**Solution:**

1. I have solved the problem using python language and I used Google Colaboratory.
2. Result analysis:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | **Confusion Matrix *(Update values here)*** | | |  |
|  |  |  |  |  |
|  |  | Predicted: Non-Persistent | Predicted: Persistent |  |
|  | Actual: Non-Persistent | 1880 | 255 |  |
|  | Actual: Persistent | 369 | 920 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  | **Accuracy Metrics *(do not enter anything here)*** | | |  |
|  |  |  |  |  |
|  | Overall Accuracy | | 82% |  |
|  | Persistent class Recall | | 71% |  |
|  | Non-persistent class Recall | | 88% |  |
|  | Persistent class Precision | | 78% |  |
|  | Non-persistent class Precision | | 84% |  |
|  |  |  |  |  |
|  | ***Update values below:*** |  |  |  |
|  |  |  |  |  |
|  | F1 Score: |  | 0.86, 0.75 |  |
|  | Area Under Curve: |  | 0.882 |  |
|  |  |  |  |  |

I will attach the excel file also.

1. I used the label encoder on the main dataset to perform the hypothesis testing and machine learning model performance. We can see the data in the colab notebook.
2. I’m attaching the excel file, colab notebook, and solution\_file doc.

For this classification problem, I used hypothesis testing for the model selection. Though I can use the Ensemble Learning algorithms to increase the model performance and accuracy but we failed to find out the coefficients and intercept for the input variables.