**Task1:**

Describe and discuss Fisher’s and Gaussian discriminant analysis learning algorithms.

Your narrative should reflect your understanding of the algorithms, thus, please try to

refrain from using quotations.

**Overview**

This task focuses on two classical supervised learning algorithms for classification: Fisher’s Discriminant Analysis (FDA) and Gaussian Discriminant Analysis (GDA). These methods are widely used in data science for tasks such as pattern recognition, medical diagnosis, and financial modeling. While both algorithms aim to classify data into distinct groups, they differ in their assumptions, mathematical foundations, and applications. This task will explore these differences and discuss the strengths and limitations of each method.

**Fisher’s Discriminant Analysis (FDA)**

Fisher’s Discriminant Analysis is a non-parametric method that does not rely on assumptions about the underlying distribution of the data. Instead, it focuses on finding a linear combination of features that maximally separates the classes in the data. FDA seeks to maximize the ratio of between-class variance to within-class variance. This means it projects the data onto a lower-dimensional space where the separation between classes is maximized, and the spread within each class is minimized.

For a **two-class problem**, FDA finds a discriminant function z=uTxz=uTx that projects the data onto a line. The direction of this line (defined by the vector uu) is chosen to maximize the separation between the projected class means while minimizing the variance within each class. Since the difference between the projected class means ,may be negative we use the squared distance.

For **multi-class problems**, the centers of two classes lie on a line, the centers of the three classes on a two dimensional plane , and the centers of J classes will always lie on a (J-1) dimensional hyperplane. FDA identifies (J−1)(J−1) discriminant functions, where JJ is the number of classes. These functions project the data onto a (J−1)(J−1)-dimensional space, where the classes are maximally separated. The variation within classes is based on the distances between each observation in the class and its class center, and is represented by a p \* p scatter matrix.

**Key Features of (FDA)**

1. FDA does not require the data to follow a specific distribution, making it a non-parametric method.
2. FDA produces linear decision boundaries between classes.
3. FDA inherently reduces the dimensional of the data by projecting it onto a lower-dimensional space.

FDA is widely used in fields like pattern recognition, image processing, and bio informatics, where class separation is critical.

**Gaussian Discriminant Analysis (GDA)**

Gaussian Discriminant Analysis is a parametric classification method that assumes the data in each class follows a multivariate normal distribution(that is Gaussian densities) of the independent variables (Welch 1939). These lead to a solution for which the probability of misclassifications is minimized. It includes two main variants: Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA).

GDA aims to minimize the probability of misclassification by modeling the class-conditional densities using Gaussian distributions. For a given class jj, the data is assumed to follow a multivariate normal distribution with mean μjμj and covariance matrix ΣjΣj​.

**Linear Discriminant Analysis** (LDA): in addition to the multivariate normality assumption , makes the assumption of homogeneity of classes co-variances matrix (Σ1=Σ2=⋯=ΣJΣ1​=Σ2​=⋯=ΣJ). Produces linear decision boundaries between classes. The classification rule assigns an observation to the class with the highest posterior probability.

**Quadratic Discriminant Analysis** (QDA): Assumes the multivariate normality , but does not make the assumptions of homogeneity of class co-variances. Allows each class to have its own covariance matrix (ΣjΣj).Produces quadratic decision boundaries, which can capture more complex relationships between classes. More flexible than LDA but requires more data to estimate the additional parameters.

**Key Features (GDA)**

1. GDA assumes multivariate normality, making it a parametric method.
2. LDA produces linear boundaries, while QDA produces quadratic boundaries.
3. GDA performs well when the normality assumption holds, but it can be less robust to violations of this assumption.

GDA is commonly used in medical diagnosis, finance, and marketing, where the normality assumption is reasonable.

**Task 2:**

Based on the information from the required reading assignments from James et al,

perform the following discriminant analysis experiments:

**Overview**

The Smarket dataset, available in the ISLR package, contains daily percentage returns for the S&P 500 stock index between 2001 and 2005. The dataset includes the following variables:

* Year: The year of the observation.
* Lag1 to Lag5: Percentage returns for the previous 1 to 5 days.
* Volume: The volume of shares traded on the previous day.
* Today: The percentage return on the current day.
* Direction: A binary variable indicating whether the market moved Up or Down.

The goal of this task is to use LDA and QDA to predict the Direction of the market based on the Volume and Lag variables.

**Task 2a: Linear Discriminant Analysis (LDA)**

The first step in this experiment was to load the Smarket dataset and explore its structure. The dataset contains 1,250 observations and 9 variables, including the dependent variable, Direction, which indicates whether the market moved Up or Down on a given day.

The independent variables include Lag1 to Lag5 (percentage returns for the previous 1 to 5 days) and Volume (the volume of shares traded on the previous day). This dataset is not high-dimensional, as the number of observations far exceeds the number of variables.

Most of the variables are continuous, except for Direction, which is a binary categorical variable. There are no missing values in the dataset, which simplifies the data preparation process. The dataset is split into training data (years 2001–2004) and test data (year 2005) to evaluate the performance of the models.

The goal of this task is to use Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) to predict the Direction of the stock market based on the Lag and Volume variables.

R code adapted from lab 4.6 of James et al. (2013):

**Load the Data and Split into Training and Test Sets**

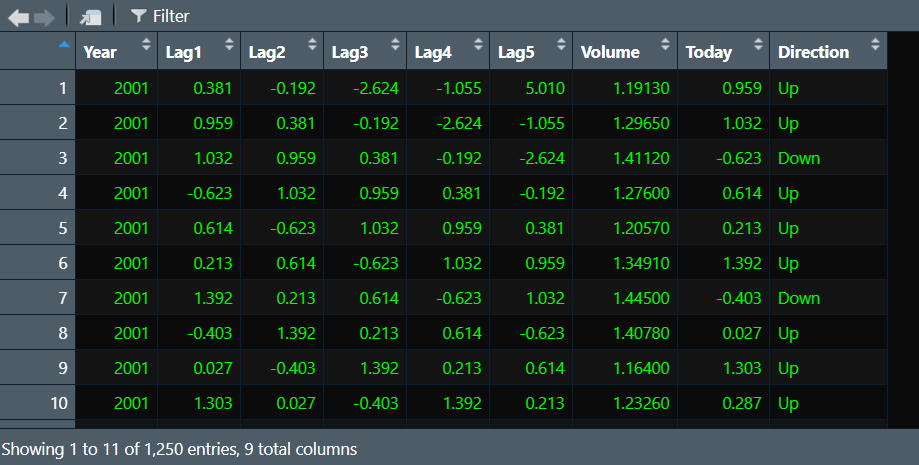
The dataset is split into training data (years 2001–2004) and test data (year 2005).

# Load the ISLR package and Smarket dataset

#install.packages("ISLR")

library(ISLR)

data(Smarket)



The above provided is the sample of the dataset Smarket from ISLR package.

**Split the data into training and test sets**

train <- (Smarket$Year < 2005)

Smarket\_train <- Smarket[train, ]

Smarket\_test <- Smarket[!train, ]

**Build the LDA model**

#lda\_model <- lda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Smarket\_train)

#Lda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Smarket\_train)

Call:

lda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Smarket\_train)

Prior probabilities of groups:

Down Up

0.491984 0.508016

Group means:

Lag1 Lag2 Lag3 Lag4 Lag5 Volume

Down 0.04279022 0.03389409 -0.009806517 -0.010598778 0.0043665988 1.371843

Up -0.03954635 -0.03132544 0.005834320 0.003110454 -0.0006508876 1.363210

Coefficients of linear discriminants:

LD1

Lag1 -0.58081056

Lag2 -0.49111007

Lag3 0.07707664

Lag4 0.06904095

Lag5 -0.04549853

Volume -1.24678716

#Step 3: Make Predictions on the Test Data

#Use the predict() function to classify the test data

# Make predictions on the test data

lda\_predict <- predict(lda\_model, Smarket\_test)

# Display the predicted classes

lda\_predict$class

> lda\_predict$class [1] Up Up Up Up Down Up Up Up Up Up Up Down Up Up Up Up Down Down

[19] Down Up Down Down Down Up Down Down Up Up Up Down Up Up Up Up Up Up

[37] Up Down Down Down Up Down Down Down Down Up Up Up Up Up Up Up Up Down

[55] Up Up Down Down Down Down Down Down Down Down Down Down Down Up Up Down Down Up

[73] Up Down Down Down Down Down Down Down Down Down Down Down Down Down Down Up Down Down

[91] Up Down Up Down Down Down Down Down Down Down Up Down Down Up Down Down Up Up

[109] Down Up Down Down Down Down Down Down Down Down Up Down Up Up Up Down Down Up

[127] Up Down Down Up Down Down Down Down Down Down Up Down Down Down Down Down Down Down

[145] Down Up Up Down Down Up Up Up Down Down Down Down Up Up Up Down Up Down

[163] Up Up Up Up Down Down Down Down Up Down Down Down Down Down Down Up Down Down

[181] Down Down Down Down Down Down Down Down Down Down Down Down Up Down Down Down Down Down

[199] Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down

[217] Down Down Down Down Down Down Down Down Down Down Down Down Up Down Down Down Down Down

[235] Down Down Down Down Down Down Down Down Down Down Down Down Down Down Up Up Up Up

Levels: Down Up

#Step 4: Evaluate the Model

#Use a confusion matrix to evaluate the performance of the LDA model.

# Create a confusion matrix

table(lda\_predict$class, Smarket\_test$Direction)

table(lda\_predict$class, Smarket\_test$Direction)

Down Up

Down 77 97

Up 34 44

# Calculate accuracy, sensitivity, and specificity

#install.packages("caret")

library(caret)

confusionMatrix(lda\_predict$class, Smarket\_test$Direction)

> confusionMatrix(lda\_predict$class, Smarket\_test$Direction)Confusion Matrix and Statistics

Reference

Prediction Down Up

Down 77 97

Up 34 44

Accuracy : 0.4802

95% CI : (0.417, 0.5437)

No Information Rate : 0.5595

P-Value [Acc > NIR] : 0.9952

Kappa : 0.0054

Mcnemar's Test P-Value : 6.062e-08

Sensitivity : 0.6937

Specificity : 0.3121

Pos Pred Value : 0.4425

Neg Pred Value : 0.5641

Prevalence : 0.4405

Detection Rate : 0.3056

Detection Prevalence : 0.6905

Balanced Accuracy : 0.5029

'Positive' Class : Down

The LDA model achieves an accuracy of 48.02%, which is worse than random guessing (50%). This indicates that the model is not effectively capturing the underlying patterns in the data to predict the Direction of the stock market.

**Task 2b: Quadratic Discriminant Analysis (QDA)**

#Build the QDA Model

#qda\_model <- qda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Smarket\_train)

#qda\_model

qda(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Smarket\_train)

Prior probabilities of groups:

Down Up

0.491984 0.508016

Group means:

Lag1 Lag2 Lag3 Lag4 Lag5 Volume

Down 0.04279022 0.03389409 -0.009806517 -0.010598778 0.0043665988 1.371843

Up -0.03954635 -0.03132544 0.005834320 0.003110454 -0.0006508876 1.363210

# Make predictions on the test data

qda\_predict <- predict(qda\_model, Smarket\_test)

# Display the predicted classes

qda\_predict$class

qda\_predict$class [1] Down Up Down Down Down Up Up Up Up Up Up Up Up Down Up Down Up Up

[19] Up Up Down Down Up Up Down Up Up Up Up Down Up Up Up Up Up Up

[37] Up Up Up Down Down Down Down Up Up Up Down Down Up Up Up Down Up Down

[55] Down Down Down Down Down Down Down Down Down Down Down Down Down Up Up Down Down Down

[73] Down Down Down Down Up Down Up Down Down Down Down Down Down Down Down Down Down Down

[91] Down Down Down Up Down Down Down Up Down Down Down Up Up Down Down Down Down Up

[109] Down Down Down Down Up Down Down Down Down Down Down Down Down Down Down Down Up Down

[127] Up Down Down Down Down Down Down Down Down Down Up Down Down Down Down Down Down Down

[145] Down Down Down Down Down Down Down Down Down Down Down Down Up Down Down Down Down Up

[163] Down Down Up Up Up Down Down Down Up Down Down Down Down Down Down Down Down Down

[181] Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down

[199] Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down

[217] Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down Down

[235] Down Down Down Down Down Down Down Down Down Down Down Down Down Down Up Up Up Up

Levels: Down Up

#Use a confusion matrix to evaluate the performance of the QDA model.

# Create a confusion matrix

table(qda\_predict$class, Smarket\_test$Direction)

> # Create a confusion matrix> table(qda\_predict$class, Smarket\_test$Direction)

Down Up

Down 82 111

Up 29 30

# Calculate accuracy, sensitivity, and specificity

confusionMatrix(qda\_predict$class, Smarket\_test$Direction)

> # Calculate accuracy, sensitivity, and specificity> confusionMatrix(qda\_predict$class, Smarket\_test$Direction)Confusion Matrix and Statistics

Reference

Prediction Down Up

Down 82 111

Up 29 30

Accuracy : 0.4444

95% CI : (0.3821, 0.5081)

No Information Rate : 0.5595

P-Value [Acc > NIR] : 0.9999

Kappa : -0.045

Mcnemar's Test P-Value : 7.608e-12

Sensitivity : 0.7387

Specificity : 0.2128

Pos Pred Value : 0.4249

Neg Pred Value : 0.5085

Prevalence : 0.4405

Detection Rate : 0.3254

Detection Prevalence : 0.7659

Balanced Accuracy : 0.4758

'Positive' Class : Down

The QDA model achieves an accuracy of  44.44% on the test data, which is slightly lower than the LDA model. However, like LDA, QDA also struggles to correctly classify the Down direction.

**Comparison of LDA and QDA**

| **Metric** | **LDA** | **QDA** |
| --- | --- | --- |
| **Accuracy** | **48.02%** | **44.44%** |
| **Sensitivity** | **69.37%** | **73.87%** |
| **Specificity** | **31.21%** | **21.28%** |

**Conclusion**

Both LDA and QDA perform poorly in terms of accuracy, with LDA achieving **48.02%**and QDA achieving **44.44%**. These results are below random guessing (50%), indicating that neither model is effective at predicting the Direction of the stock market based on the given features (Lag1 to Lag5 and Volume).

Both models have relatively high sensitivity (**LDA: 69.37%**, **QDA: 73.87%**), meaning they are better at correctly identifying the Up direction. However, this comes at the cost of poor performance in identifying the Down direction

Both models have very low specificity (**LDA: 31.21%**, **QDA: 21.28%**), indicating that they struggle to correctly classify the Down direction. This imbalance suggests that the models are biased toward predicting Up, which may be due to class imbalance or the lack of predictive power in the features.

The poor performance of both LDA and QDA highlights the difficulty of predicting stock market direction using only historical returns and trading volume. While LDA performs slightly better than QDA, neither model is reliable enough for practical use. Future work should focus on improving the model by addressing the limitations outlined above.

**Task 3**

Based on the information from the required reading assignments from Kuhn & Johnson, as well as based on what you’ve learned from Task 2, perform the following:

a) Prepare Grant data for Tasks 3b) and 3c) experiments:

To prepare Grant data for these experiments, you need file unimelb\_training.csv

and script CreateGrantData.R:

# Step 1: Install and load the AppliedPredictiveModeling package

install.packages("AppliedPredictiveModeling")

library(AppliedPredictiveModeling)

# Step 2: Locate the CreateGrantData.R script

scriptLocation()

library(readr)

grantData <- read\_csv("2025/unimelb\_training.csv")

# Step 3: Update the script (see instructions above)

# - Add stringsAsFactors=TRUE to read.csv()

grantData <- read.csv("unimelb\_training.csv", stringsAsFactors=TRUE)

# - Add options(expressions=15000) near the beginning

options(expressions=15000)

# - (Optional) Change cores <- 3 to cores <- 1

library(AppliedPredictiveModeling)

scriptLocation()

# Check the Grant data

head(grantData)

**Why these updates are needed:**

1. StringsAsFactors=TRUE: In R 4.x.x, the default behavior of read.csv() changed so that character columns are no longer automatically converted to factors. This update ensures compatibility with older code that expects factors.

2.Options(expressions=15000): This increases the limit on the number of nested expressions, which is necessary for some operations in the script to run without errors.

3.Disabling Parallel Processing: Parallel processing (doMC) may not work in newer versions of R. Disabling it simplifies the script and avoids potential errors.

**Task 3b: LDA Experiments**

Build and test a Linear Discriminant Analysis (LDA) classification model using the Grant data.