1 Part 1A - Sequential Bayesian learning

1.1 Produce contour plots of the posterior of θ:

- After observing x_1 and y_1

  The contour plot shows a broad distribution, indicating uncertainty about the parameter estimates from one observation.

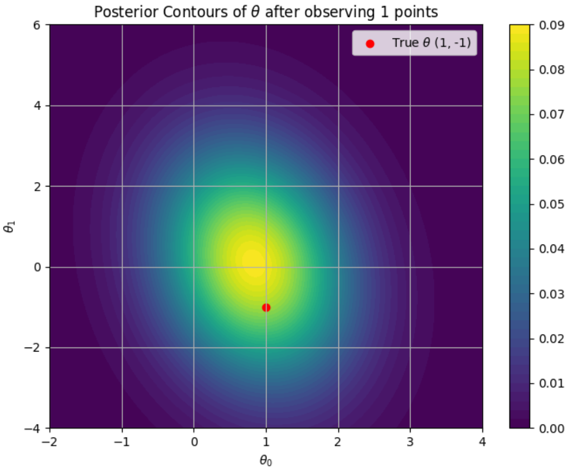

Figure 1: Posterior Counters of θ for $(\sigma_\eta^2 = 1, \sigma_\theta^2 = 4)$ after observing 1 point

- After observing x_1, y_1, x_2, and y_2

  The posterior distribution is tighter than the previous one, it shows a reduction in uncertainty.

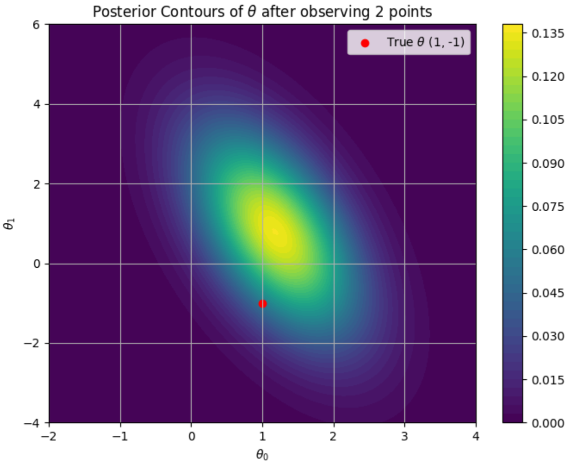

Figure 2: Posterior Counters of θ for $(\sigma_\eta^2 = 1, \sigma_\theta^2 = 4)$ after observing 2 point

- After observing all 10 data points.

  The posterior is concentrated around the true parameters $(\theta_0 = 1, \theta_1 = -1)$, it shows high certainty in the estimates.

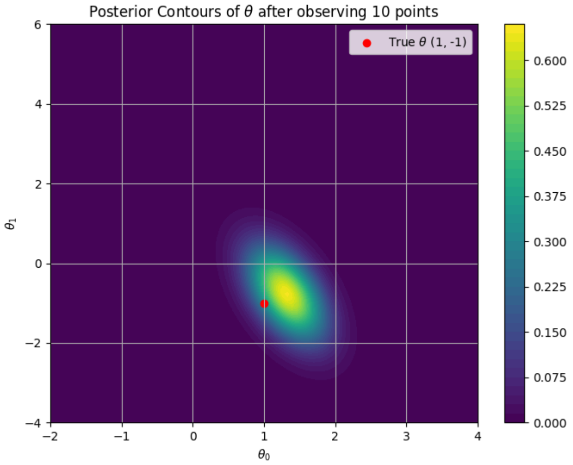

Figure 3: Posterior Counters of θ for ($\sigma_\eta^2 = 1, \sigma_\theta^2 = 4$) after observing 10 point

1.2 For each case (a)–(c), draw 10 pairs of θ from the respective posteriors and plot the obtained lines. For all drawn values of θ, plot the lines defined by these values on a same figure.

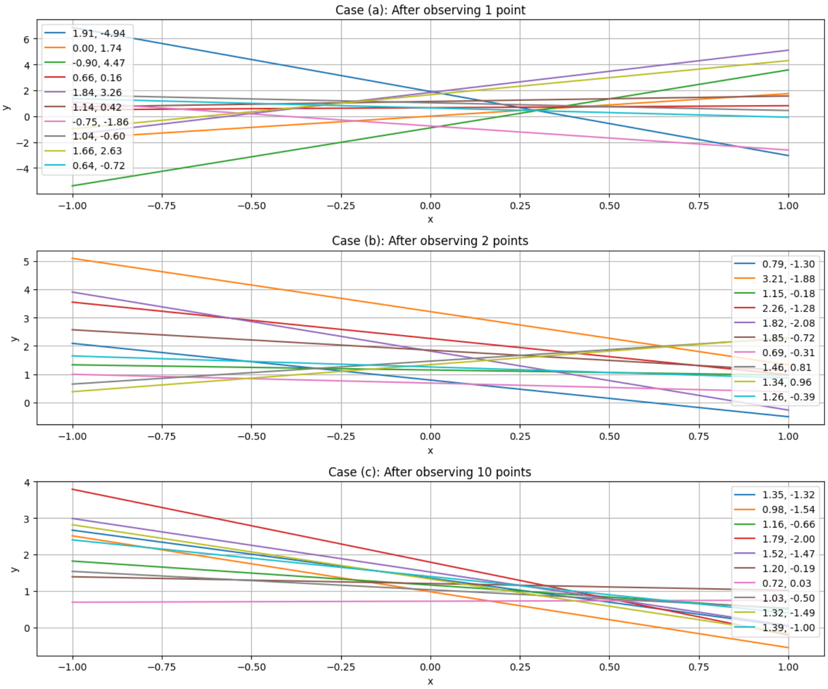

Figure 4: Plot of 10 pairs of θ after observing 1, 2, 10 pairs of points

- After observing x_1 and y_1

  The lines are varied, it shows uncertainty in the estimates of θ from single observation.

- After observing x_1, y_1, x_2, and y_2

  The lines start to converge, it shows reduced uncertainty from the previous plot.

- After observing all 10 data points.

  The lines are closely aligned, it shows the high certainty in the estimates of θ due to the larger dataset.

Each subplot shows us how data points help to stabilize the model predictions.

1.3 Use the first three pairs $\{(x_n, y_n)\}_{n=1}^{3}$ to find the predictive distribution $p(y^* \mid x^*, x_1, y_1, x_2, y_2, x_3, y_3)$, where x^* takes values from -1 to 1 in steps of 0.01. Plot the mean of the predictive distribution with one standard deviation on either side of the mean. Discuss briefly the obtained results.

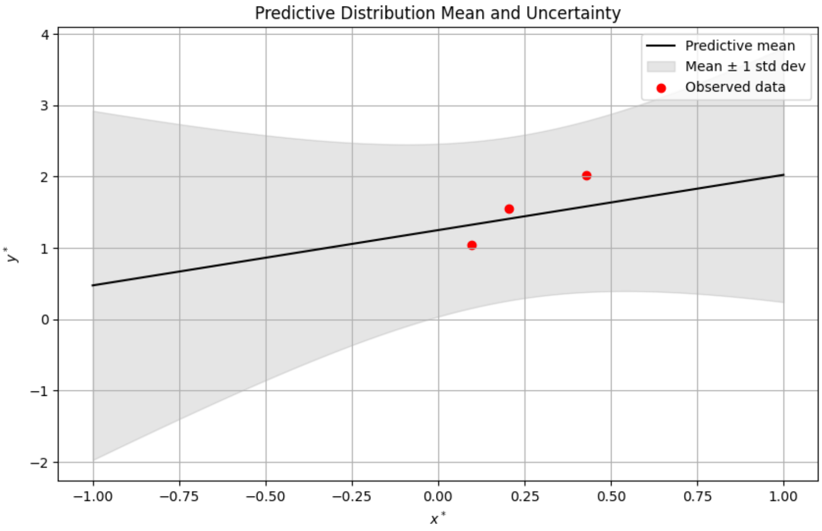

Figure 5: Mean of the predictive distribution with one standard deviation

- The black line represents the mean which is a function of x^*. This mean line is calculated by our updated beliefs about the parameters θ_0 and θ_1 after observing the three data points. The shaded area around the mean line indicates the uncertainty in our predictions.
- Near the observed data points (red dots), the uncertainty is typically smaller, indicating more confidence in the predictions. The uncertainty increases as we move away from these points.

1.4 Repeat the above tasks with $(\sigma_\eta^2 = 5, \sigma_\theta^2 = 4)$ and $(\sigma_\eta^2 = 1, \sigma_\theta^2 = 25)$. Discuss the differences with respect to the results from 1, 2, and 3.

We know, the posterior distribution is proportional to the product of the prior and the likelihood. Tighter contours suggest higher confidence (less uncertainty), while wider contours suggest lower confidence (more uncertainty). High noise or high prior variance increase uncertainty, give the broad contours in the plots.

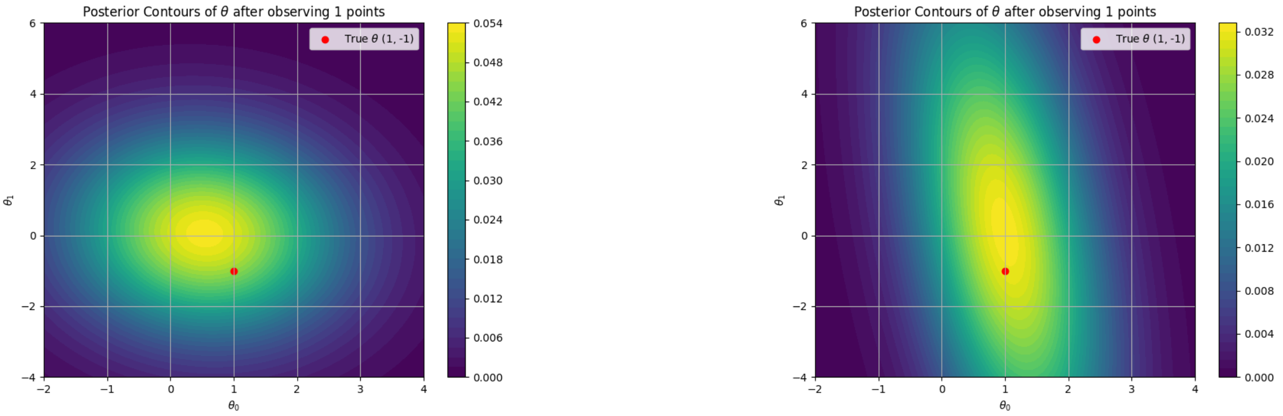

Figure 6: Posterior Counters of θ for $(\sigma_\eta^2 = 5, \sigma_\theta^2 = 4)$ in first plot and $(\sigma_\eta^2 = 1, \sigma_\theta^2 = 25)$ in second plot, after observing 1 point

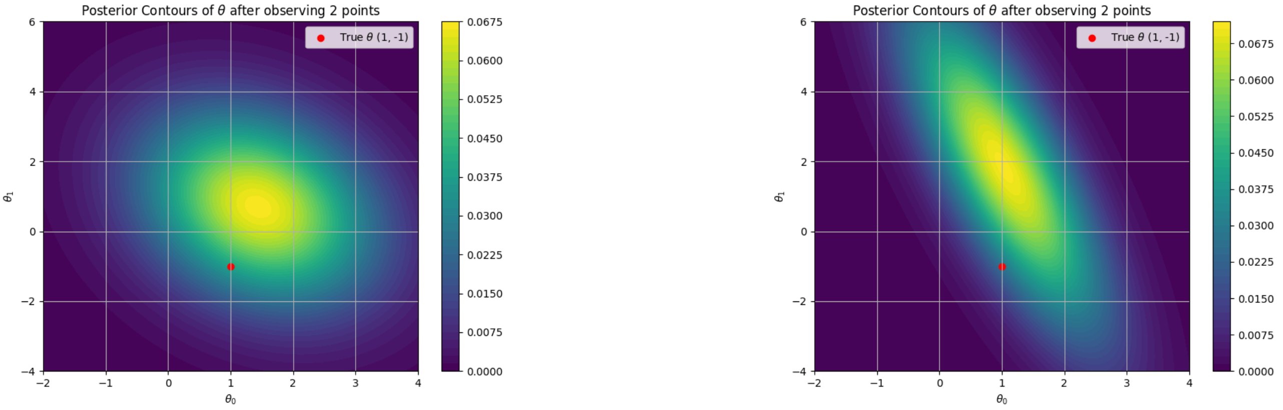

Figure 7: Posterior Counters of θ for $(\sigma_\eta^2 = 5, \sigma_\theta^2 = 4)$ in first plot and $(\sigma_\eta^2 = 1, \sigma_\theta^2 = 25)$ in second plot, after observing 2 points

Comparison of figure 1 and figure 6 - plot 1:
Lower noise variance $(\sigma_\eta^2 = 1)$ shows tighter contours around the true value (marked by the red dot) while for $(\sigma_\eta^2 = 5)$ the contours are more spread out. Higher noise variance makes the data less certain.

Comparison of figure 1 and figure 6 - plot 2:
Lower prior variance $(\sigma_\theta^2 = 4)$ shows tighter contours around the true value (marked by the red dot). This gives a stronger influence of the prior belief for posterior. For $(\sigma_\theta^2 = 25)$ the contours are wider, so does not strongly influence the parameter estimates.

Comparison of figure 1 and figure 7:
The posterior distribution is tighter because of observing another point. The comparison is similar to the above two comparisons mentioned.

Comparison of figure 1 and figure 8:
Because of observing all 10 points the contours are now centered closer to the true value θ = [1, -1].

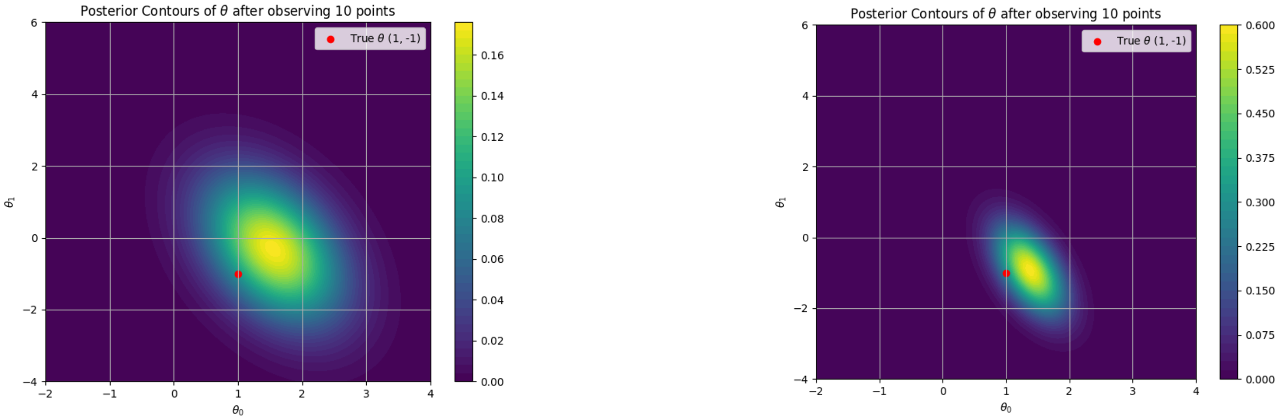

Figure 8: Posterior Counters of θ for $(\sigma_\eta^2 = 5, \sigma_\theta^2 = 4)$ in first plot and $(\sigma_\eta^2 = 1, \sigma_\theta^2 = 25)$ in second plot, after observing 10 points

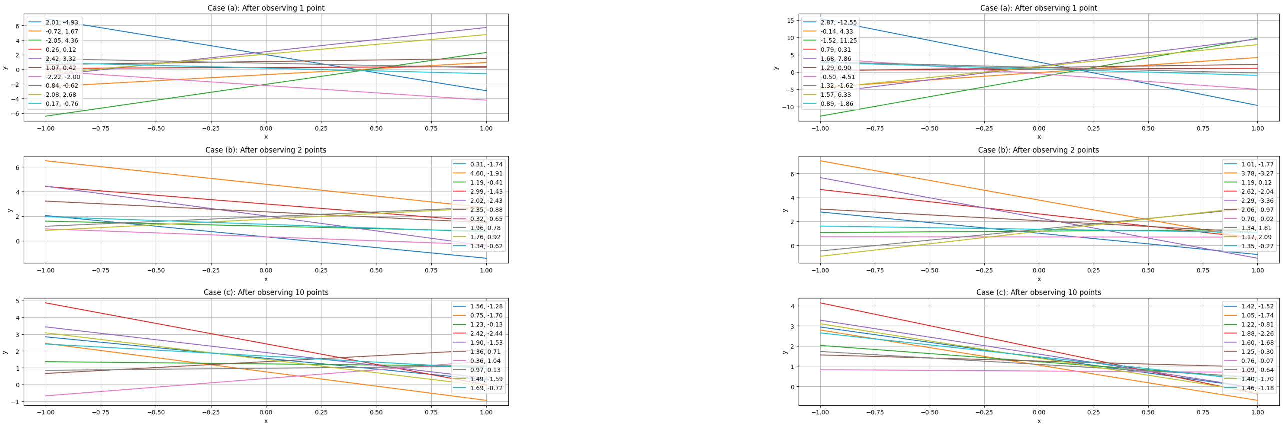

Figure 9: Plot of 10 pairs of θ for $(\sigma_\eta^2 = 5, \sigma_\theta^2 = 4)$ in first plot and $(\sigma_\eta^2 = 1, \sigma_\theta^2 = 25)$ in second plot

When the prior variance is lower, the plots are tighter. Again, when the noise variance is higher, the contours are more spread out.

Comparison of figure 4 and figure 9:

- For higher noise variance, the lines are less concentrated around the true parameter. For lower noise variance and 10 points, almost all lines closely track the true regression line. We have high confidence and precision in estimates.
- For lower prior variance and increased the number of observations, the regression lines begin to converge closer to the true parameter values. For higher prior variance, the regression lines are still more spread out but proved that with enough data, even a weak prior can be overcome.

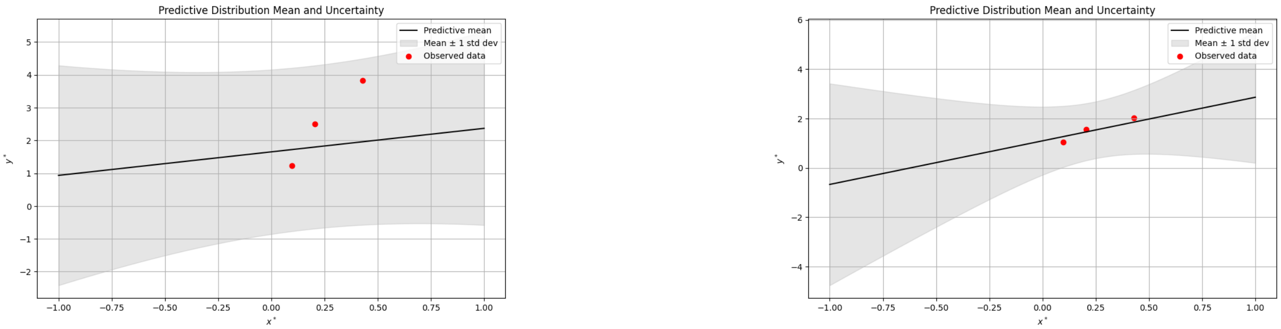

Figure 10: Mean of the predictive distribution with one standard deviation, for ($\sigma_\eta^2 = 5, \sigma_\theta^2 = 4$) in first plot and ($\sigma_\eta^2 = 1, \sigma_\theta^2 = 25$) in second plot

Comparison of figure 5 and figure 10:
Increasing noise variance ($\sigma_\eta^2 = 1 \rightarrow \ \sigma_\eta^2 = 5$) influences the uncertainty around predicted mean. So we have a wider bands around the regression line. That means individual observations are expected to vary around the predicted mean due to noise.
As prior variance ($\sigma_\theta^2 = 4 \rightarrow \ \sigma_\theta^2 = 25$) increases, the uncertainty in the slope gives a broader confidence intervals.

2 Part 1B - Gaussian basis functions

2.1 Set $s = 1$ and use the first three generated pairs $\{(x_n, y_n)\}_{n=1}^3$ to find the predictive distribution $p(y^*|x^*, x_1, y_1, x_2, y_2, x_3, y_3)$, where x^* takes values from -1 to 1 in steps of 0.01. Plot the mean of the predictive distribution with one standard deviation on either side of the mean. Discuss briefly the obtained results

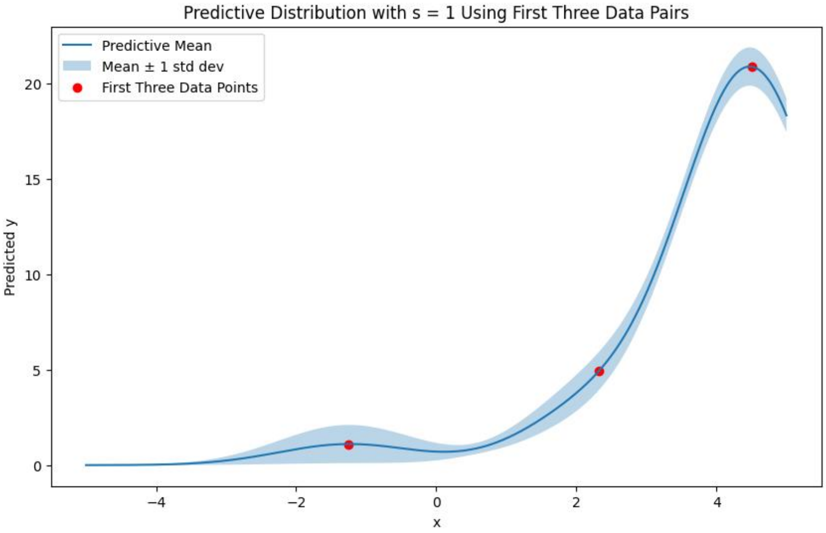

Figure 11: Predictive distribution of 3 data pairs

- Predictive Mean (Blue Line) reflects the expected value of based on the Gaussian basis functions centered around the three initial data points.
- Confidence Interval (Shaded Area) gives the range where the actual values of y are expected to fall, computed as one standard deviation from the mean. It helps us to understand uncertainty

2.2 Repeat with $s = 0.5$ and $s = 0.1$ and discuss the differences in prediction

Each plot shows the effect of changing s on the shape of the predictive mean and its uncertainty.

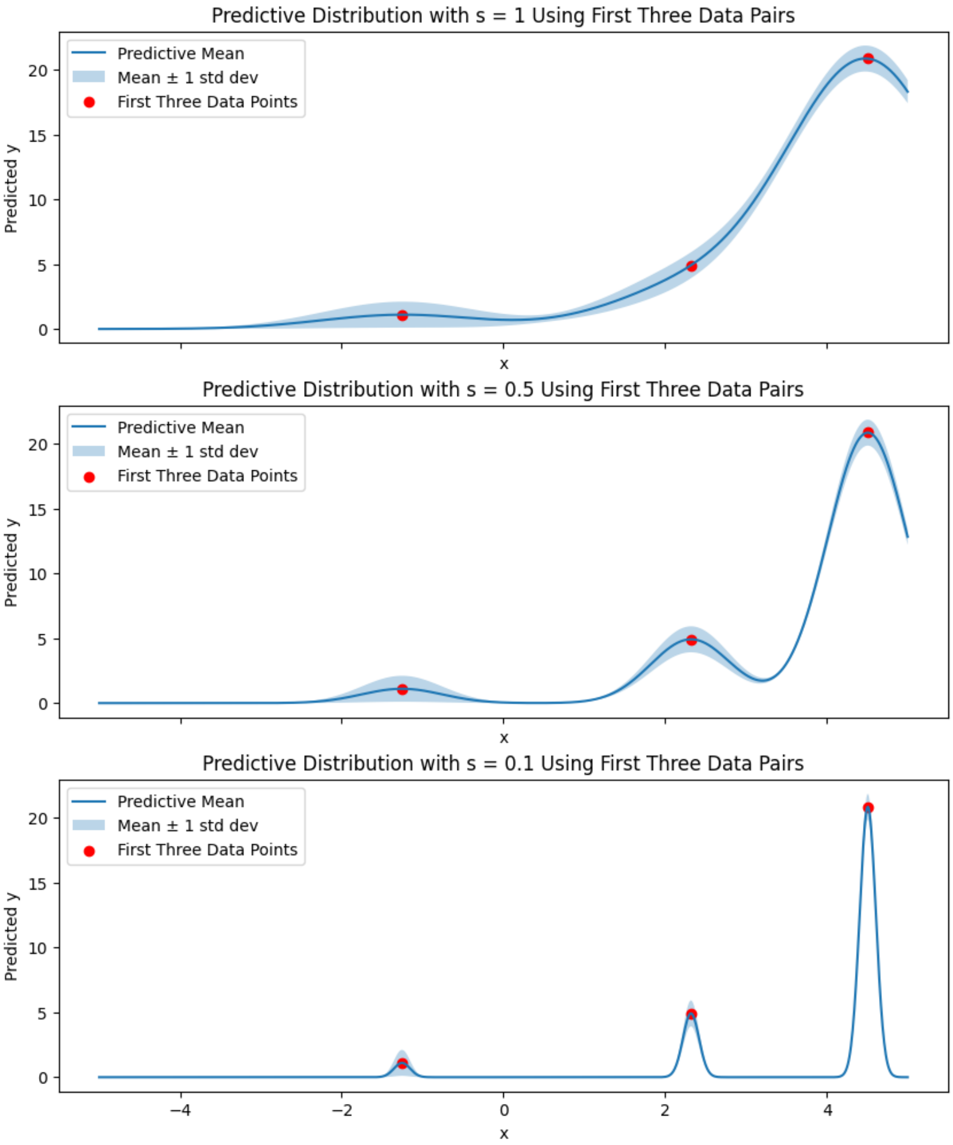

Figure 12: Predictive distribution of 3 data pairs with different s

Smoothness and Spread:
For s = 1, mean is very smooth and broad. It indicates each Gaussian function has a wider influence over the range of x values.
For s = 0.5, the curve becomes slightly more jagged, so has a reduced influence of each basis function.

It starts focusing too much on local data.
For s = 0.1, the mean shows sharpness, with narrow peaks at the data points. Gaussian function is highly localized.

Confidence Interval:
For s = 1, the confidence interval is fairly narrow and consistent across the range.
For s = 0.5, it has more increased uncertainty where data is not available.
For s = 0.1, intervals are narrow at the peaks and widen away from the data points. So very high confidence in the model's predictions close to the data points but high uncertainty in everywhere else.

As s decreases, the model shifts from a general approach to a highly localized approach. A larger s (e.g., figure 11) tends to smooth out noise and capture broad patterns, while a smaller s (e.g., figure 12 - plot 3) captures detailed local variations.

2.3 Repeat 1 with $\{(x, y)\}_{n=1}^{10}$. Compare the results with those from task 1

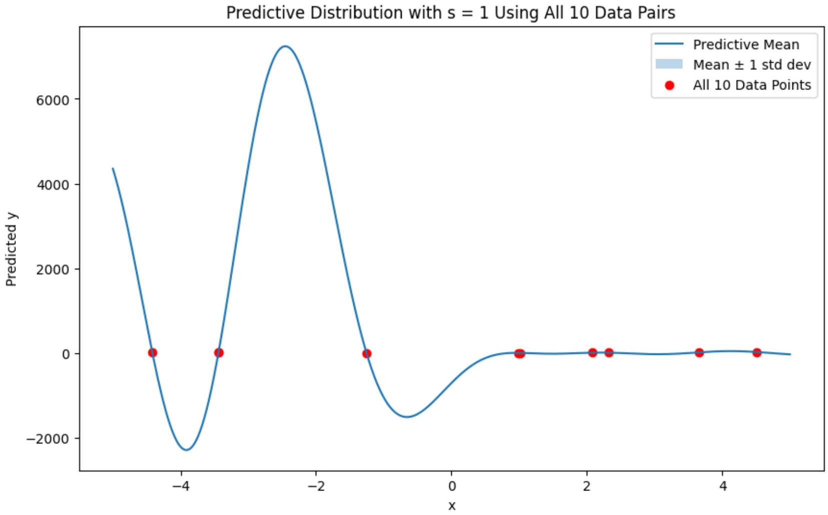

Figure 13: Predictive distribution of 10 data pairs

- With 10 data points, the model has more information to estimate the coefficients θ_n more accurately than figure 11. This reduces uncertainty of the predictive distribution.
- As more data points are used, the sum over Gaussian basis functions $\sum_{n=1}^{N} \theta_n \phi_n(x)$ can capture more complex patterns in the data.
- Sometimes more data and high number of basis functions N leads to overfitting.

3 Part 2 - Two-Class Gaussian Classification Problem

We have a two-class, two-dimensional classification problem, ω_1, and ω_2, with mean and covariance given by:

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 0.25 & 0.3 \\ 0.3 & 1 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} 2.5 \\ 2 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 0.25 & -0.3 \\ -0.3 & 1 \end{bmatrix}$$

3.1 Plotting of a data set $\mathcal{X}$

Here's the plot of the generated dataset, showing 200 points from each class ω_1 (in blue) and ω_2 (in red).

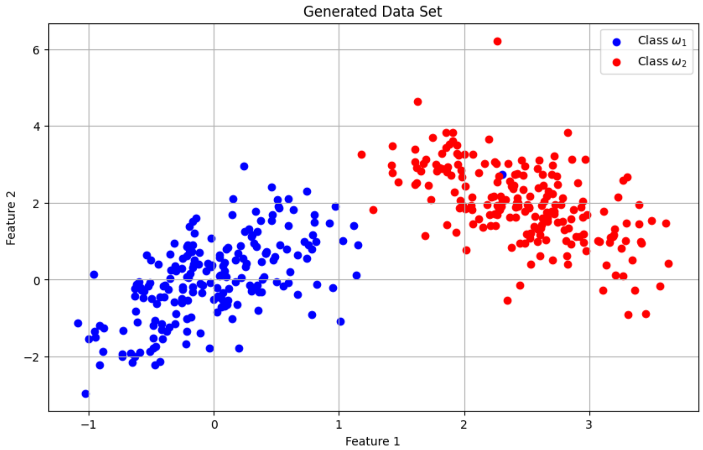

Figure 14: 200 points from each class ω_1 (in blue) and ω_2 (in red)

3.2 Use the first 100 points to learn the mean and covariance of the respective bivariate Gaussians and then assign each one of the points of the remaining samples from $\mathcal{X}$ to either ω_1 or ω_2 according to the Bayes' decision rule. Plot the points with different colors, according to the class they are assigned to, find the confusion matrix of the classifier, and compute the classification error probability. Explain how the classification is carried out.

Figure 15 shows the classified test data separated with color.

First of all, the dataset is divided into two train sets for class 1 and 2 and two test sets for class 1 and 2. Then estimated the mean and covariance from training data for both classes. Then combined test datasets. Calculated probability density functions using multivariate_normal. After that, Bayes' decision rule is applied and confusion matrix is computed.

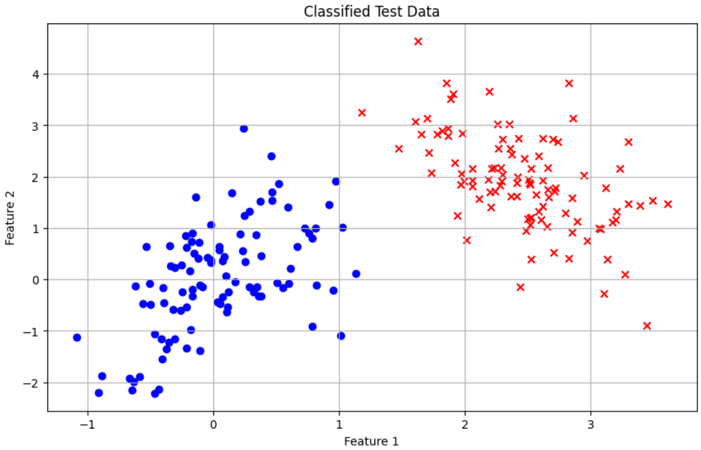

Figure 15: 100 test points from each class ω_1 (in blue) and ω_2 (in red)

The confusion matrix and the classification error probability are given by:

$$\text{Confusion Matrix} = \begin{array}{|c|c|} \hline 99 & 1 \\ \hline 0 & 100 \\ \hline \end{array}$$

$$\text{Classification Error Probability} = 0.0050000000000000044$$

This matrix shows us that all 99 points from Class ω_1 were correctly classified, and all 100 points from Class ω_2 were also correctly classified, resulting in a classification error probability of 0.005.

3.3 Classify the generated data with logistic regression.

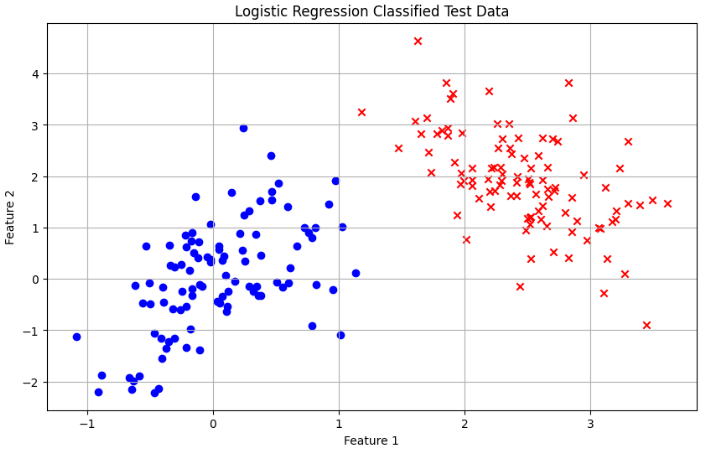

Figure 16: 100 test points from each class ω_1 (in blue) and ω_2 (in red) using logistic regression

Figure 16 shows us logistic regression classifier also performed exceptionally well. The confusion matrix for the logistic regression is:

$$\text{Confusion Matrix} = \begin{array}{|c|c|}\hline 99 & 1 \\ \hline 0 & 100 \\ \hline \end{array}$$

$$\text{Classification Error Probability} = 0.0050000000000000044$$

Same like Bayesian classifier's performance, all 99 points from Class ω_1 and all 100 points from Class ω_2 were correctly classified.

3.4 Repeating 1 and 2 with 1000 points generated from each class

Same like before, half of the dataset(500 points) is being used for training purpose.

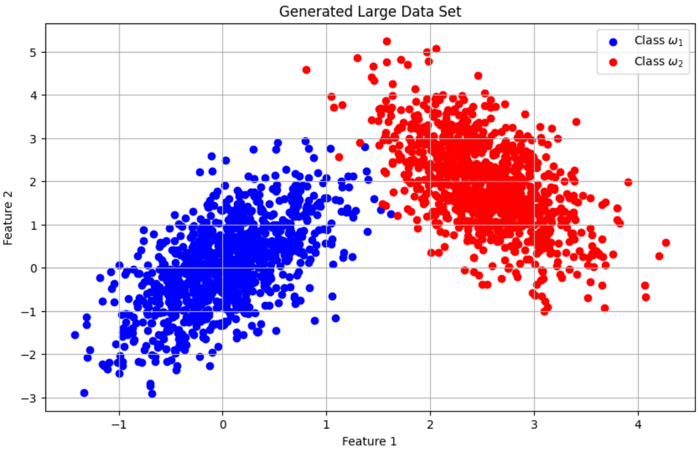

Figure 17: Generated 1000 points from each class ω_1 (in blue) and ω_2 (in red)

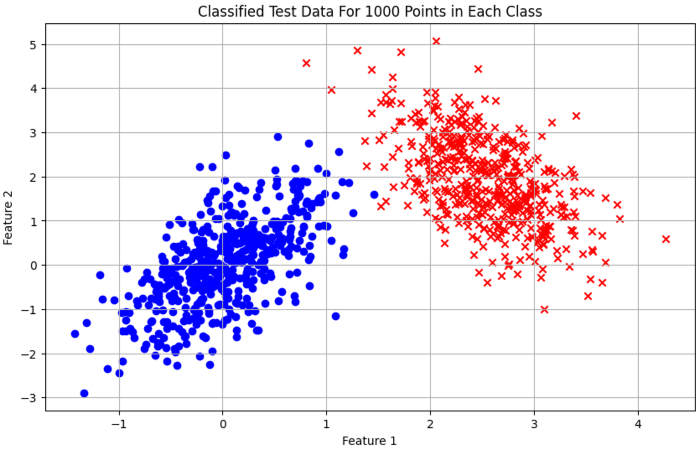

Figure 18: Classified 500 points from each class ω_1 (in blue) and ω_2 (in red)

The confusion matrix and the classification error probability are given by:

$$\text{Confusion Matrix} = \begin{array}{|c|c|}\hline 497 & 3 \\ \hline 1 & 499 \\ \hline \end{array}$$

$$\text{Classification Error Probability} = 0.0040000000000000036$$

This matrix shows that 497 out of 500 points from Class ω_1 and 499 out of 500 points from Class ω_2 were correctly classified, with only 3 and 1 misclassifications. The classification error probability is 0.004, indicating an error rate of 0.4
So, 1000 points separates the two classes with minimal error, which is a slight decrease from the previous results(0.005) where we used smaller datasets(200 points each). The classifier is robust.

3.5 Write a brief conclusion from all the results.

We can conclude that both Bayesian and logistic regression classifiers can perform well in scenarios where the classes are well-separated.

- The classification error probability slightly improves with the larger dataset.
- More data accurately estimate the underlying distributions of each class.
- Dataset has minimal overlap and clear separation between classes.

4 Part 3 - Classification of real data

The workflow in short is like below:

- Data is processed by removing columns and rows that don't contain any relevant information.
- Separated the dataset into features (X) and a target variable (y), where y is the FHR diagnosis.
- The dataset is then split into a training set(75%) and a test set(25%), using a random seed = 42 for reproducibility.
- To get a clear representation of the data, I tried Polynomial Kernel PCA to reduce the dimensionality of the feature space. Figure 19 visualizes the three principal components obtained from Kernel PCA in a 3D scatter plot.

4.1 Two machine learning methods for classification and compare their performance

I have used *RandomForestClassifier* and *RidgeClassifier* for classification. **Random forest** is a commonly-used machine learning algorithm that combines the output of multiple decision trees to reach a single result. It has ease of use and flexibility and it handles both classification and regression problems. **Ridge Regression** is a type of linear regression that includes L2 regularization, which avoids model overfitting and penalties on higher terms so as to reduce loss.

Here labels = ["Normal","Suspect", "Pathologic"] are denoted by class 1.0, 2.0, 3.0 consecutively. A comparison of their performance based on the precision, recall, f1-score are as follows -

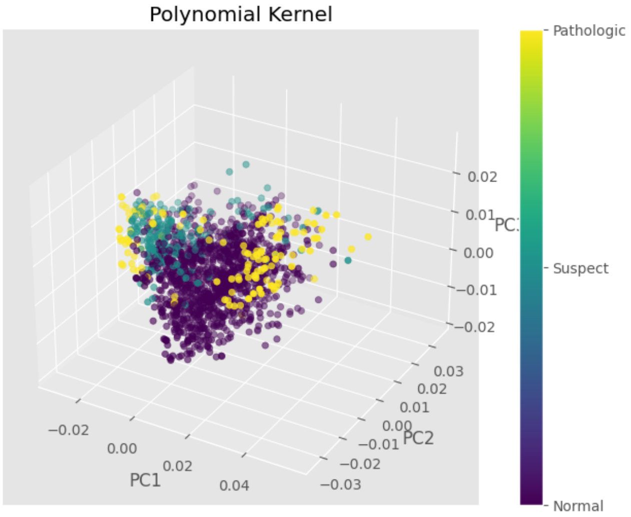

Figure 19: Visual Representation of the dataset

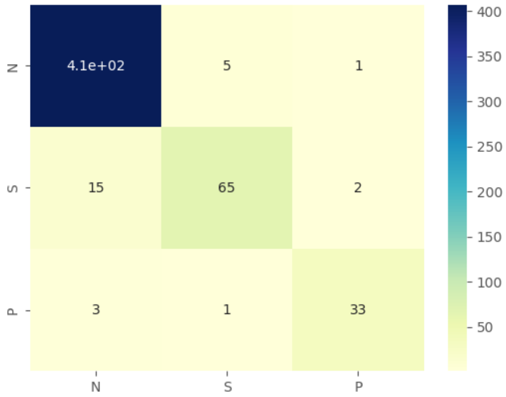

Figure 20: RandomForestClassifier confusion matrix

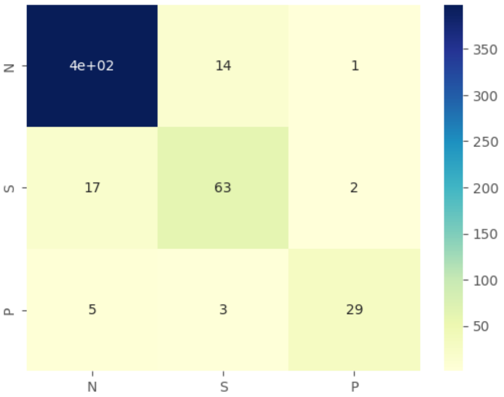

Figure 21: RidgeClassifier confusion matrix

Precision:

RandomForestClassifier has higher precision across classes. It has perfect precision (0.96) for class 1.0 and 0.92 for the others.

RidgeClassifier shows slightly lower precision values, especially for class 2.0, it is 0.79.

Recall:

RandomForestClassifier has high recall values, particularly for class 1.0 (0.99). The recall decreases for other classes but remains fairly high.

RidgeClassifier has lower recall values compared to RandomForest, especially notable for class 3.0 at 0.78.

F1-Score:

RandomForestClassifier shows strong f1-scores, with 0.97 for class 1.0 being the highest.

RidgeClassifier has lower f1-scores, with the highest being 0.96 for class 1.0 and the lowest for class

Table 1: RandomForestClassifier Classification Report

	precision	recall	f1-score	support
1.0	0.96	0.99	0.97	413
2.0	0.92	0.79	0.85	82
3.0	0.92	0.89	0.90	37
accuracy			0.95	532
macro avg	0.93	0.89	0.91	532
weighted avg	0.95	0.95	0.95	532

Table 2: RidgeClassifier Classification Report

	precision	recall	f1-score	support
1.0	0.95	0.96	0.96	413
2.0	0.79	0.77	0.78	82
3.0	0.91	0.78	0.84	37
accuracy			0.92	532
macro avg	0.88	0.84	0.86	532
weighted avg	0.92	0.92	0.92	532

2.0 at 0.78.

Support:
The support, which indicates the number of true instances for each class, is identical for both classifiers (413 for class 1.0, 82 for class 2.0, and 37 for class 3.0).

Overall Metrics:
RandomForestClassifier has an accuracy of 0.95, slightly higher than RidgeClassifier's 0.92. For Macro Avg. RandomForestClassifier outperforms with 0.93 (precision) and 0.89 (recall) compared to RidgeClassifier's 0.88 (precision) and 0.84 (recall). Similarly, RandomForestClassifier's weighted averages are higher in all metrics compared to RidgeClassifier.

For RandomForestClassifier, the confusion matrix indicates good model performance, especially in predicting class 'P' with a high number of true positives and few misclassifications. For RidgeClassifier, there is slightly more misclassifications between classes 'N' and 'P'.

4.2 Explain the selected methods and justify the choice

Random Forest is an ensemble learning method based on decision trees. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) of the individual trees. It inherently handles overfitting well due to this ensemble approach, which averages multiple deep decision trees, each trained on different parts of the same training set. I chose for its robustness and high accuracy, it is particularly useful in situations where the decision boundary is complex and the data features are non-linear. It's also beneficial when we want to reduce the likelihood of overfitting while maintaining high model performance. Now, **Ridge Regression** applies linear regression (also known as Tikhonov regularization) to classification problems. It modifies linear regression by adding a penalty equal to the square of the magnitude of the coefficients to the loss function. This method is particularly useful in cases where there is multicollinearity in the data, or when we want to prevent overfitting which can occur with simple linear regression. It is faster to train than many complex models, making it suitable for very large datasets where training time could be a concern. I chose it for it's speed and simplicity, when dealing with data where predictors are highly correlated. See random-forest, ridge-classifier.

4.3 Discuss the results comparing the performance of the methods along with specifics of their training and testing

To discuss the results comprehensively, I will compare the performance of both classifiers based on their reported metrics in table 1 and 2. First of all, let's know the equation behind Precision, Recall and F1-Score:

$$
\begin{aligned}
\text{Precision} &= \frac{TP}{TP + FP} \\
\text{Recall} &= \frac{TP}{TP + FN} \\
\text{Accuracy} &= \frac{TP + TN}{TP + FP + FN + TN} \\
\text{F1 Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\end{aligned}
$$

Confusion Matrices:

- RandomForestClassifier: The confusion matrix shows fewer misclassifications between classes compared to the other one. The matrix for Random Forest indicates strong performance for the Pathological class with only a few misclassification into Normal and Suspect classes.
- RidgeClassifier: This matrix shows a higher number of misclassifications, especially between the Pathological and Normal classes.

Classification Reports:

- RandomForestClassifier: High precision across all classes, indicating that it is less likely to label as positive when a sample is negative. Recall is very high for class 1.0, indicating excellent identification of this class with very few false negatives. High F1-Score, reflects the balanced performance between precision and recall.
- RidgeClassifier: Precision is similar to RandomForest for class 1.0 but significantly lower for class 2.0, suggesting it struggles with identifying negative samples correctly in this class. Recall

is lower than RandomForest for all classes, with a notable decrease for class 3.0. It has lower F1-Score particularly for class 2.0.

Accuracy and Averages:

- RandomForestClassifier has an overall accuracy of 95%, with a macro average F1-score of 91% and a weighted average of 95%.
- RidgeClassifier shows a lower overall accuracy of 92%, with a macro average F1-score of 86% and a weighted average of 92%.

Specific Calculations:

1. Difference in Precision for Class 1.0:
   - RandomForest: 0.96
   - Ridge: 0.95
   - Difference: $0.96 - 0.95 = 0.01$ or 1% less for Ridge.
2. Difference in Recall for Class 2.0:
   - RandomForest: 0.79
   - Ridge: 0.77
   - Difference: $0.79 - 0.77 = 0.02$ or 2% less for Ridge.
3. Difference in F1-Score for Class 3.0:
   - RandomForest: 0.90
   - Ridge: 0.84
   - Difference: $0.90 - 0.84 = 0.06$ or 6% less for Ridge.
4. Accuracy Difference:
   - RandomForest: 95%
   - Ridge: 92%
   - Difference: $0.95 - 0.92 = 0.03$ or 3% less for Ridge.
5. Weighted Average F1-Score Difference:
   - RandomForest: 0.95
   - Ridge: 0.92
   - Difference: $0.95 - 0.92 = 0.03$ or 3% less for Ridge.

Overall Efficiency Ratio:
We can also look at an efficiency ratio calculated as the ratio of accuracy to training time (hypothetically). Let's say, RandomForest takes 2 units of time to train, Ridge takes 1 unit of time to train. Then, the ratio could be calculated as:

$$\text{RandomForest Efficiency} = \frac{0.95}{2} = 0.475$$
$$\text{Ridge Efficiency} = \frac{0.92}{1} = 0.92$$

So for per unit of time, Ridge regression is more efficient.

Briefly, we need to choose RandomForestClassifier when accuracy is our top priority, and we're dealing with complex and diverse data to reduce misclassification. We need to go for RidgeClassifier for speed when computational resources are a bottleneck.