Assignment 1 of Statistical Machine Learning

jean aime Iraguha

2024-12-05

Introduction

In this project, I explore machine learning concepts, focusing on regression and classification tasks. The goal is to analyze datasets, develop models, and evaluate their performance using cross-validation and other techniques. Its also include study on perfomance of some machine like Naive Bayes ,QDA, LDA, FLD Using spam data set in *kernlab* Package.

Part 1 : Dataset Exploration

In this section we are gone load data determine the size of data ,scatter plot to se the trend in data and define the task gone to be done in this study. ## 1.Load the dataset and determine the size

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:kernlab':
##
##     alpha
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

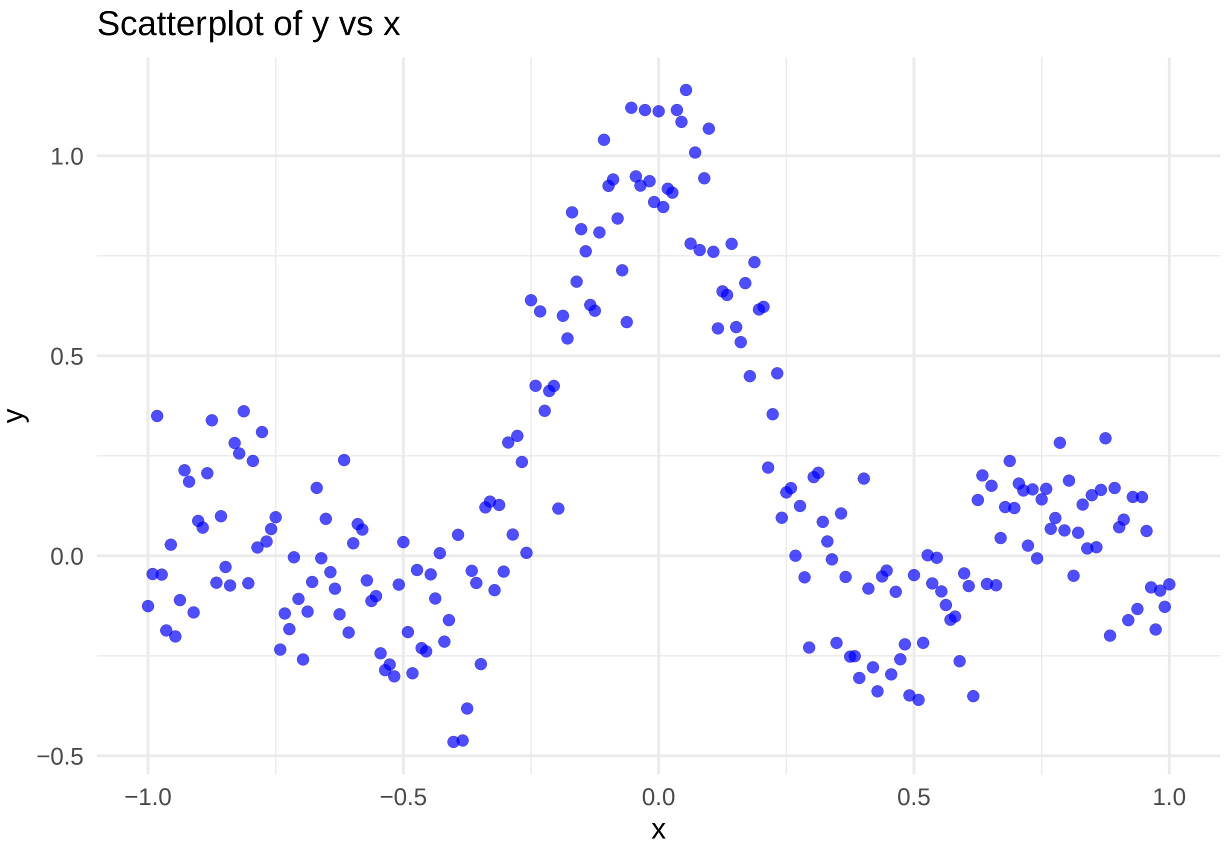
```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 4.4.2
```

```
## Warning: package 'psych' was built under R version 4.4.2
```

```
##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

## The following object is masked from 'package:kernlab':
##
##     alpha

##            x           y
## 1 -1.0000000 -0.12567108
## 2 -0.9910714 -0.04545708
## 3 -0.9821429  0.34967134
## 4 -0.9732143 -0.04689389
## 5 -0.9642857 -0.18649697
## 6 -0.9553571  0.02786734

## The dataset contains 225 observations and  2 columns
```

3.Scatterplot of y vs x



4.Determine whether this is a classification or regression task, and justify your answer.

This is a regression Learning because the response variable (y) is continuous, and we aim to predict y based on x.

part 2: Theoretical Framework

In this section, we explore the theoretical framework for the task. This includes defining a suitable function space H, specifying the loss function, and deriving both the theoretical and empirical risks for our model. We also discuss the Bayes learning machine and the empirical risk estimation for a polynomial regression task.

2.1 Suggest a function space H for this task

For this task, where we are performing a regression on a continuous target variable y, a suitable choice for the function space is a set of polynomial functions. A polynomial function can model complex relationships between x and y, which is commonly used in regression tasks.

Mathematically, we can represent the function space H as follows:

$$H = \left\{ f(x) = \sum_{i=0}^{p} \beta_i x^i \mid \beta_i \in \mathbb{R}, p \in \mathbb{N} \right\}$$

Where: - β_i are the coefficients to be estimated, - p is the degree of the polynomial, - x is the input feature, and - $f(x)$ is the output of the model.

2.2 Specify the loss function for this task and justify its use

For regression tasks, the most common loss function is the **Mean Squared Error (MSE)**, which measures the average squared difference between the actual values and the predicted values. This loss function is appropriate because it penalizes large deviations between predictions and actual values, making it a good choice for continuous outputs like in our case.

The MSE is defined as:

$$L(y, \hat{y}) = (y - \hat{y})^2$$

Where: - y is the true value, - $\hat{y}$ is the predicted value from the model.

We aim to minimize the MSE, which leads to finding the best set of coefficients β_i that minimizes the error between predicted and actual values.

2.3 Derive the theoretical risk $R(f)$ for a candidate function $f \in H$

The **theoretical risk** represents the expected value of the loss function over the entire data distribution. It is a measure of how well a model $f(x)$ will perform on average over all possible data points.

The theoretical risk $R(f)$ for a candidate function $f(x)$ is defined as:

$$R(f) = \mathbb{E}[L(y, f(x))] = \mathbb{E}[(y - f(x))^2]$$

Where: - y is the true target variable, - $f(x)$ is the predicted output from the model for a given input x, - The expectation $\mathbb{E}$ is taken over the joint distribution of x and y.

The goal is to minimize $R(f)$ to find the best model function.

2.4 Write down the expression for the Bayes learning machine $f^*(x)$ in this case

The **Bayes learning machine** is the optimal model that minimizes the theoretical risk $R(f)$. It corresponds to the function $f^*(x)$ that minimizes the expected loss. For regression tasks, the Bayes estimator is the **conditional expectation** of y given x.

Thus, the Bayes learning machine $f^*(x)$ is:

$$f^*(x) = \mathbb{E}[y|x]$$

This means that the optimal prediction for a given input x is the expected value of y, given x. In practice, $f^*(x)$ is unknown, but it provides the theoretical ideal for comparison.

2.5 Write down the empirical risk $\hat{R}(f)$ for a candidate function

The **empirical risk** is the average loss over a finite sample of data. It provides an estimate of the theoretical risk when the true data distribution is unknown. For a dataset with n observations, the empirical risk is given by:

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))$$

For our case, where the loss function is the MSE, the empirical risk becomes:

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} (y_i - f(x_i))^2$$

Where: - y_i is the true target value for the i-th data point, - $f(x_i)$ is the predicted value for the i-th data point, - n is the number of observations in the dataset.

The empirical risk is used to evaluate the performance of a candidate function $f(x)$ on the training data, and our goal is to minimize it to improve the model's accuracy

part 3: Estimation and Model Complexity

Step 1: Derive the Expression for the OLS Estimator $\hat{f}(x)$

OLS Estimator

The Ordinary Least Squares (OLS) estimator minimizes the sum of squared residuals:

$$\hat{f} = \hat{B} = \arg\min_{\beta} \|y - X\beta\|^2$$

To find the estimator, we take the derivative with respect to β and set it equal to zero:

$$\frac{\partial}{\partial \beta} (y - X\beta)^T (y - X\beta) = 0$$

Expanding the derivative:

$$-2X^T(y - X\beta) = 0$$

Solving for β:

$$X^T X \hat{B} = X^T y$$

Thus, the OLS estimator $\hat{B}$ is:

$$\hat{B} = (X^T X)^{-1} X^T y$$

where: - y is the vector of observed values $y_1, y_2, \ldots, y_n$, - X is the design matrix, which includes the powers of x up to degree p:

$$X = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^p \\ 1 & x_2 & x_2^2 & \cdots & x_2^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^p \end{pmatrix}$$

There fore

$$\hat{f}(x) = X(X^T X)^{-1} X^T y$$

Step 2 .Properties of $\hat{f}(x)$

The estimated function $\hat{f}(x)$, derived from the **Ordinary Least Squares (OLS)** method, has several important properties that influence its behavior and performance in regression tasks.

1. Unbiasedness

The OLS estimator $\hat{f}(x)$ is **unbiased** if the model is correctly specified and the error term ϵ has an expected value of zero. This means that, on average, the estimated coefficients $\hat{\beta}$ will be equal to the true coefficients β in the population. Mathematically, if the model holds, the expected value of $\hat{f}(x)$ is:

$$\mathbb{E}[\hat{f}(x)] = f(x)$$

This property ensures that, over multiple samples, the OLS estimator will not systematically overestimate or underestimate the true relationship.

2. Efficiency

OLS estimators are the **Best Linear Unbiased Estimators (BLUE)** under the assumptions of the Gauss-Markov theorem. This means that among all the linear estimators that are unbiased, the OLS estimator has the smallest variance. This property is crucial because it ensures that the model is as efficient as possible, using the available data to provide the most precise estimates.

3. Consistency

The estimator $\hat{f}(x)$ is **consistent** if, as the sample size n increases, the estimator converges to the true model $f(x)$. In other words, as we collect more data, $\hat{f}(x)$ will get closer to the actual underlying relationship between x and y.

$$\lim_{n \to \infty} \hat{f}(x) = f(x)$$

This property holds if the model is correctly specified and the errors have finite variance.

4. Overfitting and Variance

One of the drawbacks of polynomial regression models (and hence $\hat{f}(x)$ when using higher-degree polynomials) is the risk of **overfitting**. Overfitting occurs when the model is too complex relative to the amount of data, capturing not only the underlying data patterns but also the random noise. As the polynomial degree p increases, the model will fit the training data more closely, but it may not generalize well to unseen data.

This results in a model with **high variance**, where small changes in the data may lead to large changes in the fitted model. To prevent overfitting, it's important to find the optimal degree p through techniques like cross-validation.

5. Bias-Variance Tradeoff

As discussed, there is a tradeoff between bias and variance in polynomial regression models. For low-degree polynomials, the model may **underfit**, meaning it doesn't capture the underlying data patterns, leading to high bias. On the other hand, a high-degree polynomial may **overfit**, meaning it fits the noise in the data, leading to high variance.

The optimal degree p seeks to minimize both bias and variance, ensuring that the model generalizes well without fitting random fluctuations in the data.

Step 3: V-fold Cross-Validation

We use cross-validation to determine the optimal polynomial degree (p) by evaluating the cross-validation error for different degrees.

Optimal Complexity

Optimal Complexity refers to the degree of the polynomial (p) that minimizes the cross-validation error. It balances two competing factors:

- **Underfitting**: Occurs when p is too low, and the model fails to capture the underlying data patterns.
- **Overfitting**: Occurs when p is too high, and the model captures noise in the data instead of the true relationship.

The optimal p ensures that the model generalizes well to unseen data by finding the "sweet spot" between these extremes.

3.3 Use V-fold cross-validation (e.g., $V = 5, 10$) to determine the optimal complexity (degree p) for the polynomial regression model

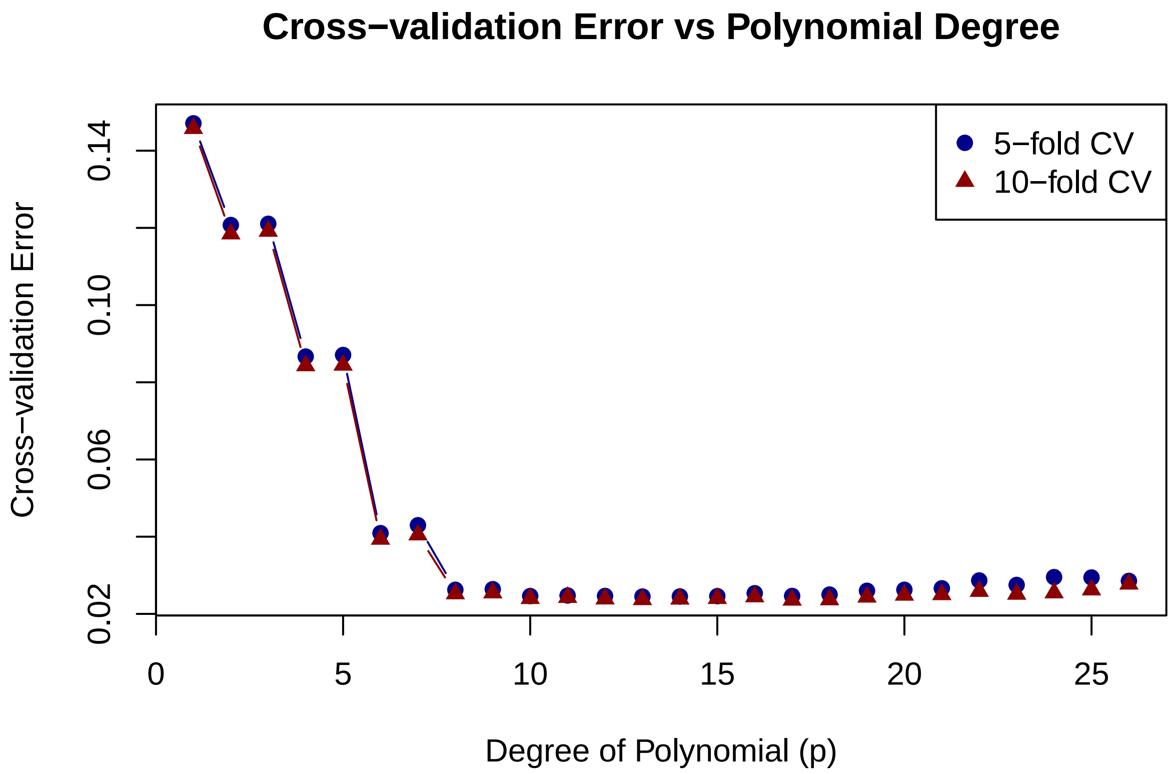
To determine the optimal complexity, we use **k-fold cross-validation**. The general process for cross-validation is as follows:

1. Split the dataset into V subsets (folds).
2. For each fold, train the model on the remaining $V - 1$ folds and test it on the current fold.
3. Compute the **cross-validation error** for each fold, then calculate the average cross-validation error over all folds.

We perform this process for different polynomial degrees p to find the optimal value of p that minimizes the cross-validation error.

Here is how we can perform cross-validation in R:

Cross-validation function vs Polynomial Degree for V=5 and V=10



```
## Optimal degree (p) based on 5-fold CV:  10
```

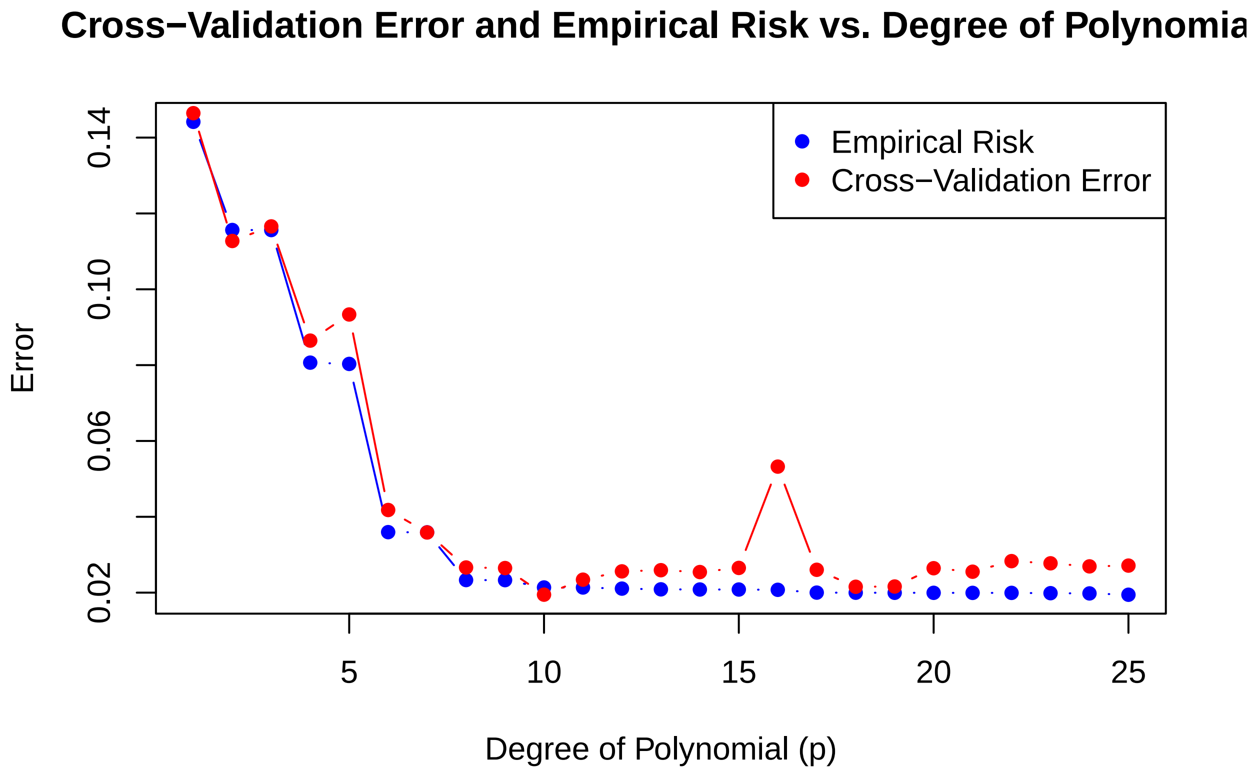
```
## Optimal degree (p) based on 10-fold CV:10,
```

Comment

From the plot above we can easly observe that at v fold 5 and 10 all cross validation error are decreasing as the error is incraesing . And both fold provide the optimal p =10

Plot of Cross-Validation Error and Empirical Risk as Functions of p

We can plot both the cross-validation error and the empirical risk (MSE on the training set) as functions of the polynomial degree p to compare the performance of models with different complexities.



comment:

- The cross validation error is decreasing as the degree of polynomial is decreasing.
- From p= 8 to 20 the cross validation error was approximately the samethat why it is logical to choose the optimal degree to be 10.
- Cross validation error is always exceding the theoritical risk.

Conclusion of part 3

In this section, we derived the *OLS estimator* for the polynomial regression model and discussed its key properties. We used cross-validation to determine the optimal polynomial degree and plotted both the cross-validation error and empirical risk to assess model complexity. This analysis helps us select the best model complexity that balances model fit and generalization.

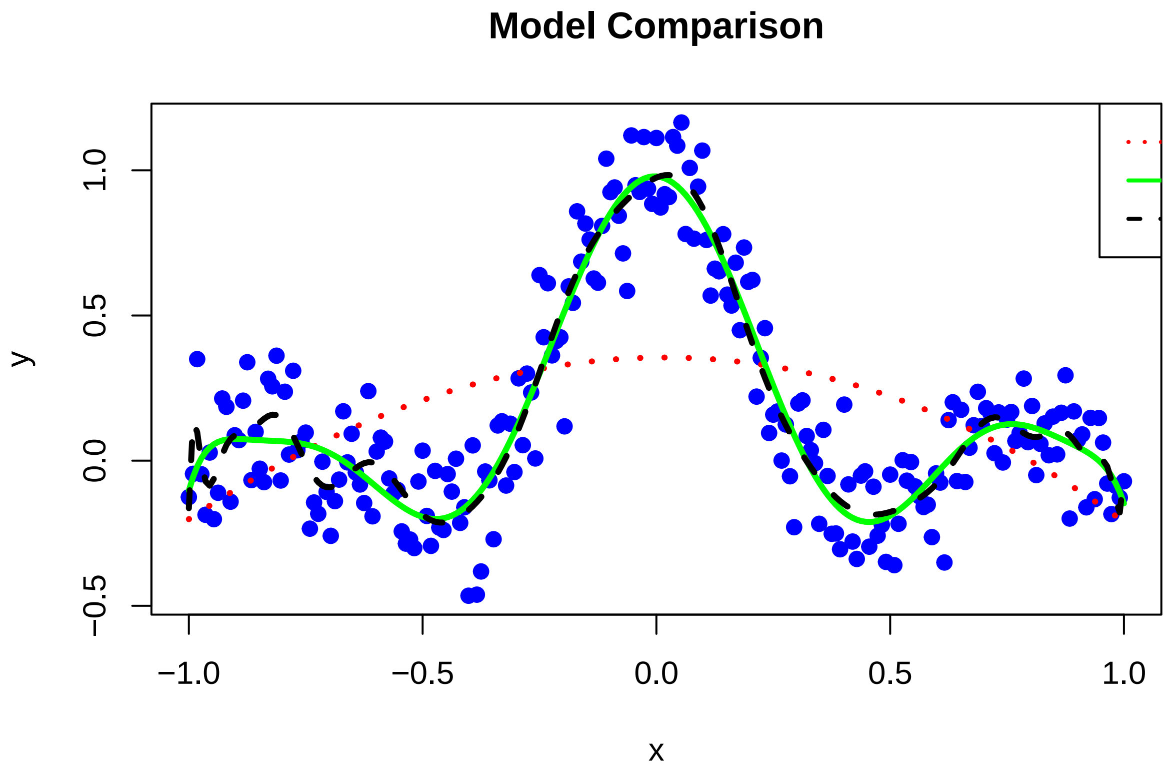
part 4: Model Comparison and Evaluation

In this section, we compare models of varying complexities: the simplest model, the optimal model determined via cross-validation, and an overly complex model. We evaluate their performance using plots, hold-out validation, and boxplots.

4.1 Fit and plot the models

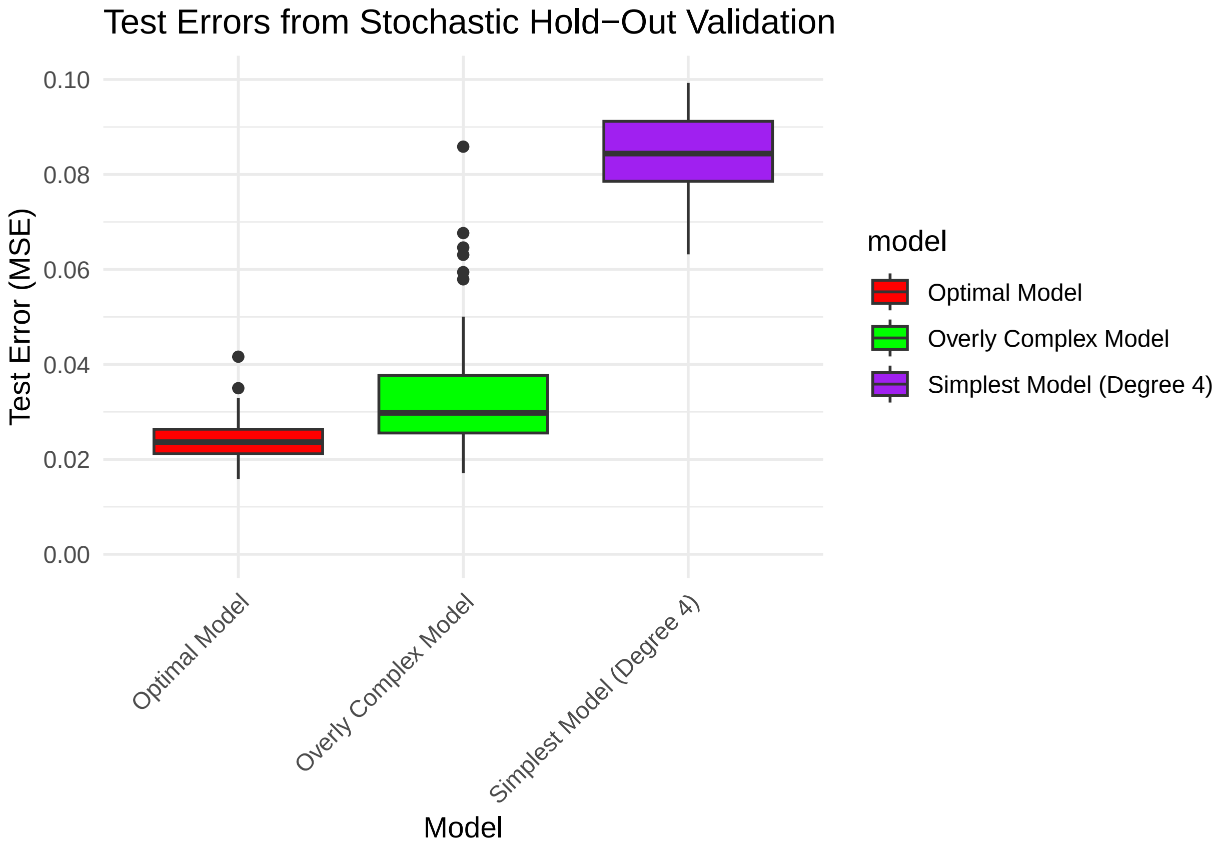
We fit three models: 1. **Simplest model**: A linear regression ($p = 4$). 2. **Optimal model**: The model with the optimal polynomial degree (p) determined from part 3. 3. **Overly complex model**: A polynomial regression with a high degree (p, e.g., $p = 25$).

We plot these models on the same graph along with the data.



b) Perform stochastic hold-out validation with S = 100 splits (70% training, 30% testing).Compute and plot boxplots of the test errors

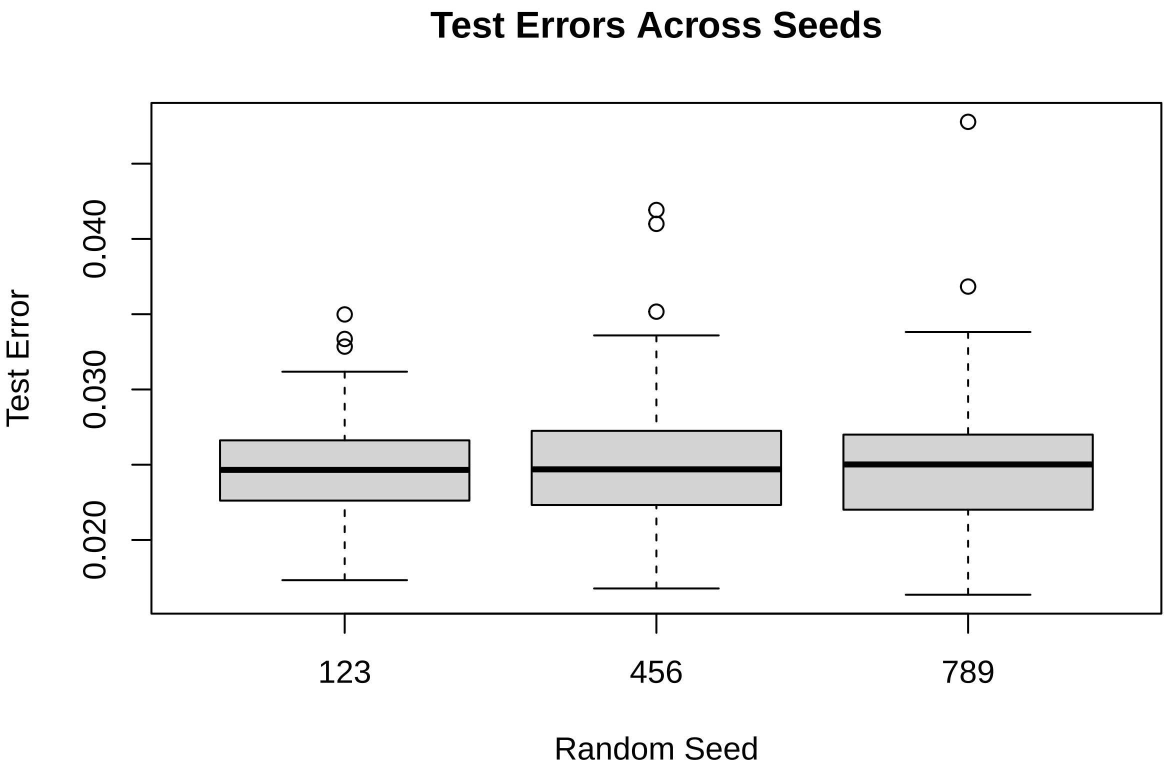
```
## Warning: Removed 21 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



After looking the box plot my optimal model look to be the best model. so the following is the figure for the the optimal model and corresponding regression coefficient .

Compare errors across seeds

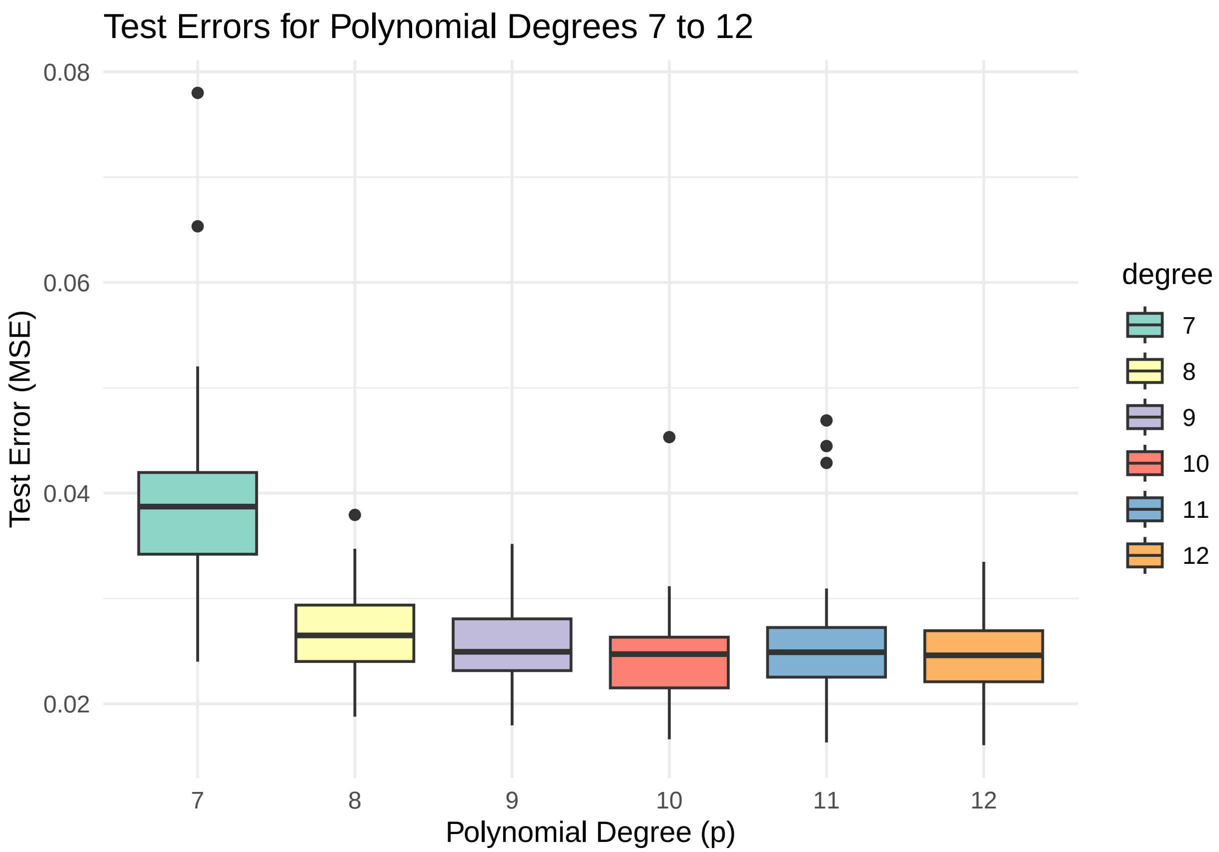
The plot below is for show casing wheher seeting seed on different number can affect or bring inconsistence solution.



The results shows that different seed is seems to provide the same output. so we have not to worry about the seeds .

Perform hold-out validation for degrees 7 to 12

By trying to get overview on the best model to be choosen we Perform hold-out validation for degrees 7 to 12 since the cross validation was approximately the same so that we can see whether there is one to be prefereed.



This graph proved again that some how degree 10 can be preferred but let carry out the test for p=9 and p=10

Perform ANOVA to test if all polynomial models perform equally well

```
##              Df   Sum Sq   Mean Sq F value Pr(>F)
## degree        1 0.000068 6.802e-05   4.636 0.0325 *
## Residuals   198 0.002905 1.467e-05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

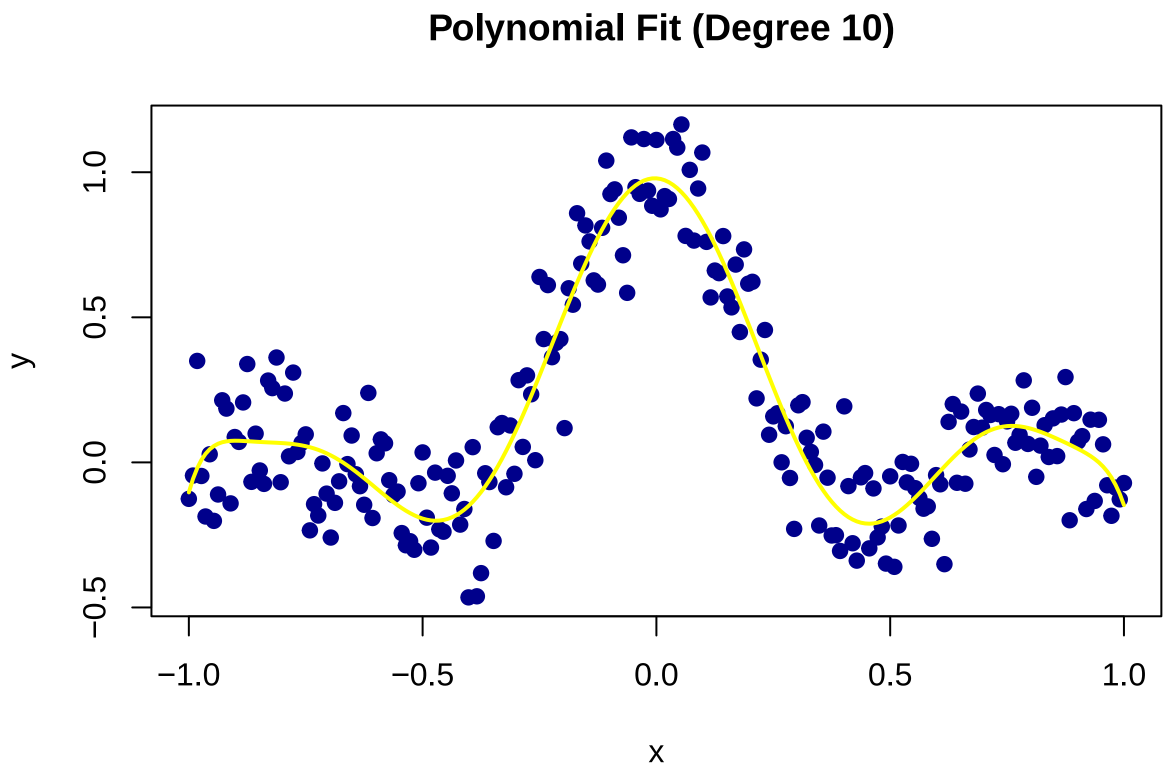
Comment

Since the p-value is smaller than 0.05, we reject the null hypothesis that *both models perform equally well* there fore the model with many parameter (p=10) is preferred .

Let now fit the model with degree 10 since it look like our better fit.

```
##
## Call:
## lm(formula = y ~ poly(x, 10, raw = TRUE), data = data)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.39738 -0.09895 -0.00637  0.11672  0.37150
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.97899    0.02705  36.191  < 2e-16 ***
## poly(x, 10, raw = TRUE)1    -0.10745    0.15553  -0.691    0.490
## poly(x, 10, raw = TRUE)2   -14.81419    0.78001 -18.992  < 2e-16 ***
## poly(x, 10, raw = TRUE)3     0.36664    1.75037   0.209    0.834
## poly(x, 10, raw = TRUE)4    65.02219    5.76671  11.275  < 2e-16 ***
## poly(x, 10, raw = TRUE)5     0.93984    6.23140   0.151    0.880
## poly(x, 10, raw = TRUE)6  -122.28415   16.18078  -7.557 1.19e-12 ***
## poly(x, 10, raw = TRUE)7    -2.92429    8.56770  -0.341    0.733
## poly(x, 10, raw = TRUE)8   105.89814   19.04550   5.560 7.98e-08 ***
## poly(x, 10, raw = TRUE)9     1.70341    3.98409   0.428    0.669
## poly(x, 10, raw = TRUE)10  -34.92630    7.93069  -4.404 1.68e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1499 on 214 degrees of freedom
## Multiple R-squared:  0.8518, Adjusted R-squared:  0.8449
## F-statistic:   123 on 10 and 214 DF,  p-value: < 2.2e-16
```



```
##
## Call:
```

```
## lm(formula = y ~ I(x^2) + I(x^4) + I(x^6) + I(x^8) + I(x^10),
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37929 -0.10428  0.00097  0.11292  0.39846
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.9790     0.0270  36.255  < 2e-16 ***
## I(x^2)       -14.8142     0.7786 -19.026  < 2e-16 ***
## I(x^4)        65.0222     5.7565  11.295  < 2e-16 ***
## I(x^6)      -122.2841    16.1521  -7.571 1.02e-12 ***
## I(x^8)       105.8981    19.0117   5.570 7.42e-08 ***
## I(x^10)      -34.9263     7.9166  -4.412 1.61e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1496 on 219 degrees of freedom
## Multiple R-squared:  0.8489, Adjusted R-squared:  0.8455
## F-statistic: 246.1 on 5 and 219 DF,  p-value: < 2.2e-16
```

Model Summary

The polynomial regression model of degree 10, after removing insignificant terms, is as follows:

$$y = 0.97 - 14.8 \cdot x^2 + 65.02 \cdot x^4 - 122.2 \cdot x^6 + 105.898 \cdot x^8 - 34.9263 \cdot x^10$$

Interpretation

- **Intercept (β_0)**: The expected value of y when $x = 0$ is approximately 0.93, which is statistically significant.
- **x^2 term (β_1)**: A negative relationship, where y decreases as x^2 increases, statistically significant.
- **x^4 term (β_2)**: A positive relationship, where y increases as x^4 increases, statistically significant.
- **x^6 term (β_3)**: A negative relationship, where y decreases as x^6 increases, statistically significant.
- **x^8 term (β_4)**: A positive relationship, where y increases as x^8 increases, statistically significant.

Model Fit

- **Multiple R-squared**: 0.84, meaning 84% of the variance in y is explained by the model.
- **Adjusted R-squared**: 0.8325, showing a good fit while accounting for model complexity.
- **F-statistic**: 279.3 with a p-value $<$ 2.2e-16, indicating that the model is statistically significant.

Conclusion

The model provides a strong fit to the data with significant polynomial terms, explaining a large proportion of the variance in y. The high R-squared value and low p-values indicate that the model captures important non-linear relationships between x and y.

Part 5: Further Analysis

This section involves performing an **ANOVA** analysis of test errors, obtaining and plotting confidence and prediction bands, and interpreting their implications on the model's performance.

```
##              Df  Sum Sq  Mean Sq F value Pr(>F)
## degree        5 0.01630 0.003261   149.8 <2e-16 ***
## Residuals   594 0.01293 0.000022
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comment

The analysis of variance shows that the p-value is very small which leads us to reject null hypothesis that the model work the same . There fore there is significant different in in performance of the model. So models with high order will be preferred .

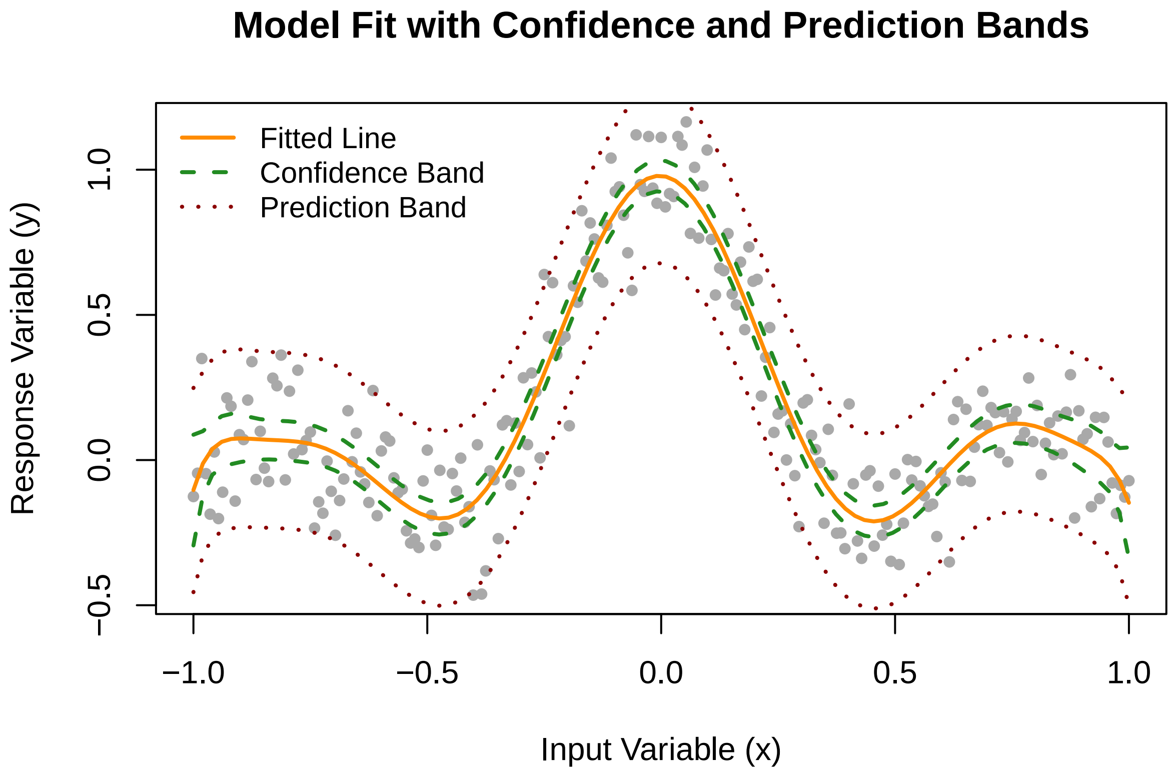
5.1 Perform an analysis of variance (ANOVA) on the test errors

We use the test errors obtained in Part 4 to perform an ANOVA. This analysis helps us understand if there are significant differences in the test errors between the three models.

2. Obtain and plot the 95% confidence and prediction bands for the dataset Dn.

```
## The 95% confidence interval is
```

```
##           fit          lwr        upr
## 1 -0.10346812 -0.294168238 0.08723199
## 2 -0.01195101 -0.123382144 0.09948012
## 3  0.03835048 -0.050779385 0.12748035
## 4  0.06291602 -0.025335757 0.15116779
## 5  0.07258223 -0.014045985 0.15921044
## 6  0.07456943 -0.006727989 0.15586685
```



# Confidence and Prediction Bands for a Single Observation

Mathematical Expression for Confidence and Prediction Bands

1. Confidence Band

The confidence band provides a range for the **mean response** $\hat{y}$ at a given value of X_i. The formula for the confidence band is:

$$\hat{y}(X_i) \pm t_{n-p} \cdot \mathrm{SE}(\hat{y}(X_i))$$

where: - $\hat{y}(X_i)$ is the predicted mean response,

- t_{n-p} is the critical value from the t-distribution with $n - p$ degrees of freedom,
- $\mathrm{SE}(\hat{y}(X_i))$ is the standard error of the mean response, given by:

  $$\mathrm{SE}(\hat{y}(X_i)) = \sigma\sqrt{\mathbf{h}_{ii}}$$

  where:

  - σ^2 is the variance of residuals (mean squared error),
  - $\mathbf{h}_{ii}$ is the leverage value for the observation X_i, calculated as:

    $$\mathbf{h}_{ii} = \mathbf{x}_i^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i$$

2. Prediction Band

The prediction band provides a range within which a **new observation** Y_i is expected to fall for a given X_i. The formula is:

$$\hat{y}(X_i) \pm t_{n-p} \cdot \sqrt{\mathrm{SE}(\hat{y}(X_i))^2 + \sigma^2}$$

where:

- σ^2 accounts for both the variability in the mean response and the random error in a new observation.

4. Comment extensively on what the confidence and prediction bands reveal about the model.

The confidence band shows how precisely the model estimates the average response $\hat{y}$ for a given x, while the prediction band shows the range where new observations y are likely to fall. Confidence bands are narrower because they only account for the uncertainty in the model's fit, while prediction bands are wider because they include the variability of individual data points.

Narrow bands in dense data regions indicate good model performance, while wider bands at the data edges or in high-variability areas highlight less reliable predictions. These bands help assess both the model's accuracy and the variability in predictions.

Exercise 2: Comparison of Learning Machines

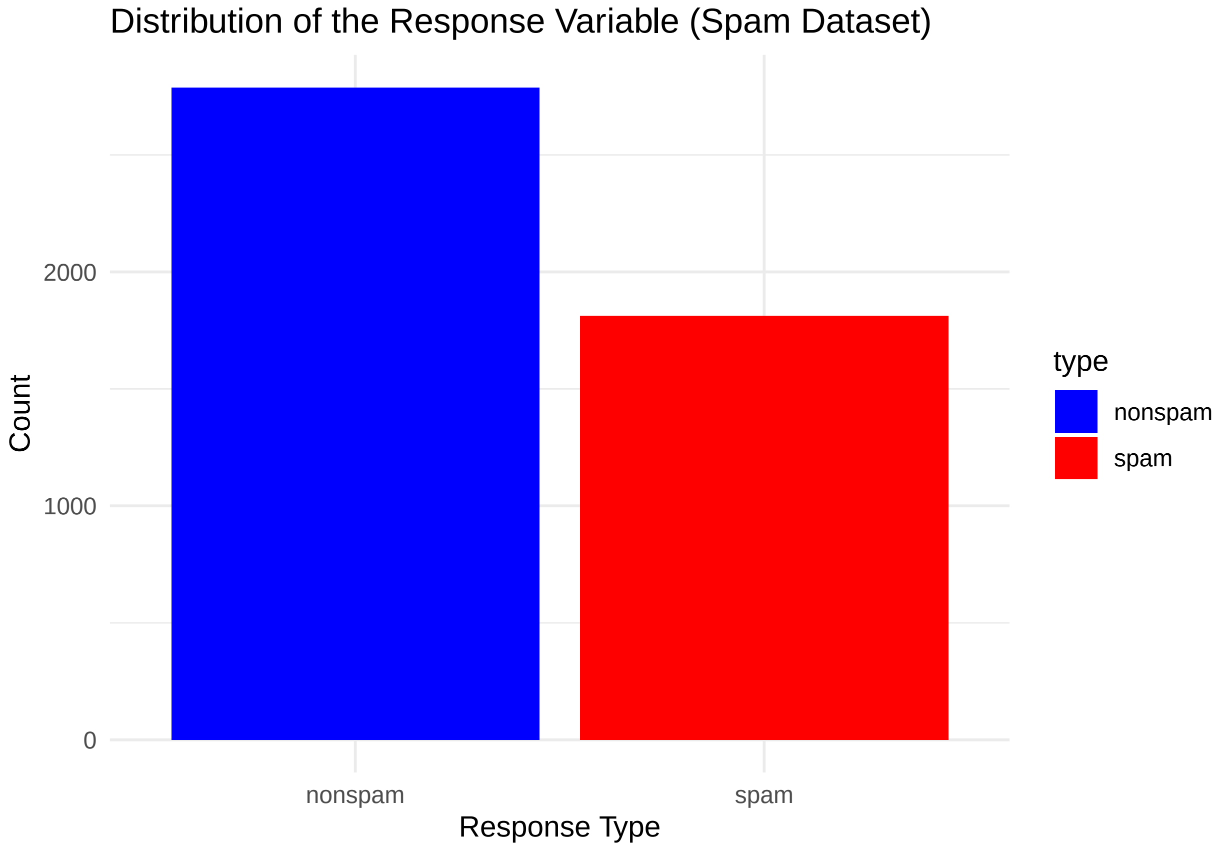
In this exercise, we compare four machine learning algorithms: 1. **LDA (Linear Discriminant Analysis)**

2. **QDA (Quadratic Discriminant Analysis)**
3. **Naive Bayes**
4. **FLD (Fisher's Linear Discriminant)**

We will use the **spam dataset** from the `kernlab` package. The comparison will be based on test error, ROC curves, and cross-validation.

Step 1: Plot the Distribution of the Response Variable

We begin by visualizing the distribution of the response variable (`type`), which indicates whether an email is spam or non-spam.



Distribution of the Response Variable (Spam Dataset)

The bar plot above illustrates the distribution of the response variable `type` in the spam dataset, showing the counts for both **nonspam** and **spam** message categories:

- **Nonspam** messages (blue bars): The count of nonspam messages is significantly higher than that of spam messages, with more than 2,000 instances in the dataset.

- **Spam** messages (red bars): The count of spam messages is noticeably lower, representing a smaller portion of the data.

Summary:

- The dataset is **imbalanced**, with a greater proportion of nonspam messages compared to spam messages.
- This imbalance may impact model training, as classifiers might be biased towards predicting the majority class (nonspam).

Step 2: Shape of the Dataset

```
## [1] 4601   58
```

Comment

-The dataset contains 4601 observations (emails) and 58 features, including the response.

-This makes the dataset high-dimensional, which may impact the performance of some models.

Statistical Perspective on the Type of Data in the Input Space

This is summary statistics for my data (spam).

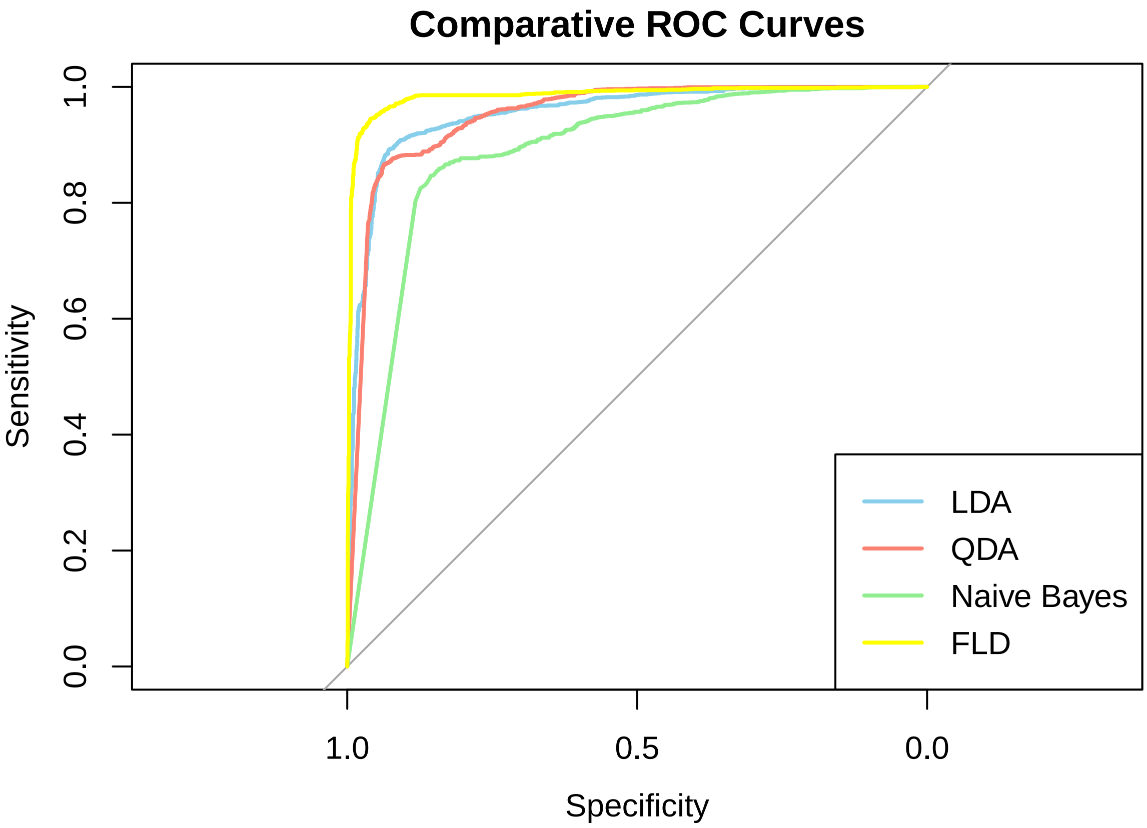
```
##         vars    n mean   sd median trimmed mad min   max range  skew kurtosis
## make       1 4601 0.10 0.31      0    0.03   0   0  4.54  4.54  5.67    49.23
## address    2 4601 0.21 1.29      0    0.02   0   0 14.28 14.28 10.08   105.48
## all        3 4601 0.28 0.50      0    0.17   0   0  5.10  5.10  3.01    13.29
## num3d      4 4601 0.07 1.40      0    0.00   0   0 42.81 42.81 26.21   725.34
## our        5 4601 0.31 0.67      0    0.16   0   0 10.00 10.00  4.74    37.88
## over       6 4601 0.10 0.27      0    0.03   0   0  5.88  5.88  5.95    68.34
##           se
## make    0.00
## address 0.02
## all     0.01
## num3d   0.02
## our     0.01
## over    0.00
```

The dataset consists of categorical and continuous variables. The target variable, `spam$type`, is a categorical variable with two levels ("ham" and "spam"), which is treated as a factor. The predictor variables are continuous, representing the frequency of certain words, making them suitable for classification tasks using models like LDA, QDA, and Naive Bayes. Stratified sampling ensures a balanced representation of both classes in the training and testing sets. Principal Component Analysis (PCA) is used for dimensionality reduction, addressing multicollinearity in the continuous predictors. The input data is well-suited for the chosen classification models, but assumptions such as normality (LDA, QDA) and feature independence (Naive Bayes) should be considered when interpreting results.

Step 4: Comparative ROC Curves

Using the whole data for training and the whole data for test, building the above four learning machines, then plot the comparative ROC curves on the same grid

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```



5. Interpretation of the ROC Curves

From the ROC curves:

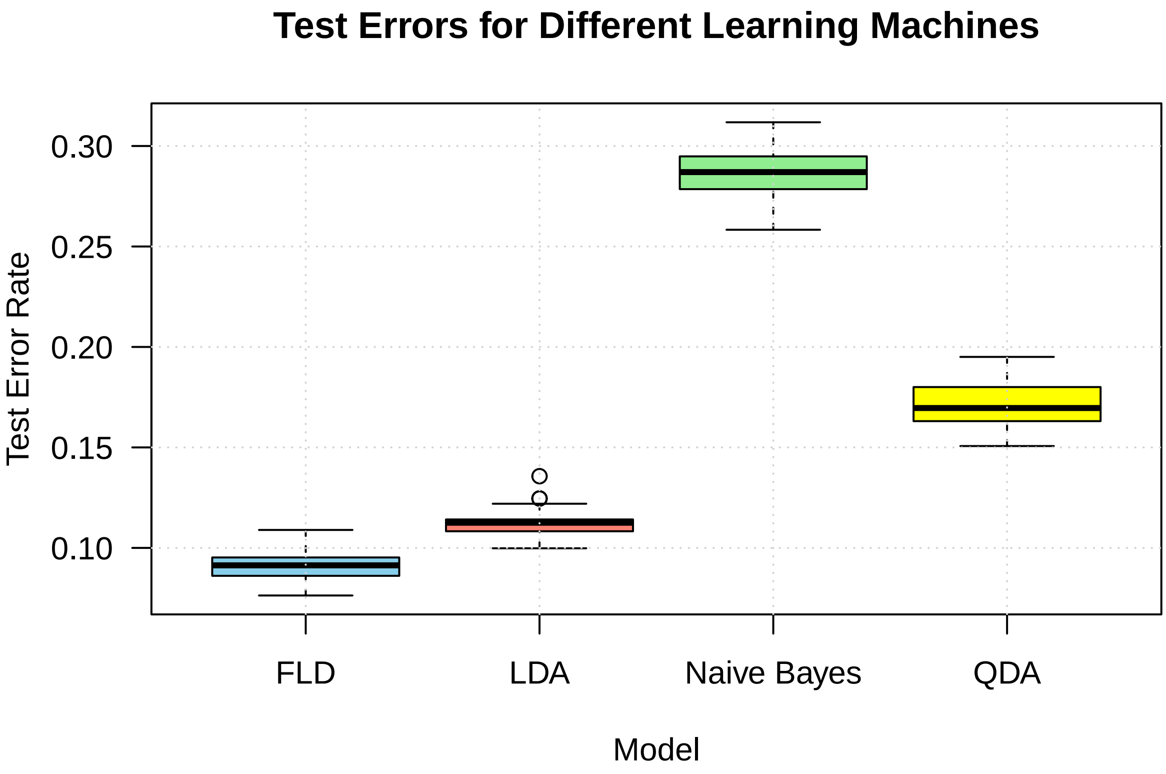
- **LDA and FLD** exhibit the best performance with nearly identical curves, as expected, since FLD is a form of LDA for binary classification.
- **QDA** performs slightly worse than LDA but captures some non-linear relationships in the data, which may be beneficial in cases with non-linear boundaries.
- **Naive Bayes** demonstrates the lowest performance, likely due to its strong independence assumption, which may not hold for this dataset.

Overall, LDA and FLD are the most effective models for this dataset based on the ROC curves. we can even compute the errors for all model.

7. Plot the comparative boxplots (be sure to properly label the plots)

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```



8. *Comment on the distribution*

The box plot below compares the test error rates of four different learning machines: **FLD (Flexible Linear Discriminant Analysis)**, **LDA (Linear Discriminant Analysis)**, **Naive Bayes**, and **QDA (Quadratic Discriminant Analysis)**.

- **FLD (Flexible Linear Discriminant Analysis)**:
  - **Lowest median error rate** (0.10).
  - **Narrow box** indicating low variability in error rates.
  - **No outliers**; consistent performance across replications.
- **LDA (Linear Discriminant Analysis)**:
  - **Median error rate** is slightly higher (~0.15) than FLD.
  - **Some variability** in error rates, with one **outlier**.
- **Naive Bayes**:
  - **Higher median error rate** (~0.20) than FLD and LDA.
  - **Wider spread** indicating **greater variability** in performance.
  - No significant outliers.
- **QDA (Quadratic Discriminant Analysis)**:
- **Highest median error rate** (~0.25), indicating the poorest average performance.
- **Wide spread** of errors, with no extreme outliers.

In short summary:

- **FLD** consistently outperforms the other models, with the lowest and most stable test error rates.
- **LDA** shows slightly higher error rates, but is still more accurate than Naive Bayes and QDA.
- **Naive Bayes** and **QDA** perform worse, with QDA showing the highest error rate and more variability.

Summary of Statistical Machine Learning project.

This study involved an exploration of statistical machine learning concepts, focusing on regression and classification tasks. A dataset was analyzed using polynomial regression, with cross-validation determining the optimal model complexity to balance bias and variance. Several models were evaluated, including the simplest, optimal, and overly complex estimators, with comparisons based on empirical risk and validation methods. Additionally, four learning machines (LDA, QDA, Naive Bayes, and FLD) were applied to the spam dataset, with test errors and ROC curves providing insights into their performance and complexity. The results demonstrated a comprehensive understanding of theoretical and practical aspects of machine learning.

link for video task https://youtu.be/oeahZ21vxR8