COMP30027 Assignment 1 Report

He Shen (1297447)

2. 1-NN classification

Since the label quality is a Boolean of numerical 0 and 1, I simply compare my prediction label's value with the actual ones, sum the difference, then divide by overall row counts to get a mean value, which results in an accuracy about 0.764 (76.4%).

The 76.4% of accuracy is not a very high number, this implies that this dataset is not quite suitable for a 1-NN classifier. On the one hand, this may be because 1-NN itself only considers the nearest neighbor, and for our current wine quality dataset of 1350 data points, only one of them was used for prediction, which did not have a very good utilization rate. On the other hand, the nearest neighbor may not be reliable and should be considered an outlier. For example, in the below Figure 1, I choose the feature "totalSulfurDioxide" as x, "density" as y, and randomly select 100 samples from the training dataset. If the black dot is one of our test instances that needs to predict its label, from 1-NN, it seems to be low quality since its distance to its neighbor blue dot is shorter. However, from the overall distribution, there are more orange dots nearby, hence, labeling itself as high quality has much more possibility under this consideration.

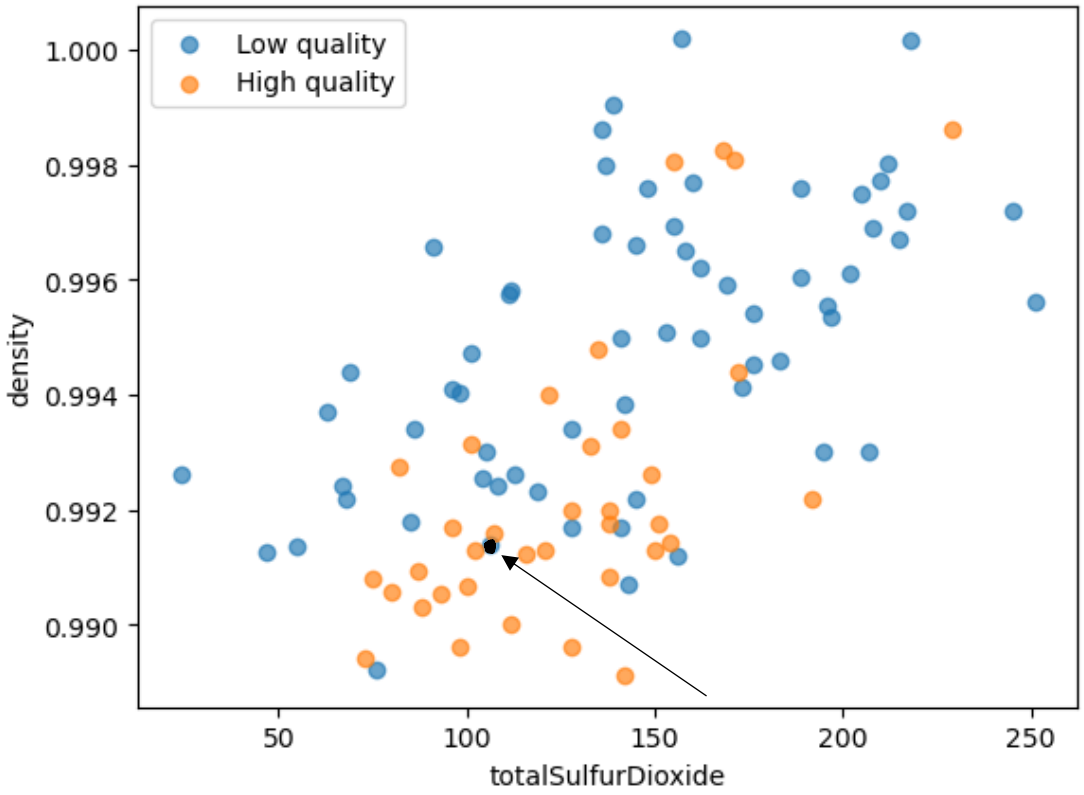


Figure 1

3. Normalization

Min-max normalization gives an accuracy of 0.850 (85.0%), while standardization gives an accuracy of 0.867 (86.7%). Both of them are much higher than the 1-NN prediction accuracy of 0.764 (76.4%) before normalization.

The whole 1-NN prediction itself is very sensitive to Euclidean distance, and some features such as “totalSulfurDioxide” may occupy larger values than others. In this case, features with larger values may have more dominance, resulting in some features such as “density” with smaller values but actually having a greater impact on the label almost ignored in distance calculation. If we redraw the last graph in question 2 using equal axis spacing, it looks like there is only a horizontal line (Figure 2). Compared to the data size and distribution of the "totalSulfurDioxide" on the x-axis, the value and interval of "density" on the y-axis are almost close to 0. That is to say, when calculating distance, "totalSulfurDioxide" occupies the majority of the calculation, thereby weakening the dependence on "density".

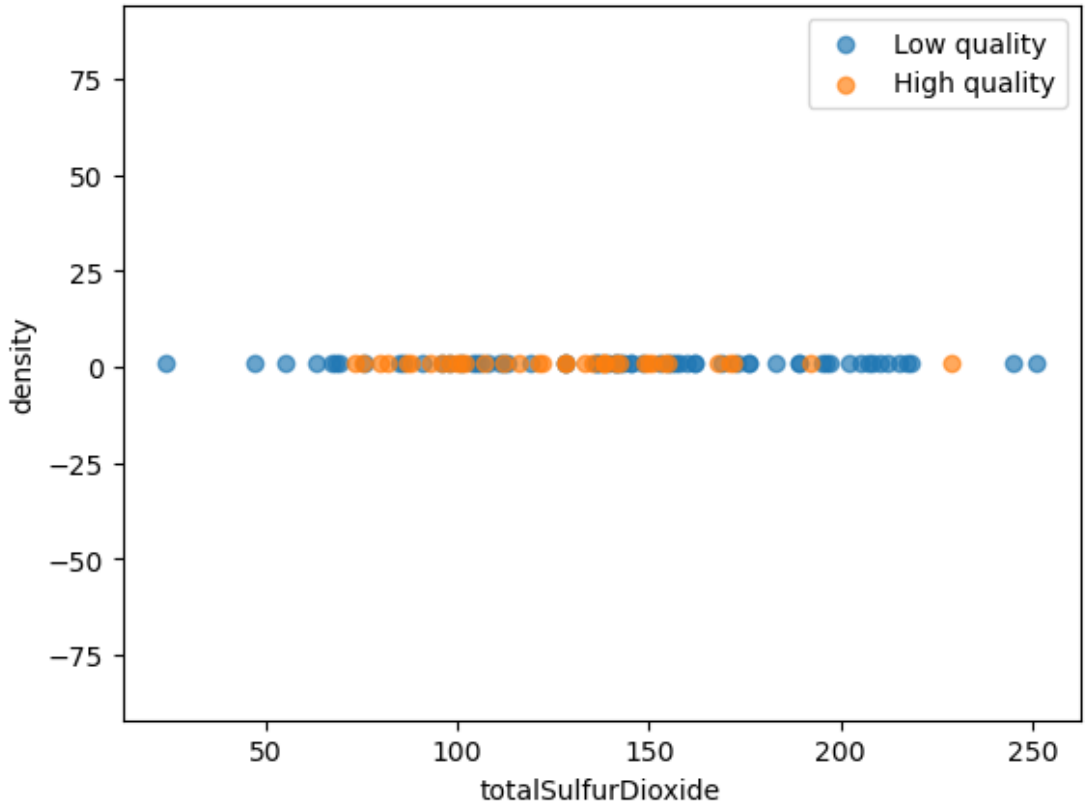


Figure 2 – Unnormalized with equal axis

After executing two normalization methods, the data for these two features were regularized, meaning that they were almost always maintained and distributed within the same and reasonable interval range. In this way, as shown in Figures 3 and 4, the data intervals and sizes of the x and y axes are more consistent, so that when calculating the distance in the end, each feature can have a role, and therefore, improves accuracy.

Figure 3 Min-max

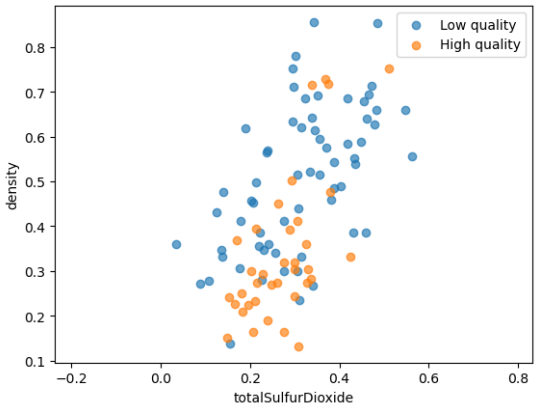
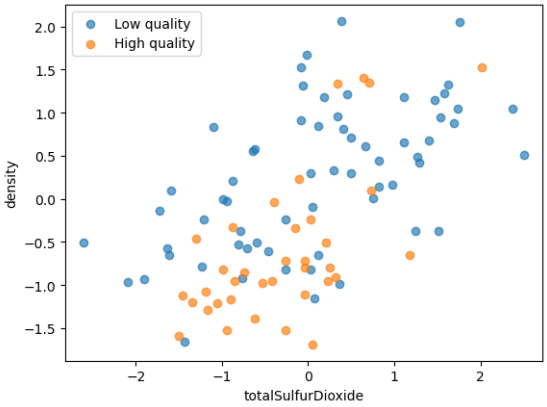


Figure 4 Standardization



4. Model extensions

(1) Compare the performance of your best 1-NN model from Question 3 to a Gaussian naive Bayes model on this dataset. In your write-up, state the accuracy of the naive Bayes model and identify instances where the two models disagree. Why do the two models classify these instances differently?

The best performance of my 1-NN from question 3 is using standardization, with an accuracy of 0.867 (86.7%). However, the Gaussian naïve Bayes' accuracy is 0.774 (77.4%), which is much smaller.

The main reason caused this difference is that out dataset does not satisfy normal distribution, which GNB greatly relies on. If we draw the graph (Figure 5) on standardized feature "density", we can find that the numbers of label low quality (0) are more than high quality (1). And at the same time, the distribution of both low quality and high quality are not symmetric, their centers of gravity are all tilting to the right, which will further cause the mean value to not remain at the center of the highest peak. Thereby affecting GNB models trained on these values.

Figure 5

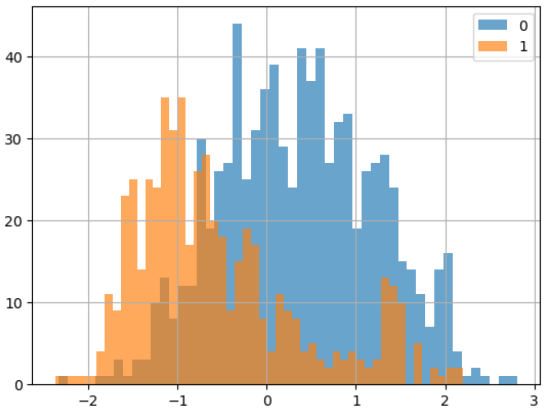


Figure 6

```
count    272.00
mean      -0.01
std        0.94
min       -2.27
25%       -0.68
50%       -0.21
75%        0.44
max        2.20
Name: density, dtype: float64
```

Also, using some pandas' tools, we can specifically see the difference between two models in "density" respect to mean and medium (Figure 6), which there are about 0.21 on both halves. When x is nearby -0.3, the GNB would regard the quality as high since it considers itself very close to its mean. However, it should actually be close to low quality due to the fact it is near its center peak, as well as how 1-NN with standardization makes the prediction.