**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:

**Similarities**

1. They are the same in which they are both a probability classifier and they predict from the same feature label via Ymap = argmax P(Y|X)

**Difference**

They are both different in the way they build the classifier, Naïve Bayes uses the Bayes Equation to predict the likelihood of P(X|Y) and compute the posterior probability P(Y|X) using Bayes theorem. While, logistic regression calculate directly to P(Y|X) and does not attempt to model the distribution of the data.

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:**

* **Probability Player will play tennis if it is sunny: 2/5**
* **Probability players will play tennis when sunny and windy: 1/2**

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.

Ans:



I find the digit 5 to be the hardest to recognize, because it looks like 3 or 9 with it given outlook.

1. 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.
2. 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.

Ans:

Based on a cycle of 150 times of running the model. The accuracy ranking in terms of their average accuracy of SVM: 0.960222, LogisticRegression:0.938148, Gaussion:0.823363. Running it over multiple runs and averaging makes it more credible and justifiable in a way to stimulate running in a real life-scenario.

1. 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.

Ans:

Based on the syntax for the confusion matrix:

SVM Confusion Matrix

A chart of a confused matrix

Description automatically generated with medium confidence

Logistic Regression:

A chart of a confused matrix

Description automatically generated with medium confidence

Gaussian:

A diagram of a confused matrix

Description automatically generated

Based on the diagonal values of the correctly predicted instances: it can be concluded that across classifiers the digit 4 was the hardest to recognize.

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>