**Instructions**: Please complete and submit your work to the appropriate folder in LumiNUS. You may work in study groups, but each student must be responsible for their own submission.

Please submit all the following documents as a single zip file named StudentID-Name-H5.zip:

1. Completed Word file named as StudentID-Name-H5.docx (with all results)
2. Print preview of ipynb file named as StudentID-Name-H5.pdf (with results)
3. Working ipynb file named as StudentID-Name-H5.ipynb
4. Naïve Bayes and Logistic Regression are both probabilistic classifiers. (i) Describe how they are the same and how they are different. (ii) Describe the even closer connection between Gaussian Naïve Bayes and Logistic Regression. (iii) It is often said that Logistic Regression is the Linear Regression idea applied to Classification problems. Explain why people would say that statement.

**Ans 1:**

1. **Naive Bayes:** Naïve Bayes is based on Bayes’ Theorem with the ”naive” assumption that features are independent. The Naive Bayes is linear classifier using Bayes Theorem and strong independence condition among features. Given a data set with n features represented by , Naive Bayes states the probability of output: Y from features are:

This requires that the features Fn are conditionally independent. From Bayes Theorem:



**Logistic Regression:** Logistic Regression is based on applying the idea of Linear Regression to classification problems. It is mainly used in cases where the output is Boolean. For features  , outcome Y can take two values from {0,1} and the probability of the out is assumed to follow a parametric model given by the sigmoid function.





This gives us a simple linear expression for classification. We predict the outcome to be Y=1 if: 

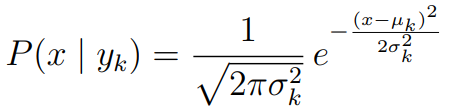
**How Both Are The Same (Similar To Each Other):**

* Both Naive Bayes and Logistic regression are linear classifiers and are of inherently probabilistic in nature, and hence, used for classification problems.
* Both approaches yield a probability distribution over a set of classes for each input sample. To classify, one would typically return the class with highest probability and hence, both models can be used for classification problems.
* Logistic Regression is consistent with the Naive Bayes assumption that the input features Xi are conditionally independent of each other, given Y, and both classifies input data by taking highest likelihood class.

**How Both Are Different:**

* Logistic Regression makes a prediction for the probability using a direct functional form whereas Naive Bayes figures out how the data was generated given the results.
* Naive Bayes is a generative model as it models the joint distribution of the feature X and target Y, and then predicts the posterior probability given as P(X|Y). However, Logistic Regression is a discriminative model. As it directly models the posterior probability of P(Y|X) by learning the input to output mapping by minimizing the error.
* Naïve Bayes assumes all the features to be conditionally independent. Logistic regression splits feature space linearly.
* Naïve Bayes is known to have a higher bias and lower variance, and Logistic regression is the exact opposite.

1. When your data set’s features are all continuous, the Gaussian Naive Bayes classifier is appropriate. In this classifier, the assumption is that data from each label is drawn from a Gaussian distribution. Suppose x is a continuous feature. Let µk be the mean of the values in x associated with class yk, and let σk2 be the bias corrected variance of the values in x associated with class yk. Then P(x|yk) is given by’



The closer connections between Gaussian Naïve Bayes and Logistic Regression are as follows:

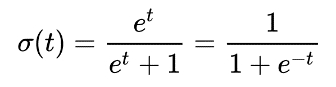
* Both operate well on continuous features and separates features in a linear fashion.
* The equations governing both of them are pretty similar. The Naïve Bayes Gaussian model can be modelled to a linear classifier, a hyperplane that separates the two classes,  which takes the following form where both can categorize continuous input features well as both forms a continuous function to relate the features and target where w w is a vector of coefficients that define the separating hyperplane and b is the hyperplane’s intercept. W and b are functions of the Gaussian moments:

P(y|x) = \frac{1}{1+e^{-y(w^T x+b)}}

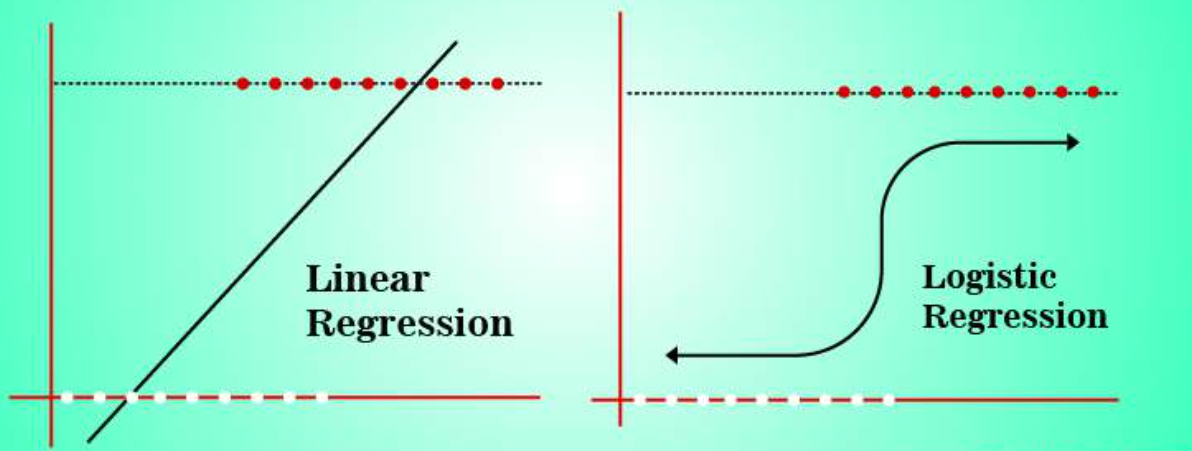
* Logistic Regression is a linear classifier over X. The linear classifiers produced by Logistic Regression and Gaussian Naive Bayes are identical in the limit as the number of training examples approaches infinity, provided the Naive Bayes assumptions hold.

1. In my opinion, the main reason why people would say that Logistic Regression is the Linear Regression idea applied to Classification problems because of the following reasons:

* Both generate continuous value of the “Y-output” i.e., the dependent variable. In my opinion, the sigmoid curve of the Logistic Regression curve runs from (0,1) which shows the probability estimation and generates a “continuous” output from input continuous features. This is what makes logistic regression a classification algorithm that classifies the value of linear regression to a particular class depending upon the decision boundary, which is defined by the equation below:



* Linear Regression predicts Y or output values for the given set of rules as input to the model, and is governed by the general line equation Y=mX+c and will output continuous values which aren’t suitable for categorizing within one of the two classes nor identifying it as a probability value to predict a class. However, if the same logic is applied for the sigmoid curve of Logistic Regression, the sigmoid curve varies from 0 to 1 probability values which returns the overall class of the output variable, and hence, is used for classification problems as the output does not “vary” with different input values as it sticks to {0,1}, so it scales better for the problems of likelihood with [0,1] values for its lower and upper bound.
* Both linear and logistic regression has the same working principle which is the minimize the loss function.
* Hence, the graph below summarizes it all: Logistic Regression can be a Linear Regression, but a classification version of it which returns a class {0,1} based on probability values of continuous input variables.



1. Consider the Play Tennis / Don’t Play Tennis dataset. (i) Compute the probability that players will play tennis if it is sunny. (ii) Compute the probability that players will play tennis if it is sunny and windy.

**Ans 2:**

Text, letter

Description automatically generated

1. In this problem, we will look at the Digits dataset available in SKLearn. You can start with the C08 code I have uploaded to LumiNUS/Files/Code, and you can use the dataset from SKLearn. The dataset is a set of 8x8 images of handwritten digits, so there are 10 classes (0 through 9), with about 180 images per class.
2. Look through the dataset and assess for yourself which handwritten digits are the hardest to recognize for you and your friends. This will involve you visualizing the data. My code shows you how to do that.
3. Split the data into 75% training and 25% test sets. Run a supervised training and classification test using the SVM, Naïve Bayes and Logistic Regression classifiers. Display out the accuracy, some sample image predictions, and the classification reports for each classifier.
4. Report the accuracy scores of each classifier and rank the classifiers in terms of their accuracy scores. Is it necessary to average the accuracy over multiple runs? Explain why.
5. For each classifier, determine which was the “hardest” digit for each classifier to categorize. You can do this by looking at the confusion matrix. You can look at one of the past code sample or the SKLearn documentation to figure out the syntax for the confusion matrix.

Please refer to the following documentation for more information about SKLearn syntax.

* You can read about performance metrics at: <https://en.wikipedia.org/wiki/Confusion_matrix>
* SKLearn contains functions to compute these metrics:

<https://scikit-learn.org/stable/modules/classes.html#sklearn-metrics-metrics>

* SKLearn digits dataset information:

<https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html>

**Ans 3:**

1. In my own eyes, the handwritten digits which are the hardest to recognize are the numbers 2, 3 and 5 & 8 respectively as their edges are not easily discernable (especially for the number 5 and 8 seems blurry). The same applies for my friends as well when I asked them.
2. Code is given in the PDF and .ipynb file codes.
3. Ranking of accuracy scores of each classifier:

|  |  |  |
| --- | --- | --- |
| **Supervised Classifier** | **Average Accuracy Score** | **Classifier Ranking Based On Accuracy Score** |
| SVM | 0.9822222222222223 | 1 |
| Logistic Regression | 0.9577777777777777 | 2 |
| Multinomial Naïve Bayes | 0.9022222222222224 | 3 |
| Gaussian Naïve Bayes | 0.8644444444444443 | 4 |
| Bernoulli Naïve Bayes | 0.8488888888888889 | 5 |

I think it is important to average out the accuracy over multiple runs to take into consideration, the randomness of the data due to the splitting of the testing and training data, so an average over multiple runs will give a more reliable result.

1. For SVM, the hardest seems like 8. For Logistic regression, the hardest seems 2, 8 and 9. For Gaussian, Multinomial & Bernoulli Naïve Bayes, the hardest looks like 2, 9 and 8 as well. This is represented by the number of misclassifications shown in the confusion matrix and the poor images depicted.