Danmarks Tekniske Universitet

02450 Introduction to Machine Learning and Data Mining Project 2

Group 12
Aleksandar Lukic - (AL) - s194066
Marco Andreis - (MA) - s243116
Martina Bellini - (MB) - s243118

Lecturer

Georgios Arvanitidis

Section No. & Title	AL	MA	MB
1 Regression Part A	20%	40%	40%
2 Regression Part B	20%	40%	40%
3 Classification	60%	20%	20%
4 Discussion	40%	40%	20%
5 Exam problem solutions	33%	33%	33%

Contribution % to report sections per group member.

November 14th, 2024 at 17:00

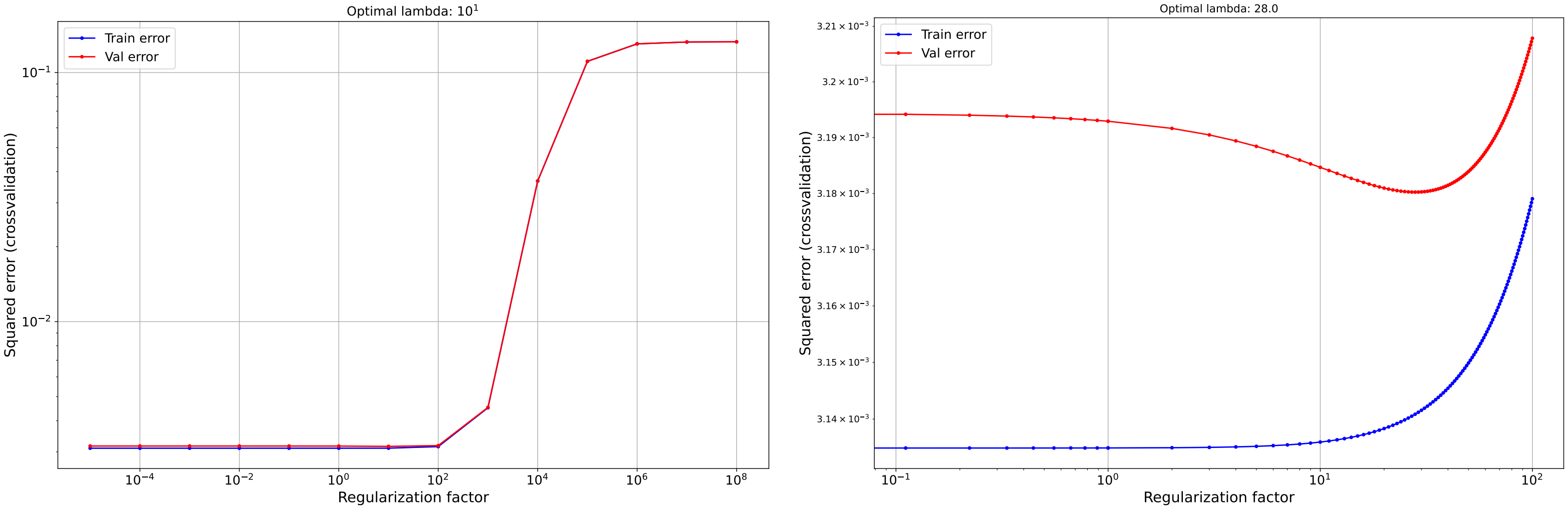
Summary

1 Regression Part A

To determine the target on which to perform regression we rely on the analysis made for the report 1 of the course, during which we verified, using PCA, that all the physical attributes of the abalones are linearly correlated. Additionally these measurements can be divided between those of the dimensions of the abalones and those regarding, instead, the weight of the specimen. The correlation between the attributes of each group was found to be really strong - PCC mostly between 0.90 and 0.95. The same can also be seen across the two, but in this case the relation was shown to follow an exponential law. This was also observed by the authors of the original papers [1].
Based on this consideration we decided to focus on predicting the *Whole Weight* from the dimension measurements by taking their logarithm to reconduce the relation to a linear one. To perform the analysis we used the processed dataset created previously, meaning that the distribution of all the features are normalized.

As a comparison for the performance of the regularized regression models, we utilized an upper ceiling, consisting in a model which does not have access to any of the features and, therefore, is limited to always guessing the mean of the target attribute. On the other hand, we used an unregularized regression model, which achieved high performance both on the training and test set. We will be taking into account this behavior in the following paragraphs.

The initial testing used λs that varied in the 10^{-5} - 10^{9} range. To obtain an estimation of the generalization error we performed 10-fold cross-validation, therefore each value of λ was used to estimate the weights of a regression model on 10 different variations of the dataset, the train and test errors were then averaged along the folds, resulting in the curve shown in *Figure 1a*.



(a) 10^{-5} - 10^{9} *interval*

(b) 10^{-1} - 10^{2} *interval*

Figure 1: *Changes in prediction performance based on the regularization.*

The behavior observed for large λs is the expected one: the regularization becomes too significant, constraining the weights to assume small values, eventually all reaching zero (*Figure 2*). This causes the model to always predict a single value, represented by the intercept term, which is not subjected to regularization and thus can still be correctly learned. The value for this term that minimizes the loss is the mean of the target attribute, which makes the model equivalent to the baseline.

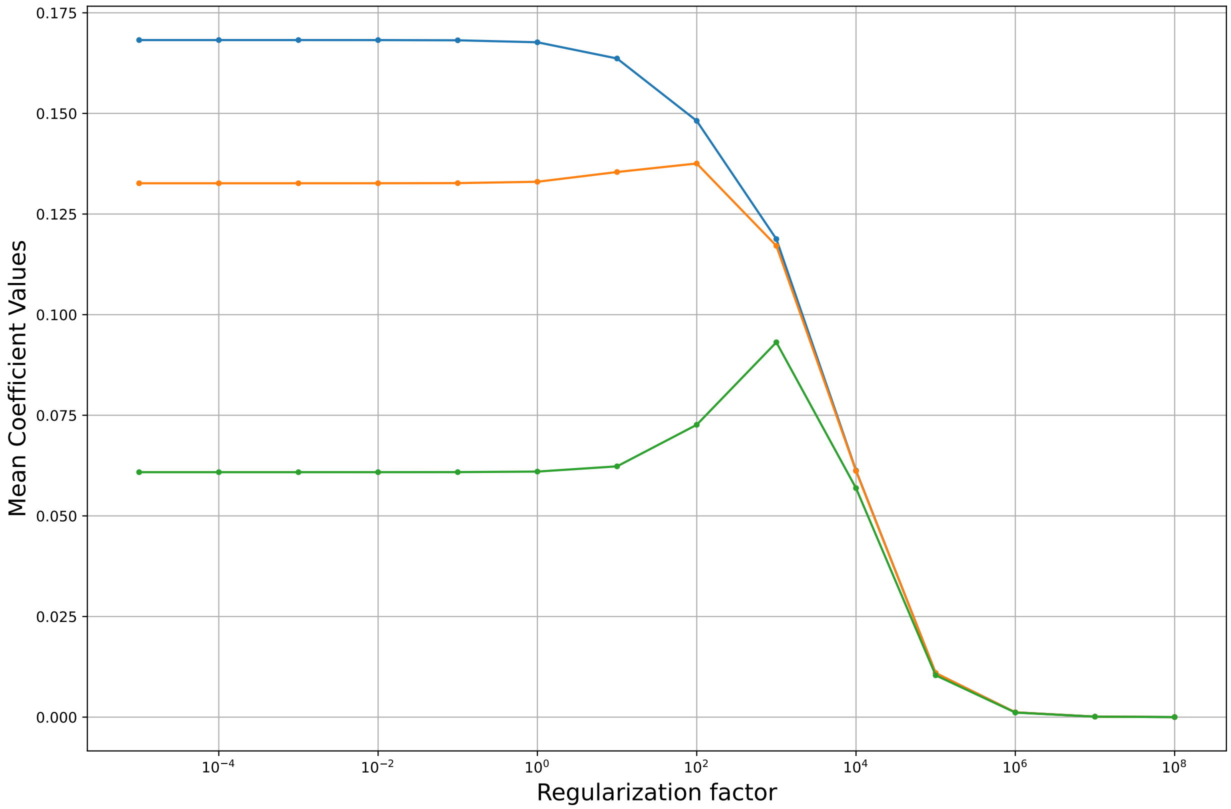


Figure 2: *Changes in the weights based on regularization*

The results for the smaller λ values are more difficult to interpret, the expectation is to observe overfitting: having the train error be equal to that of the unregularized model (as regularization is not meaningful), and also significantly lower than the test error. This does not happen, instead both errors remain similar across all the regularization factors.
To investigate further and verify that what was obtained was not due to a scale problem, testing was extended to focus with higher resolution on the 10^{-1} - 10^{2} interval (results shown in *Figure 1b*). This allows to see that, even if really small, there is improvement which can be attributed to the regularization.

Estimating an optimal value for λ is difficult as by running the cross validation multiple times the results change slightly in the 20-40 range.
The estimated generalization errors of the utilized model can be found in *Table 1*.

Model	**Regularized linear regression**	**Unregularized linear regression**	**Baseline**
Test error	0.00306	0.00301	0.13156

Table 1: *Test errors of the different linear models*

The reason why the difference between train and test error is so small can be attributed to the distribution of target attribute. As was shown in the previous report, the correlation between the dimension measurements (specifically the length) and the whole weights is really strong, also we found that the data does not have any significant outliers from this correlation. Based on this consideration, we hypothesize that overfitting is not significant as even an overfitted model would still successfully capture the underlying truth.

To elucidate the importance of each individual attribute for the performance of the prediction we repeated the experiment with the same setup, but removing each feature. The resulting errors (shown in *Table 2*) demonstrate that ignoring one of the features does not have a significant impact on the test error, this is in agreement with the observation that the three attributes are linearly dependent and strictly correlated.

Removed attribute	**None**	**Length**	**Diameter**	**Height**
Test error	0.00306	0.00378	0.00342	0.00373

Table 2: *Test errors of the models trained without one of the attributes*

2 Regression Part B

Building upon the results obtained using regularized linear regression to predict the whole weight of the abalones. Testing proceeded by investigating how artificial neural networks (ANNs) compared in this task to the previous approach. To do so we utilized two level cross-validation using ten folds for both the inner and outer levels, using again the mean squared error as the performance metric.

Using as reference the previously estimated best λ values, the interval utilized was from 10^{-1} to 10^{2}. While, for the ANN we limited the complexity of the model by using only a single hidden layer. To analyze how different architecture impacted the overall performance we tested ANNs having an increasing number of hidden nodes (h) from 1 to 14.

In *Table 3* are shown the optimal values for λ and h found in the outer folds, as well as their respective test errors and those of a baseline model (which is the same as before).
The optimal λs are compatible with the range found in the previous analysis and the errors are also comparable. It can be seen that these values, as well as those found for h* fluctuate,

this is acceptable as the dataset is splitted in different ways across the outer folds, meaning that the data distribution between train and test set (for both outer and inner fold) is also different - this could result in varying optimal values being selected. Another factor affecting this variance is the fact that, as shown before, there are large intervals of regularization strength where the test errors change only slightly.

Outer fold	Linear regression		ANN		baseline
i	λ_i^*	E_i^{test}	h_i^*	E_i^{test}	E_i^{test}
1	40.0	0.00518	10	0.00642	0.15591
4	40.0	0.00580	13	0.00562	0.14196
5	40.0	0.00463	10	0.00359	0.13217
2	40.0	0.00458	13	0.00401	0.12247
3	40.0	0.00412	8	0.00477	0.13399
6	50.0	0.00576	11	0.00812	0.12182
7	40.0	0.00517	13	0.00629	0.11638
8	1.0	0.00448	11	0.00982	0.13253
9	50.0	0.00561	10	0.00666	0.12883
10	50.0	0.00529	14	0.00378	0.12955

Table 3: *Optimal λ and h values with respective errors in each outer fold*

From the table it appears that the optimal ANNs tend to achieve slightly worse results than the optimal linear models, also both perform significantly better than the baseline. To give a statistical basis to these observations, the best value for λ^* and h^* are computed by taking the averaged and the estimate of their generalization errors across the outer folds are also calculated (*Table 4*). For h it has to be said that obviously an ANN can only have an interger value of hidden nodes, therefore the best performing network would have 11 of them. Then a t-test is used to determine if the difference in estimated generalization errors is statistically significant, the test also repeated to confirm that both models are better than the trivial baseline. The results are shown in *Table 5*.

	Average Value	E^{gen}
h^*	39.1	0.00506
λ^*	11.3	0.00591
baseline	-	0.13156

Table 4: *Estimated optimal values for λ and h*

Null Hypothesis	P-value	Confidence Interval	
		Low	High
$E^{\text{gen}}_{\lambda^*} - E^{\text{gen}}_{h^*} = 0$	0.04325	5.69×10^{-5}	0.00296
$E^{\text{gen}}_{\lambda^*} - E^{\text{gen}}_{\text{baseline}} = 0$	9.11×10^{-10}	0.11546	0.13754
$E^{\text{gen}}_{h^*} - E^{\text{gen}}_{\text{baseline}} = 0$	1.10×10^{-9}	0.11445	0.13685

Table 5: *T-test results for differences in generalization errors*

The p-value allow us to reject the null hypothesis of the two models being identical (using 0.05 as the significance threshold) - additionally, from the confidence intervals, we can say that there is statistical evidence supporting that the regularized linear regression model performs better than the ANNs, as the interval is positive. As expected both models are better than the trivial baseline, this is important as observing the opposite would mean that there was a problem in the application of the methods that would lead the models to not learn the relation between the analyzed attributes.

The approach used is an example of Setup I, this is because the inner folds are only used to select the best complex-controlling factors for the linear regression and the ANN independently, while the comparison between the models is made after the selected models are tested on the outer test set. This means our results are limited by the specific splits that where generated for the outer folds. To make our results more general it would need an additional cross-validation level, but we reserved from following to this path as it would multiply by a factor of 10 the already long computation time needed to train the ANNs.

3 Classification

3.1 The classification problem

Using the parameters *Length*, *Diameter*, *Height*, *Whole weight*, *Shucked weight*, *Viscera weight* and *Shell weight*, we will attempt to classify the age of the abalones, divided into 3 classes based on the number of rings. Thus, we are trying to solve a *multi-class classification* problem. We will set up 3 different models: baseline, Logistic regression and K-Nearest-Neighbours (KNN).

Baseline In order to benchmark the different models, we use a baseline classifier, which simply classifies every point as the mode class, i.e. the class with the most entries in the data.

Logistic regression Using a regularized linear regression classifier (as covered in section 14.1), we are able to train the model to find the optimal weights for each feature. The complexity controlling parameter is the regularization factor, λ, equally logarithmically distributed between 10^{-3} to 10^{2}.

K-Nearest-Neighbours The complexity controlling parameter in the KNN classifier, is the optimal number of neighbours, k_i, which is chosen for the model yielding the lowest generalization error. The parameter was tested for $k_i \in [1 \; ... \; L]$, where $L = 40$ was the maximum no. of neighbours.

3.2 Two-level cross-validation

The two-level cross-validation is used with $K1 = K2 = 10$ for the folds. In the inner folds, the classification error rate is calculated for each parsed complexity-controlling parameter. These error rates are then parsed to the outer fold, where the optimal parameter is chosen by the lowest error rate. The minimum error rates and optimal parameters for each outer fold are shown in Table 6.

The median optimal parameters and mean error rates are thus shown in Table 7. From this, we get the impression that both of our classifiers perform slightly better than the baseline. However, with a minor improvement of ≈ 8% for the logistic regression over the KNN, it is difficult to conclude anything just yet.

Outer fold	KNN		Logistic regression		baseline
i	k_i	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	1	47.4	0.0036	34.2	65.6
2	1	47.4	0.0036	39.0	65.1
3	1	42.4	0.0010	36.0	66.7
4	1	42.4	0.0010	31.9	67.6
5	1	42.9	0.0010	36.2	66.9
6	1	45.8	0.0010	36.2	62.8
7	1	40.3	0.0010	32.1	65.5
8	1	43.2	0.0010	37.2	63.8
9	1	38.8	0.0010	30.9	62.4
10	1	41.0	0.0010	35.5	67.1

Table 6: Two-level cross-validation used for benchmarking model performances.

KNN		Logistic regression		Baseline
k	E_μ^{test}	λ^*	E_μ^{test}	E_μ^{test}
1	43.16	0.0010	34.92	65.35

Table 7: Summary of benchmark data.

3.3 Statistical evaluation of models

We have applied a *correlated t-test for cross-validation* from *Setup II* for our statistical evaluation. The null-hypothesis in this test, is that the two compared models yield the same performance. For the hypothesis to be rejected, the p-value must be $p \leq 0.05$. With each p-value seen in Table 8 being well below $p \leq 0.01$, there is strong evidence against their performances being the same. Specifically, we can rate the performances from worst to best as: **Baseline < KNN < Logarithmic regression**.

Model 1	Model 2	t-score	p-value	95%-CI (lower)	95%-CI (upper)
KNN	Baseline	-38.673100	0.000668	-0.255400	-0.204300
KNN	Logistic Regression	26.090900	0.001466	0.064900	0.090500
Baseline	Logistic Regression	38.194800	0.000685	0.272900	0.342200

Table 8: Setup II statistical comparison evaluation.

3.4 Training the logistic regression model

As we can conclude, that the logistic regression performed the best out of our models, we can continue analysing the model by computing the weights of each of the abalones' features. By using the optimal values of λ^*, we get the weights shown in Table 9.

Class (rings)	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight
Young (0-8)	0.352100	-0.456000	-0.468000	**-1.625200**	**1.901600**	-0.026500	**-1.316800**
Mature (9-10)	0.238000	0.157100	0.085500	**-0.607700**	**0.218600**	0.233100	**0.192100**
Old (11-29)	-0.590100	0.299000	0.382500	**2.233000**	**-2.120200**	-0.206600	**1.124700**

Table 9: Average optimal weights for each feature, computed through training the logistic regression classifier using two-level cross-validation at $K1 = K2 = 10$.

It is shown, that the features with the highest weights, are the *weight*-based features. Interestingly, I seems that there happens a switch in the growth of the abalones as they age. From the weights, it appears that the shell weight grows as the abalones age, while their shucked weight decreases. Also, the total weight grows over time.

4 Discussion

In section 1 and section 2 we investigated the possibility of predicting an abalone *Whole Weight* using measurements of the specimen dimensions, specifically with by comparing linear regression, artificial neural networks and baseline. Using cross-validation we firstly determined that the optimal values for the regularization factor and number of hidden nodes (complexity-controlling parameters) are respectively 39.1 and 11. Both these approaches are capable of remarkable performance achieving errors inferior to 0.01, this confirms is in agreement with the observation that there is a strong correlation between the dimension and weights of the specimens.
To find what model was ultimately the best, we compared their the estimated generalization errors - the results allow us to say that both non-trivial models are definitely better that the baseline and, more of interest, that regularized linear regression is superior to ANNs in the prediciton task.

Despite being well-established in the machine learning field, The abalones dataset has been predominantly utilized as benchmark for classification task consisting in age prediction on the basis of the physical measurements, which we investigated in the classification section. The interest in the relation between the other attributes is scarse, with the main source being the original 1994 tecnhical report [1] from which for which the data was originally collected. Here the authors studied the correlation of the different morphometrics taken in couples. Between these, the most similar to our analysis is the one showing the presence of a strong power-law relation between the *diameter* and the *Whole Weight*.

In section 3 we trained 3 different classifiers; a baseline, K-nearest neighbours (KNN) and Logistic regression. We found that the complexity controlling parameters for KNN and logistic regression were $k = 34.92$ and $\lambda^* = 0.0010$ respectively. Using these, we were able to conduct a statistical analysis using correlated t-test for cross-validation. This showed, that the best performing classifier was the logistic regression followed by KNN, both outperforming the baseline. Through the optimal parameters, we were able to calculate the weights of the abalones' features. These findings showed, that as the abalones grow older, so does their shell weight and total weight. However, their shucked weight decreases.

In this University of British Columbia project[2], the group applied logistic regression like the one in this report. By looking at their feature weight magnitudes, they were able to conclude the same findings as we did:

> *"Based on the coefficients, shuckedweight influences the model the most towards predicting that an abalone is young, whereas whole_weight influences the model the most towards predicting that an abalone is old."*

Whereas this report and the research paper mentioned in the project[3] used a 3-category classification problem, the project joined the two younger groups into one, making theirs a 2-category classification problem. This however, would not change the results deducted in this report.

References

[1] Warwick J. Nash et al. *Abalone*. UCI Machine Learning Repository. 1994. DOI: https://doi.org/10.24432/C55C7W.

[2] UBC Master of Data Science (MDS). *Abalone Age Classification*. Accessed: 2024-11-14. 2023. URL: https://ubc-mds.github.io/abalone_age_classification/README.html.

[3] David Clark, Zoltan Schreter, and Anthony Adams. "A Quantitative Comparison of Dystal and Backpropagation". In: *Proceedings of the Australian Conference on Neural Networks*. Australia, 1996.

5 Exam problem solutions

5.1 Question 1: OPTION A

From the ROC curve in Figure 1 we can see that the first 25% of the test set, i.e. $N = 8 \rightarrow 8 \cdot 0.25 = 2$ observations are true positives, i.e. categorized as red crosses. This is only the case for option A, which is thus the correct answer.

5.2 Question 2: OPTION C

We can use the *purity gain* and *classification error impurity* as defined in Formulae (9.1) and (9.4) respectively in the textbook:

$$\Delta = I(r) - \sum_{k=1}^{K} \frac{N(v_k)}{N(r)} I(v_k) \quad , \quad I(r) \Rightarrow \text{ClassError}(v) = 1 - \max_c p(c|v)$$

For the split $x_7 = 2$, we have 1 true class and 134 false classes out of 135 total. This gives us the following impurity gain:

$$\Delta = I(r) = 1 - \max\left(\frac{1}{135}, \frac{134}{135}\right) \approx \underline{\underline{0.0074}}$$

5.3 Question 3: OPTION A

The ANN takes in 7 attributes as input, i.e. $n_{in} = 7$. It has a hidden layer of $n_h = 10$ units. Finally, the model returns a congestion level from 1 to 4, i.e. $n_{out} = 4$. Using these layers, we are now able to calculate the weights and biases.

$$n_{weights} = n_{in} \cdot n_h + n_h \cdot n_{out} = 70 + 40 = 110 \tag{1}$$

The Sigmoid activation function implies a bias for each node in the hidden and output layer, thus the number of biases is:

$$n_{biases} = n_h + n_{out} = 10 + 4 = 14 \tag{2}$$

The total number of parameters thus become:

$$n_{total} = n_{weights} + n_{biases} = 110 + 14 = \underline{\underline{124 \text{ parameters}}} \tag{3}$$

5.4 Question 4: OPTION D

By comparing the decision tree with the PCA classification, we can deduct that option D is the only rule assignment, where congestion level 4 is reachable.

- **Node A**: If $b_1 \geq -0.76$ then Con. level $\in [1, 3, 4]$, else Con. level $\in [1, 2]$
- **Node B**: If $b_2 \geq 0.03$ then Con. level $= 2$, else Con. level $= 1$
- **Node C**: If $b_1 \geq -0.16$ then Con. level $= 4$, else Con. level $\in [1, 3]$
- **Node D**: If $b_2 \geq 0.01$ then Con. level $= 1$, else Con. level $= 3$

5.5 Question 5: OPTION C

To calculate the total time taken to compose the table, we can define the values needed for the Two-layer Cross-validation:

- Inner folds: $K_2 = 4$
- Outer folds: $K_1 = 5$
- Models to validate: $S = 5$
- Training time (ANN): $T_{ann,train} = 20$
- Testing time (ANN): $T_{ann,test} = 5$
- Training time (Log.reg.): $T_{lr,train} = 8$
- Testing time (Log.reg.): $T_{lr,test} = 1$

Firstly, we calculate the number of models trained in the cross-validation:

$$\mathcal{M}_{total} = K_1 \cdot (K_2 \cdot S + 1) = 5 \cdot (4 \cdot 5 + 1) = 105 \tag{4}$$

Secondly, we calculate the total time per model trained:

$$T_{\mathcal{M}_*} = T_{ann,train} + T_{ann,test} + T_{lr,train} + T_{lr,test} = 20 + 5 + 8 + 1 = 35 \text{ ms} \tag{5}$$

The total time taken for composing the table is thus the product of all models trained and their training and testing times:

$$T_{total} = \mathcal{M}_{total} \cdot T_{\mathcal{M}_*} = 105 * 35 = \underline{\underline{3570 \text{ ms}}} \tag{6}$$

5.6 Question 6: OPTION B

The observation assigned to class $y = 4$ must be the one with the highest probability. Firstly, we calculate the per-class probabilities for $k = 1, 2, 3$:

$$\hat{y}_1 = \begin{bmatrix} 1 \\ -0.6 \\ -1.6 \end{bmatrix}^T \begin{bmatrix} 1.2 \\ -2.1 \\ 3.2 \end{bmatrix} = -2.66, \quad \hat{y}_2 = \begin{bmatrix} 1 \\ -0.6 \\ -1.6 \end{bmatrix}^T \begin{bmatrix} 1.2 \\ -1.7 \\ 2.9 \end{bmatrix} = -2.42, \quad \hat{y}_3 = \begin{bmatrix} 1 \\ -0.6 \\ -1.6 \end{bmatrix}^T \begin{bmatrix} 1.3 \\ -1.1 \\ 2.2 \end{bmatrix} = -1.56$$

Secondly, we calculate the exponentials for $k = 1, 2, 3$, followed by the sum of these:

$$e^{\hat{y}_1} = 0.07, \quad e^{\hat{y}_2} = 0.09, \quad e^{\hat{y}_3} = 0.21, \quad \sum_{k=1}^{3} e^{\hat{y}_k} = 0.37$$

Finally, we can use the sum of exponentials and calculate the per-class probability of $y = 4$:

$$P(y = 4 | \begin{bmatrix} 1 & -0.6 & -1.6 \end{bmatrix}) = \frac{1}{1 + 0.37} \approx \underline{\underline{0.73}}$$

Following the same procedure for Option A, C and D yields much smaller probabilities:

$$P(y = 4 | \begin{bmatrix} 1 & -1.4 & 2.6 \end{bmatrix}) \approx 3.0 \times 10^{-6}, \quad P(y = 4 | \begin{bmatrix} 1 & 2.1 & 5.0 \end{bmatrix}) \approx 1.8 \times 10^{-6}, \quad P(y = 4 | \begin{bmatrix} 1 & 0.7 & 3.8 \end{bmatrix}) \approx 4.7 \times 10^{-6}$$