**Nazmul Hasan Rabbi report for Task 2 Quality Control**

**Task 2.1 Determine batch quality** [10 Points]

*Approach*: I read data from the file “quality\_control.defective.csv” into a *pandas* data frame called *defective*, count the number of defective items per batch that appear in *defective*, and identify those batches that contain more than 1 defective item poor as *poor\_ batches*. A batch in the file “quality\_control.measurements.csv” is labeled “poor” if it appears in *poor\_ batches*; it is labeled “good” otherwise.

I used the following SQL query to confirm that *poor\_ batches* were correctly identified:

select batchID

from defective

group by batchID

having count(\*) > 1;

* How many of the 5000 batches are of *poor* quality?

Number of poor quality batches = 2635

Quality (***good*** or ***poor***) of the first 10 batches in the file “quality\_control.measurements.csv” are presented in the table below:

|  |  |
| --- | --- |
| **batchID** | **quality** |
| B008457 | poor |
| B022992 | poor |
| B030845 | good |
| B027829 | good |
| B018931 | poor |
| B017167 | poor |
| B033830 | poor |
| B039127 | poor |
| B042136 | good |
| B041049 | good |

**Task 2.2 Train and validate a Classification Tree to predict batch quality** [15 Points]

I used 5-fold cross-validation with the first 4000 labeled samples to obtain a decision tree model with the fewest leaf nodes that predicts batch quality reliably based on input features. The model was validated using the last 1000 labeled batches.

Results with training samples:

Accuracy with training data = **95.90** % (round to 2 decimal places)

Confusion matrix for training data:

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of batches** | Predicted Good Quality | Predicted Poor Quality | Row Total |
| Label Good Quality | 1811 | 96 | 1907 |
| Label Poor Quality | 68 | 2025 | 2093 |
| Column total | 1879 | 2121 | 4000 |

Results with validation samples:

Accuracy with validation data = **100.00** % (round to 2 decimal places)

Confusion matrix for validation data:

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of batches** | Predicted Good Quality | Predicted Poor Quality | Row Total |
| Label Good Quality | 458 | 0 | 458 |
| Label Poor Quality | 0 | 542 | 542 |
| Column total | 458 | 542 | 1000 |

**Task 2.3 Interpret results** [5 Points]

How many rules did you need to achieve the results reported above? **4 rules**

Rule 1. IF test16 > 798.5 THEN QUALITY = good

Rule 2. IF test16 ≤ 798.5 AND test9 > 822.5 THEN QUALITY = good

Rule 3. IF test16 ≤ 798.5 AND test9 ≤ 822.5 AND test27 > 813.5 THEN QUALITY = good

Rule 4. IF test16 ≤ 798.5 AND test9 ≤ 822.5 AND test27 ≤ 813.5 THEN QUALITY = poor

Satisfactory classification accuracy may be achieved using the following **3 features (tests)** that were used for classification**: test9, test16, test27.** The figure below displays the decision tree.

A diagram of a test

AI-generated content may be incorrect.

**Task 2.4 Classify new examples** [5 Points]

The quality for the 40 batches in the file “quality\_control.new\_batches.csv” predicted using the trained decision tree with 4 rules are presented in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| batchID | quality |  | batchID | quality |
| B034661 | good |  | B045573 | poor |
| B025934 | good |  | B023920 | poor |
| B003756 | good |  | B002185 | poor |
| B001011 | good |  | B023469 | poor |
| B028427 | good |  | B028235 | poor |
| B038321 | good |  | B022332 | poor |
| B031049 | good |  | B047388 | poor |
| B048603 | good |  | B037418 | poor |
| B006088 | good |  | B026500 | poor |
| B004005 | good |  | B018968 | poor |
| B002095 | good |  | B019522 | poor |
| B033135 | good |  | B008613 | poor |
| B031075 | good |  | B030383 | poor |
| B038719 | good |  | B036104 | poor |
| B047296 | good |  | B033163 | poor |
| B039446 | good |  | B046386 | poor |
| B014608 | good |  | B030860 | poor |
| B009859 | good |  | B046809 | poor |
| B016923 | poor |  | B045449 | poor |
| B049499 | poor |  | B027080 | poor |