**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.
   2. 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.
   3. 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.
5. 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.
6. 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.

Ans & Questions:

1a) Accuracy is not always a good predictor of accuracy as when the dataset is imbalanced or one class outweighs the other, the accuracy predictor might give a false sense of the model performance as it might classify the majority class more then the minority classes. Given for an example the dataset contains 1000 patients, and only 10 of them have the disease (i.e., 990 healthy patients, 10 diseased patients).

Numerical Examples:

**True Positives (TP)**: 1 (correctly predicted diseased)

**True Negatives (TN)**: 900 (correctly predicted healthy)

**False Positives (FP)**: 90 (incorrectly predicted diseased)

**False Negatives (FN)**: 9 (incorrectly predicted healthy)

**Accuracy Calculation:**

Accuracy=TP+TNTP+TN+FP+FN=1+9001+900+90+9=9011000=90.1%\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{1 + 900}{1 + 900 + 90 + 9} = \frac{901}{1000} = 90.1\%Accuracy=TP+TN+FP+FNTP+TN​=1+900+90+91+900​=1000901​=90.1%

In this case the model accuracy will be incorrect as it only predicts only 1 out 10 have disease it but in fact the other minority was not detected. Therefore, it is better to use precision, recall to measure.

1b) 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.

Ans: For example, using the case of an email filtering system problem where the algorithm must classify between spam or not spam or True Positive or False Positive and for this situation, we want to reduce the amount of False Positive in the prediction. Hence, models such as Logistics Regression can work well with this type of classification problem, and usually metrics such as precision are more preferred to filter between spam or not a spam mail.

Dataset: 1000 emails - Inbox

Classes:

Spam (Negative Class) – True Negative - 100  
 Not Spam (Positive) – True Positive - 900

Precision Percentage of identified correctly: 90%

Comment: We want to increase the precision percentage to make sure the model can predict more accurately without any False Positive Class.

1c)

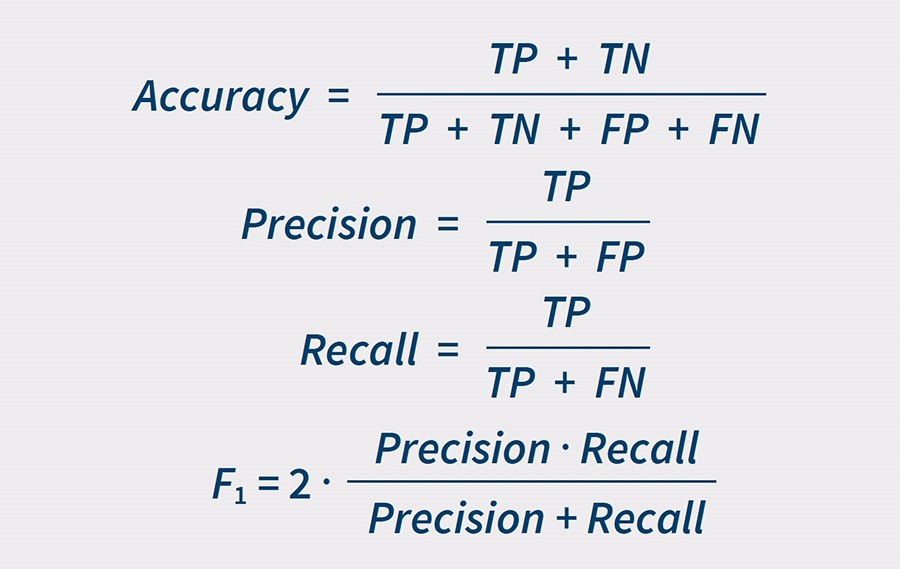
Problem Definition: The purpose is to determine whether internet transactions are fraudulent or real. In this case, we want to make certain that as many fraudulent transactions as possible are correctly identified, even if it means identifying a few valid transactions as potentially fraudulent. This makes recall the most crucial metric.  
  
Classification: Fraudulent transactions (positive class).  
Legitimate transaction (negative class).  
  
Why Higher Recall is Important: In the case of fraud detection, the expense of missing a fraudulent transaction (false negative) far outweighs the inconvenience caused by mistakenly identifying a genuine transaction (false positive). If fraudulent transactions go unnoticed, they can result in considerable financial losses for both clients and the organization.

Note:   
True Positives (TP): Successfully identified fraudulent transactions.  
False Negatives (FN) are fraudulent transactions that were improperly classified.

2) 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.

Ans:

Definitions of Terms:  
Q3) – Thread Tire Problems



**Data Set is based on a set of data – 100 data points**

**Confusion Matrix 1(Higher Accuracy/ Lower Recall(Precision)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Confusion Matrix**  **for Classifier 1** | | **Actual Class** | |
| **Positive** | **Negative** |
| **Predicted Class** | **Positive** | TP = 20 | FP = 30 |
| **Negative** | FN = 28 | TN = 22 |

**Confusion Matrix 2(Lower Accuracy) / Higher Precision and Recall)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Confusion Matrix**  **for Classifier 2** | | **Actual Class** | |
| **Positive** | **Negative** |
| **Predicted Class** | **Positive** | TP = 30 | FP = 42 |
| **Negative** | FN = 18 | TN = 10 |

Classifier Matrix 1 : Values

* Accuracy: 0.42
* Precision: 0.4
* Recall: 0.416

Classifier 2 Outputs :

* Accuracy: 0.4
* Precision: 0.416
* Recall: 0.625

Then we can see that Classifier 1 has a higher accuracy rate, but classifier 2 has higher precision and recall rates.

3) DataSet is given below

|  |  |
| --- | --- |
| 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 |

Qn)

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 it 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.

\*Need to calculate the value of m and b) – and fit it into the equations

Q3a) Linear regression formula =

Y= mx + b – Sub in formulas

Formula to calculate b)

[(ΣY)(ΣX2) – (ΣX)(ΣXY)]  /  [n(ΣX2) – (ΣX)2]

[n(ΣXY) – (ΣX)(ΣY)]  /  [n(ΣX2) – (ΣX)2]

Q3b)

A line graph with dots

Description automatically generated

Tire Thread

Mileage

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Mileage** | **Tire Tread Depth** | **X\*Y** | **X^2** | **Y^2** |
|  | (in 1000 miles) | (in mils) |  |  |  |
|  |  |  |  |  |  |
|  | 0 | 394.33 | 0 | 0 | 155496.1 |
|  | 4 | 329.5 | 1318 | 16 | 108570.3 |
|  | 8 | 291 | 2328 | 64 | 84681 |
|  | 12 | 255.17 | 3062.04 | 144 | 65111.73 |
|  | 16 | 229.33 | 3669.28 | 256 | 52592.25 |
|  | 20 | 204.83 | 4096.6 | 400 | 41955.33 |
|  | 24 | 179 | 4296 | 576 | 32041 |
|  | 28 | 163.83 | 4587.24 | 784 | 26840.27 |
|  | 32 | 150.33 | 4810.56 | 1024 | 22599.11 |
| Summation | 144 | 2197.32 | 28167.72 | 3264 | 589887.1 |
|  |  |  |  |  |  |
| bo(m) - b |  | 360.63 |  |  |  |
| b1(b) -m |  | -7.281 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| SSE Calculation: |  |  |  |  |  |
| SSE=∑(yi​−(mxi+b))2 –  SSE = SS\_Total - SS\_R = 53418.7298 - 50887.2004 = 2531.5294 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Therefore, subbing the values of bo and b1 into the equation y = b1x + b the best fit line expressions is given as y = -7.281x + 360.63# and the Sum of the squared error = 2531.529 for the SSE

Comment: The fit of the line is decently good with the data points not separating too far away from the fitted line.

3c)

Leaving out the last test sample (x = 32, and y = 150.33) we have the best fit line expression with y = -7.89x +366.30 with a slight deviation from a better fitted line. The training error is 1705.33 and the Test Error is 1327.82

3d) Running it multiple times and I got almost identical results with not much difference, but in a way how it might be different is due to the numerical precision used between the manual linear regression. Whereas ScikitLearn has a higher precision due to its optimization algorithms and may involved regularization and minor tweaks to ensure stability and efficiency in the output results.

A graph of a fit line

Description automatically generated