softmax transformation as given obtaining the following results:

P(y=4|**y**)= 3,02532E-06 for observation A
P(y=4|**y**)= 0,73046 for observation B
P(y=4|**y**)= 1,7675E-06 for observation C
P(y=4|**y**)= 1,44996E-05 for observation D

We choose than observation B as the one with highest probability of being classified as y=4.

Appendix A.

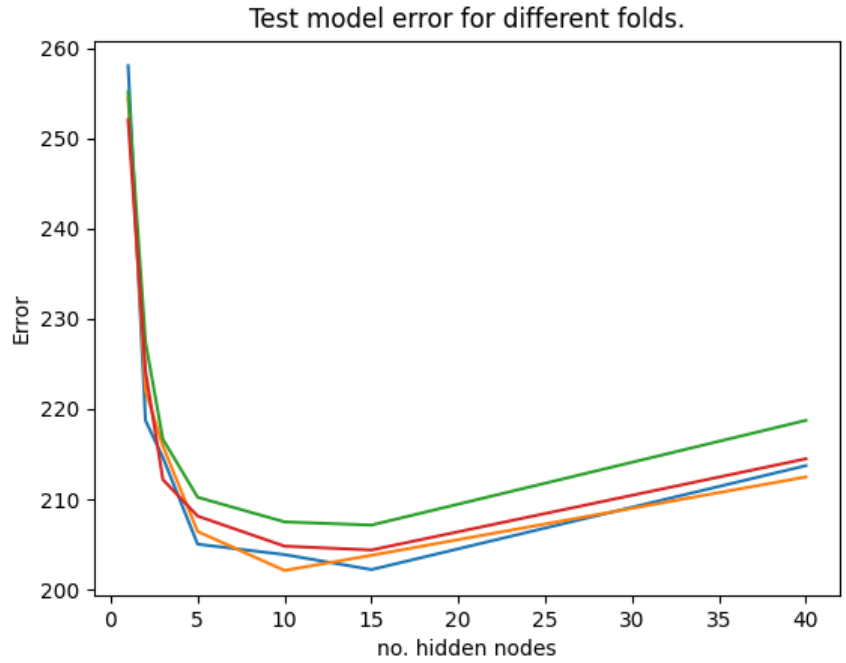


Figure A.1, Trail run to determine suitable number of hidden nodes.

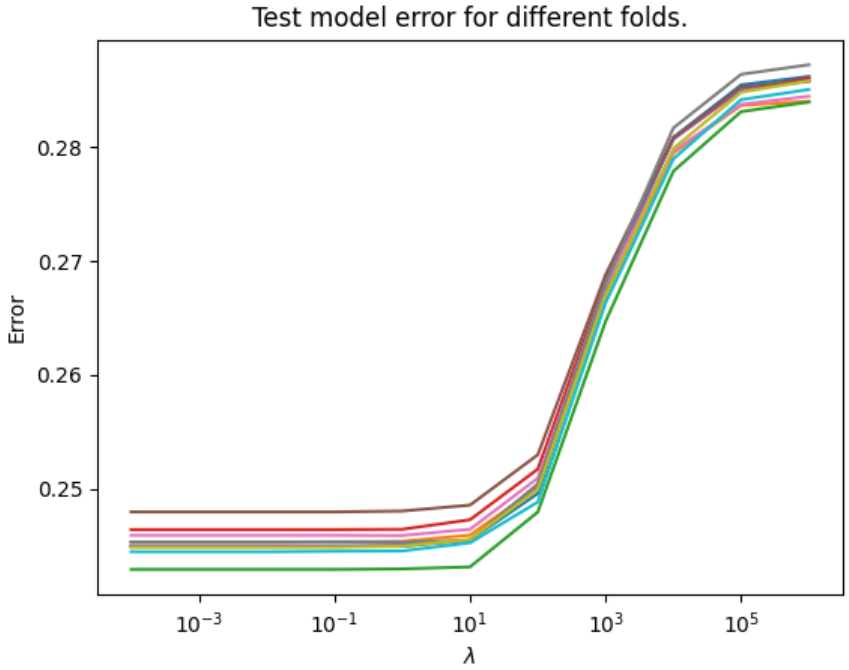


Figure A.2, Trail run to determine suitable regularization constant for classification.