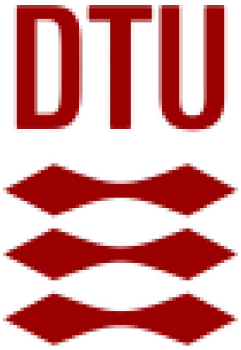

02450 - Introduction to Machine Learning and Data Mining

Project 2 Report

Group 9

Name	**Student Number**
FOO YAN RONG	S231651
KAI JIE, JARED LIANG	S231648

Contributions to Report

Report Section	**Contributions (Yan Rong - Jared)**
Regression, part a	60% - 40%
Regression, part b	40% - 60%
Classification	60% - 40%
Discussion	40% - 60%
Exam Questions	60% - 40%

1. Regression Part A

1. Description and Overview

The target variable selected for regression is Cholesterol which measures the level of cholesterol in a body in mm/dL. The regressors used are Age, Sex, RestingBP, FastingBS, ChestPainType, RestingECG, ExerciseAngina, MaxHR, Oldpeak, ST_Slope and HeartDisease. By conducting the regression, it may be possible to develop a predictive model that estimates a person's cholesterol level based on the regressors. This can be useful for assessing individual health and risk factors for other kinds of potential health problems such as diabetes, atherosclerosis or even stroke.

The feature transformation choice used is one-hot encoding for categorical variables. After which, z-standardization is applied to all variables where the mean and standard deviation across

1.1.1 Feature Transformation - One-hot Encoding for Categorical Variables

One-hot encoding is performed to convert categorical data into a numerical format that can be fed into machine learning models. To do this, n-1 dummy variables were created for each categorical variable with n representing the number of unique categories in each variable. A value of 0 is given for the absence of that category and a value of 1 is given for the presence of that category. For example, the sex variable has 2 distinct values: M or F. A dummy variable is created, Sex_M, where a value of 1 indicates that the patient is a male and a value of 0 indicates that the patient is a female. The same steps are performed for all the categorical variables, namely: Sex, ChestPainType, RestingECG, ExerciseAngina and ST_Slope. Note that the FastingBS variable is already expressed as a binary variable. Hence, the categorical variables are replaced by the following dummy variables.

Sex_M	ChestPainType_TA	ChestPainType_ATA	ChestPainType_NAP	RestingECG_N	RestingECG_ST	ExerciseAngina_Y	ST_Slope_U	ST_Slope_F
1	0	1	0	1	0	0	1	0
0	0	0	1	1	0	0	0	1
1	0	1	0	0	1	0	1	0
0	0	0	0	1	0	1	0	1
1	0	0	1	1	0	0	1	0
...	...	...	...	...	...	...	...	...
1	1	0	0	1	0	0	0	1
1	0	0	0	1	0	0	0	1
1	0	0	0	1	0	1	0	1
0	0	1	0	0	0	0	0	1
1	0	0	1	1	0	0	1	0

Figure 1: Dummy variables replacing categorical variables

1.1.2 Standardization of Variables

Depending on the model applied, feature scaling may be important to ensure that features with a large range and higher magnitude do not necessarily have a larger influence over the model due to their larger raw variance value. This is especially important for models such as logistic regression. Hence, it is important for features to be on a similar scale. However, it is clear that the features are on different scales. For example, the "Age" variable has values ranging from about 30 to 80 while the "Cholesterol" variable has values ranging from about 100 to 600 (excluding 0 values). Hence, the values are standardised with mean = 0 and standard deviation = 1 using the z-index standardisation formula:

$$Z = \frac{x - \mu}{\sigma}$$

where z is the Z-index, x is the data point, μ is the variable mean and σ is the variable standard deviation.

Note that for both regression and classification, the target variable does not need and will not be standardized.

1.1.3 Regularization

Introducing a regularization parameter λ helps prevent overfitting and fine-tunes the model to achieve a better balance between fitting the training data and generalizing to unseen data. The loss function E with the regularization parameter can be expressed as:

$$E_{\lambda} = ||\hat{y} - \hat{X}w||^2 + \lambda||w||^2$$

The following λ values were considered: 0.1,1, 10, 100, 1000, 10000, 100000. By using K = 10 fold cross-validation to estimate the generalization error, the following graphs were produced.

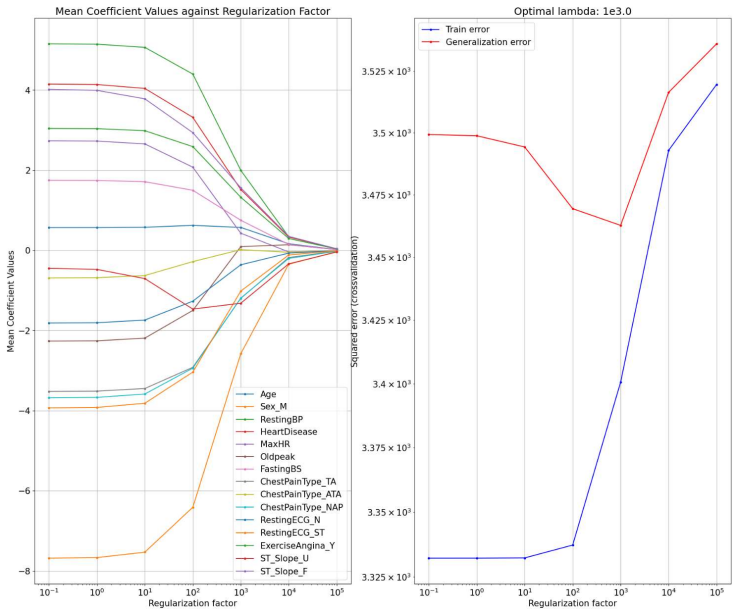

Figure 2: Graph of mean coefficient values against regularization factor (left)
Graph of training and generalization error against regularization factor (right)

It is observed that as λ increases from 0.1 to 1000, the generalization error decreases. When λ increases above 1000, the generalization error increases. When λ is relatively small at 0.1, the regularization term has minimal influence on the model's cost function. The model is essentially allowed to fit the training data with little constraint. As λ increases, it imposes stronger regularization by penalizing the magnitude of the model's coefficients. This encourages the model to generalize better, preventing it from overfitting the training data.
A moderate level of regularization at λ = 1000, there is a balance between fitting the training data and generalizing to new data. This results in a reduction in generalization error because it prevents the model from overfitting, making it more robust to unseen data. However, as λ continues to increase beyond a certain point at λ > 1000, the regularization term becomes dominant in the equation. This causes the coefficients to converge closer and closer to 0 as seen in Figure 1.1 (Left). The model is now penalized to an extent where it cannot fit the training data effectively. Hence, very strong regularization causes the model to underfit the training data. It excessively shrinks the coefficients and makes the model too simple, leading to high bias.
This high bias results in a decrease in the model's ability to capture the underlying patterns in the data, ultimately leading to an increase in the generalization error.

Hence, λ = 1000 is chosen for the model.

1.1.4 Linear Regression Results

The linear equation for the model is as shown:

$$\begin{aligned} Cholesterol = {} & w0 + w1 * Age + w2 * Sex\_M + w3 * RestingBP + w4 * HeartDisease + \\ & w5 * MaxHR + w6 * Oldpeak + w7 * FastingBS + w8 * ChestPainType\_TA + \\ & w9 * ChestPainType\_ATA + w10 * ChestPainType\_NAP + w11 * RestingECG\_N + \\ & w12 * RestingECG\_ST + w13 * ExerciseAngina\_Y + w14 * ST\_Slope\_U + \\ & w15 * ST\_Slope\_F \end{aligned}$$

The weights are determined by performing linear regression with the regularization parameter $\lambda = 1000$ that produced the lowest generalization error. The following weights/coefficients and intercept are determined:

	Feature Name	Coefficient Term	Coefficient Value
0	Age	w1	2.419827
1	Sex_M	w2	-7.318833
2	RestingBP	w3	3.906442
3	HeartDisease	w4	2.338611
4	MaxHR	w5	3.083832
5	Oldpeak	w6	0.184333
6	FastingBS	w7	1.041506
7	ChestPainType_TA	w8	-3.501805
8	ChestPainType_ATA	w9	0.398438
9	ChestPainType_NAP	w10	-3.749943
10	RestingECG_N	w11	-3.905399
11	RestingECG_ST	w12	-3.24493
12	ExerciseAngina_Y	w13	2.574776
13	ST_Slope_U	w14	1.702447
14	ST_Slope_F	w15	4.436751
15	Intercept (Bias)	w0	244.7988

Figure 3: Table of feature names with their coefficient term and value

For a given input, the features would have to undergo the same pre-processing steps before being input into the linear model. I.e: one-hot encoding is performed on the categorical variables before z-standarization is carried out for all the regressors. After the features have been transformed, these values will be inserted into the linear equation and multiplied by their respective weights before summing up to give a final value for the target variable, Cholesterol.

The effect of an individual attribute in x in the output y of the linear model can be represented by its weight, also known as the coefficient, determined in the linear model. It indicates the strength and direction of the relationship between that regressor and the target variable.

Magnitude
The magnitude shows how much the target variable is expected to change for a one-unit change in the regressor while holding all other variables constant. It represents the strength of the relationship. A large magnitude of the weight corresponds to a larger influence in the model in determining the value of the target variable.

Sign
The sign represents the direction of the relationship. A positive weight indicates a positive relationship between the regressor and the target variable. When the regressor increases, the target variable is expected to increase as well. A negative weight suggests a negative relationship. An increase in the regressor is associated with a decrease in the target variable.

Observation
By observing the weights of the regressors, the effect of most individual attributes makes sense. For example, the coefficient value for Age is positive indicating that older patients are likely to have higher cholesterol levels, ceteris paribus. Sex_M has a large negative coefficient value indicating that males are likely to have much lower cholesterol levels compared to females, ceteris paribus. This observation has been backed up by numerous studies. HeartDisease has a large positive coefficient value indicating that patients with heart disease are likely to have higher cholesterol levels than those who do not, ceteris paribus.

1.2.1 Model Comparisons for Linear Regression

1.2.1.1 Two-level Cross-validation

Two-level cross-validation is a robust evaluation technique that involves two nested loops of cross-validation. The purpose is to provide a more reliable estimate of a model's performance, especially in situations where hyperparameter tuning is involved.

Inner Loop

In the inner loop, the dataset is split into training and validation sets. Multiple models with different hyperparameter settings are trained on the training set. The models are evaluated on the validation set, and their performance metrics are recorded. The hyperparameters of the model that performs the best on the validation set are selected.

Outer Loop

In the outer loop, the dataset is split into training and test sets. The model selected in the inner loop is trained on the training set. The trained model is then evaluated on the test set, and its performance metric is recorded. This process is repeated for each fold in the outer cross-validation.

Performing two-level cross validation can help prevent overfitting of the hyperparameters to a specific dataset during model selection, giving unbiased performance estimation.

1.2.1.2 Models Selected

a) Artificial Neural Network

For the artificial neural network (ANN) model, a 3-layer model was created with 1 input, 1 hidden and 1 output layer. The input layer has 15 units since there are 15 input attributes. The hidden layer has h units with $h = \{1, 10, 20, 30, 50\}$ which was decided based on a few test-runs. The non-linear activation function used in the hidden layer is the rectified linear unit (ReLU). The output layer has one unit which returns the predicted output of the model. The optimizer used for the model is the stochastic gradient descent (SGD) with learning rate = 0.001. The cost function used is the mean-squared error since the output is a continuous variable.

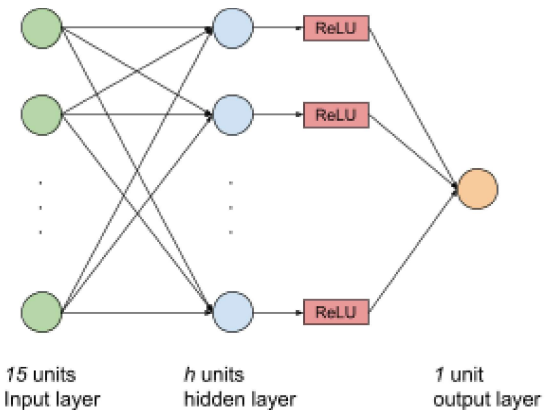

Figure 4: Layout of the Neural Network Feedforward Process

b) Linear Regression

For the linear regression model, the ridge regression model was created with regularization parameter $\lambda = \{100, 250, 500, 750, 1000\}$ which was based on a few test runs.

c) Baseline Model

A baseline model was created that returned the mean value of the inner loop training set that gave the lowest validation error. This mean value is then used to test against the outer loop testing set.

1.2.2 Performance of Models

The error measure/performance metric E to help determine the performance of the model is the squared loss per observation.

$$E = \frac{1}{N^{test}} \sum_{i=1}^{N^{test}} (y_i - \hat{y}_i)^2$$

Outer fold	ANN		Linear Regression		Baseline
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	30	3018.22	500	1987.83	2103.43
2	50	4032.98	500	3964.60	4239.73
3	30	2898.44	1000	2016.78	2782.98
4	20	2681.59	1000	2410.85	2844.97
5	50	3922.63	100	3622.79	3783.75
6	50	4123.35	250	3363.53	3444.96
7	50	4278.73	1000	3847.02	3942.83
8	30	3474.57	750	3040.89	3022.23
9	20	5020.43	500	4114.41	4372.88
10	50	3592.08	100	2924.26	2982.07

Figure 5: Results of two-level cross validation using ANN, linear regression and baseline models

From the table of results, there seems to be no one model that is performing significantly better than other models. The error measure also seems to fluctuate by a significant amount, likely indicating that the models are sensitive to the specific subsets of data in each fold. This could be due to the dataset being rather small, hence there may be some level of overfitting.

For the hyperparameters, it seems that there is no one best-fit-all value for h and λ as each outer fold can have a different optimal h/λ value depending on what subset of the dataset the model was trained on. As seen from the table, λ = 1000 may not be the most optimized hyperparameter value as it may highly depend on the training data. λ * = 1000 was only selected at the start to help gauge the range and magnitude of λ * to use for the actual model comparison and testing. By narrowing down the range of λ to be around 1000 such as values [100, 250, 500, 750, 1000, 1250], it is then observed that λ * can take on different values.

1.2.3 Statistical Evaluation using Setup I (Paired T-Test)

For statistical evaluation, paired t-test was performed to evaluate if there is a significant difference in performance between the fitted ANN, linear regression model and baseline model. The t-test formula is as shown:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}$$

where $\bar{d}$ is the mean difference between paired mean-squared-errors of the models, s_d is the standard deviation of the differences and n is the number of mean-squared-errors for each model

Model 1	**Model 2**	**T-statistics**	**95% Confidence Interval**	**p-value**
Baseline	Linear Regression	0.657	[-489.71, 935.08]	0.520
Baseline	Neural Network	-1.091	[-1030.74, 326.10]	0.290
Neural Network	Linear Regression	1.707	[-132.73, 1282.74]	0.105

Figure 6: Table of results from the paired t-tests between models

Comparing the baseline model and linear regression model, the t-statistics is positive, indicating that the baseline has on average higher mean-squared errors than the linear regression, indicating that the linear regression model performs better. However, since p-value = 0.52 > 0.05, this difference in mean-squared errors between the 2 models is statistically insignificant. Hence, it can be concluded that these 2 models have about the same performance.

Comparing the baseline model and ANN model, the t-statistics is negative, indicating that the baseline has on average lower mean-squared errors than the ANN model, indicating that the baseline model performs better. However, since p-value = 0.290 > 0.05, this difference in mean-squared errors between the 2 models is statistically insignificant. Hence, it can be concluded that these 2 models have about the same performance.

Comparing the ANN model and linear regression model, the t-statistics is positive, indicating that the ANN model has on average higher mean-squared errors than linear regression, indicating that linear regression model performs better. However, since p-value = 0.105 > 0.05, this difference in mean-squared errors between the 2 models is statistically insignificant. Hence, it can be concluded that these 2 models have about the same performance.

From the 3 paired t-test results, all models seem to have roughly the same performance level based on p-value. This can be seen from the fact that 0 lies in all of the 95% confidence intervals. The greatest difference in performance is between ANN and the linear regression model which has the highest magnitude of the t-statistics and lowest p-value. For such a dataset, it is recommended to use linear regression as it performs the best but not significantly better than other models such as ANN and the baseline model.

2. Classification

2.1 Overview and Description

The classification problem to solve is to determine if a patient will have heart disease given information on their age, sex, cholesterol levels, chest pain type, resting blood pressure, whether they have fasting blood sugar, their resting ECG results, max heart rate, their exercise angina, old peak and ST slope. This is a binary classification problem where a patient is predicted to either have heart disease or lack thereof.

2.2 Model Selection

a) Artificial Neural Network

Similar to linear regression, a 3-layer ANN model was created with 1 input, 1 hidden and 1 output layer. The input layer has 15 units since there are 15 input attributes. The hidden layer has h units with h = {1, 10, 20, 30, 50} which was decided based on a few test-runs. The non-linear activation function used in the hidden layer is the rectified linear unit (ReLU). The output layer is one unit which returns the predicted output of the model. Since the target variable is binary with classes = {0,1}, the sigmoid function was used as the non-linear activation function for the output layer. The optimizer used for the model is the stochastic gradient descent with learning rate = 0.001. The cost function used is the binary cross-entropy since the output is a binary variable.

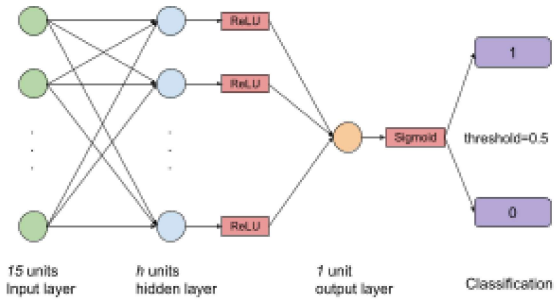

Figure 7: Layout of the Neural Network Feedforward Process

b) Logistic Regression

For logistic regression, the logistic regression model was created with regularization parameter $\lambda = \{0.01, 0.05, 0.1, 0.5\}$ which was based on a few test runs. The log-odds (logit) form of the logistic regression equation is:

$$ln(\frac{P(HeartDisease=1)}{1-P(HeartDisease=1)}) = w0 + w1 * Age + w2 * Sex\_M + w3 * RestingBP + w4 * Cholesterol + w5 * MaxHR + w6 * Oldpeak + w7 * FastingBS + w8 * ChestPainType\_TA + w9 * ChestPainType\_ATA + w10 * ChestPainType\_NAP + w11 * RestingECG\_N + w12 * RestingECG\_ST + w13 * ExerciseAngina\_Y + w14 * ST\_Slope\_U + w15 * ST\_Slope\_F$$

c) Baseline Model

A baseline model was created that returned the majority class label of the inner loop training set that gave the lowest validation error. This class label is then used to test against the outer loop testing set.

2.3 Performance of Models

Similar to regression, the two-level cross validation is performed to evaluate the performance of the models. The error measure E to help determine the performance of the model is the misclassification rate.

$$E = \frac{\{Number\ of\ misclassified\ observations\}}{N^{test}} = 1 - accuracy\ rate$$

Outer fold	ANN		Logistic Regression		Baseline
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	50	0.720	0.01	0.146	0.360
2	30	0.507	0.01	0.107	0.453
3	50	0.467	0.1	0.107	0.520
4	50	0.546	0.1	0.187	0.520
5	50	0.427	0.5	0.107	0.507
6	50	0.480	0.1	0.067	0.480
7	50	0.459	0.01	0.162	0.500
8	50	0.568	0.1	0.162	0.473
9	50	0.419	0.01	0.149	0.486
10	50	0.622	0.5	0.162	0.473

Figure 8: Results of two-level cross validation using ANN, logistic regression and baseline models

From the table of results, the logistic regression model seems to be the best performing model that consistently gives the lowest error value across all the outer folds.

For the hyperparameters, it seems that h = 50 is the most optimized value for the ANN model after hypertuning. Even so, the performance of the ANN model is not that great with an average misclassification rate of 50% across the outer folds, which performs quite similarly to the baseline model. For λ, the optimal value varies across different values in {0.01, 0.1, 0.5} for different outer folds. Even with different λ* values, the logistic regression model continues to perform among the models across the different folds.

2.4 Statistical Evaluation using Setup I (McNemera's Test)

Outer fold	ANN vs Logistic Regression	ANN vs Baseline	Logistic Regression vs Baseline

i	Theta (ANN - Log Reg	CI	p-value	Theta (ANN - Baseline)	CI	p-value	Theta (Log Reg - Baseline)	CI	p-value
1	-0.333	(-0.459, -0.201)	1.09e-05	0	(-0.197, 0.197)	1.10	0.333	(0.191, 0.469)	4.13e-05
2	-0.200	(-0.325, -0.0713)	0.00592	0.0933	(-0.0767 , 0.261)	0.360	0.293	(0.144, 0.436)	0.000472
3	-0.173	(-0.285, -0.0599)	0.00720	-0.0267	(-0.209, 0.157)	0.888	0.147	(0.00463, 0.286)	0.0708
4	-0.44	(-0.559, -0.313)	1.02e-08	-0.0133	(-0.194, 0.168)	1.00	0.427	(0.295, 0.550)	6.68e-08
5	-0.373	(-0.510, -0.228)	8.36e-06	0.0133	(-0.168, 0.194)	1.0	0.387	(0.257, 0.509)	4.18e-07
6	-0.493	(-0.630, -0.343)	5.73e-08	-0.213	(-0.363, -0.0583)	0.0139	0.280	(0.132, 0.421)	0.000753
7	-0.446	(-0.593, -0.286)	1.96e-06	-0.135	(-0.323, 0.0578)	0.220	0.311	(0.164, 0.451)	0.000191
8	-0.324	(-0.445, -0.198)	8.43e-06	0.0946	(-0.0054 4, 0.194)	0.118	0.419	(0.270, 0.558)	1.64e-06
9	-0.257	(-0.373, -0.137)	0.000157	0.122	(-0.0734 , 0.312)	0.281	0.378	(0.248, 0.502)	7.66e-07
10	-0.581	(-0.711, -0.434)	7.08e-10	-0.135	(-0.279, 0.0118)	0.110	0.446	(0.318, 0.566)	1.02e-08

Figure 9: Table of results from the McNemera's test between models (Rounded to 3s.f.)

Comparing the ANN model and the Logistic Regression Model, we obtain values of theta which are all negative and significantly different from zero across all 10 folds. This implies that the Logistic Regression model is more favourable than the ANN model since in this case theta is defined as the difference in performance of the ANN model compared to the Logistic Regression model. Likewise across all folds, the confidence intervals for theta do not contain zero and the p-values are all extremely small and less than 0.05 (when we have set alpha = 0.05). Thus, there is a statistically significant difference between the two models' performances. This supports our model performance evaluation earlier, where we can conclude that the Logistic Regression Model performs better than the ANN model.

Comparing the ANN model and the baseline model, we obtain values of theta which are not significantly different from zero, and vary in polarity, where in this case theta is defined as the difference in performance of the ANN model compared to the baseline model. In all except one of the folds, the confidence intervals contain zero and the p-values are larger than 0.05. This implies that the difference in the performance of both models is not statistically significant. This supports our model performance evaluation earlier, and we conclude that these two models have similar performance.

Comparing the Logistic Regression model and the baseline model, we obtain values of theta which are all positive and different from zero, which implies the Logistic Regression model performs more favourably than the baseline model, since in this case theta is defined as the difference in performance of the Logistic Regression model compared to the baseline model. The confidence intervals for theta all do not contain zero, and in all except one of the folds, the p-values are all extremely small and less than 0.05. Thus, there is a statistically significant difference between the two models' performances. This supports our model performance evaluation earlier, where we can conclude that the Logistic Regression Model performs better than the baseline model.

As such from the three pairwise comparisons above, we can deduce that the Logistic Regression model performs significantly better as compared to both the ANN model and baseline model, which have similar performance. Thus for our classification model, we would recommend using the Logistic Regression model.

2.5 Logistic Regression Interpretation

Given the logistic regression equation in log-odds form:

$$ln(\frac{P(HeartDisease = 1)}{1 - P(HeartDisease = 1)}) = w0 + w1 * Age + w2 * Sex\_M + w3 * RestingBP + w4 * Cholesterol + w5 * MaxHR + w6 * Oldpeak + w7 * FastingBS + w8 * ChestPainType\_TA + w9 * ChestPainType\_ATA + w10 * ChestPainType\_NAP + w11 * RestingECG\_N + w12 * RestingECG\_ST + w13 * ExerciseAngina\_Y + w14 * ST\_Slope\_U + w15 * ST\_Slope\_F$$

The weights are determined by performing linear regression with regularization parameter that produced the lowest generalization error. The following weights/coefficients and intercept are determined:

Feature Name	Coefficient Term	Coefficient Value
Age	w1	0.133
Sex_M	w2	0.308
RestingBP	w3	0.0925
Cholesterol	w4	0.0670
MaxHR	w5	-0.151
Oldpeak	w6	0.290
FastingBS	w7	0.0857
ChestPainType_TA	w8	-0.0835
ChestPainType_ATA	w9	-0.279
ChestPainType_NAP	w10	-0.236
RestingECG_N	w11	-0.0636
RestingECG_ST	w12	0.00303
ExerciseAngina_Y	w13	0.312
ST_Slope_U	w14	-0.404
ST_Slope_F	w15	0.364
Intercept (Bias)	w0	-0.164

Figure 10: Table of feature names with their coefficient term and value

For a given input, the features would have to undergo the same pre-processing steps before being input into the model. I.e: one-hot encoding is performed on the categorical variables before z-standarization is carried out for all the regressors. After the features have been transformed, these values will be inserted into the logistic equation and multiplied by their respective weights before summing up to give a final value for the log-odds of HeartDisease. Taking the exponential and performing basic manipulation would give the probability of a patient having heart disease given their corresponding input features.

Magnitude

The magnitude shows how much log-odds of the target variable is expected to change for a one-unit change in the regressor while holding all other variables constant. It represents the strength of the relationship. A large magnitude of the weight corresponds to a larger influence in the model in determining the value of the target variable.

Sign
The sign represents the direction of the relationship. A positive weight indicates a positive relationship between the regressor and the target variable. When the regressor increases, the target variable is expected to increase as well. A negative weight suggests a negative relationship. An increase in the regressor is associated with a decrease in the target variable.

Observation
A unit increase in age is associated with an increase of 0.133 log-odds of a patient having heart disease, ceteris paribus. The odds of a male patient having a heart disease is $e^{0.308} = 1.36$ times as likely compared to a female patient, ceteris paribus. Some coefficient values are unexpected such as ChestPainType_TA/ATA/NAP which all showed a negative value, indicating that patients with chest pain have lower odds of having a heart disease compared to patients with no chest pain, ceteris paribus. This is rather counter-intuitive.

3. Discussion

3.1 Findings from Regression and Classification

In both regression and classification, the artificial neural network (ANN) does not perform as well compared to the linear regression and logistic regression respectively. The poor performance of the ANN could be due to the lack of sufficient data as there are only 746 entries after cleaning, which may have caused the neural network to overfit and perform poorly on new and unseen data.

The logistic regression model performed significantly better compared to other models in the classification problem. This may be partly due to the dataset size. Additionally, the relationships between regressors and the target variable may be linear or simple which can be easily captured by the logistic regression model. On the other hand, ANNs are more capable of capturing non-linear and complex relationships which might not have been necessary given the classification problem.

The linear regression model did not perform significantly better compared to the baseline model. This may be because the dataset is used primarily as a binary classification problem for heart disease. By choosing cholesterol as the target variable, there may be presence of confounding variables that is not captured by the model, leading to poorer performance.

There is a lack of interpretability using ANN models for both classification and regression. The ANN can have a large number of parameters due to the number of units in the hidden layer. Furthermore, the non-linear activation functions introduce complexities that are harder to interpret. Thus, interpreting the impact of each parameter on the output becomes increasingly difficult as the model complexity grows. On the other hand, linear regression and logistic regression are inherently linear models. This linearity makes it straightforward to interpret the impact of changes in predictor variables on the response. The coefficients in linear regression directly represent the change in the mean response for a one-unit change in the predictor, allowing for the comparison of effect sizes. In logistic regression, exponentiating the coefficients gives the odds ratio, which represents the multiplicative change in odds for a one-unit change in the predictor.

3.2 Comparison with other analysis

When sourcing for other studies and notebooks which also analyse this data, we are only able to gain access to publicly available notebooks which were mainly focused on classification, correlation analysis, exploratory data analysis, as compared to just regression. Here we will focus our discussion on one of the many analysis notebooks using the same Heart Disease data set found on Kaggle (Link: https://www.kaggle.com/code/ahmadbarkat/cardiovascular-disease)

This study did their classification analysis using several models, which include Logistic Regression, AdaBoost, XGBClassifier, K Nearest Neighbours, Random Forest Classifier. Among these models, Logistic Regression displayed an average performance in terms of test set accuracy, while other models like XGB Classifier performed much better with over 90% test set accuracy. In our current analysis, Logistic Regression is our best performing model, so perhaps this gives us room for further exploration with other classifiers. Nonetheless, we do feel that our classification analysis provides deeper depth and insight as given our implementation of the two-level cross validation (absent in the other report), regularisation, and statistical evaluation efforts.