CS5691 Pattern Recognition and Machine Learning Programming Assignment 2

Team 16 - Akshith CS19B005, Shashank CS19B043

March 3, 2022

1 Regression

1.1 1D Dataset

The given training and development datasets have 200 points each. Fig 1 shows their scatter plots.

Fig 2 shows the solution curves for different model complexities (degree of the polynomial), by estimating the weights of the model using Maximum Likelihood Estimation. We can observe that as the model complexity increases, it keeps getting closer to the actual data.

Fig 3 shows the plot of error obtained when different model complexities are chosen. In Least Squares regression, the objective is to find the optimal weight vector to minimize the sum-of-squares error. From the plot, it is clear that minimum error occurs when degree is 7.

Note: As instructed, we only considered degree up to 7, even though the input data gives better results for higher degrees. This model under-fits the data.

Since degree 7 gives the best fit for the input data, we use that degree in all the future results.

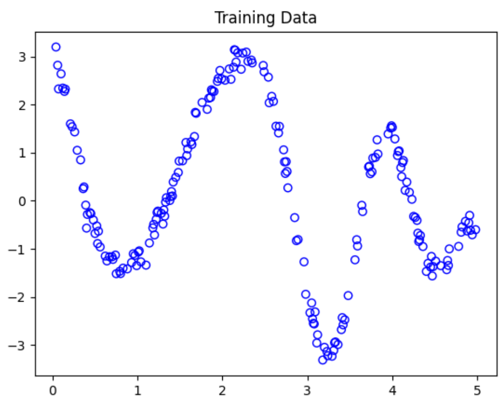
Fig 4 shows the solution curves when the model is trained with smaller training samples ($N = 20, 50, 100, 150$). As evident from the plots, smaller subsets of data are not sufficient to completely understand the pattern in the input data. By training with larger sample sizes, our model can better approximate the input data.

Regularization term $\lambda\|w\|^2$ is generally used to prevent the model from over-fitting the data. In our case, we are using regularization to constrain the weights from getting too large. Fig 6 shows the error obtained (both training & development data) as λ varies. We chose a logarithmic scale for λ for better comprehension. From the plot, the choice $\ln \lambda = -10$ seems to give the least error. Though the error becomes smaller as $\ln \lambda$ becomes more negative, it does not make sense to make $\ln \lambda \to -\infty$ or $\lambda \to 0$, as the model becomes equivalent to Least Squares regression.

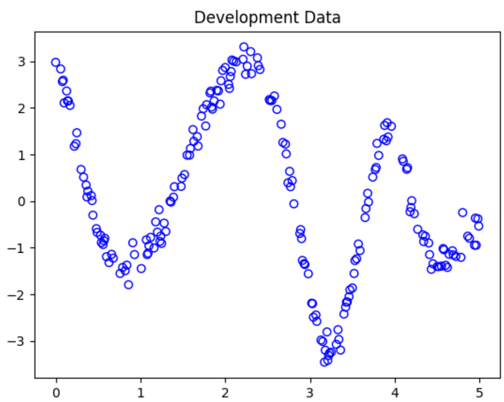
To summarize, the best curve is obtained when degree = 7 and $\ln \lambda = -10$

1.2 2D Dataset

Fig 7 shows the scatter plot of the input datasets (train and development).
The concepts applied in the case of 1D datasets hold true even in the case of 2D Dataset. However,

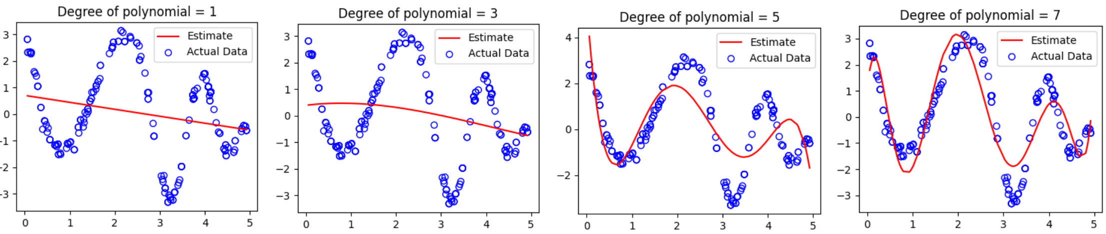


(a) Training Data



(b) Development Data

Figure 1: Input Training and Development Data



(a) Degree = 1 (b) Degree = 3 (c) Degree = 5 (d) Degree = 7

Figure 2: Solution Plots for different Model Complexities

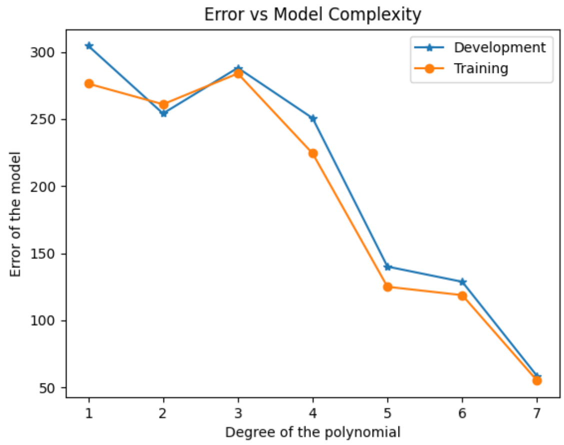
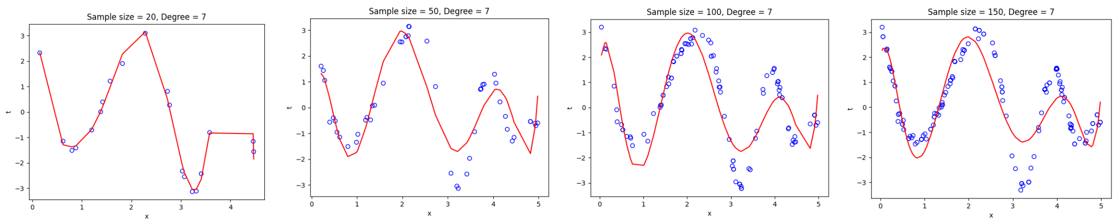


Figure 3: Error for different model complexities



(a) Sample Size = 20 (b) Sample Size = 50 (c) Sample Size = 100 (d) Sample Size = 150

Figure 4: Solution curves trained from different sample sizes

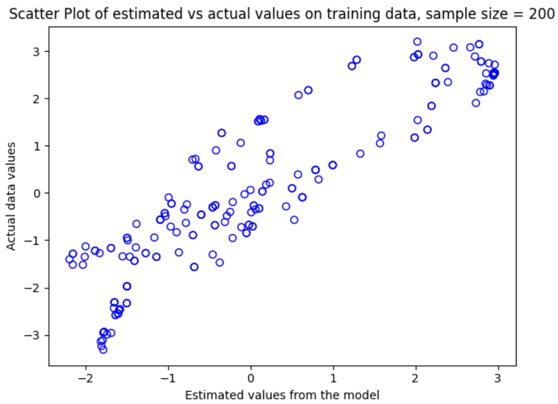


Figure 5: Actual data vs Data estimated from the model

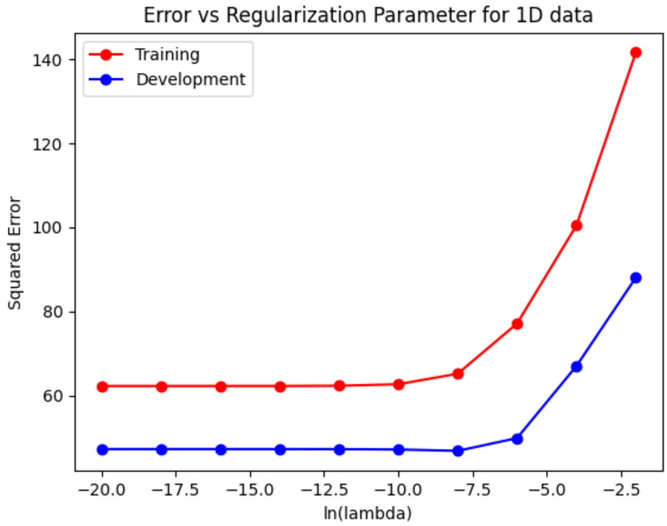
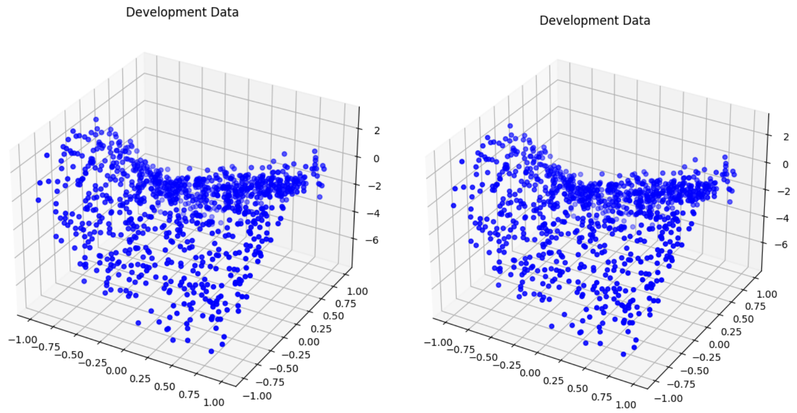


Figure 6: Error vs ln(Regularization Parameter)



(a) Training Data

(b) Development Data

Figure 7: Input Training and Development Data (2D)

the calculation of the weight vector differs, as we now have to consider basis functions over all possible terms $x_1^\alpha x_2^\beta$, satisfying $0 \leq \alpha \leq MC, 0 \leq \beta \leq MC$, where MC is the model complexity (degree of the polynomial chosen).

Fig 8 shows the error plot obtained by training models of different complexities on the entire training set. From this diagram, it is evident that degree 7 gives the least error. Hence, all the future inferences are made by assuming the degree to be 7.

Fig 9 shows the scatter plot of actual values (training data) vs values estimated by the model. All the values are close the line $y = x$, indicating a great accuracy in training.

tFig 10 shows the plot of Error vs ln(Regularization Parameter λ) for training data. Deciding similar to the 1D case, here $\ln \lambda = -18$ seems to give the lowest error.

2 Classification

For Linearly Separable data: 100% accuracy was obtained in all the five cases.

Formulae used for estimation of parameters in Classification:

$$\hat{\pi}_j = \frac{n_j}{n} \qquad \hat{\mu}_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i$$

Case 1 : Bayes with Co-variance same for all classes

$$\Sigma = \frac{1}{n-3} \sum_{j=1}^{3} \sum_{x_i \in C_j} (x_i - \mu_j)(x_i - \mu_j)^T$$

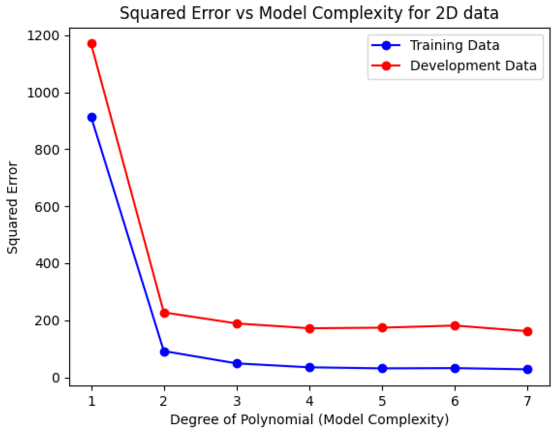


Figure 8: Error for different model complexities

Figure 9: Actual data (training) vs Model estimation

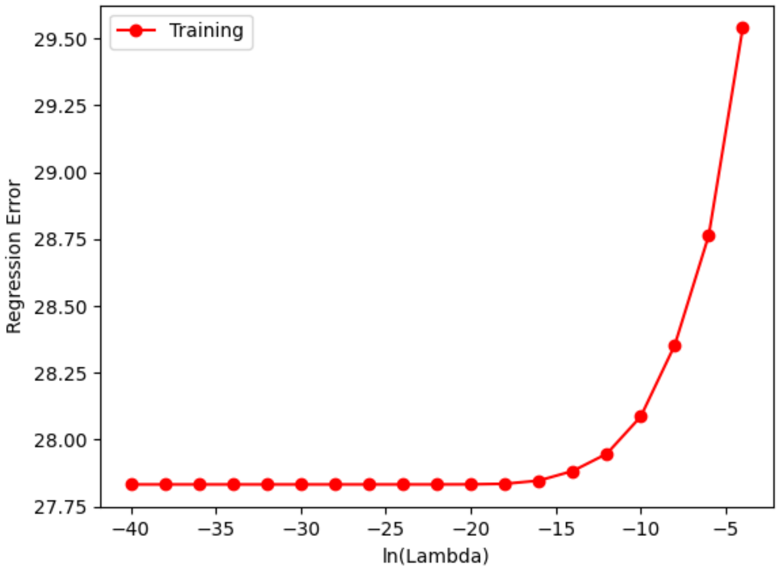
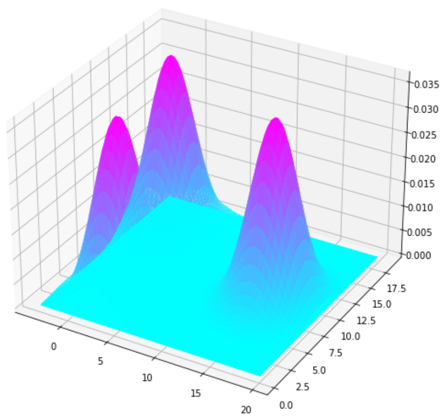
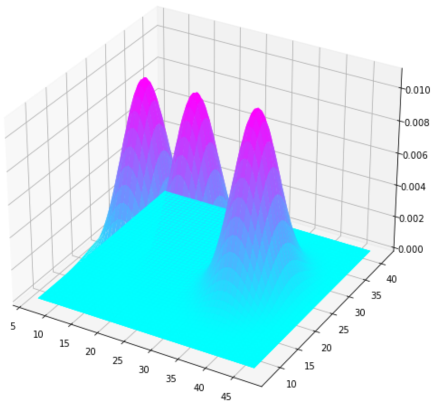


Figure 10: Error vs ln(Regularization Parameter)



(a) PDF for Linearly Separable (model 3)



(b) PDF for Non Linearly Separable (model 4)

Figure 11: PDF plots

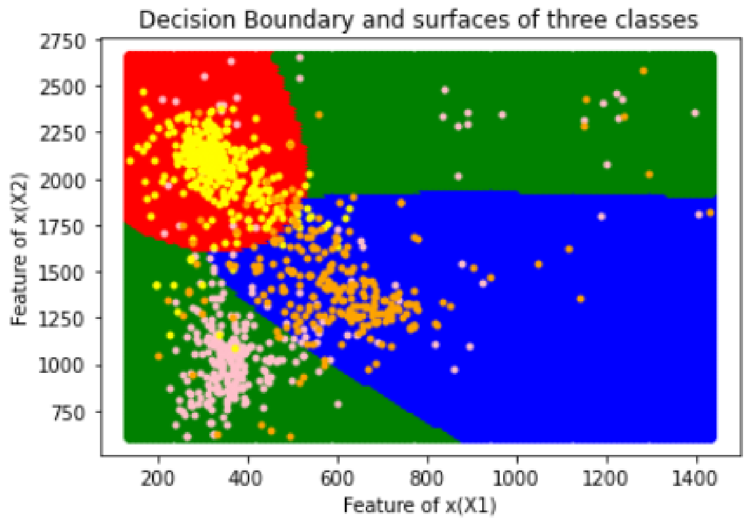


Figure 12: Decision Surface for Real Data (model 2)

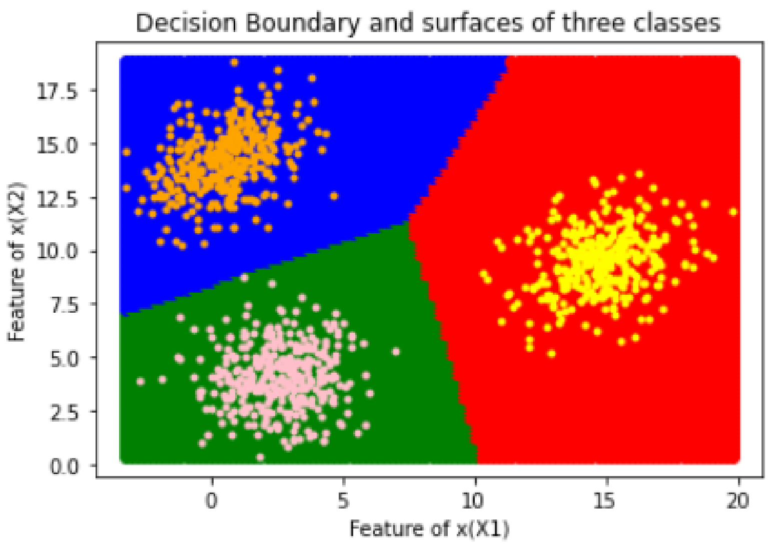


Figure 13: Decision Boundary for Linearly Separable Data (model 3)

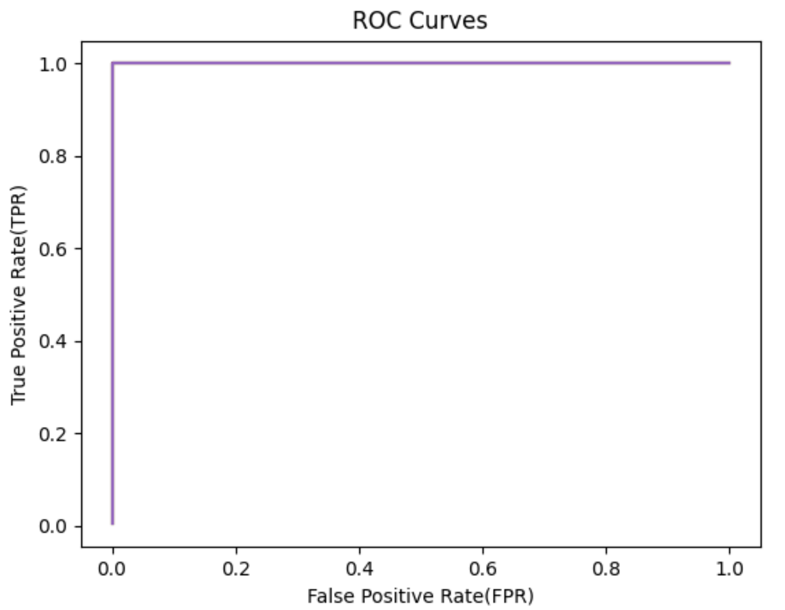
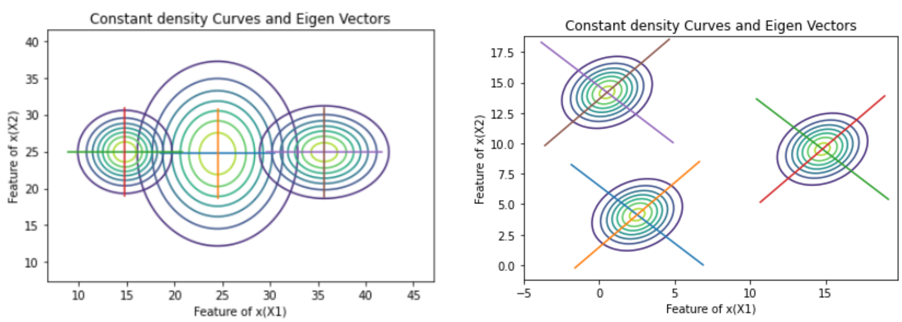


Figure 14: ROC curve for Linearly Separable Data



(a) Constant Density curves and Eigen Vectors for Non Linearly separable data (model 5)

(b) Constant Density curves and Eigen Vectors for Linearly separable data (model 1)

Figure 15: Constant Density curves

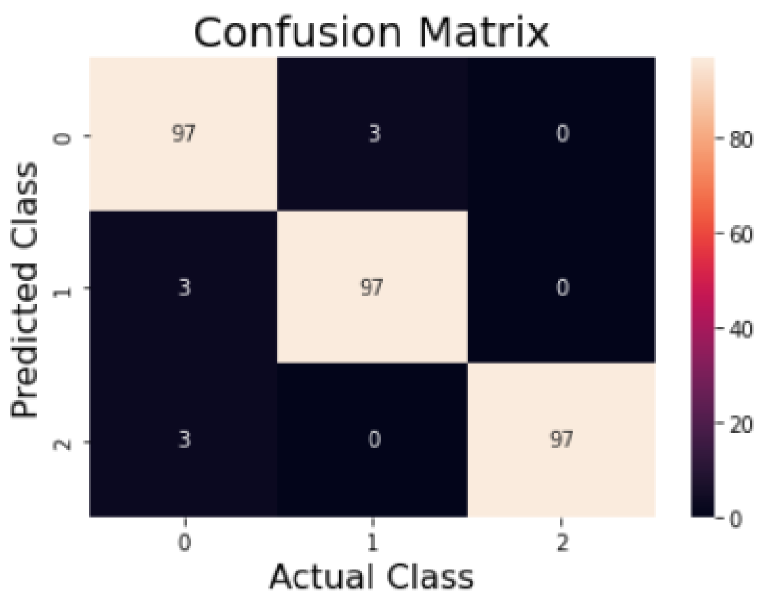
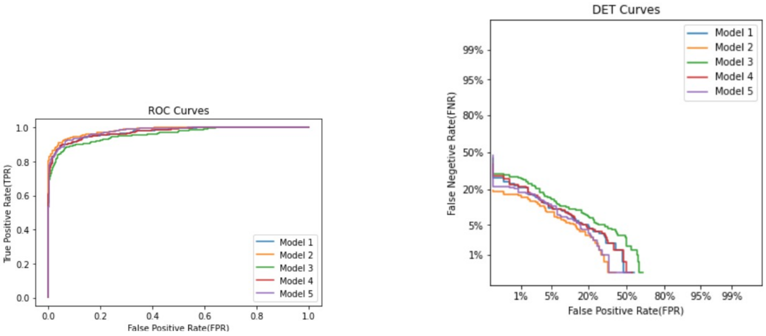
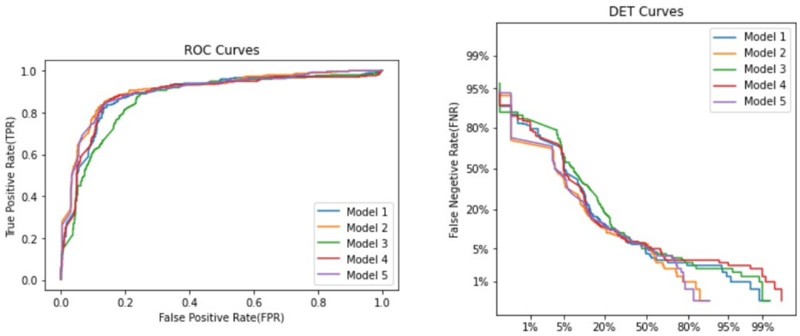


Figure 16



(a) ROC for Non Linearly Separable Data (b) DET for Non Linearly Separable Data

Figure 17: Non Linearly Separable Data curves



(a) ROC for Real Data (b) DET for Real Data

Figure 18: Real Data curves

Case 2 : Bayes with Co-variance different for all classes

$$\Sigma_j = \frac{1}{n_j - 1} \sum_{x_i \in C_j} (x_i - \mu_j)(x_i - \mu_j)^T$$

Case 3 : Naive Bayes with $C = \sigma^2 I$

$$\sigma^2 = \frac{1}{n} \sum_{j=1}^{3} \left(\sum_{x_i \in C_j} (x_i - \mu_{j,1})^2 + \sum_{y_i \in C_j} (y_i - \mu_{j,2})^2 \right)$$

Case 4 : Naive Bayes with $C_1 = C_2$:

$$\sigma_1^2 = \frac{1}{n} \sum_{j=1}^{3} \sum_{x_i \in C_j} (x_i - \mu_{j,1})^2 \qquad \sigma_2^2 = \frac{1}{n} \sum_{j=1}^{3} \sum_{y_i \in C_j} (y_i - \mu_{j,2})^2$$

Case 5: Naive Bayes with $C_1 \neq C_2$:

$$\sigma_{j,1}^2 = \frac{1}{n_j} \sum_{x_i \in C_j} (x_i - \mu_{j,1})^2 \qquad \sigma_{j,2}^2 = \frac{1}{n_j} \sum_{y_i \in C_j} (y_i - \mu_{j,2})^2$$

Points:-

1. In case of linear seperable data gives the 100 accuracy.
2. EigenVectors of covariance matrix are the eigen vectors plotted in constant density curves and eigen vector figures.
3. Boundary and surface plots are obtained by plotting the meshgrid and considering the maximum probability among the classes present.
4. In case of linear seperable curves for all models coincides.
5. Det curve is plotted using inbulit library function.
6. Projection of gussian curve on 2D plane gives density curves.