**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-H4.zip:

1. Completed Word file named as StudentID-Name-H4.docx (with all results)
2. Print preview of ipynb file named as StudentID-Name-H4.pdf (with all results)
3. Working ipynb file named as StudentID-Name-H4.ipynb
4. Consider the metrics accuracy, precision, and recall.
   1. Give one example when accuracy would not be a good performance metric. Give a numerical example.

990 negatives and 10 positives in 1000 samples

|  |  |  |  |
| --- | --- | --- | --- |
| Total population | | Predicted condition | |
| Positive (PP) | Negative (PN) |
| Actual condition | Positive(P) | 0 | 10 |
| Negative(N) | 0 | 990 |

The classifier predicts all samples as negative with 99% accuracy. However, it is useless, while it can not find any positive ones.

* 1. Given one example of a supervised machine learning classification problem when higher precision is desired. Please give a different example than the ones given in class. This need not be a numerical example but must be clearly defined classification problem and dataset.

Drunken driving is illegal. When drunk is positive, to test whether diver is drunk, precision is important. Because we don’t want to arrest a good man wrongly, the false positive should be as little as possible. Therefore, the higher precision is desired.

* 1. Given one example of a supervised machine learning classification problem when higher

recall is desired. Please give a different example than the ones given in class. This need not be a numerical example but must be clearly defined classification problem and dataset.

In faculty, all products need to be examined before selling, which means disqualified products need to be found. Set the products below standard positive while over the standard negative. To make sure there isn’t any broken products sold, the false negative, mistaking disqualified product as qualified, should be as little as possible.

Therefore, the higher recall is desired.

1. Suppose you are given the same test dataset and two binary classifiers. Give a numerical example such that Classifier 1 has higher accuracy than Classifier 2, but Classifier 2 has both higher precision and higher recall than Classifier 1? Hint: Give a hypothetical 2x2 confusion matrix for each classifier.

200 negatives and 100 positives in 300 samples

Classifier 1

|  |  |  |  |
| --- | --- | --- | --- |
| Total population | | Predicted condition | |
| Positive (PP) | Negative (PN) |
| Actual condition | Positive(P) | 50 | 50 |
| Negative(N) | 60 | 140 |

Accuracy1=63.33% recall1=50% precision1=45.46%

Classifier 2

|  |  |  |  |
| --- | --- | --- | --- |
| Total population | | Predicted condition | |
| Positive (PP) | Negative (PN) |
| Actual condition | Positive(P) | 70 | 30 |
| Negative(N) | 81 | 119 |

Accuracy2=63% recall2=70% precision2=46.35%

I give a mathematical derivation as following: (I use a/b/c/d instead of TP/FN/FP/TN) 表格

描述已自动生成

文本, 信件

描述已自动生成

there

1. Consider the tire tread versus mileage problem we discussed in the lecture. The problem is to predict the tire tread depth from the mileage. The dataset, which has nine pairs of points, is reproduced below. This is an individual assignment to be done by every student. You may work in a group, but I expect every student to solve the problem and implement the code themselves.

|  |  |
| --- | --- |
| Mileage  (in 1000 miles) | Tire Tread Depth  (in mils) |
| 0 | 394.33 |
| 4 | 329.50 |
| 8 | 291.00 |
| 12 | 255.17 |
| 16 | 229.33 |
| 20 | 204.83 |
| 24 | 179.00 |
| 28 | 163.83 |
| 32 | 150.33 |

1. Compute the linear regression solution (i.e., best fit line) for this dataset. Use the entire dataset to train and find the best fit line. Give the expression for the best fit line and compute the error performance on the training dataset. Recall that the error performance is measured by the sum of squared errors. For this question, you can use Python to do the computations, but you may not use Scikit-learn to do the regression for parts a, b, & c.
2. Plot the best fit line over the data points and comment on whether the fit is good.
3. Leave out the last sample (x=32) and use it as a test data point. Use the remaining samples to train and find the best fit line. Give the expression for the best fit line. Compute the training error and the test error performance.
4. Using the entire dataset to train, find the linear regression solution using Scikit-learn and compare to the solution you got in part (a). The two solutions might be different. Can you explain why? Try doing this several times if the answers are the same.