UMC 203 Assignment 1

Debanjan Saha
23607

Question 1

Part 1.1

Estimation is a core branch of statistics. The FLD 'estimates' the class conditional mean and variance with the samples in the dataset. We study how the estimates change with the number of samples chosen. For each class, study the change in the estimates with $n = 50, 100, 500, 1000, 2000, 4000$. Report this with a plot of the l_2 norms of the mean vectors for each class and the Frobenius Norm of the covariance matrices for each class.

Answer

The following tables show the l_2 norms of the mean vectors and the Frobenius norms of the covariance matrices for each class for different values of n:

Mean norms of each class for different n:

	c0	c1	c2	c3
n=50	24.3725	24.4161	26.0036	25.0284
n=100	25.3359	24.5907	26.2031	25.0048
n=500	24.8841	23.9361	25.3278	25.3108
n=1000	24.4111	23.9369	25.6027	25.2434
n=2000	24.3727	24.0799	25.6493	25.0484
n=4000	24.3547	24.1941	25.6513	24.946

Covariance norms of each class for different n:

	c0	c1	c2	c3
n=50	79.2707	128.033	92.0379	89.4493
n=100	88.4562	105.289	90.1574	81.981
n=500	87.7814	98.7399	84.0281	89.5576
n=1000	91.1905	103.974	86.1804	87.9867
n=2000	92.4639	105.116	88.024	88.267
n=4000	94.2818	105.772	87.6579	88.1412

The **figures 1** and **2** show the plots of the l_2 norms of the mean vectors and the Frobenius norms of the covariance matrices for each class:

From the plots, we can observe that as n increases, the l_2 norms of the mean vectors and the Frobenius norms of the covariance matrices tend to stabilize which indicates that the estimates become more consistent as the sample size increases.

Part 1.2

You will now implement the multi-class FLD for the provided dataset.

(a) Find the weights of the FLD for each of the 4 classes with $n = 2500, 3500, 4000, 4500, 5000$ samples with 20 different subsets (except 5000). Plot box plots for the multi-class objective value for different values of n. Plot the projection of points of each class onto the 'linear' discriminant. Based on the above plot, choose suitable choices for the threshold for each classifier. Report the chosen thresholds and draw insight on the projected points.

(b) (2 marks) Report the accuracy of the model on the test set with the classifiers with different number of samples. You will not be graded solely based on the performance of the classifier, but on the soundness and correctness of the design of the classifier based on the multi-class FLD.

Answer (a)

The box plot is in **figure 3**. The projections of train and test data are in **figure 4** and **5** respectively.

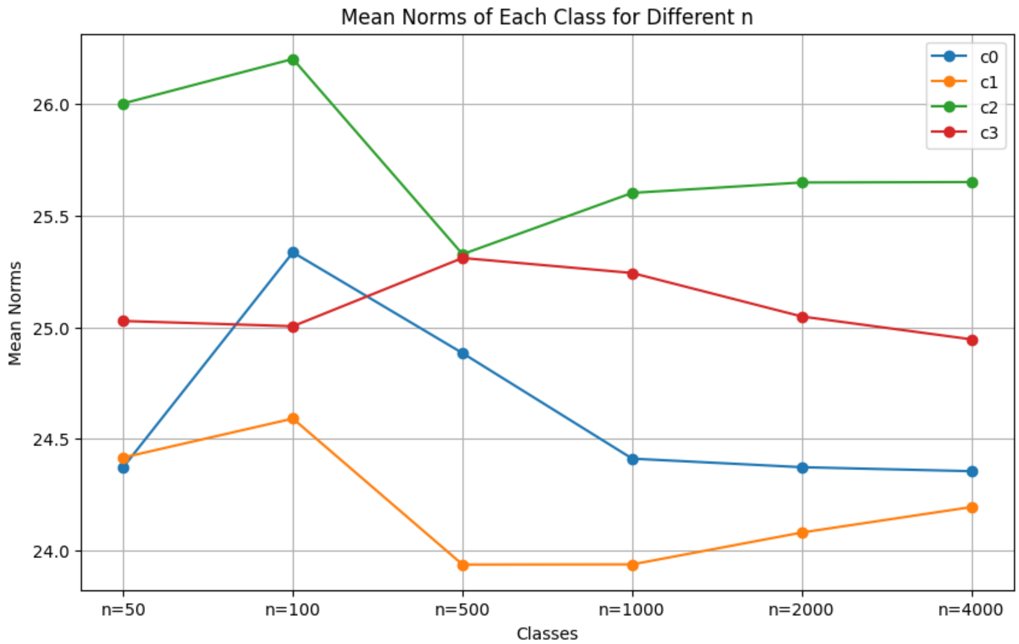

Figure 1: l_2 norms of the mean vectors for each class (Q.1.1)

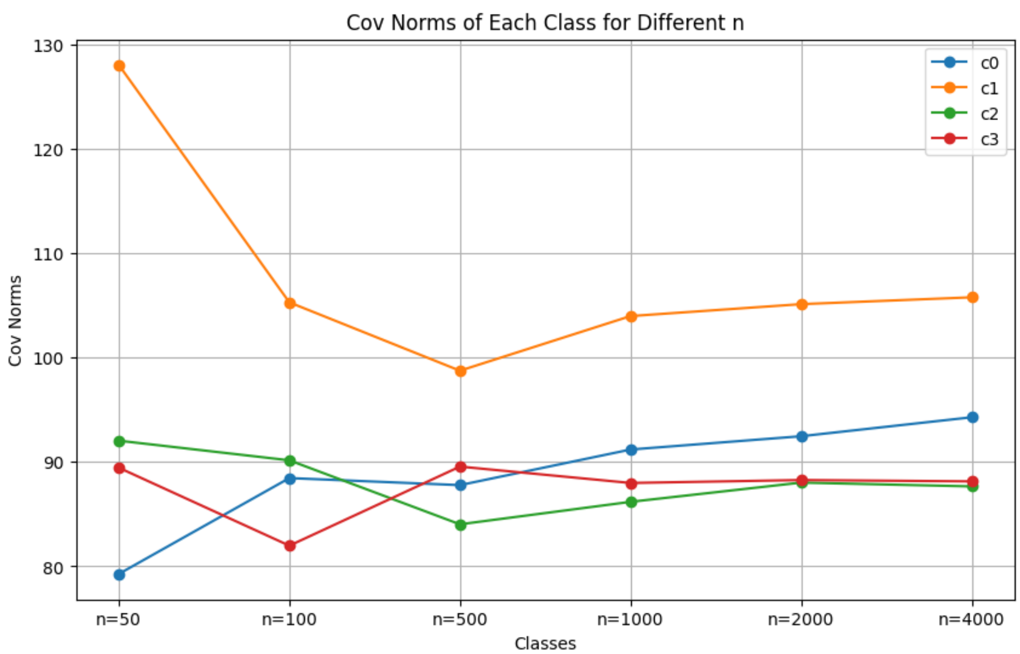

Figure 2: Frobenius norms of the covariance matrices for each class (Q.1.1)

Calculation of Objective values

First I calculated the scatter matrices S_i, for $i = 1, 2, 3, 4$, and thus calculated S_W. Then calculated the class means and thus S_B. Now for finding the objective value, we take the sum of eigen values of ${S_W}^{-1} \cdot S_B$ (the largest 3 eigen values actually, since theoretically the rest are 0).

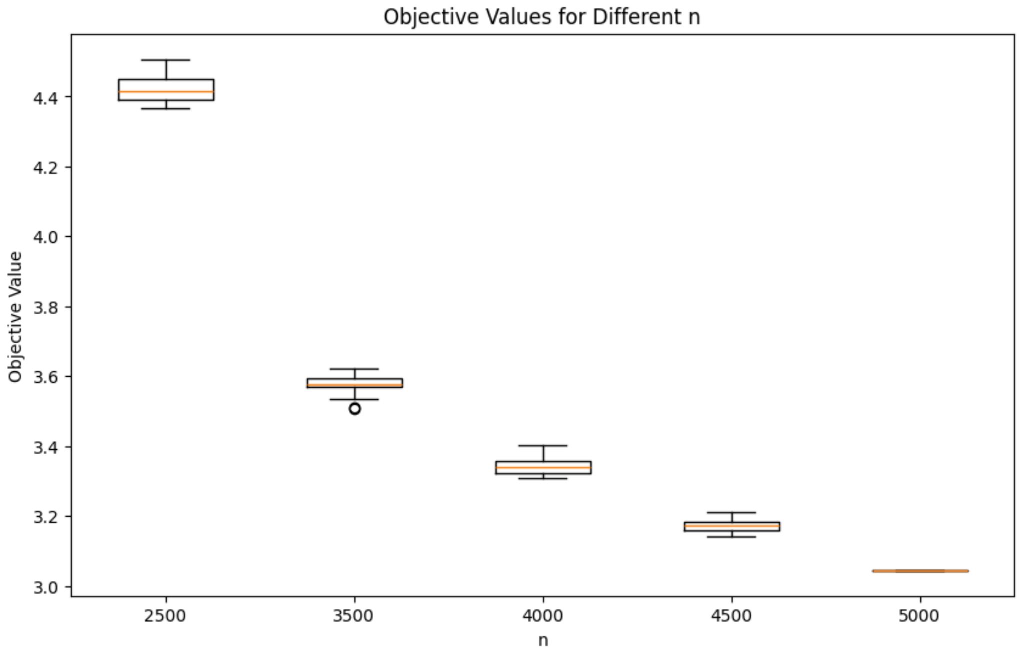

Figure 3: Box plots for the multi-class objective value for different values of n (Q.1.2.a)

Calculation of the projections

For the training set, I calculated S_B and S_W and thus eigen values of ${S_W}^{-1} \cdot S_B$. Now the eigen vectors corresponding to the largest three eigen values make W. For any given data-point $\mathbf{x}$ we take the projection as $W^T \cdot \mathbf{x}$. Taking the projections of all data points we plot those in 3-D.

Trend in box-plot

The box plot shows, objective values vary more for lower sample size and stabilizes with increasing number of samples.

Calculation of the thresholds

I could not do the threshold calculation directly. One idea is to plot the projections of the class-means and infer that for a point we can take take the *argmin* of distances from mean for each class and assign corresponding class. Otherwise, we can calculate posterior probabilities involving projected mean, projected datapoints' covariance, and the test datapoint's projection itself. Then we take *argmax* of the probabilities to get the class. Refer to **figure 6** for mean projections.

Insight on projected points

The projected points does not seem to be well separated and especially the test set is very very intermixed, which probably results in very low accuracy.

Answer (b)

For n = 5000, I got **55.1%** accuracy. I also implemented combination of two binary FLDs. In that case I got an overall accuracy of **54.6%**. Multiclass FLD accuracy values for different n are given in below table.

Implementation of combined binary FLDs

Since the classes 0-4 corresponds to no/no, no/yes, yes/no and yes/yes correspondingly for attribute 1/2 so I divided the test set for binary FLD classifications of attribute 1 and 2 separately. Performing FLD I got accuracy of 60.7% for big-lips attribute and 89.1% for wearing-lipstick attribute. Now recombining the predicted labels I got 54.6% accuracy on the final test set. (I have included code for both of my implementations in the '.py' file. (Refer to **Table 1**).

Table 1: Accuracy for different values of n (Q.1.2.b)

n	Accuracy (%)
200	27.300
500	38.500
1000	37.900
1500	45.700
2000	51.400
2500	49.200
3500	54.000
4000	55.700
4500	54.500
5000	55.900

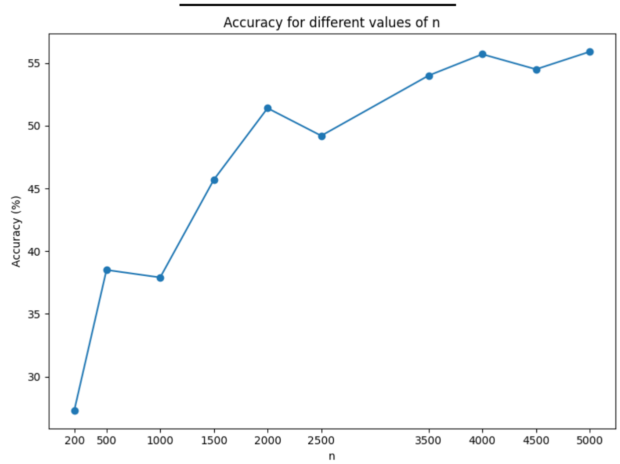

Figure 8: Accuracy comparison of table and plot (Q.1.2.b)

Question 2

2.1

Train the Modified Bayes Classifier for different values of $\epsilon \in 0.01, 0.1, 0.25, 0.4$ on training data and evaluate how decision confidence affects the performance on test data.

Misclassification Loss among the non-rejected sample

For all values of ϵ same misclassification loss of **0.02625** or error rate of **2.625**% was observed. [Even taking $\epsilon = 0.499$ no rejection occurred]. Probable cause of this answered in next sub-part.

Number of rejected sample

No rejected samples observed.
I could not observe any effect of incorporation of ϵ. I got the classes 5 and 27. I calculated $log\ \eta(x)$, which had

an average in the range of -3000 to -800 and the value of $log\ \eta_5/log\ \eta_{27}$ for any x was averagely in the order of $10^{\pm 50}$ to $10^{\pm 100}$. This might be a strong reason for no data-point getting rejected.

2.2

You are provided a dataset with a 50-50 split between priors. Subsample the dataset to create datasets with a 60-40, 80-20, 90-10, 99-1 split. Now compute the Modified Bayes Classifier under these modified priors for $\epsilon = 0.1, 0.25, 0.4$
Observations:

(i) **No values of ϵ has any effect.** (So, no graph shown separately for different values of ϵ)

(ii) Increasing train data samples for one class drastically reduces accuracy, if test sample contains 50-50 split.

Table 2: Misclassification Loss, test split 50-50 (Q.2.2)

Ratio	Misclassification Loss
40/60	0.026
20/80	0.484
10/90	0.500
1/99	0.500

(iii) Misclassification loss increases due to decreasing prior, going from 50-50 to 60-40 to 80-20. But then decreases probably due to less overall data points in the later class, so less misclassifications in total.

Table 3: Misclassification Loss, for same split in both train and test (Q.2.2)

Ratio	Misclassification Loss
40/60	0.030
20/80	0.198
10/90	0.099
1/99	0.010

(Both tables visualized in **figure 7**)

2.3

(a) Perform K-Fold Cross Validation with $K = 5$ and run Modified Bayes classifier with $\epsilon = 0.25$. For each fold, construct the confusion matrix on the validation set. Then, report the following performance metrics (you may treat 'reject' as a third category in the confusion matrix, or simply exclude it and focus on the 2×2 confusion of 'predicted 0 vs. predicted 1 whenever the classifier does not reject):

i. Recall: $\frac{TP}{TP+FN}$

ii. Precision: $\frac{TP}{TP+FP}$

iii. Accuracy: Proportion of correctly classified instances among the total instances

iv. F1-score: Harmonic mean of precision and recall, providing a balanced measure of model performance.

You are allowed to use scikit-learn for **K-fold cross validation**.

(b) Apply the trained model to the test data and Report:

- Number of rejected samples.
- Report the misclassification loss among the non-rejected samples

(a) K-fold cross validation results

We perform K-fold cross validation with $K = 5$ Please refer to **table 4** for performance metrics of each fold. **Table 5** contains average of the metrics.

Table 4: Performance Metrics for Each Fold (Q.2.3.a)

Fold	Accuracy	Recall	Precision	F1-Score
1	0.9583	0.9333	0.9824	0.9572
2	0.9563	0.9231	0.9881	0.9545
3	0.9583	0.9333	0.9824	0.9572
4	0.9708	0.9458	0.9956	0.9701
5	0.9635	0.9333	0.9933	0.9624
Average	0.9615	0.9338	0.9883	0.9603

Table 5: Average Cross-Validation Performance Metrics (Q.2.3.a)

Accuracy	Recall	Precision	F1-Score
0.9615	0.9338	0.9883	0.9603

(b) Test Data Performance

Even taking $\epsilon = 0.499$ gives zero rejected samples. (Tested with $\epsilon = 0.4999999999999$ and got only 2 rejected test data points.

Table 6: Test Data Performance Metrics (Q.2.3.b)

Metric	Value
Number of rejected samples	0
Misclassification Loss (on non-rejected samples)	0.02875

Question 3

3.1

Visualize the decision tree you trained using *dtreeviz*. Please refer to the **figure 8**.

Class	Precision	Recall	F1-Score	Support
0	0.91	0.79	0.85	38
1	0.71	0.87	0.78	23
Accuracy	0.82			
Macro Avg	0.81	0.83	0.81	61
Weighted Avg	0.84	0.82	0.82	61

Table 7: Decision Tree Classification Report (Q.3.2)

3.2

Report the precision, accuracy, recall, and F1 score of the above decision tree.
We use the *accuracy_score()* function to get the **table 7.**

3.3

What is the most important feature for predicting heart disease according to your tree?

From the **figure 8** we can infer that **cp** is the most important feature. The very first split at the root of the tree is based on the "cp" feature. This indicates it has the highest information gain and is the most significant factor in distinguishing between the two classes.

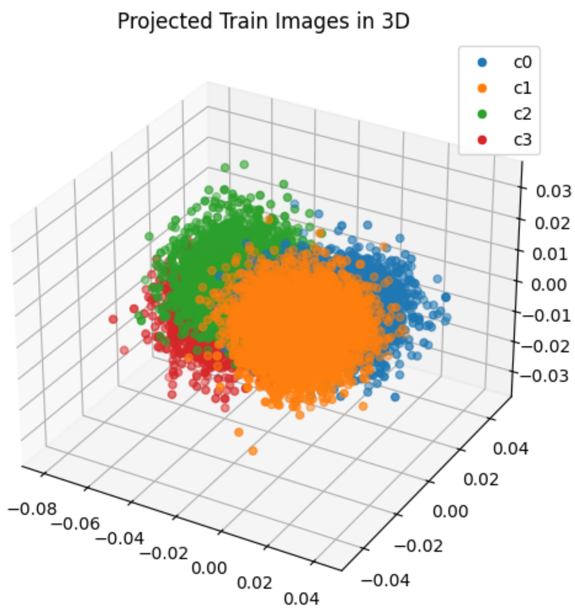

Figure 4: Train Data projection (Q.1.2.a)

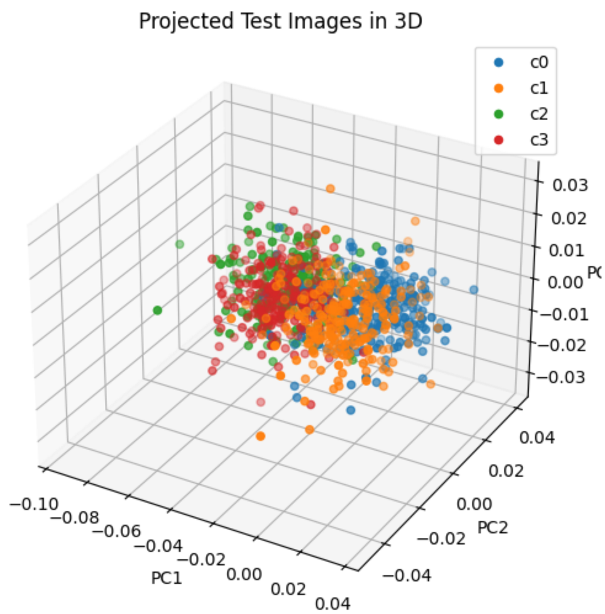

Figure 5: Test Data projection (Q.1.2.a)

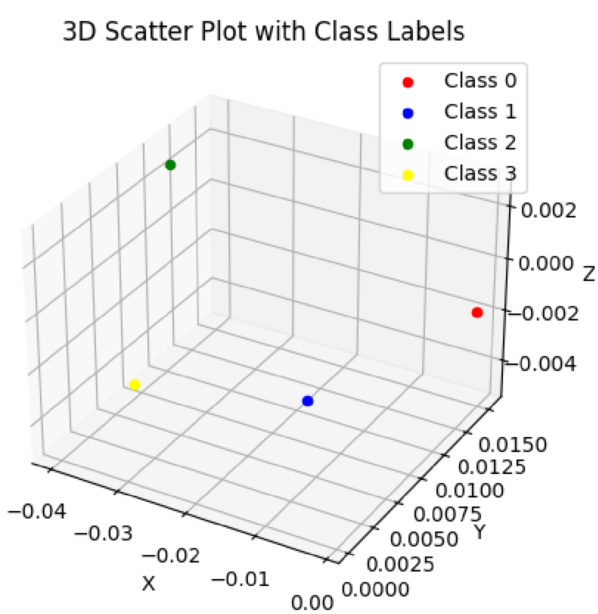

Figure 6: Projections of means of classes (Q.1.2.a)

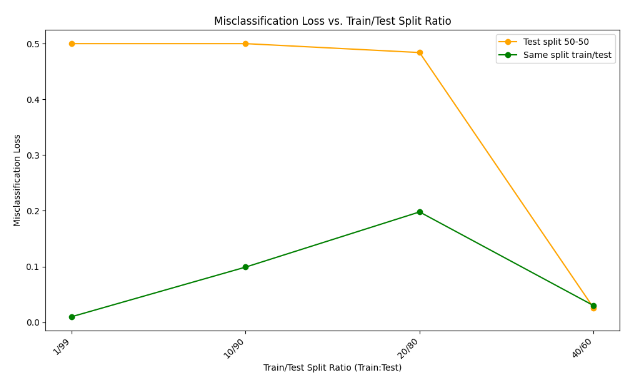

Figure 7: Misclassification loss (Q.2.2)

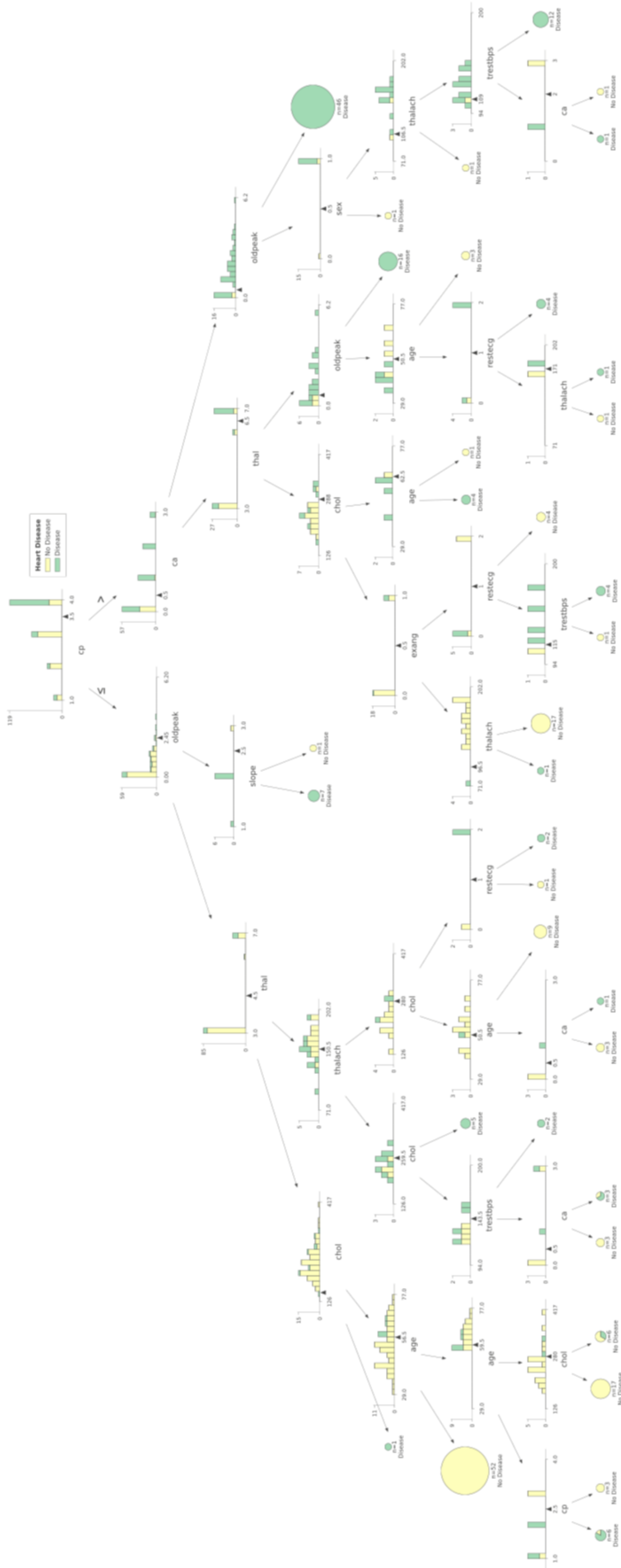

Figure 9: Decision tree visualization using dtreeviz (Q.3.1). See image