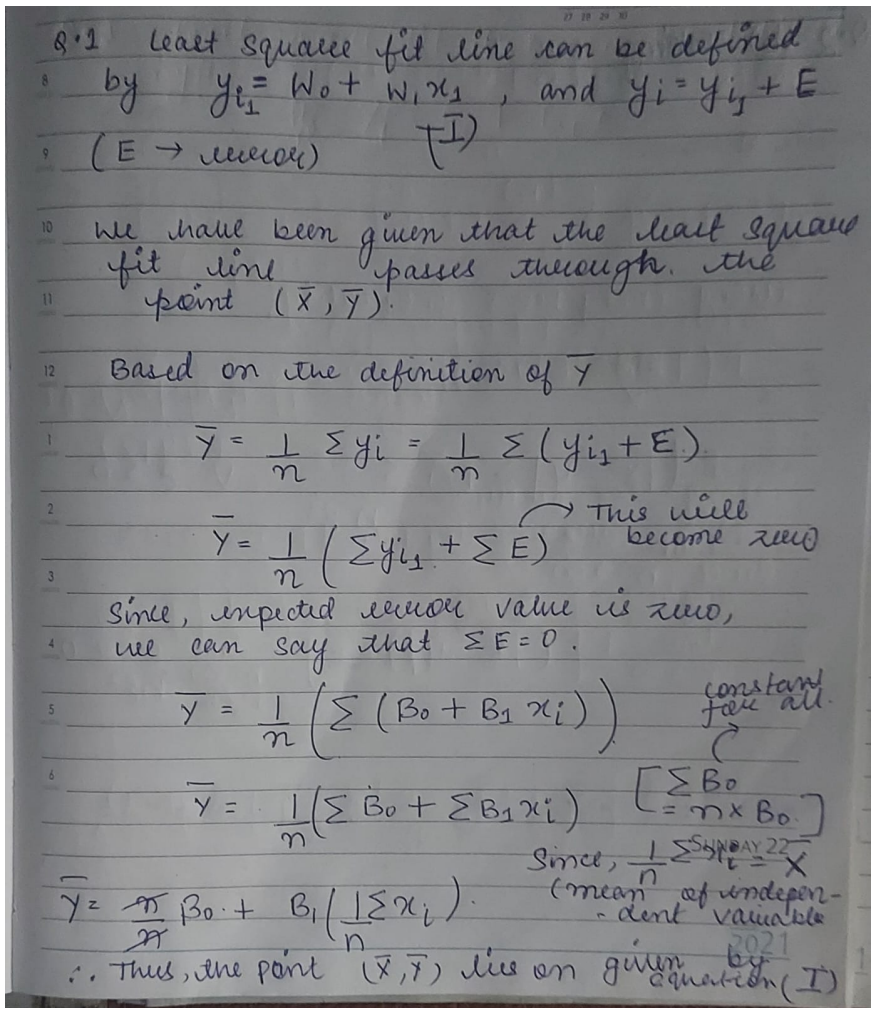
Assignment 1 ML

Vaibhav Jaiswal (2020547)

Question 1:

A) The proof can be viewed in the picture below :



B) Thus, the best way to analyse how the correlation will function is to take an example. On taking an example of variables X, Y and Z. Now, Y is highly correlated to X and Z. But, X and Z are not relatively correlated to each other. Taking the example, that the correlation between Y and X is 0.7 and correlation between Y and Z is 0.75. Between X and Z, correlation is 0.05.

C)

PseudoCode :

1. Take a empty list x
2. Take a for loop for values ranging from 0, python
3. the value inside y will be generated randomly using random function in
4. Apply gaussian probability distribution formula and append it onto the list.
5. initialise another variable n
6. take another loop, and add the index of that loop into variable n. n = n + x[_].
7. We return n at the end.

Q.1 (c) WEAK LAW OF LARGE NUMBERS:

Given n independent and identical distribution, $X_1, X_2 \ldots X_n$. Standard error σ, $E[X_R] = \mu$

$$\bar{X} = \frac{(X_1 + X_2 + X_3 \ldots X_n)}{n}$$

$$\lim_{n \to \infty} P[(|\bar{X} - \mu|) \geq E] = 0.$$

Let $X_1, X_2, X_3 \ldots X_n$ be random variables.

$E[X_i] = \mu$ $\therefore \bar{X} = \frac{(X_1 + X_2 + X_3 \ldots X_n)}{n}$

$$\bar{X} = \frac{n\mu}{n} = \mu$$

$var(X_i) = \sigma^2$

$$M_n = \frac{1}{n} \sum_{i=1}^{n} X_i$$

$$E[M_n] = E\left[\frac{1}{n} \sum_{i=1}^{n} X_i\right] = \frac{1}{n} E[X_1 + X_2 \ldots + X_n)$$

$$= \frac{n\mu}{n} = \mu.$$

2021

$$var(\bar{X}) = \frac{var(X_1 + X_2 \ldots X_n)}{n^2}$$

$$= \frac{\sigma(n)}{n^2} = \frac{\sigma^2}{n}$$

using chebyshev's inequality,

$$P(|\bar{X} - E[M_n]| \geq \varepsilon) \leq \frac{var(\bar{X})}{E^2} \quad \forall E > 0.$$

$$P(|\bar{X} - \mu| \geq E) \leq \frac{\sigma^2}{n\varepsilon^2}, \quad \forall \varepsilon > 0.$$

Since, $n \to \infty$.

$$\lim_{n \to \infty} P[(|\bar{X} - \mu|) \geq \varepsilon] = \lim_{n \to \infty} \frac{\sigma^2}{n\varepsilon^2} = 0.$$

D)

THURSDAY AUGUST 26

Q.1

(d) $y = w^T x + \varepsilon$ (as taught in class).

$\varepsilon \sim N(0, \sigma^2)$.

(noise follows normal distribution).

$y \sim N(w^T x, \sigma^2)$.

Probability of gaussian distribution, when calculating probability of each response variable.

$$P(y|x, w) = N(y | w^T x, \sigma^2).$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-1/2\left[\frac{(y - w^T x)^2}{\sigma^2}\right]}$$

Given data

$D = \{(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)\}$.

Assuming, gaussian prior distribution over the weight w.

$$P(w) = N(w | 0, \lambda^{-1} I)$$

$$= \frac{1}{\sqrt{2\pi D}} e^{-1/2[\lambda w^T w]}$$

$$\log P(w|D) = \log \frac{P(w) P(D/w)}{P(D)} = \log P(w) + \log P(D/w) - \log P(D)$$

Maximum - A - Posterior solution:

(a)

$$P(w|\vec{x}_i y_i) = \sum_i^n \frac{P(y_i|\vec{x}_i, w)\,P(w)}{P(y_i|X)}$$

$$\underset{w}{\text{argmax}} \prod_{i=1}^{n} P(w|\vec{x}_i, y_i)$$

$$= \underset{w}{\text{argmax}} \frac{\prod P(y_i|\vec{x}_i w)\,P(w)}{P(y|x)}$$

Taking log - posterior:

$$w_{MAP} = \underset{w}{\text{argmax}} \log P(w|D)$$

$$= \text{argmax} \left\{ \sum \log \left(\frac{1}{\sqrt{2\pi}\sigma^2} e^{\left(\frac{-1}{2} \left(y_i - \frac{w_i^T x_i}{\sigma} \right)^2 \right)} \right) + \log(P(w)) \right\}$$

$$= \text{argmax} \left\{ \sum \frac{1}{2} \left(y_i - \frac{w^T x}{\sigma} \right)^2 + \frac{1}{2} \log(2\pi) - \frac{\lambda}{2} w^T w \right.$$

Simply ignoring Independent terms

$$= \text{argmin} \frac{1}{2\sigma^2} \sum_{n=1}^{\infty} (y_n - w^T x_n)^2 + \frac{\lambda}{2} w^T w$$

Question 2 :

Part A →

The optimal value of K is at K = 5. The values of K with the Average RMSE can be seen in the table given below.

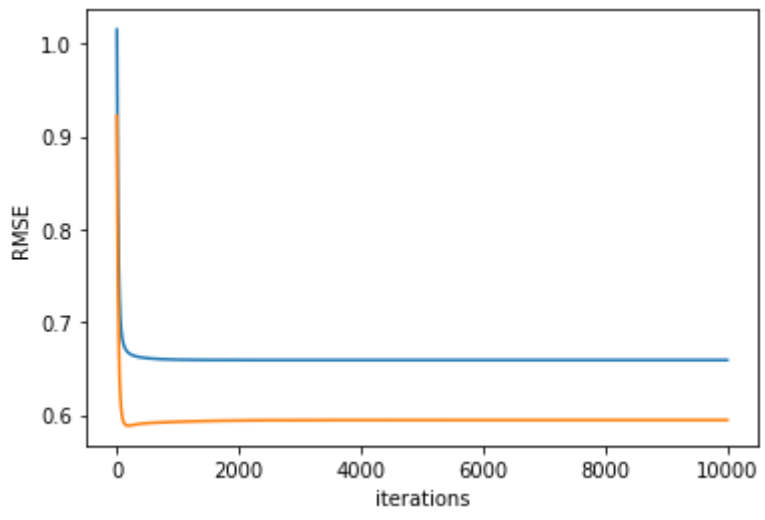
K	RMSE Means for K
2	1.0000046405180678
3	0.9999995260347649
4	0.9999995260347649
5	0.9999995260347649

```
RMSEKvalues = []
for i in range(2,6):
    print(i)
    RMSEKvalues.append(train(kfold(df,i),i))

print(RMSEKvalues)
```

```
.0000046405180678, 0.9999995260347649, 1.0001641050721342, 0.9995186621625347]
```

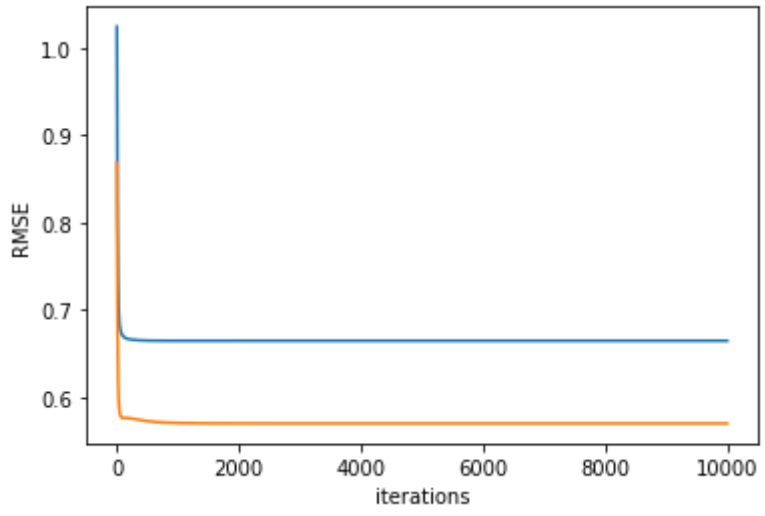
Part B →



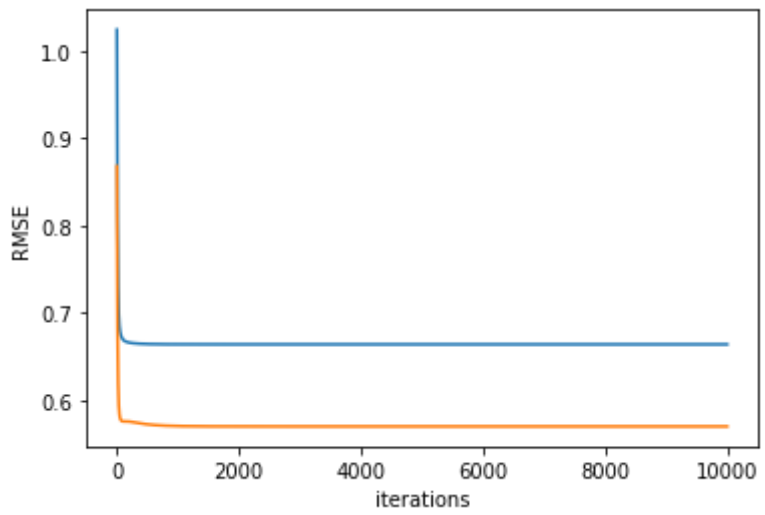
Part C →

Tried different values of alpha, the values of alpha (or a comes) to be equal to 0.01

Lasso Regression :



Ridge Regression :



Part D →

The normal form directly gives the optimal parameter $wparmeter$ for the equation. This will be informed for the optimal value of K. The optimal Value is 5.

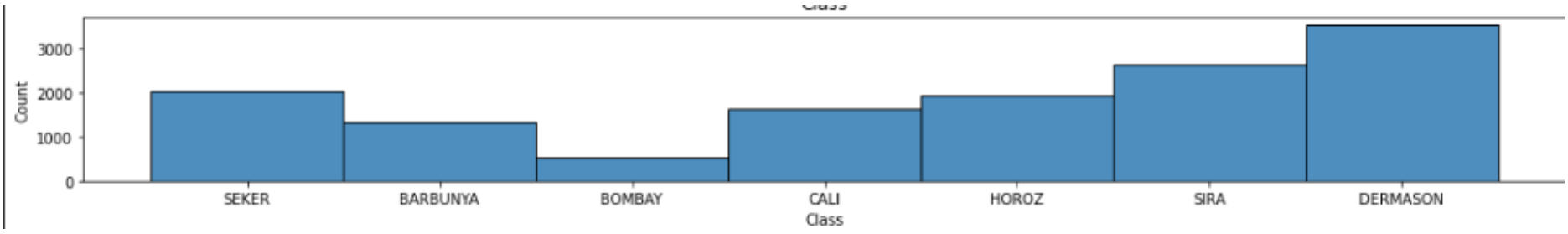
For the 5 validation sets generated using K = 5, we get the set as shown below.

```
[1.045289278556397,
 1.0738903617321636,
 1.1330584321885546,
 1.259860933441022,
 1.0302762014828666]
```

Question 3 :

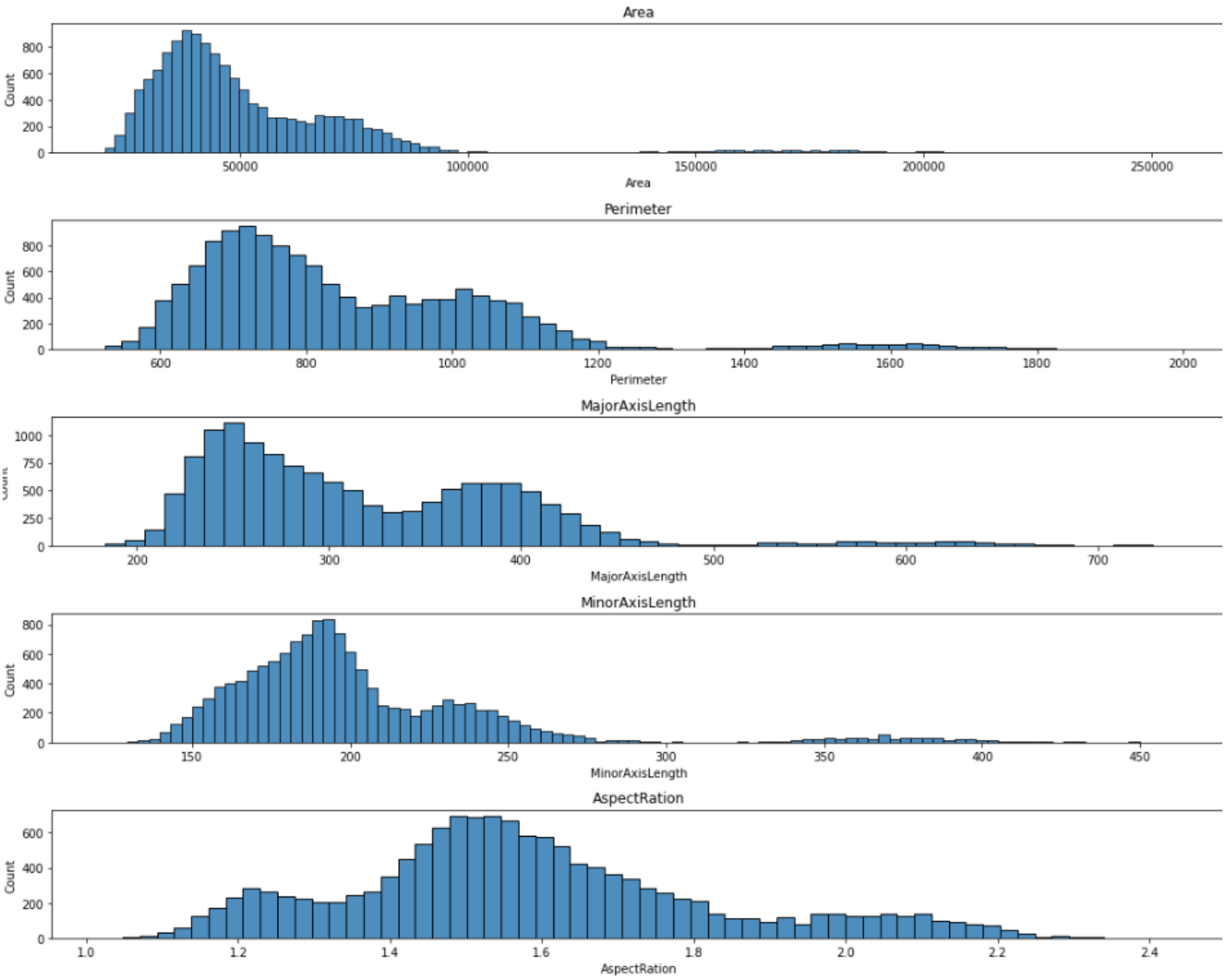
Part A →

The class distribution for different dry beans set was plotted



It was observed that the dermason class of dry beans had a greater number of count in the dataset. The count for the different types is not equally divided, and is unequally distributed. Other than this more distributions were made on other features

```
'Area' 'Perimeter' 'MajorAxisLength' 'MinorAxisLength' 'AspectRation'
 'Eccentricity' 'ConvexArea' 'EquivDiameter' 'Extent' 'Solidity'
 'roundness' 'Compactness' 'ShapeFactor1' 'ShapeFactor2' 'ShapeFactor3'
 'ShapeFactor4']
```



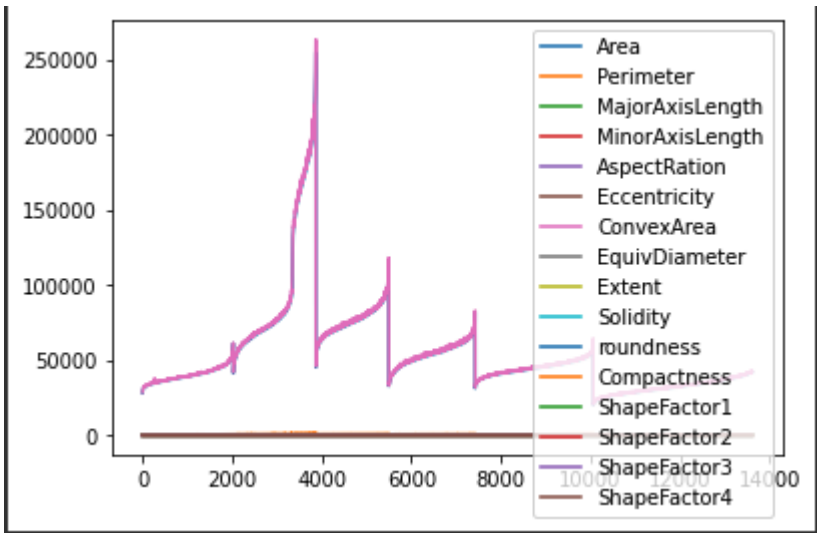
The observation was that for most of the classes the values are distribution is skewed in nature. Data is not equally distributed.

Part B (Exploratory Data Analysis)→

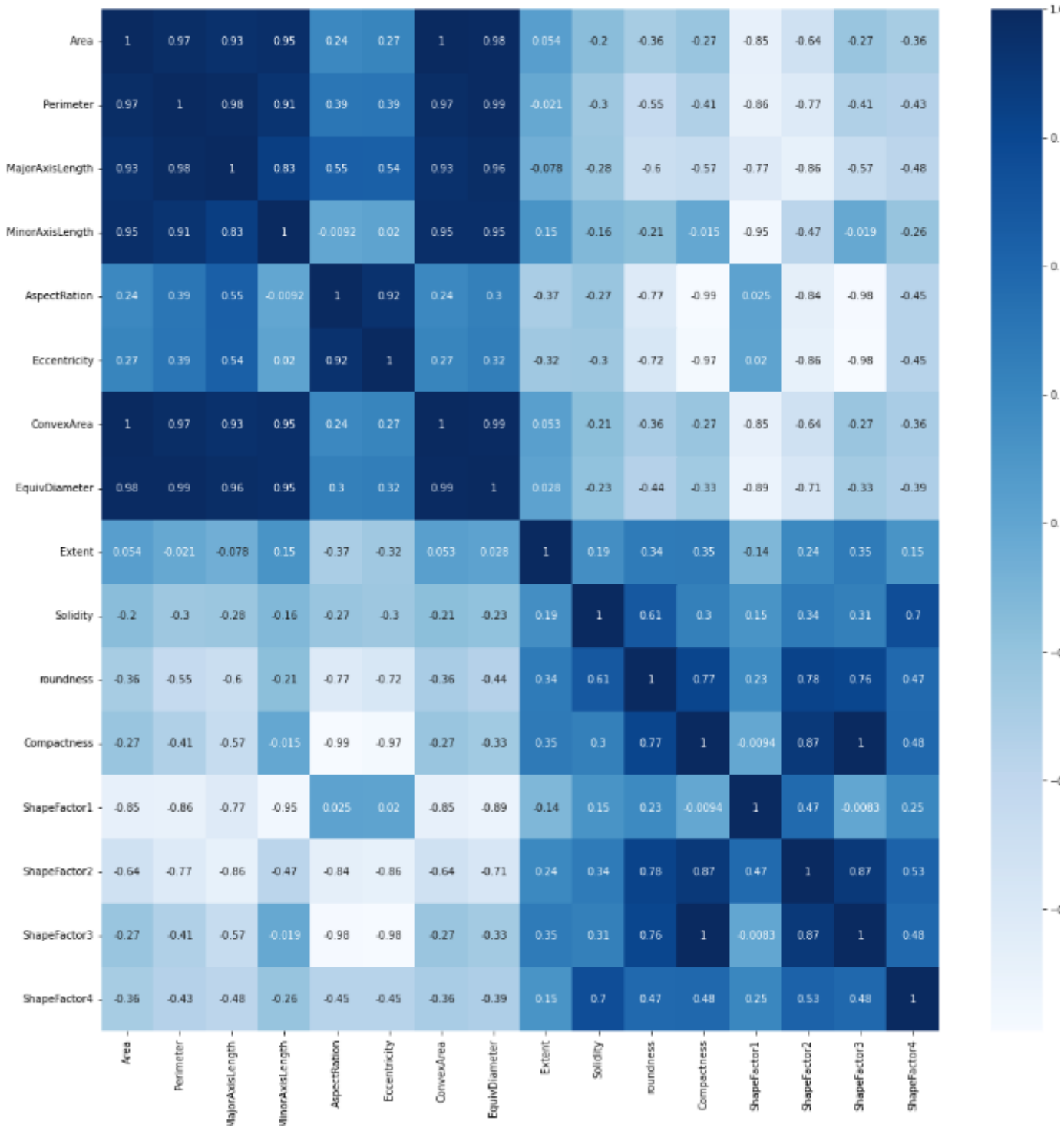
Data Insight 1 : Plotting →

On plotting the data, it can be seen that the data is skewed, and is not consistent. The convex area and area, are related can also be also observed from the graph itself. The plotting for various graphs are consistent as

well. The max values can be seen from the graph as well.

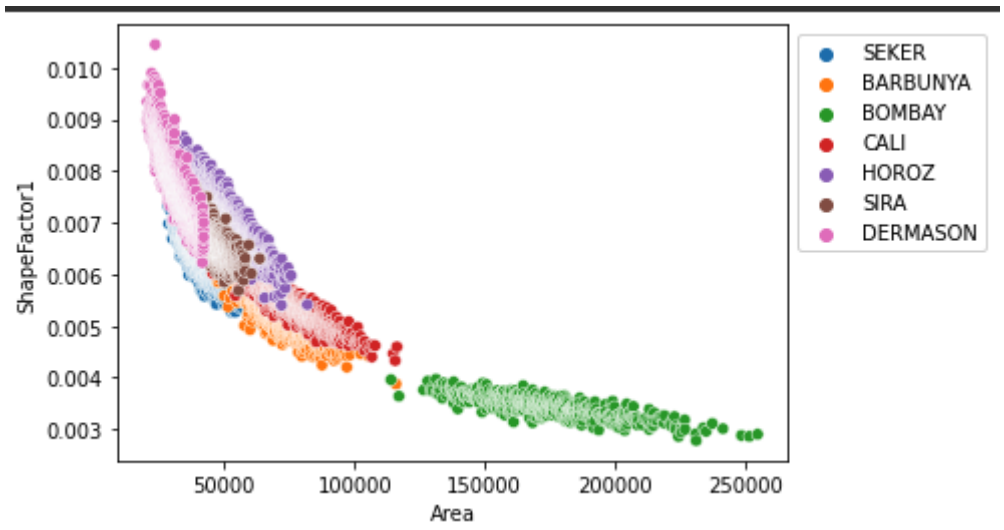


Data Insight 2 : Correlation Map →



- On Using heatmap, correlation in the data can be clearly observed. It is well understood that features like convex area, and area will be correlated. Features like perimeter, major axis length, convex area, equivdiameter are related to each other. Other than this, shapefactor 3, 2 Shape and Compactness have a strong correlation. Roundness and Shape factor 2,3 also have a strong correlation.

- Similar observations became more apparent for specific conditions when using scatter plot. This also shows it in respect of various types of classes.

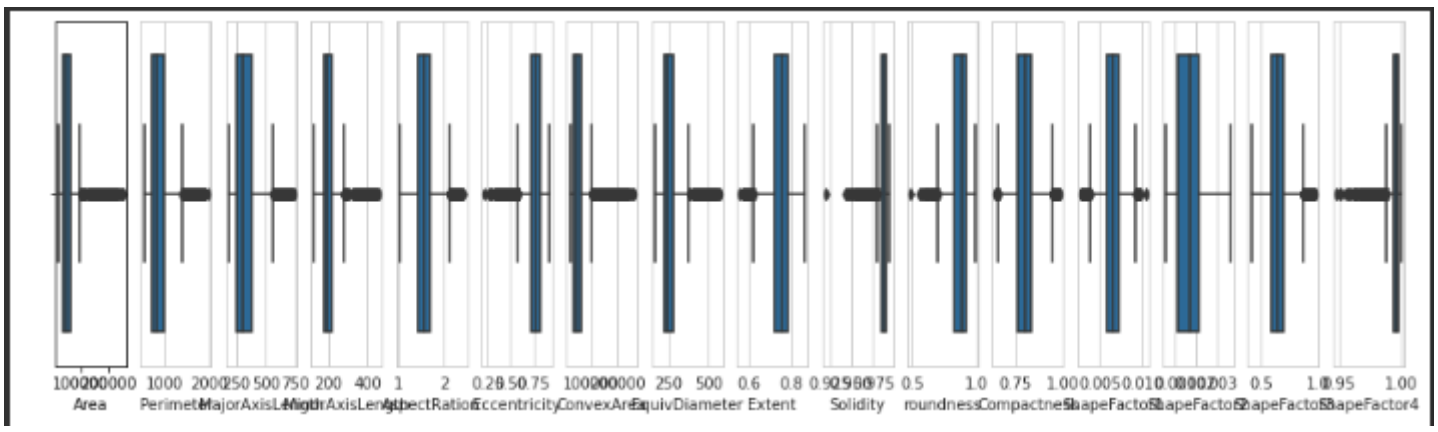


Data Insight 3 : Plotting Data Distribution →

Using Histogram, we can see the different distributions of data and its classes. It is observed that the data is skewed in nature. Specifically shape factor 4 and more are highly skewed in nature.

Data Insight 4 : Box Plot →

The box plot shows that the data has a massive number of outliers, which is represented by the black dots in the graph. The features like extent, still have reduced number of outliers, whereas things like shape-factor have massive number of features.

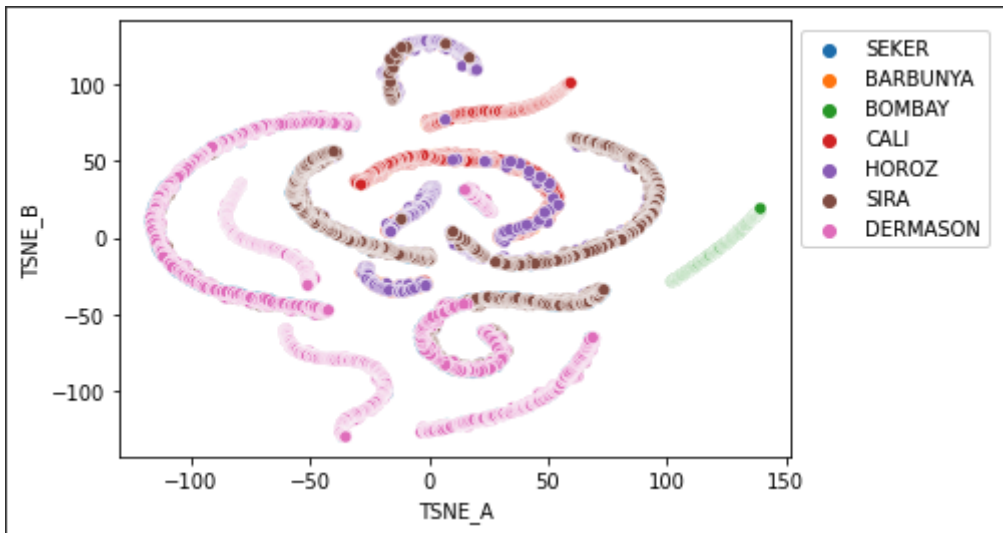


Data Insight 5 : Exploring the classes and data →

Using commands like df.head and df.info(), df.shape, on the command, what we get the data, which gives things like the dimensions. The non-null value reflects that there are NO MISSING VALUE in the dataset. It can also be observed that all the values are in the form of Dtpe of float64, things like area and convex area are in the form of int64. And one is in the form of object. The Data is of the dimensions 13611 entries to 17 columns. Memory usage is 1.8+ MB.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13611 entries, 0 to 13610
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Area             13611 non-null  int64
 1   Perimeter        13611 non-null  float64
 2   MajorAxisLength  13611 non-null  float64
 3   MinorAxisLength  13611 non-null  float64
 4   AspectRation     13611 non-null  float64
 5   Eccentricity     13611 non-null  float64
 6   ConvexArea       13611 non-null  int64
 7   EquivDiameter    13611 non-null  float64
 8   Extent           13611 non-null  float64
 9   Solidity         13611 non-null  float64
 10  roundness        13611 non-null  float64
 11  Compactness      13611 non-null  float64
 12  ShapeFactor1     13611 non-null  float64
 13  ShapeFactor2     13611 non-null  float64
 14  ShapeFactor3     13611 non-null  float64
 15  ShapeFactor4     13611 non-null  float64
 16  Class            13611 non-null  object
dtypes: float64(14), int64(2), object(1)
memory usage: 1.8+ MB
```

Part C (T-SNE)→



On using a perplexity value of 50 (since the data is decently large), and the for a number iterations = 4000. The data is separable, with maximum of that of dermason and bombay. The values still have outliers, that overlap with other classes of dry beans

Part D (Naive-Bayes Algorithm)→

Using the sklearn, we train different naive bayes algorithms, and then report precision, recall, f-score and accuracy of different values. Note that we have used the 80:20 split itself.

For Gaussian Naive Bayes :

	precision	recall	f1-score	support
BARBUNYA	0.66	0.52	0.59	246
BOMBAY	1.00	1.00	1.00	101
CALI	0.69	0.81	0.75	325
DERMASON	0.89	0.83	0.86	716
HOROZ	0.81	0.78	0.80	404
SEKER	0.69	0.72	0.71	406
SIRA	0.73	0.79	0.76	525
accuracy			0.77	2723
macro avg	0.78	0.78	0.78	2723
weighted avg	0.78	0.77	0.77	2723

For Multinomial Naive Bayes:

	precision	recall	f1-score	support
BARBUNYA	0.64	0.65	0.64	246
BOMBAY	1.00	1.00	1.00	101
CALI	0.77	0.77	0.77	325
DERMASON	0.88	0.83	0.85	716
HOROZ	0.85	0.79	0.82	404
SEKER	0.76	0.76	0.76	406
SIRA	0.72	0.81	0.76	525
accuracy			0.79	2723
macro avg	0.80	0.80	0.80	2723
weighted avg	0.80	0.79	0.79	2723

For Bernoulli Naive Bayes :

	precision	recall	f1-score	support
BARBUNYA	0.00	0.00	0.00	246
BOMBAY	0.00	0.00	0.00	101
CALI	0.00	0.00	0.00	325
DERMASON	0.26	1.00	0.42	716
HOROZ	0.00	0.00	0.00	404
SEKER	0.00	0.00	0.00	406
SIRA	0.00	0.00	0.00	525
accuracy			0.26	2723
macro avg	0.04	0.14	0.06	2723
weighted avg	0.07	0.26	0.11	2723

Since, the data is categorical and more skewed in nature. The gaussian naive bayes is able to give some good accuracy of 77 percent. Since, the data is multinomial (different classes of beans have been given), hence using a Bernoulli distribution is not able to give a good accuracy (26 percent) (barely above baseline random) The precision is only present for dermason due to its larger value. Multinomial naive bayes performs the best with 79 percent accuracy.

Part E (Principal Component Analysis)→

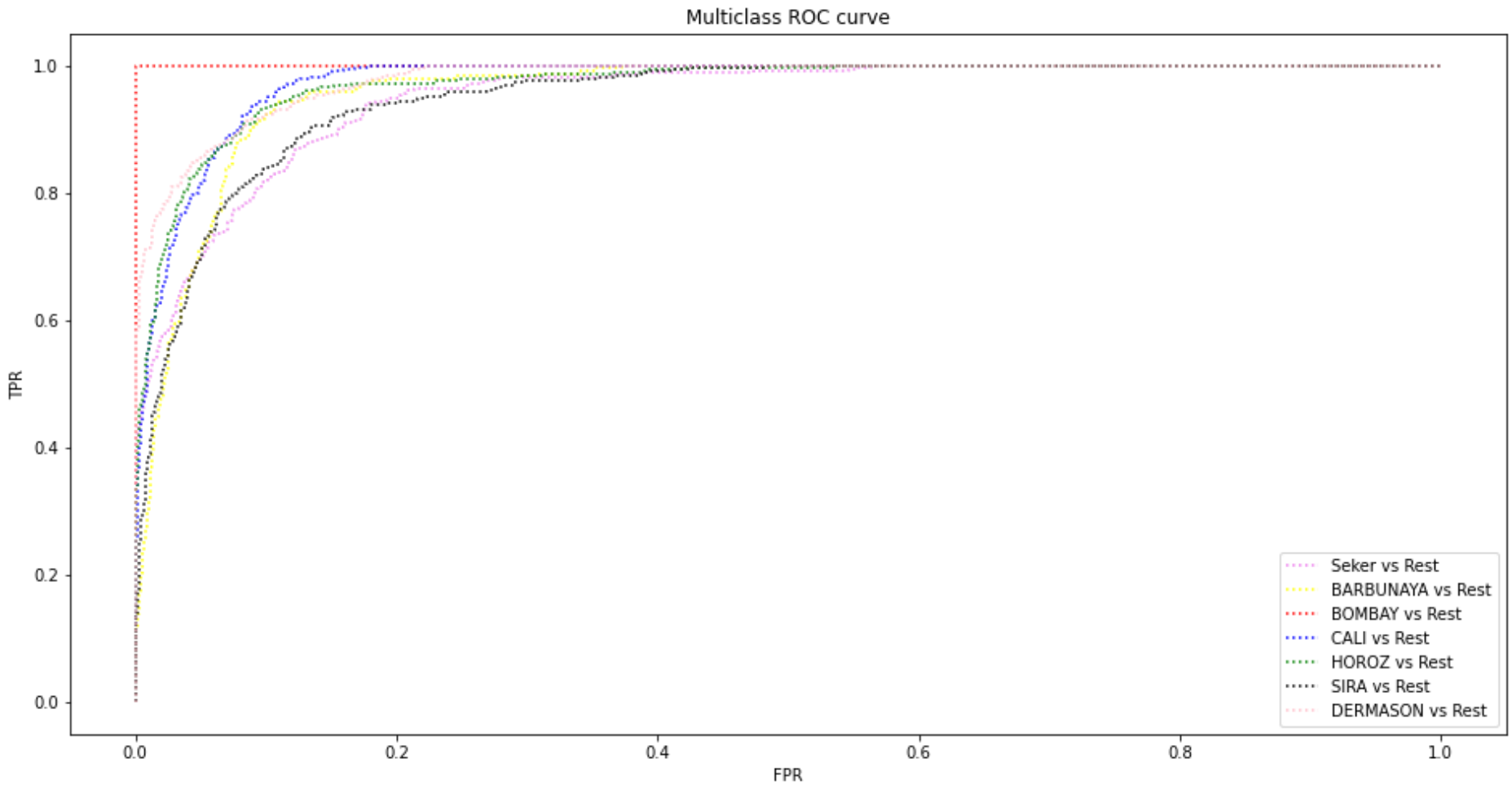
We can do the PCA for different components, 4,6,8,10,12. This is shown in this way : Using the gaussian naive bayes distribution, we will make the comparison.

S.No.	N Components	Accuracy Score (round off)	Precision Score	Recall Score	F1 Score
1	4	89.05	0.89	0.89	0.89
2	6	88.68	0.89	0.88	0.88
3	8	89.05	0.89	0.89	0.89
4	10	89.4	0.89	0.89	0.89
5	12	89.9	0.89	0.89	0.89

Accuracy has increased steadily as soon as the number of iterations are increased, with a drop and then an increase from 4 → 6 → 12 features. This will give the values required.

Part F (ROC-AUC curve)→

Since, ROC_AUC curves helps us figure the complete fitting of the classification model. We can say that the TPR values are higher than FPR values for the given scenario, that is why we get a ROC like that. (FPR→ false positive rate) (TPR→ true positive rates)



Part G (Logistic Regression) →

The number of iterations used were = 10,000. This was so that the model gets enough iterations to reach the point of convergence. The alpha (learning rate is selected automatically). The algorithm was also throwing “failed to converge error on using lower number of iterations.” Increasing the iterations increases accuracy, till a single point. Increasing its enormously, does not return comparatively greater amount of accuracy.

The class chosen was multi-class due to a simple reason that the the multinomial loss is minimised across the entire probability distribution. The newton_cg solver gives a better accuracy with the l2 loss, function. l1 loss would have performed better, since the data has higher amount of outliers, but the newton_cg covers up by giving better values in total. The accuracy comes out to be 92 percent. Which is far better than the naive bayes models trained on data on which PCA was used.

Model has greater accuracy than gaussian naive bayes, with 2

percent better accuracy. This can be credited to logistic regression and sigmoid function being used in it

	precision	recall	f1-score	support
BARBUNYA	0.95	0.91	0.93	246
BOMBAY	1.00	1.00	1.00	101
CALI	0.93	0.94	0.94	325
DERMASON	0.92	0.91	0.91	716
HOROZ	0.96	0.96	0.96	404
SEKER	0.94	0.93	0.94	406
SIRA	0.83	0.87	0.85	525
accuracy			0.92	2723
macro avg	0.93	0.93	0.93	2723
weighted avg	0.92	0.92	0.92	2723

S.No.	Penalty	Solver	multi_class	Accuracy
1	l2	newton_cg	mutlinomial	92
2	l1	liblinear	—	90.1
3	l2	sag	—	67
4	l2	saga	—	62