Group Members: Tom Zhi Hern, Peter Qian Ziyu

Question 1

The two methods implemented in Question 1 are micro-averaging and macro-averaging. For micro-averaging, precision and recall are calculated by making the sum of TP, FP, FN of each class and compute the formation.

$$Precision_{\mu} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FP_i}$$

$$Recall_{\mu} = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + FN_i}$$

For macro-averaging, we calculate precision and recall for each class and take the mean.

$$Precision_{M} = \frac{\sum_{i=1}^{C} Precision(i)}{C}$$

$$Recall_{M} = \frac{\sum_{i=1}^{C} Recall(i)}{C}$$

F-score is calculated by taking the value of beta as 1.

$$F_1 = \frac{2PR}{P + R}$$

The difference between micro and macro averaging is that micro-averaging weights each sample equally while macro-averaging weights each class equally.

	Micro-averaging	**Macro-averaging**
Precision	0.75	0.719
Recall	0.75	0.736
F1-score	0.75	0.727

Table 1.1 Micro & Macro evaluation

According to *Table 1.1*, macro-averaging gives worse results than micro-averaging. The reason behind this is because classes other than "Mountain" have smaller proportion and their results average down the score in micro-averaging. For instance, "Tree", "Trianglepose" and "Warrior1" have only 6, 4, and 5 instances in the test dataset, which are much smaller than other classes such as "Mountain" which has 30 instances. Inversely, 'Mountain' class have relatively larger proportion and its score has averaged up the overall result in micro-averaging. Thus, micro-averaging evaluation will be preferred over micro-averaging as the dataset is slightly imbalanced as we have class imbalance in this assignment.

Question 2

In order to see the distribution of the dataset, we grouped every feature with missing values discarded and have selected 3 classes (Mountain, Downwarddog, Childs) to draw their QQ-plots. The QQ-plot will form a straight line if the data is normally distributed. Please note that the dataset we used is from *all.csv*, which combined all data from both *train.csv* and *test.csv*.

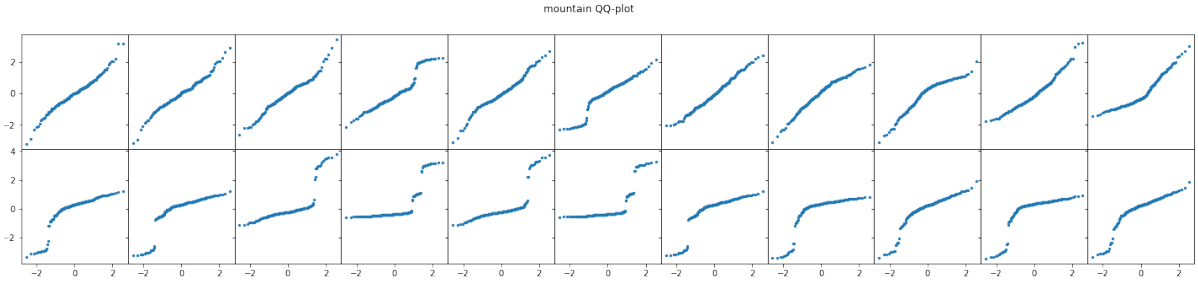

Figure 2.1: Mountain

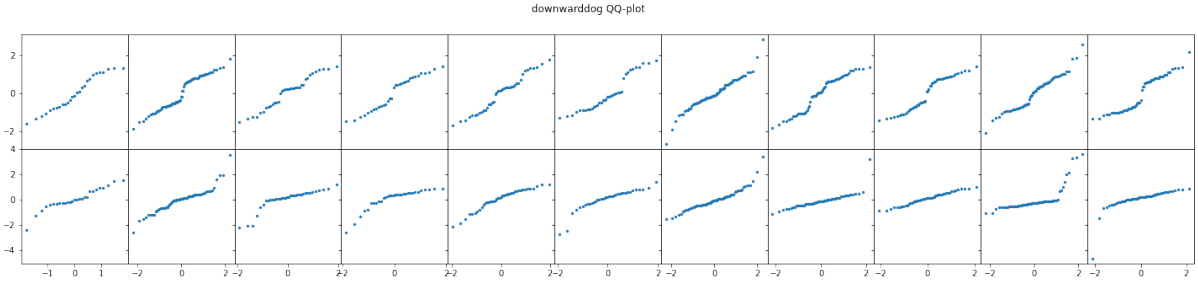

Figure 2.2 Downwarddog

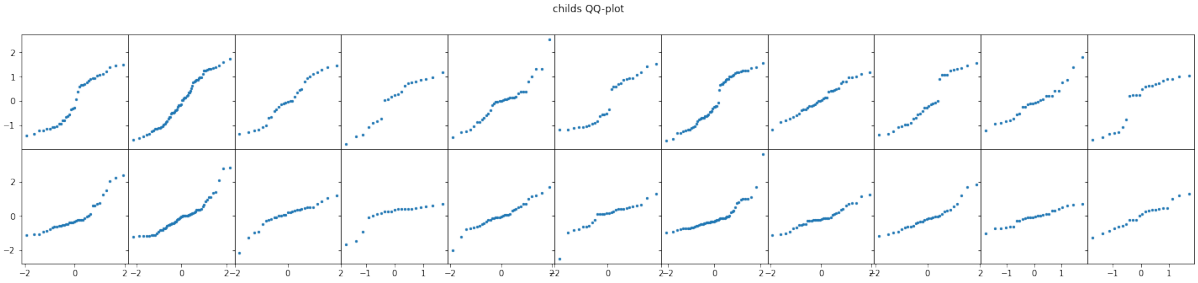

Figure 2.3 Childs

According to the plots (*Figure 2.1*, *Figure 2.2* and *Figure 2.3*) above, there is a bunch of features having right or left skewed distribution in the dataset and the distribution of each feature is not always consistent for every class. For instance, even though *x1* in Mountain class forms a straight line in QQ-plot, it forms 2 peaks (which is also known as a bimodal distribution) in Childs class. This applied to *x7* as well. Also, all the plots above did not show

a certain straight line, some of them are either right-skewed, left skewed, light tailed or heavy tailed. The reason behind this may be the size of dataset for each feature in class is not large enough to generalise a Gaussian distribution. Therefore, the assumption that the numeric data comes from a Gaussian distribution is not always true in this dataset.

Question 3

	Accuracy
Gaussian Naïve Bayes	0.750
KDE Naïve Bayes（sigma=0.1）	0.607
KDE Naïve Bayes（sigma=5）	0.795

Table 3.1 KDE and Gaussian NB accuracy

According to *Table 3.1*, the performance of KDE (sigma=0.5) is slightly higher than Gaussian Naïve Bayes classifier. Naïve Bayes would actually perform well if the assumption of the Gaussian distribution of attribute values is valid. In this dataset, some values do not follow the Gaussian distribution as stated in *Question 2*. KDE does not care about what actual distribution the dataset has, it only needs to compare the test values and train values.

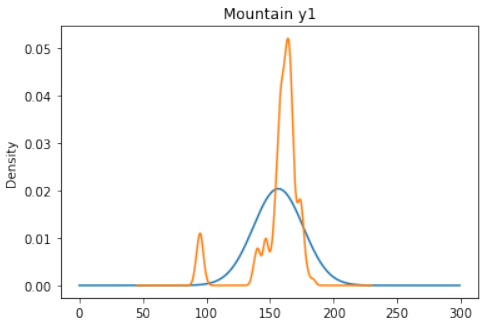

Figure 3.1 Gaussian NB and KDE NB (sigma=0.1)

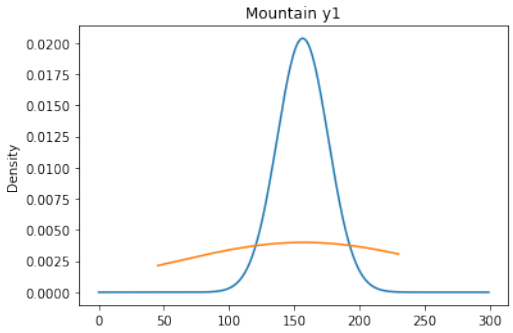

Figure 3.2 Gaussian NB and KDE NB (sigma=5)

As shown in *Figure 3.1* and *Figure 3.2*, Gaussian Naïve Bayes assumes most of the values lie between $\mu \pm 2\sigma$, while KDE creates a smooth curve given a set of data (varies by given sigma). The reason why KDE performed better might be the assumption that the numeric data comes from a Gaussian distribution is not always true.

Question 4

In order to find the best kernel bandwidth parameter, 20 sigmas were tested using random hold-out. The dataset was split into train and test set with 8:2 ratio and was tested 5 times. The results were recorded into the table beneath.

SIGMA	Accuracy 1	Accuracy 2	Accuracy 3	Accuracy 4	Accuracy 5	Average
5	0.834	0.783	0.779	0.775	0.775	0.789
6	0.827	0.788	0.779	0.780	0.777	0.790
7	0.863	0.797	0.786	0.792	0.772	0.797
8	0.841	0.795	0.789	0.787	0.772	0.797
9	0.841	0.802	0.786	0.787	0.777	0.799
10	0.841	0.807	0.789	0.790	0.770	0.799
11	0.836	0.807	0.784	0.790	0.782	0.800
12	0.841	0.802	0.784	0.790	0.784	0.800
13	0.839	0.800	0.786	0.794	0.777	0.799
14	0.836	0.800	0.789	0.797	0.779	0.800
15	0.832	0.795	0.784	0.799	0.777	0.797
16	0.827	0.797	0.784	0.794	0.770	0.794
17	0.827	0.795	0.779	0.792	0.768	0.792
18	0.815	0.800	0.779	0.787	0.765	0.789
19	0.815	0.797	0.779	0.792	0.763	0.789
20	0.810	0.795	0.779	0.790	0.763	0.787
21	0.806	0.792	0.779	0.783	0.763	0.785
22	0.799	0.792	0.774	0.783	0.754	0.780
23	0.799	0.790	0.772	0.780	0.754	0.779
24	0.794	0.788	0.767	0.775	0.754	0.776
25	0.789	0.788	0.767	0.773	0.756	0.775

Figure 4.1 Random Hold-out (SIGMA from 5 to 25)

According to the table above, we can see that the best kernel bandwidth was in the range between 11 to 14. The average accuracy starts to decrease beyond this boundary. The reason why random hold-out is preferred is to avoid over-fitting. As a matter of fact, overfitting occurs when one model is too closely fit to a limited set of data points. In other words, it will perform poorly on other data points. If an arbitrary kernel bandwidth was chosen to predict the test set, an over-fitted result might be obtained. As proof, if only Accuracy 1 was taken, SIGMA=7 will be the best kernel bandwidth for the model, but in fact if the dataset in other way, it will perform worse than other SIGMAs do. As conclusion, a kernel bandwidth between 11 to 14 will be the best choice for this KDE Naïve Bayes model.