DTU ML Project 2

Jai Murhekar s237013
Nicolai Pavliuc s240366
Alice Coimbra s237108

April 10, 2024

	Regr Part a	Regr Par b	Classification	Discussion	Exercises
Jai (s237013)	10%	10%	80%	10%	40%
Nicolai (s240366)	80%	50%	10%	10%	30%
Alice (s237108)	10%	40%	10%	80%	30%

Table 1: Contributions

1 Regression

1.1 Part a

The main regression task for the Abalone dataset is predicting the number of rings of an abalone from it's physical measurements.

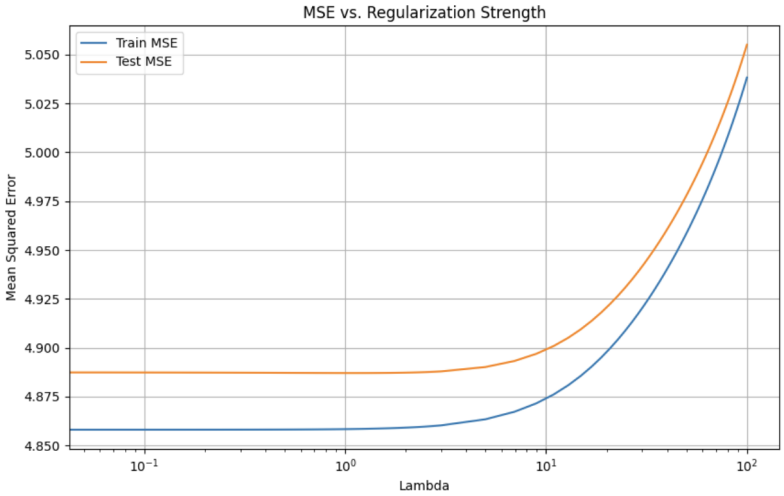
Before getting started with regression, we normalize the values in X. We do this because linear models are sensitive to magnitude of features.

Next step is applying *one-of-K* encoding. The *Sex* attribute is split into *Sex_I*, *Sex_M*, *and Sex_F*. One-of-K is applied since only numerical values can be used in regression. Also, simply using values like 0, 1, 2 is not ok, because the model would incorrectly start treating the attribute as some ratio.

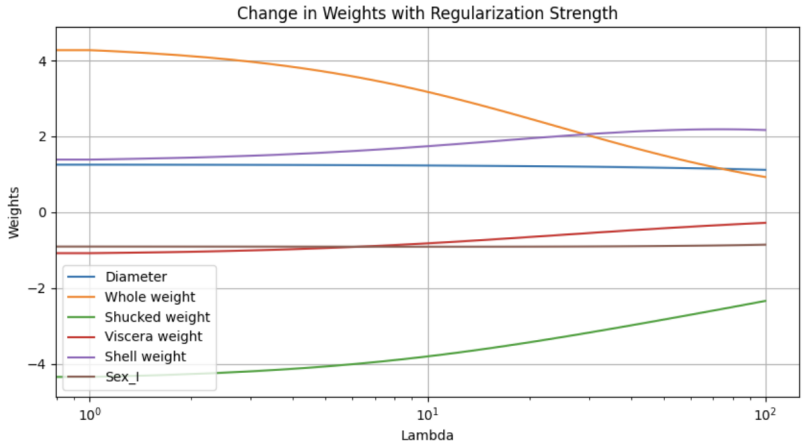
Additionally, since we will predict the number of rings, which is originally a discrete attribute, we convert this attribute to continuous type.

The following attributes are selected for the following L2 regularization: *Diameter*, *Whole weight*, *Shucked weight*, *Viscera weight*, *Shell weight*, *Sex_I* (k=6). These were picked after a very basic 2-level cross-validation. 1) On the outer level, we use 1 holdout (20% test data) to estimate the performance a linear regression model. 2) On the inner level, we use 5-fold cross-validation to perform sequential feature selection. The linear regression model with those 6 selected attributes gives us a gen. err. of 4.927%.

The next step is picking the best lambda value for L2 regularization. The lambdas for selection is 100 values in range [0;100]. The figure below shows the gen. err. for different lambda values.



Although it is hard to see on the graph, the test MSE actually decreases slightly at lambda close to 1. The point where the test MSE is at its lowest represents an optimal balance between bias and variance (model is complex enough to capture the underlying patterns in the data without overfitting). For larger lambda values, test MSE only grows. The train MSE, on the other hand, never decreases at any point. As regularization becomes higher, the models get more biased and simple. They start to perform similar on both train and test data.



The figure above shows the behaviour of coefficients of models for different values of lambda. As lambda grows, coefficients are penalized more and more, leading to underfitting models.

After cross-validation, we discover that the optimal lambda value is 1.1020. For this value, the test MSE is 4.88. The table below shows the coefficients for the features from the model with optimal lambda. Table shows the average for the folds with that lambda.

Attribute	Diameter	Whole weight	Shucked weight	Viscera weight	Shell weight	Sex_I
Weight	1.25226264	4.25767539	-4.33605309	-1.07657924	1.39443432	-0.90647298

The positive coefficients (e.g. Diameter, Whole weight, Shell weight) lead to increase of number of rings, while negative coefficients lead to decrease. It's worth mentioning that the coefficients for Whole weight and Shucked weight are unusually large. This is possibly because of their high correlation (0.97) on the covariance matrix. Apart from that, everything else looks reasonable.

1.2 Part b

A relevant question when applying regression to a dataset is to evaluate different models. In this case, three different models were compared: an Artificial Neural Network, the regularized linear regression from the previous section and a baseline. For this purpose, 2-levels cross validation was applied with K_1=K_2=10 folds to compare between the models. The inner loop is used to select the optimal lambda (in the case of the linear regression) and the optimal number of hidden layers (in the case of the ANN), while in the outer loop the generalized error is estimated. The error measure used is the squared loss per observation. The table below shows the results of cross-validation for regression models. For linear regression with regularization, we tested 30 lambda values in the range between [0;3]. This is because in the previous part, we have seen that we get the lowest gen. errors for lambda being around 1. For the ANN model, different number of hidden layers were tested in the range 1-5. As for the baseline model, a linear regression model with no features was used.
From the results shown in Table 1, it can be concluded that both the ANN and linear regression models perform better than the baseline, as expected since this model is very simple and just computes the average of the data and uses this for the prediction. Between the first two models, the generalized errors for each fold are quite similar, so it is not possible to clearly distinguish the performance of these two models based om these results. Furthermore, there does not seem to be a relation between the complexity parameter lambda and the corresponding generalization error. On the other hand, in the ANN model it seems that for a number of hidden layers equal to 4 the generalization error is higher than when the number of hidden layers is three. This might be due to overfitting, as the model becomes more complex.

Table 2: Cross-validation for regression models

Outer fold	ANN		Linear Regression		Baseline
i	E_i^{test}	h_i	E_i^{test}	λ	E_i^{test}
0	4.46	4	4.82	0.517	11.24
1	4.49	4	5.05	0.517	10.40
2	4.03	4	4.60	0.517	8.86
3	4.13	4	4.61	0.310	9.18
4	4.72	4	5.95	0.413	12.59
5	4.26	4	5.16	0.620	10.65
6	5.11	4	5.74	1.241	12.55
7	4.45	3	4.63	1.034	11.33
8	3.25	3	3.74	1.137	8.39
9	3.85	3	4.52	2.68	8.755

1.2.1 Comparing accuracy of regression models

The summary of differences between model's accuracy are given in the table below (alpha = 0.05).

Table 3: Comparing accuracy of regression models

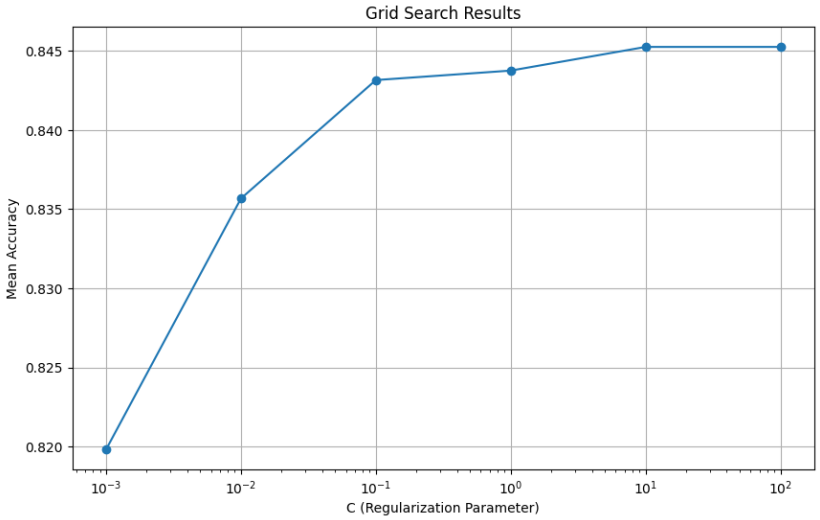
Model Comparison	**p-value**	**Confidence Interval (%)**
Linear Regression vs Baseline	5.961×10^{-26}	$(-6.96 - -4.83)$
ANN vs Baseline	3.462×10^{-8}	$(0.03 - 0.07)$
ANN vs Linear Regression	4.579×10^{-26}	$(4.88 - 7.02)$

The table can be interpreted the following. LR is significantly better than baseline. ANN is also better than baseline. However, ANN is superior of the 3. The low p-values suggest that results are accurate.

2 Classification

We classified the abalone into three age categories (young, medium, old) based on ring counts (young - 7 rings or less, medium - 8 to 15 rings, old - more than 15 rings). This is a multi class classification

problem. We compare logistic regression, naive bayes, and a baseline method. In logistic regression, we use grid search as a test run to find out the optimal value of the regularation parameter, C. We find that the best value of C is 100 ($\lambda = 0.01$) with an accuracy of 84.5 percent (see figbure below)



Hence in our comparison, we choose the range of λ to be $0.005 - 0.015$ for logistic regression. For Naive Bayes, we choose b as the complexity controlling parameter with values ranging from 0 to 0.5. The baseline method involves predicting the largest class observed in the training data for all instances in the test data. In order to compare these three methods, we use two-level cross-validation using 5 outer folds. Each fold splits the data for training and testing, and models are trained on one part and evaluated on the other. The error rate is then computed for each model across all folds.

The results of the cross validation are given below

Table 4: Cross-validation results

Outer fold	Naive Bayes		Logistic Regression		Baseline
i	E_i^{test}	b_i	E_i^{test}	λ	E_i^{test}
0	20.8	0.5	15.2	0.005	28.0
1	19.6	0.5	16.3	0.010	26.7
2	22.4	0.4	16.4	0.005	28.1
3	22.2	0.3	16.5	0.005	26.1
4	19.4	0.5	13.3	0.005	22.8

From the above table we see that both Naive Bayes and Logistic Regression outperform the baseline method in terms of error rate. Logistic Regression has a slightly lower average error rate compared to Naive Bayes, indicating that it may be a better choice for this classification task. Additionally, the average value of λ for Logistic Regression is close to 0.006, suggesting a moderate level of regularization is preferred for this dataset.

We use the McNemar's test to compare the three models. The McNemar's test is a statistical test used to compare the performance of two models on paired data. The test examines whether the differences in performance between the two models are significant or not. The results are as follows :

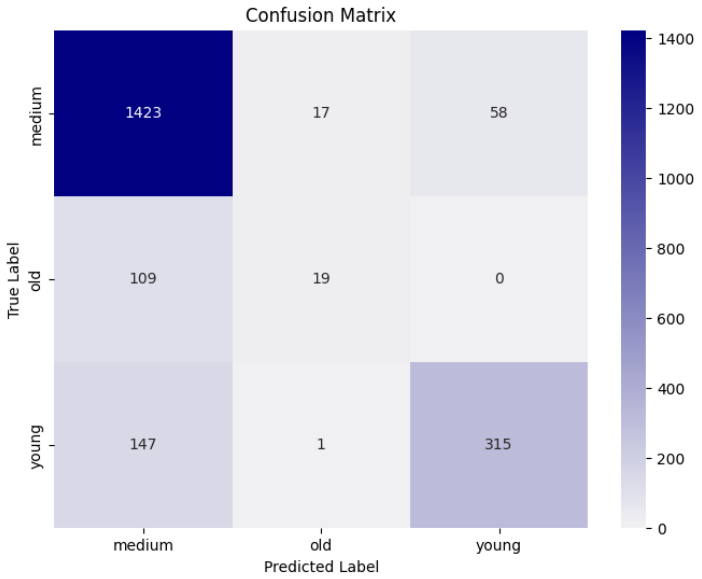
Table 5: Results of McNemar's Test

Model Comparison	**Statistic**	**p-value**	**Confidence Interval (%)**
Logistic Regression vs Naive Bayes	4.00	5.65×10^{-8}	$(53.0 - 65.0)$
Logistic Regression vs Baseline	15.00	2.04×10^{-4}	$(62.0 - 76.0)$
Naive Bayes vs Baseline	0.00	2.22×10^{-16}	$(72.0 - 87.0)$

We can interpret the results of the McNemar's test as follows : Both Logistic Regression and Naive Bayes models have significantly lower p-values ($p < 0.05$). compared to the baseline, suggesting that

they both significantly outperform the baseline method. We suggest using the Logistic Regression in comparison to Naive Bayes because the difference in performance is statistically significant and it also has a lesser error rate as can be seen from Table 4.

Logistic regression works by fitting a sigmoid function to the training data, enabling classification into distinct categories based on input features. We run logistic regression with $\lambda = 0.005$, and use 50 percent of the data for training and the remaining for testing. We get a very good accuracy of 84 percent which is quite close to the accuracy we obtained in grid search. We also display the plot of the confusion matrix which shows the performance of the logistic regression in classifying the age of the abalones. Additionally, we use the same features (number of rings) in the regression.



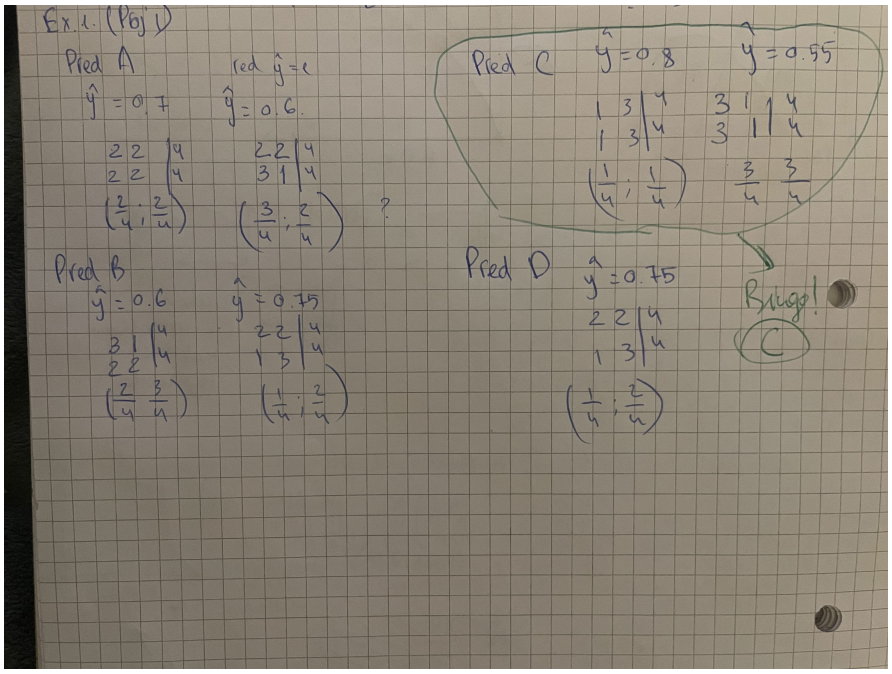
3 Discussion

This report focused on applyting regression and classification to our Abalone dataset. We started by looking at the effect of the regularization parameter lambda. As this parameter increases, the model becomes less complex, which reduces the risk of overfitting. However, lambda cannot be too high as this oversimplifies the model. We confirmed this when we plotted lambda against the weights and we saw that the weights tended to be closer to zero when lambda was increased. In this report, we were able to find the optimal value for lambda that is complex enough without overfitting. We then proceeded to compare the performance of different models through two-level cross validation where we saw that, as expected, more complex models have a better performance than a simple baseline model. We then saw the importance of doing statistical analysis to be able to confidently compare between models.

There are numerous studies that analyse the regression and classification tasks applied to the Abalone dataset. Many of these studies look at different score metrics and parameters tuning. For instance, when using ANN for regression the authors of this paper [1] looked into a number of different parameters: number of hidden layers, batch size, number of epochs and noise levels. Similarly to the work done in this report, the authors found the optimal number of hidden layers to be 3. Besides the models mentioned in this report, many other models can be used for classification and regression, namely backpropagation feed-forward neural network (BPFFNN), K-Nearest Neighbors (KNN), Decision Tree, Random Forest and Support Vector Machine [2]. In terms of the performance metrics used, although they vary between authors, in general Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are the most widely used. Morevover, the accuracy levels for classification are similar to what we have achieved (around 85 percent). It is interesting to see that depending on the dataset one model might be more accurate than the other, highlighting the importance of using different models and not use exclusively one model and apply it to all datasets.

4 Exercises

4.1 E1



4.2 E3

The correct answer is option A. The neural network consists of:

- 7 input neurons
- 10 hidden units with sigmoid activation
- 4 output neurons with softmax activation

Each connection contributes to a parameter, totaling 124:

$$7 \times 10 + 10 + 10 \times 4 + 4 = 124$$

Thus, the correct answer is A) 124 parameters.

4.3 E5

For each of the K1 outer folds, K2 models need to be trained and tested, for both the ANN and logistic regression models, on the 5 λ and n_h values and then a single model needs to be trained and tested to estimate the generalization error for that fold. Then, for each of the two types of models, the total number of models to be trained and tested is:

$$K_1 \times (K_2 \times M) + K_1 = 5 \times (4 \times 5) + 5 = 105 \tag{1}$$

The time it takes to compose the table is approximately the time it takes to train and test all the models:

$$105 \times (20 + 5) + 105 \times (8 + 1) = 3570\,\text{ms} \tag{2}$$

4.4 E6

The correct answer is option B. We calculate the probability for being assigned to class 4 for each option using the below code. We see that option B has the largest probability.

```
def probability(b1, b2):
    w1 = np.array([1.2, -2.1, 3.2])
    w2 = np.array([1.2, -1.7, 2.9])
    w3 = np.array([1.3, -1.1, 2.2])

    y1 = np.dot(w1, [1, b1, b2])
    y2 = np.dot(w2, [1, b1, b2])
    y3 = np.dot(w3, [1, b1, b2])
    
    denominator = 1 + np.exp(y1) + np.exp(y2) + np.exp(y3)
    return 1 / denominator

print("Option A:", probability(-1.4, 2.6))
print("Option B:", probability(-0.6, -1.6))
print("Option C:", probability(2.1, 5.0))
print("Option D:", probability(0.7, 3.8))
```

```
PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE

PS C:\Users\jaimu> python -u "C:\Users\jaimu\AppData\Local\Temp\tempCodeRunnerFile.python"
Option A: 3.0253229959961156e-06
Option B: 0.7304570363062247
Option C: 1.7674983200204532e-06
Option D: 4.6563844858874605e-06
PS C:\Users\jaimu>
```

References

[1] M. Misman, A. A Samah, N. Aziz, H. Majid, Z. Ali Shah, H. Hashim, and M. F. Harun, "Prediction of abalone age using regression-based neural network," pp. 23–28, 09 2019.

[2] S. Guney, I. Kilinc, A. Hameed, and A. Jamil, *Abalone Age Prediction Using Machine Learning*, pp. 329–338. 04 2022.