Introduction to Machine Learning and Data Mining

Panagiotis Papadamos - s205637
Dimitrios Loukas - s220514
Ziyu Zhang - s212657

Introduction

The main goal of this report is to demonstrate the application of Classification and Regression on our dataset with title *IoT Based Risk Level Prediction Model For Maternal Health Care In The Context Of Bangladesh*. In this dataset the main interest lies within the attribute of risk factor, which describes the risk level on the basis of risk factors in pregnancy (*low risk, mid risk, high risk*). In this report we attempt to predict this class using both Classification and Regression methods, as well as statistically evaluate our result. We use nested cross validation in both methods and we compare the results of different models in order to find the most suitable one. Finally we compare Logistic Regression method in classification and Linear Regression method in regression in terms of feature selection. It is important to note that in Linear Regression part we chose to approximate the attribute risk level, instead of predicting a continuous attribute, by seeing it as a continuous variable.

Regression A

Feature Selection

Our goal is to predict the risk level of a new observation, by training a linear regression model and predict a new observation's y value. In this way we will approximate the class (the value of the class) that the observation belongs in. It is actually an approximation and the results are not optimal as the linear regression will predict a continuous real values contrary to the discrete values that the classes take. For our dataset we could have chosen to work with all the independent variables (attributes) as:

- The number of attributes is not relatively large to the number of observations
- All the independent variables could potentially have relations with a health issue, as they are all related to human physiology, and thus we could have some interesting results.

However, we did feature selection using 10-fold cross validation and the results can be found in Figure 1. Based on these results we trained our linear regression model using 4 out of 6 independent variables, namely **Systolic BP**, **Blood Glucose**, **Body Temperature** and **Heart Rate**. We should note that the difference between the generalization error when the model trained used these attributes and the generalization error when the model trained on another combination of attributes were not big.
When it comes to feature transformations, we did the standardization of the data, because we want to apply regularization. We didn't apply one-of-K-coding to the vector y, as the ordinal values of y represent the natural order that the classes have (low, mid, high). In case that ordered numbers wouldn't give a natural point to the classes, we would have applied one-of-K-coding.

Regularization Parameter

In order to select the optimal regularization parameter we used hold-out cross validation for different values of regularization strength, compute the test error for each regularization parameter and picked the value that gave the lowest test error. Figure 2 shows the error for different values of λ. One can see that the the lowest test error is given for λ=10 that is the value of select for λ. In order to train the model and calculate the train and test errors we used the **Ridge** method of the **sklearn.linear_model** package that is used to train reguralized linear regression models. We should note that because if our code we have the data split shuffling, each time that one

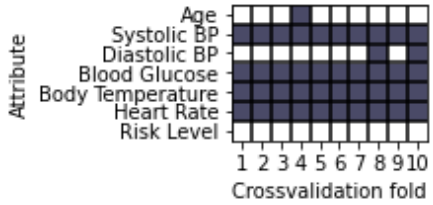


Figure 1: *Feature Selection*

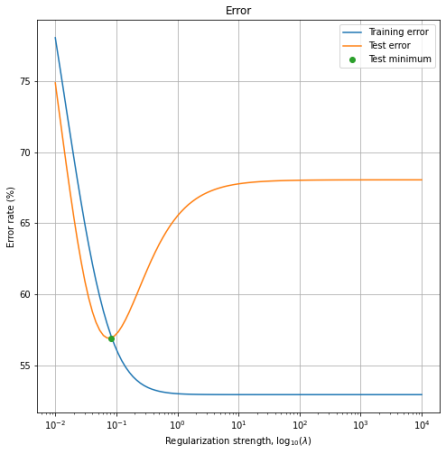


Figure 2: *Errors for different regularization strength*

runs the code the diagram will be different. However, we always select the value of λ that give the lowest test error, as after this value the models start overfitting the data and the test error increases. The selected value for λ gives an test error rate of around 57%.

Predict New Observations

In order to make a prediction using the model we trained using the selected parameter λ, we take a new observation x and we firstly apply the same transformation to the values, that is standardization using the mean and the standard deviation of the features in the training dataset. Then we take the the sum of the linear combination of the standardized observation $\bar{x}$ with the optimal parameters w^* of the trained model and the expected value of y [3]. However, this computation will give us a standardized result. In order to get a result close to the real value of the class we can do the reversed procedure, multiply with the standard deviation and add the mean of the column that represents the classes.

Regression B

Model Evaluation

In this section two-level cross-validation is implemented to compare three different models: the regularized linear regression model, an artificial neural network (ANN) and a baseline. And

statistically evaluate if there is a significant performance difference between the fitted three models using **Setup I** method. Here we use feature-selected attributes from Regression A part.

Initialization and parameter set up

In this project, using two-level cross-validation with $K_1 = K_2 = 10$ folds. For K_1-fold cross-validation iterations, split the data D into a training set and a test set. For each iteration, find the optimal model using K_2-fold cross-validation on training set. Using the selected model structure train on the full outer fold training set. Get test error for every outer fold iteration, finally, estimate the generalization error based on all test error.

Error calculation and hyperparameter tuning

In this project, we calculate the squared loss per observation, and divide squared loss by the number of observation in the test set, as the error measure.To control the learning process, we gave hyperparameters for linear regression model and ANN respectively.
For regression model, we altering the cost function to have a stronger preference small weights, which introducing a regularization factor λ. Here, we choose 10^{-5} to 10^{9} as the range of values of λ , where the generalization error first drop and then increases, shows in figure 3.

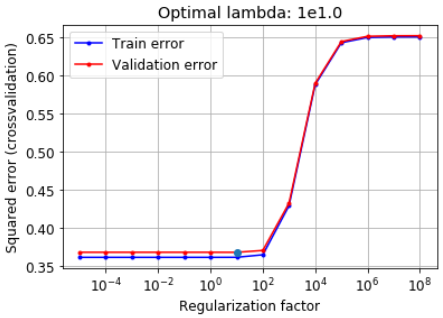
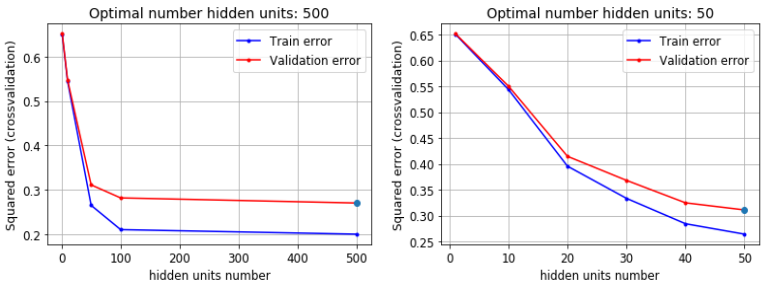


Figure 3: *Regularization factor for linear regression model*

We use ANN with an input layer, one layer of hidden units and an output layer. In hidden layer, more than one hidden units can be much more flexible, however, a reasonable interval should be selected. Firstly we set large interval for the number of hidden units, then select a reasonable small interval by weighting the pros and cons. Figure 4(a) shows there is no significant improvement in model performance when hidden units more than 50, therefore we choose 1 to 50 as the interval of number of hidden units when training.



(a) (b)

Figure 4: Hidden units number for ANN

As a baseline model, we apply a linear regression model with no features, i.e. it computes the mean of y on the training data, and use this value to predict y on the test data.

Results

Train models with all the preset parameters integrated, results are found in table 1. It shows when hidden units is 50 and regularization factor λ is 10(same value as in the previous section), ANN and linear regression models have lowest test error, 0.229433 and 0.313636 respectively. Comparing the overall test errors of the three models, ANN has relatively good performance, followed by the linear model, and the baseline model is the worst.

Table 1: Two-level cross-validation table used to compare the three regression models.

Outer fold	ANN		Linear Regression		Baseline
i	h_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	40	0.338806	10	0.344988	0.633272
2	50	0.260171	10	0.313636	0.658493
3	50	0.229433	10	0.334462	0.689461
4	50	0.353644	10	0.397938	0.675144
5	50	0.264631	1	0.393574	0.631842
6	50	0.323192	10	0.316993	0.663925
7	50	0.264124	10	0.352600	0.618517
8	50	0.319688	10	0.428477	0.661442
9	50	0.265198	10	0.425030	0.670451
10	50	0.258585	10	0.343591	0.599803

Statistical evaluation

In past sections, we train models on our dataset and obtain the fitted ANN, linear regression model and baseline, then estimate their generalization error. Here we want to statistically evaluate if there is a significant performance difference between them using the **Setup I** method.
To do so, we re-use two-level cross-validation to estimated generalization error for fitted models, which are ANN with h = 50, linear regression model with $\lambda = 10$ and baseline. Statistical evaluation is preformed pairwise. For each two models, calculate the difference between generalization errors based on same splits. Set $\alpha = 0.05$ and the 95% confidence interval for true difference

value is obtained by function 11.52. Assuming the null hypothesis is the two models have the same performance, through function 11.53 get p-value for this event, the p-value captures how unlikely H_0 is true. In other word, confidence interval shows the estimated difference value, and small p-value indicates two models have difference performance.
The estimated difference in generalization error is computed using the square loss, and we computed a confidence interval based on the student-t distribution with mean and variance. Based on this approximation, a 95% confidence interval is shown in table 2. In conclusion, both ANN and linear regression model are better than baseline, while ANN and linear regression model have identical performance. Consider the computing cost, we recommend linear model as optimal regression model for our question.

Table 2: Statistical evaluation for regression models

Fold	ANN vs Baseline		Linear regression vs Baseline		ANN vs Linear regression	
i	p-value	Confidence Interval	p-value	Confidence Interval	p-value	Confidence Interval
1	$3.1{\cdot}10^{-7}$	(0.22,0.47)	$5.1{\cdot}10^{-5}$	(0.12,0.34)	$1.4{\cdot}10^{-2}$	(0.02,0.20)
2	$4.2{\cdot}10^{-14}$	(0.27,0.44)	$9{\cdot}10^{-12}$	(0.20,0.33)	$2.7{\cdot}10^{-3}$	(0.03,0.15)
3	$3.7{\cdot}10^{-26}$	(0.33,0.47)	$1{\cdot}10^{-21}$	(0.24,0.34)	$2{\cdot}10^{-5}$	(0.05,0.13)
4	$1.1{\cdot}10^{-30}$	(0.31,0.43)	$1.9{\cdot}10^{-26}$	(0.24,0.34)	$7.7{\cdot}10^{-6}$	(0.05,0.12)
5	$2.9{\cdot}10^{-36}$	(0.31,0.41)	$4.1{\cdot}10^{-31}$	(0.24,0.32)	$2.6{\cdot}10^{-6}$	(0.04,0.11)
6	$4.6{\cdot}10^{-45}$	(0.32,0.41)	$1.1{\cdot}10^{-3}$	(0.24,0.32)	$9.5{\cdot}10^{-8}$	(0.05,0.12)
7	$1.8{\cdot}10^{-54}$	(0.32,0.41)	$1.4{\cdot}10^{-41}$	(0.24,0.32)	$9.5{\cdot}10^{-9}$	(0.05,0.11)
8	$2.9{\cdot}10^{-58}$	(0.32,0.40)	$4.4{\cdot}10^{-48}$	(0.25,0.32)	$7.6{\cdot}10^{-8}$	(0.04,0.10)
9	$1.7{\cdot}10^{-61}$	(0.32,0.40)	$3.4{\cdot}10^{-52}$	(0.25,0.31)	$2.6{\cdot}10^{-8}$	(0.05,0.10)
10	$7.9{\cdot}10^{-70}$	(0.32,0.40)	$1.7{\cdot}10^{-58}$	(0.25,0.31)	$6.0{\cdot}10^{-10}$	(0.05,0.10)

Classification

In this section we compared three different models, using two-level cross validation, namely *Logistic Regression*, *K-Nearest Neighbors* and *baseline* and then we solve a classification problem, using *Logistic Regression*. The baseline method is done using a model, which computes the largest class on the training data and then predicts the test-data as belonging to that class.

Classification problem

The present classification problem is a multiclass one, as the desired class to be predicted is *binary* and has multiple values. More specifically this class describes the risk level on the basis of risk factors in pregnancy (low risk, mid risk, high risk).

Initialization and parameter set up

As mentioned above,in this project three different classification methods are compared using two-level cross validation[2] *Logistic Regression*, *K-Nearest Neighbors* and *baseline*.The current cross validation consists of two loops, inner and outer, of 10 iteration each using K-Fold method. First the standardized data are imported and then are split into a training set and a test-set using K-Folds cross validator ($K = 10$). Afterwards follows the part of initialization of the classifiers. More specifically in Logistic Regression method the solver *Newton-CG* is used, while in K-Nearest Neighbors the algorithm *Ball Three* with *Leaf Size = 50* is selected. Next follows the setting of the parameter grid, in which the desired hyperparameters for tuning are set. In the case of Logistic Regression method C the regularization parameter $\mathbf{C}$ ($C = 1/\lambda$) is set, where

λ defines the complexity of the model, while in the case of K-Nearest Neighbors method the parameter concerning the number of neighbors $n\_neighbors$ is set.

Hyperparameter tuning and error calculation

Then the best hyperparameters are being selected using **GridSearchCV**[4] command from *sklearn-.model_selection* module for each model, which is essentially the inner loop. It is worth mentioning that **GridSearchCV** calculates the average error of all inner iterations for each model and not the error of each iteration. Next the error rate for each outer iteration is calculated using the following equation:

$$E = \frac{\{\text{Number of misclassified observations}\}}{N^{test}}$$

Results

After the hyperparameter tuning the best parameters are selected for each model. In the case of Logistic Regression the value of parameter C with the lowest error is $C = 0.01 => \lambda = 100$ and in the case of K-Nearest Neighbors the best value for the number of nearest neighbors is $n\_neighbors = 1$. It worth noting that for every inner iteration during the hyperparameter tuning the best parameters for both Logistic Regression and K-Nearest Neighbors are the same respectively. That means that no further analysis is needed to be done, regarding best parameter selection. Otherwise, if there were more than one parameter to choose, then the one with the lowest error rate would be selected. In Table 3 are presented the best parameters extracted from hyperparameter tuning, as well as the calculated error rates from the outer validation splits for each model used in the two-level cross validation.

Table 3: Two-level cross-validation table used to compare the three classification models.

Outer fold	KNN		Logistic Regression		Baseline
i	k_i^*	E_i^{test}	λ_i^*	E_i^{test}	E_i^{test}
1	1	0.147	100	0.352	0.549
2	1	0.156	100	0.470	0.686
3	1	0.117	100	0.352	0.607
4	1	0.147	100	0.323	0.686
5	1	0.158	100	0.405	0.524
6	1	0.168	100	0.346	0.564
7	1	0.118	100	0.386	0.623
8	1	0.158	100	0.297	0.504
9	1	0.158	100	0.336	0.673
10	1	0.188	100	0.306	0.574

Statistical Evaluation

For our classification task we chose three models, namely a baseline model, logistic regression (regularized) and the K-nearest neighbor classification method. As a baseline model we chose one that classifies all the new observation in the bigger class (of the set that was trained on). We trained the models using two-layer cross validation and got some interesting results, but in order to determine whether a model is better than another one we should perform statistical evaluation.

To do so, we chose to perform McNemar tests and compare the models pairwise. The significance level we chose is a=0.05. In 4 one can see the summary of results. The p-value was used in order to determine if there is difference between the performance of two classifiers and the confidence interval in order to see where this difference lies (if of course p-value shows a difference). Essentially, with the McNemar tests we performed, we tested the null hypothesis $H_0 : \theta = \theta_{M_1} - \theta_{M_2}$=0, where M_1 and M_2 the models we compared. In cases that the null hypothesis was rejected, we have difference between the two models with significance level 0.05. In the other cases we didn't have enough evidence to reject the null hypothesis and thus we cannot determine whether there is difference in models' performance with significance level a.

McNemar's test estimates the difference between the accuracy of two classifiers. So, when there is difference between the accuracy of two classifiers, the estimation $\hat{\theta}$ gives us evidence about which classifier is better, essentially which is correct more times, and thus evidence about which classifier to choose. Note that we performed McNemar's test 10 times for each pair of models. The parameters of the models were determined in the inner layer of the two-layer cross validation and the models were tested in the outer layer of the of the the two-layer cross validation. Below we present the results more detailed for the pairs of models.

K-Nearest Neighbors against Baseline

As one can see in 4 in all the tests we performed to compare K-Nearest Neighbors and the Baseline, the p-value was smaller than the significance level a(0.05) and thus the null hypothesis is rejected. This means that there is difference between the accuracy of the two models that are compared each time. Based on the confidence interval we can conclude that in general K-Nearest Neighbor is better that the Baseline model in general, a fact we anticipated. This is because the test showed that θ's estimation is always positive, so KNN's accuracy is bigger than Baseline's.

K-Nearest Neighbors against Logistic Regression

Between K-Nearest Neighbors and Logistic Regression the statistical evaluation gives us evidence to prefer KNN to Logistic Regression. 9 out of 10 tests we rejected the null hypothesis and so we had difference in the accuracy of the classifiers. The estimation of the difference was always positive as can be seen in the column of confidence intervals and thus we can say that with confidence level 0.05 KNN model has better performance than Logistic Regression.

Logistic Regression against Baseline

Finally, between Logistic Regression and the Baseline model the statistic evaluation gives us evidence to prefer Logistic Regression, as expected. In all the tests we performed the null hypothesis was rejected and the estimation of θ (difference in accuracy) was positive, so we can conclude that Logistic Regression is better model.

We should note that the results are based both in the sets we used to train the models and the significance level we choose. That's because if we change the set we train the models the performance may change and if we change the significance level the p-value and the confidence intervals will change as well.

Logistic Regression model and feature selection comparison

In the classification process a Logistical Regression model was trained on the maternal dataset using the paramater $\lambda = 100$, which was derived after the hyperparameter tuning in the inner

loop of the two level cross validation.

Logistic Regression is essentially a linear model used for classification, which uses different solvers in order to solve the desired optimization problems. In this case, which is a multinomial problem, the *newton-cg* solver is used in order to minimize the following cost function:

$$\min_{(w,c)} \frac{1}{2} w^T w + C \sum_{i=1}^{n} log(exp(-y_i(X_i^T w + c)) + 1)$$

.

Feature selection is implemented in the outer fold of two level cross validation in both Classification and Regression with the method of Logistic Regression and Linear Regression respectively, using the algorithm provided from Toolbox, resulting in the same outcome. The important features are ***Systolic BP***, ***Blood Glucose***, ***Body Temperature***, ***Heart Rate***, ***Risk Level***, while the less important are ***Age*** and ***Diastolic BP***.

Table 4: Statistical Evaluation for Classification Models

Fold	KNN vs Baseline		KNN vs Logistic Regression		Logistic Regression vs Baseline	
i	p-value	Confidence Interval	p-value	Confidence Interval	p-value	Confidence Interval
1	$1.3{\cdot}10^{-8}$	(0.28,0.51)	$5{\cdot}10^{-4}$	(0.09,0.31)	$3{\cdot}10^{-4}$	(0.09,0.29)
2	$7.3{\cdot}10^{-12}$	(0.40,0.64)	$9{\cdot}10^{-6}$	(0.18,0.43)	$1{\cdot}10^{-5}$	(0.12,0.3)
3	$2.9{\cdot}10^{-11}$	(0.36,0.6)	$6.9{\cdot}10^{-5}$	(0.12,0.33)	$2.4{\cdot}10^{-5}$	(0.14,0.36)
4	$4.8{\cdot}10^{-13}$	(0.42,0.64)	$2{\cdot}10^{-3}$	(0.07,0.27)	$1.4{\cdot}10^{-10}$	(0.26,0.45)
5	$1.2{\cdot}10^{-7}$	(0.24,0.48)	$7{\cdot}10^{-5}$	(0.13,0.35)	$2.2{\cdot}10^{-2}$	(0.02,0.21)
6	10^{-8}	(0.27,0.5)	$5{\cdot}10^{-3}$	(0.06,0.29)	$1{\cdot}10^{-4}$	(0.11,0.31)
7	$1.6{\cdot}10^{-12}$	(0.38,0.61)	$1.4{\cdot}10^{-6}$	(0.16,0.36)	$3.8{\cdot}10^{-5}$	(0.13,0.33)
8	$1.2{\cdot}10^{-6}$	(0.22,0.46)	$1.6{\cdot}10^{-2}$	(0.03,0.24)	$4.9{\cdot}10^{-5}$	(0.11,0.29)
9	$5.7{\cdot}10^{-11}$	(0.38,0.63)	$6{\cdot}10^{-3}$	(0.05,0.29)	$1.9{\cdot}10^{-8}$	(0.23,0.43)
10	$8{\cdot}10^{-8}$	(0.26,0.5)	$5.01{\cdot}10^{-2}$	(0.01,0.22)	$1.4{\cdot}10^{-6}$	(0.16,0.36)

Discussion

In this project we engaged with the fundamentals of multinomial Classification and Regression process, as well as with the statistical evaluation of their outcomes. More specifically we started from simple implementation such as using Linear Regression model on our dataset, to predict the desired class, to two level cross validation so that we distinguish the best classification and regression methods and models. More in depth we learned how to run models like Logistic Regression, KNN algorithm and Artificial Neural Network, as well as use K-fold algorithm for training purposes in order to acquire trusted results. Moreover we practiced hyperparameter tuning for the above methods and we got familiar with regularization parameters. Finally we tested our statistics skills by executing statistical evaluation on both regression and classification's results using the McNemar algorithm.
The present dataset was part of a research paper[1] by Marzia Ahmed and Mohammod Abul Kashem. The main goal of their research was to analyzed the health data and risk factors of pregnant women to identify the risk intensity level by using classification methods. Despite the fact that our pool of classifiers for comparison differ considerably from theirs, Logistic Regression method was a common one. We both deduced that Logistic Regression method was performing above average, but still not accurately enough.

Exercises

Exercise 1

The correct answer is C. The ROC is defined by considering all conceivable thresholds and plotting the true positive rate (TPR) against the false positive rate (FPR). For the threshold at 0.5 we have no FP thus FPR = 0, whereas TP = 1 and FN = 3, thus TPR=0.25. Raise the threshold we get when threshold at 0.6 that TP = 1, FP = 2, TN = 2, FN = 3, thus FPR = 0.5, TPR = 0.25. All ROC situations in C match the given curve, so choose C.

Exercise 2

Exercise 3

Exercise 4

The correct answer is D. We can see that split 'C' creates a branch including only the congestion level 4. So, in the Figure 4, split 'C' must be related to b_1. Split 'C' is the one that determines the border for congestion level in Figure 4. The value can be found in Figure 4 (a little lower than 0). And because b_1 is entirely responsible for congestion 4 level's border we can conclude that the correct answer is D.

Exercise 5

The correct answer is **C**. For each outer fold we have 4 inner folds and for each inner fold we train and test 5 different ANN and 5 different regression (based on the parameters). So in order to determine the parameter we need 5·4·5(25+9)=3400ms. On top of that we need time for the train and test of the models with optimal parameters on the outer levels, that is 5(25+9)=170ms. So in total we need 3570ms which is answer C.

Exercise 6

In order to find which observation is classified in class number 4 we need to calculate the probabilities for each class for each observation and see in which class each observation belongs. The way we do it is using probabilities (transformation, softmax in this example), because probabilities belong to [0,1] and the predictions may not belong in the same interval. The correct observation is **B** $\mathbf{b}= [-0.6, -1.6]^T$. To see this we first compute the vector y=$[\bar{y_1},\bar{y_2},\bar{y_3}]^T$.

It is:

$\bar{y_1} = 1 \cdot 1.2 + 0.6 \cdot 2.1 - 3.2 \cdot 1.6 = -2.66$

$\bar{y_2} = 1 \cdot 1.2 + 0.6 \cdot 1.7 - 2.9 \cdot 1.6 = -2.42$ and

$\bar{y_3} = 1 \cdot 1.3 + 0.6 \cdot 1.1 - 2.2 \cdot 1.6 = -1.56$

So the probability we want to calculate is $P(y = 4|\bar{y}) = \frac{1}{1+e^{-2.66}+e^{-2.42}+e^{-1.56}} = 0.73$. This is the highest probability that a class for this observation can have, so we can conclude that observation **B** will be classified in class 4.

Table 5: Work distribution

Section	**Papadamos (s205637)**	**Loukas (s220514)**	**Zhang (s212657)**
Introduction		✓	
Feature Selection	✓		
Regularization Parameter	✓		
Predict New Observation	✓		
Model Evaluation			✓
Initialization and Parameter Set Up (Reg)			✓
Error Calculation and Hyperparameter Tuning (Reg)			✓
Results (Reg)			✓
Statistical Evaluation (Reg)			✓
Classification		✓	
Classification Problem		✓	
Initialization and Parameter Set Up (Cls)		✓	
Hyperparameter Tuning and Error Calculation		✓	
Results (Cls)		✓	
Statistical Evaluation (Cls)	✓		
Discussion		✓	
Coding	✓	✓	✓
Exam Question 1			✓
Exam Question 4	✓		
Exam Question 5	✓		
Exam Question 6	✓		

References

[1] Marzia Ahmed and Mohammod Abul Kashem. "IoT Based Risk Level Prediction Model For Maternal Health Care In The Context Of Bangladesh". In: (2020), pp. 1–6. DOI: 10.1109/STI50764.2020.9350320.

[2] *Cross Validation*. URL: https://github.com/rasbt/stat451-machine-learning-fs20/blob/master/L11/code/11-eval4-algo__nested-cv_verbose1.ipynb.

[3] Tue Herlau, Mikkel N. Schmidt, and Morten Mørup. *Introduction to Machine Learning and Data Mining*.

[4] *scikit-learn*. URL: https://scikit-learn.org/stable/.