Breast Cancer

I trained Breast Cancer dataset using the following 6 classifiers built by me:

1. Naïve Bayes
2. Bayes (Gaussian Class Conditional Density)
3. Bayes (Smoothened Parzen Window)
4. K-Nearest Neighbour Classifier
5. K-means Clustering
6. Logistic Regression

First, I splitted the dataset into training and test sets.

For **Naïve Bayes classifier**, all the features are assumed to be independent. I estimated the class conditional density using gaussian distribution, where the mean and variance were the ML estimates.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

For **Bayes Classifier**, I estimated class conditioned density using two approaches:

- Multivariate Gaussian Distribution:

Multivariate Gaussian distribution:

$$f(x) = \frac{1}{\sqrt{(2\pi)^D \det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

μ: mean, Σ: covariance matrix, D: dim(x)

  where, the μ and ∑ were the ML estimates.

- Parzen Window Estimate:

$$p_n(\vec{X}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\vec{X}-\vec{X}_i}{h_n}\right)$$

  where ϕ was chosen to be a gaussian kernel.

For, remaining three classifiers, I again splitted the training set into training and validation sets.

In **K-NN**, after training, I tried various values of k and chose the one which gave the highest F1-score on the validation data. After that, I used that value of k on the test dataset.

Similar procedure was applied for **K-means clustering**.

Hardest part was to train using **Logistic Regression**. I tried training it many times using different learning rates and weight scale. For lr = 1e-5 and weight scale = 1e-2, I got convergence. But when I ran it again for same lr and weight scale, classifier didn't converge. I ran it many times, and when I got convergence again, I saved the weights and biases into a csv file. Now, I am using these saved weights and biases as initial weights and biases for the logistic classifier.

I computed various Classification Performance Parameters such as Accuracy, Recall, Precision and F1-Score.
Accuracy is obtained by dividing the number of correctly classified labels by total number of labels.
Precision and accuracy were computed with the help of confusion matrix. Shape of matrix is C X C, where C is the number of classes. M_{ij} represents the number of labels for which the actual label is i and the predicted label is j. Precision and recall for class i were obtained by the following formulas:

$$\text{Precision}_i = \frac{M_{ii}}{\sum_j M_{ji}}$$

$$\text{Recall}_i = \frac{M_{ii}}{\sum_j M_{ij}}$$

F1 score for class i is the harmonic mean of precision and recall for class i.

Here are the results obtained:

Classifier	Accuracy	Precision (Averaged over all classes)	Recall (Averaged over all classes)	F1-score (Averaged over all classes)
Naïve Bayes	92.0	92.49	90.31	91.21
Bayes (Gaussian)	95.0	94.87	94.35	94.61
Bayes (Parzen)	92.0	92.49	90.31	91.21
K-NN	96.0	95.71	95.71	95.71
K-Means	90.0	91.94	87.04	88.69
Logistic	94.0	94.753	92.45	93.41

Note that the results are subject to change on running the jupyter notebook again. This is because I am randomly sampling the training, validation and test data from given dataset.

Thus, the order of performance of various classifiers according to F1-score is:

K-Means < Naïve Bayes = Bayes (Gaussian) < Logistic Regression < Bayes (Gaussian) < K-Means.

ZOO

I trained Zoo dataset using the following 5 classifiers built by me:

1. Naïve Bayes
2. Bayes (Smoothened Parzen Window)
3. K-Nearest Neighbour Classifier
4. K-means Clustering
5. Logistic Regression

First, I splitted the dataset into training and test sets.

For **Naïve Bayes classifier**, all the features are assumed to be independent. Since the dataset is discrete, I estimated the class conditioned density of each class and feature using the **Bernoulli distribution**. The parameters were found using ML technique.

I tried to run the **Bayes Classifier** with **multivariate gaussian density** but failed because the covariance matrix was coming out to be singular. I replaced the zeros of covariance matrix by a small number such as 1e-5, but then also the matrix was singular.

Then, I tried the **Bayes Classifier** with class conditioned density estimated using **Parzen Window Approach**. I used the Gaussian Kernel and tuned the hyperparameter "h" accordingly.

Then, I splitted the training set into training and validation sets.

In **K-NN**, after training, I tried various values of k and chose the one which gave the highest F1-score on the validation data. After that, I used that value of k on the test dataset.

Similar procedure was used for **K-means clustering**.

For **Logistic Regression**, I tuned the learning rate and weight scale. And I chose the model which performed best on validation data.

I computed various Classification Performance Parameters such as Accuracy, Recall, Precision and F1-Score.
Accuracy is obtained by dividing the number of correctly classified labels by total number of labels.
Precision and accuracy were computed with the help of confusion matrix. Shape of matrix is C X C, where C is the number of classes. M_{ij} represents the number of labels for which the actual label is i and the predicted label is j. Precision and recall for class i were obtained by the following formulas:

$$\text{Precision}_i = \frac{M_{ii}}{\sum_j M_{ji}}$$

$$\text{Recall}_i = \frac{M_{ii}}{\sum_j M_{ij}}$$

F1 score for class i is the harmonic mean of precision and recall for class i.

Here are the results obtained:

Classifier	Accuracy	Precision (Averaged over all classes)	Recall (Averaged over all classes)	F1-score (Averaged over all classes)
Naïve Bayes	85.71	97.14	80.953	86.03
Bayes (Parzen)	100	100	100	100
K-NN	95.24	95.24	98.81	96.52
K-Means	90.48	90.48	91.667	88.90
Logistic	95.24	95.238	95.238	94.28

Note that the results are subject to change on running the jupyter notebook again. This is because I am randomly sampling the training, validation and test data from given dataset.

Thus, the order of performance of various classifiers according to F1-score is:

Naïve Bayes < K-Means < Logistic Regression < K-NN < Bayes (Parzen).

Statlog (Heart)

I trained Statlog (Heart) dataset using the following 6 classifiers built by me:

1. Naïve Bayes
2. Bayes (Gaussian Class Conditional Density)
3. Bayes (Smoothened Parzen Window)
4. K-Nearest Neighbour Classifier
5. K-means Clustering
6. Logistic Regression

First, I splitted the dataset into training and test sets. I performed the feature scaling and mapped all the features to [0, 1] range.

For **Naïve Bayes classifier**, all the features are assumed to be independent. I estimated the class conditional density using gaussian distribution, where the mean and variance were the ML estimates.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

For **Bayes Classifier**, I estimated class conditioned density using two approaches:

- Multivariate Gaussian Distribution:

Multivariate Gaussian distribution:

$$f(x) = \frac{1}{\sqrt{(2\pi)^D \det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

μ: mean, Σ: covariance matrix, D: dim(x)

  where, the μ and ∑ were the ML estimates.

- Parzen Window Estimate:

$$p_n(\vec{X}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\vec{X}-\vec{X}_i}{h_n}\right)$$

  where ϕ was chosen to be a gaussian kernel.

For, remaining three classifiers, I again splitted the training set into training and validation sets.

In **K-NN**, after training, I tried various values of k and chose the one which gave the highest F1-score on the validation data. After that, I used that value of k on the test dataset.

Similar procedure was applied for **K-means clustering**.

For **Logistic Regression**, I tuned the learning rate and weight scale. And I chose the model which performed best on validation data.

I computed various Classification Performance Parameters such as Accuracy, Recall, Precision and F1-Score.
Accuracy is obtained by dividing the number of correctly classified labels by total number of labels.
Precision and accuracy were computed with the help of confusion matrix. Shape of matrix is C X C, where C is the number of classes. M_{ij} represents the number of labels for which the actual label is i and the predicted label is j. Precision and recall for class i were obtained by the following formulas:

$$\text{Precision}_i = \frac{M_{ii}}{\sum_j M_{ji}}$$

$$\text{Recall}_i = \frac{M_{ii}}{\sum_j M_{ij}}$$

F1 score for class i is the harmonic mean of precision and recall for class i.

Here are the results obtained:

Classifier	Accuracy	Precision (Averaged over all classes)	Recall (Averaged over all classes)	F1-score (Averaged over all classes)
Naïve Bayes	94.0	92.88	94.03	93.41
Bayes (Gaussian)	86.0	84.74	83.69	84.17
Bayes (Parzen)	88.0	89.40	83.77	85.71
K-NN	86.0	85.91	82.26	83.64
K-Means	74.0	70.96	70.320	70.60
Logistic	84.0	82.17	82.17	82.17

Note that the results are subject to change on running the jupyter notebook again. This is because I am randomly sampling the training, validation and test data from given dataset.

Thus, the order of performance of various classifiers according to F1-score is:

K-Means < Logistic Regression < K-NN < Bayes (Gaussian) < Bayes (Parzen) < Naïve Bayes.

HEPATITIS

I trained Hepatitis dataset using the following 5 classifiers built by me:

1. Naïve Bayes
2. Bayes (Smoothened Parzen Window)
3. K-Nearest Neighbour Classifier
4. K-means Clustering
5. Logistic Regression

First, I splitted the dataset into training and test sets. The data consists of both continuous as well as discrete valued features. I filled the missing values of features by the mean of that feature. In case of discrete valued features, I rounded the mean to nearest integer and filled at the appropriate places.

Also, I deleted 19th column because it had a lot of missing values. Then, I performed the feature scaling and mapped all the features to [0, 1] range.

For **Naïve Bayes classifier**, all the features are assumed to be independent. I faced a problem in estimating the class conditioned density using gaussian distribution as standard deviation was coming out to be zero for some classes and features. Also, I couldn't use the discrete Naïve Bayes classifier built for Zoo dataset as there are some continuous valued features. So, I built Naïve Bayes classifier using bins. I divided the range of each feature into some number of intervals. After that, I found the probability of each class and each feature in all bins.

I tried to run the **Bayes Classifier** with **multivariate gaussian density** but failed because the covariance matrix was coming out to be singular. I replaced the zeros of covariance matrix by a small number such as 1e-5, but then also the matrix was singular.

Then, I tried the **Bayes Classifier** with class conditioned density estimated using **Parzen Window Approach**. I used the Gaussian Kernel and tuned the hyperparameter "h" accordingly.

Then, I splitted the training set into training and validation sets.

In **K-NN**, after training, I tried various values of k and chose the one which gave the highest F1-score on the validation data. After that, I used that value of k on the test dataset.

Similar procedure was used for **K-means clustering**.

For **Logistic Regression**, I tuned the learning rate and weight scale. And I chose the model which performed best on validation data.

I computed various Classification Performance Parameters such as Accuracy, Recall, Precision and F1-Score.
Accuracy is obtained by dividing the number of correctly classified labels by total number of labels.
Precision and accuracy were computed with the help of confusion matrix. Shape of matrix is C X C, where C is the number of classes. M_{ij} represents the number of labels for which the actual label is i and the predicted label is j. Precision and recall for class i were obtained by the following formulas:

$$\text{Precision}_i = \frac{M_{ii}}{\sum_j M_{ji}}$$

$$\text{Recall}_i = \frac{M_{ii}}{\sum_j M_{ij}}$$

F1 score for class i is the harmonic mean of precision and recall for class i.

Here are the results obtained:

Classifier	Accuracy	Precision (Averaged over all classes)	Recall (Averaged over all classes)	F1-score (Averaged over all classes)
Naïve Bayes	85	92.5	50	45.94
Bayes (Parzen)	85	92.5	50	45.94
K-NN	90	83.33	73.53	77.14
K-Means	90	79.69	87.25	82.68
Logistic	82.5	42.30	48.53	45.20

Note that the results are subject to change on running the jupyter notebook again. This is because I am randomly sampling the training, validation and test data from given dataset.

Thus, the order of performance of various classifiers according to F1-score is:

Logistic Regression < Naïve Bayes = Bayes (Parzen) < K-NN < K-Means

HABERMAN

I trained Haberman dataset using the following 5 classifiers built by me:

1. Naïve Bayes
2. Bayes (Smoothened Parzen Window)
3. K-Nearest Neighbour Classifier
4. K-means Clustering
5. Logistic Regression

First, I splitted the dataset into training and test sets.

For **Naïve Bayes classifier**, all the features are assumed to be independent. I estimated the class conditional density using gaussian distribution, where the mean and variance were the ML estimates.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

I tried to run the **Bayes Classifier** with **multivariate gaussian density** but failed because the covariance matrix was coming out to be singular. I replaced the zeros of covariance matrix by a small number such as 1e-5, but then also the matrix was singular.

Then, I tried the **Bayes Classifier** with class conditioned density estimated using **Parzen Window Approach**. I used the Gaussian Kernel and tuned the hyperparameter "h" accordingly.

Then, I splitted the training set into training and validation sets.

In **K-NN**, after training, I tried various values of k and chose the one which gave the highest F1-score on the validation data. After that, I used that value of k on the test dataset.

Similar procedure was used for **K-means clustering**.

For **Logistic Regression**, I tuned the learning rate and weight scale. And I chose the model which performed best on validation data.

I computed various Classification Performance Parameters such as Accuracy, Recall, Precision and F1-Score.
Accuracy is obtained by dividing the number of correctly classified labels by total number of labels.
Precision and accuracy were computed with the help of confusion matrix. Shape of matrix is C X C, where C is the number of classes. M_{ij} represents the number of labels for which the actual label is i and the predicted label is j. Precision and recall for class i were obtained by the following formulas:

$$\text{Precision}_i = \frac{M_{ii}}{\sum_j M_{ji}}$$

$$\text{Recall}_i = \frac{M_{ii}}{\sum_j M_{ij}}$$

F1 score for class i is the harmonic mean of precision and recall for class i.

Here are the results obtained:

Classifier	Accuracy	Precision (Averaged over all classes)	Recall (Averaged over all classes)	F1-score (Averaged over all classes)
Naïve Bayes	88	93.61	66.66	71.59
Bayes (Parzen)	76	59.35	59.35	59.35
K-NN	76	59.35	59.35	59.35
K-Means	86	92.71	61.11	64.25
Logistic	84	91.84	55.55	55.55

Note that the results are subject to change on running the jupyter notebook again. This is because I am randomly sampling the training, validation and test data from given dataset.

Thus, the order of performance of various classifiers according to F1-score is:

Logistic Regression < K-NN = Bayes (Parzen) < K-Means < Naïve Bayes.