**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.
5. Same: both of them predict the label from feature via 文本

   中度可信度描述已自动生成

Different: Naïve Bayes builds the joint model via Bayes equitation. P(x|y)= P(x,y)/ P(y);

Logistic Regression build the P(x|y) directly.

1. Gaussian Naïve Bayes and Logistic Regression are the same.
2. Logistic Regression classifier is 文本

   描述已自动生成 which is same as LR function. Actually, Logistic Regression using sigmoid function maps the label. Using this, it regresses the boundary of the training data. So it using LR idea regress the classifier.
3. 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.
4. P(x=play|y=sunny)=P(x,y)/P(y)=2/5
5. P(x=play|y=sunny,windy)= P(x,y)/P(y)=1/2
6. 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.
7. 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.

I find 5 is hardest to recognize. Because it looks similar to 3 or 9

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.

Cycle 100 times with average accuracy in SVC:0.957000 LogisticRegression:0.935000 Gaussion:0.820600. In my opinion, it is necessary to run multiple times. Running times over times is a kind of Monte Carlo method to simulink the application senario. The average result after mutiple runs is more credible.

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.

SVC:

日历

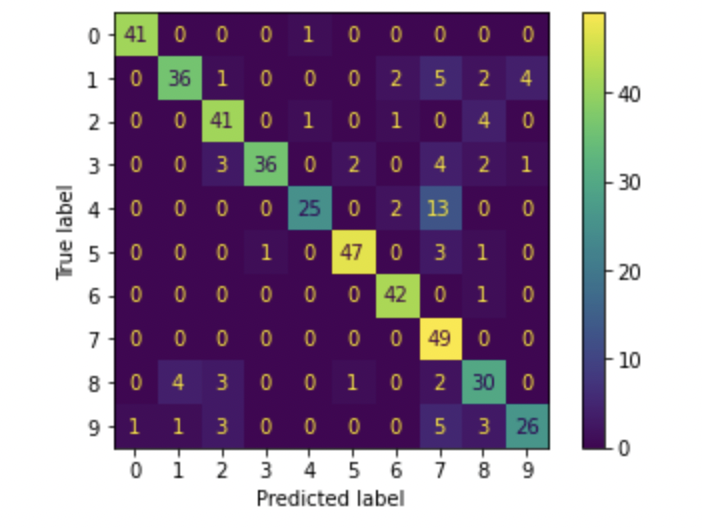
描述已自动生成

LogisticRegression:

日历

描述已自动生成

Gaussian:



In conclusion, 4 is the ‘hardest’ digit for each classifier to categorize.

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>