DATA SCIENCE LABORATORY

Term Work

Deepankar Sharma
Roll No: 233512013
Program: MCA
Year: 2025
Semester: 4

Exercise 1: Data Cleaning and Visualization (Air Quality)

Experiment No.: 1

Date:

Problem Definition:

Identify and handle missing values in the Air Quality dataset and visualize pollution trends.

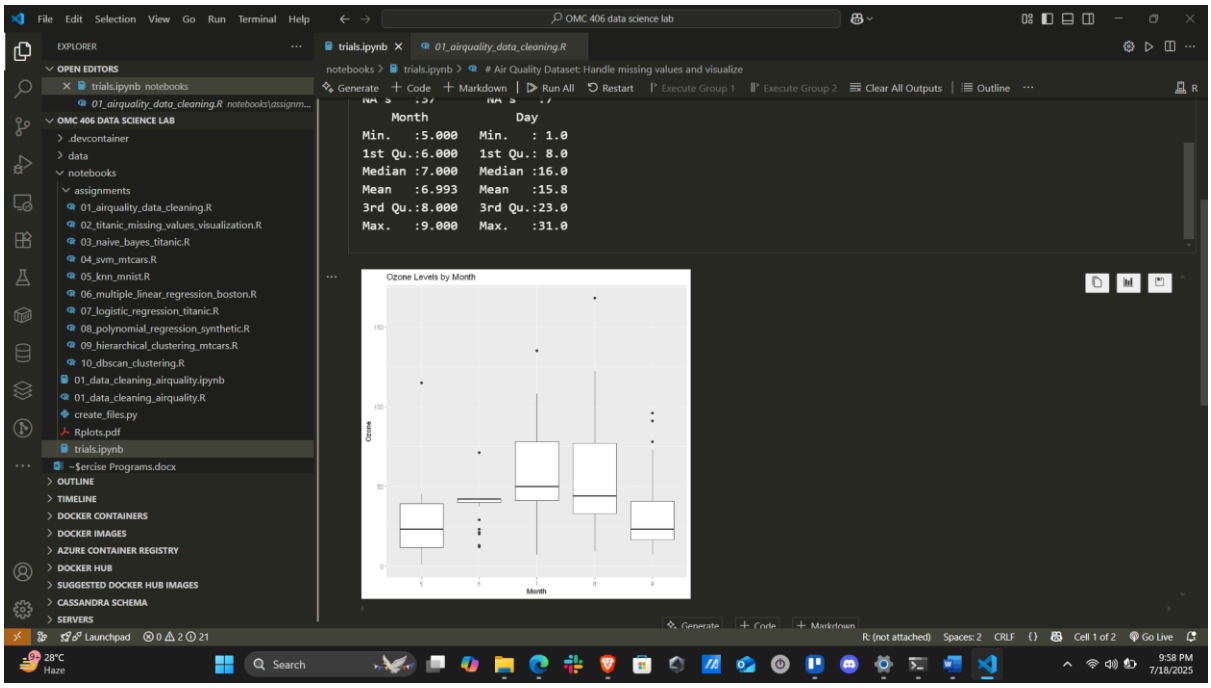
Theory Background:

- **Missing Values Handling:** Removal or imputation (mean/median).
- **Visualization:** Line charts, histograms, and boxplots.

R Program:

```
data(airquality)
summary(airquality)
airquality$Ozone[is.na(airquality$Ozone)] <-
mean(airquality$Ozone, na.rm = TRUE)
airquality$Solar.R[is.na(airquality$Solar.R)] <-
mean(airquality$Solar.R, na.rm = TRUE)
library(ggplot2)
ggplot(airquality, aes(x = factor(Month), y = Ozone)) +
  geom_boxplot() +
  labs(title = "Ozone Levels by Month", x = "Month", y =
"Ozone")
```

OUTPUT :


File Edit Selection View Go Run Terminal Help
OMC 406 data science lab
EXPLORER
OPEN EDITORS
trials.ipynb notebooks
01_airquality_data_cleaning.R notebooks\assignm...
OMC 406 DATA SCIENCE LAB
.devcontainer
data
notebooks
assignments
01_airquality_data_cleaning.R
02_titanic_missing_values_visualization.R
03_naive_bayes_titanic.R
04_svm_mtcars.R
05_knn_mnist.R
06_multiple_linear_regression_boston.R
07_logistic_regression_titanic.R
08_polynomial_regression_synthetic.R
09_hierarchical_clustering_mtcars.R
10_dbscan_clustering.R
01_data_cleaning_airquality.ipynb
01_data_cleaning_airquality.R
create_files.py
Rplots.pdf
trials.ipynb
~$ercise Programs.docx
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
SUGGESTED DOCKER HUB IMAGES
CASSANDRA SCHEMA
SERVERS
trials.ipynb
01_airquality_data_cleaning.R
notebooks > trials.ipynb > # Air Quality Dataset: Handle missing values and visualize
Generate
Code
Markdown
Run All
Restart
Execute Group 1
Execute Group 2
Clear All Outputs
Outline
Month Day
Min. :5.000 Min. : 1.0
1st Qu.:6.000 1st Qu.: 8.0
Median :7.000 Median :16.0
Mean :6.993 Mean :15.8
3rd Qu.:8.000 3rd Qu.:23.0
Max. :9.000 Max. :31.0
Ozone Levels by Month
Ozone
Month
Launchpad
R: (not attached)
Spaces: 2
CRLF
Cell 1 of 2
Go Live
28°C
Haze
Search
9:58 PM
7/18/2025

Exercise 2: Data Cleaning and Visualization (Titanic Dataset)

Experiment No.: 2

Date:

Problem Definition:

Handle missing values in the Titanic dataset and visualize survival patterns.

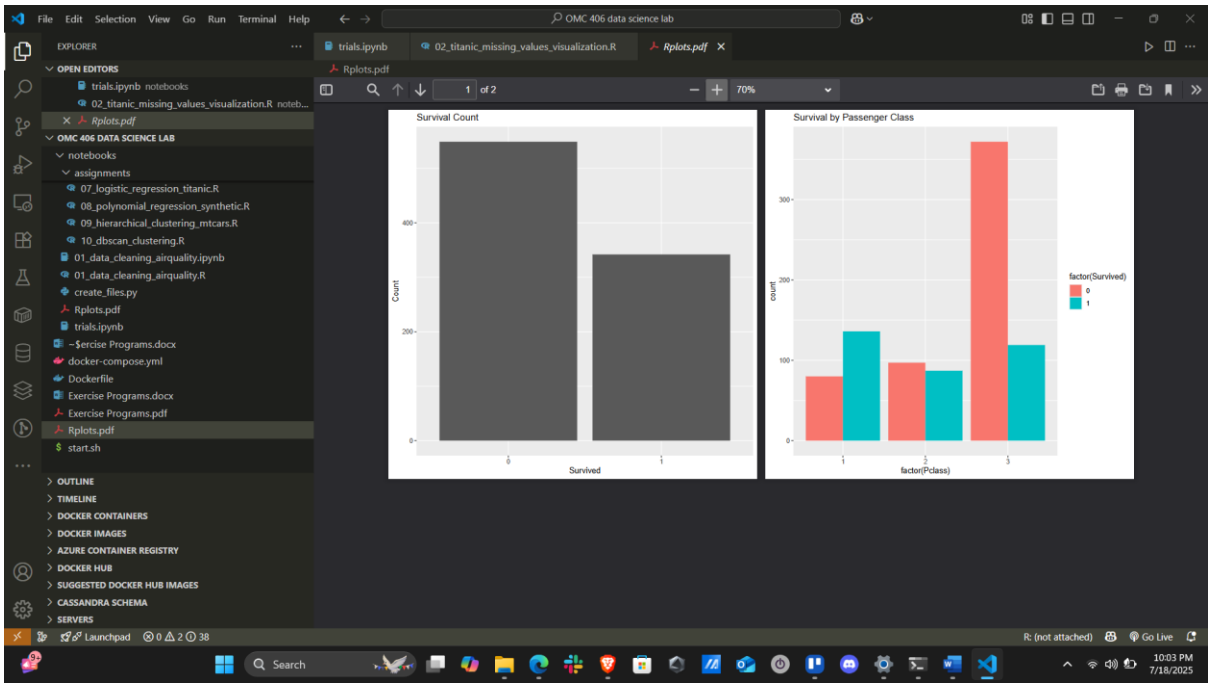
Theory Background:

- **Missing Values Handling:** Imputation, dropping missing data.
- **Visualization:** Bar plots, pie charts, histograms.

R Program:

```
# install.packages("titanic")
library(titanic)
data <- titanic_train

# Check missing values
summary(data)

# Impute missing age with mean
data$Age[is.na(data$Age)] <- mean(data$Age, na.rm=TRUE)

# Drop unnecessary columns
data <- subset(data, select = -c(Cabin))

# Visualize survival
library(ggplot2)
ggplot(data, aes(x = factor(Survived))) + geom_bar() +
  labs(title="Survival Count", x="Survived", y="Count")

ggplot(data, aes(x = factor(Pclass), fill = factor(Survived)))
+
  geom_bar(position = "dodge") + labs(title="Survival by
Passenger Class")
```

OUTPUT


File Edit Selection View Go Run Terminal Help
OMC 406 data science lab
EXPLORER
OPEN EDITORS
trials.ipynb notebooks
02_titanic_missing_values_visualization.R
Rplots.pdf
OMC 406 DATA SCIENCE LAB
notebooks
assignments
07_logistic_regression_titanic.R
08_polynomial_regression_synthetic.R
09_hierarchical_clustering_mtcars.R
10_dbscan_clustering.R
01_data_cleaning_airquality.ipynb
01_data_cleaning_airquality.R
create_files.py
Rplots.pdf
trials.ipynb
~$ercise Programs.docx
docker-compose.yml
Dockerfile
Exercise Programs.docx
Exercise Programs.pdf
Rplots.pdf
start.sh
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
SUGGESTED DOCKER HUB IMAGES
CASSANDRA SCHEMA
SERVERS
Rplots.pdf
1 of 2
70%
Survival Count
Count
Survived
Survival by Passenger Class
count
factor(Pclass)
factor(Survived)
0
1
Launchpad
R: (not attached)
Go Live
Search
10:03 PM
7/18/2025

Exercise 3: Naïve Bayes Classifier (Titanic)

Experiment No.: 3

Date:

Problem Definition:

Predict survival using Naïve Bayes Classifier.

Theory Background:

- Naïve Bayes algorithm.
- Use of the e1071 package.

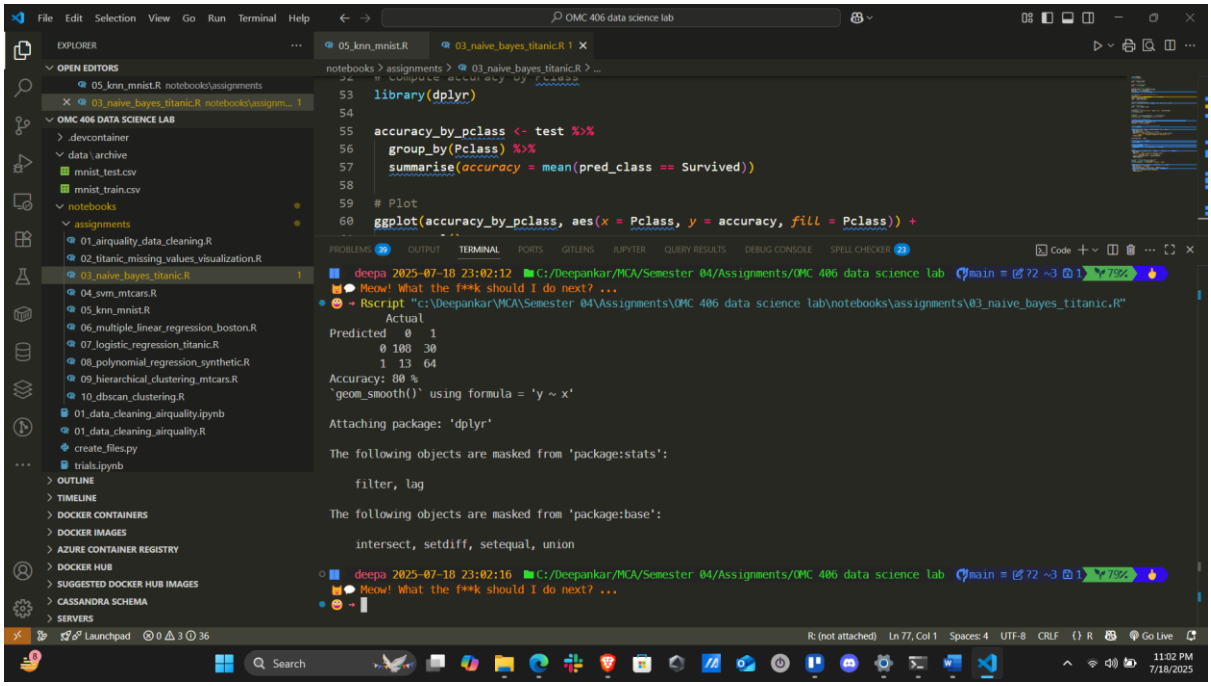
R Program:

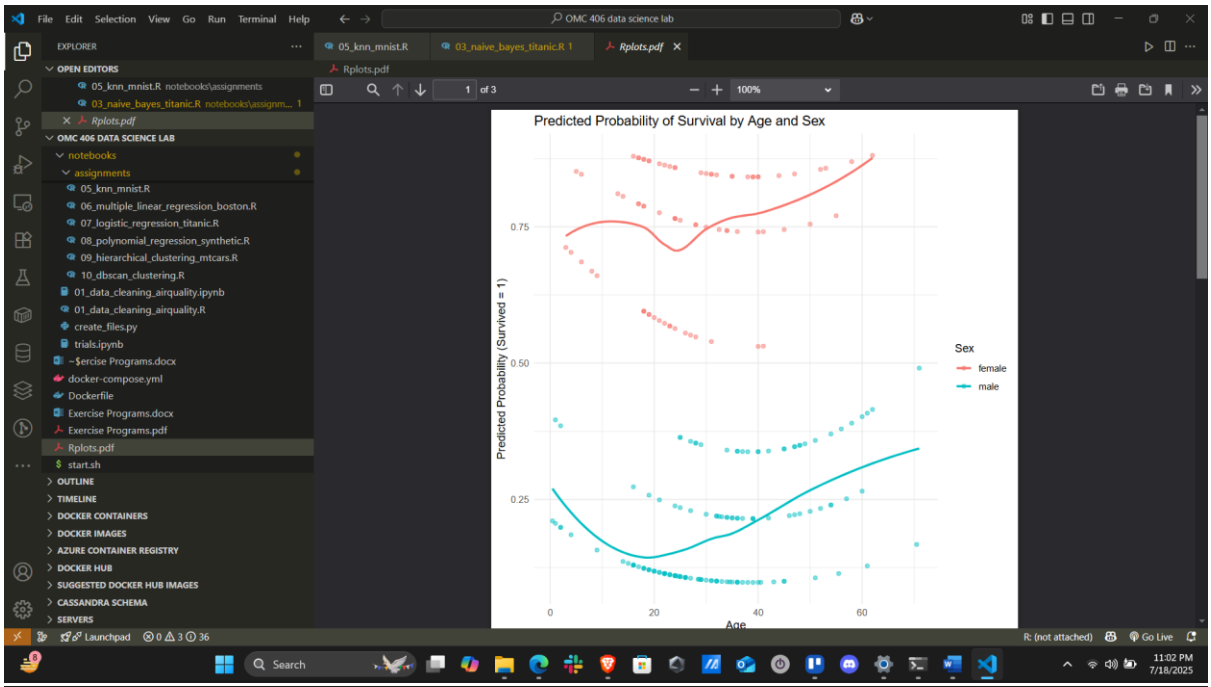
```
library(e1071)
library(titanic)

# Load Titanic dataset
data <- titanic_train

# Remove missing values
data <- na.omit(data)

# Convert variables to factors
data$Survived <- factor(data$Survived)
data$Sex <- factor(data$Sex)
data$Pclass <- factor(data$Pclass)

# Train-test split (70% train, 30% test)
set.seed(123)
train_idx <- sample(1:nrow(data), 0.7 * nrow(data))
train <- data[train_idx, ]
test <- data[-train_idx, ]

# Train Naive Bayes model
model <- naiveBayes(Survived ~ Pclass + Sex + Age, data =
train)

# Make predictions
pred <- predict(model, test)
```

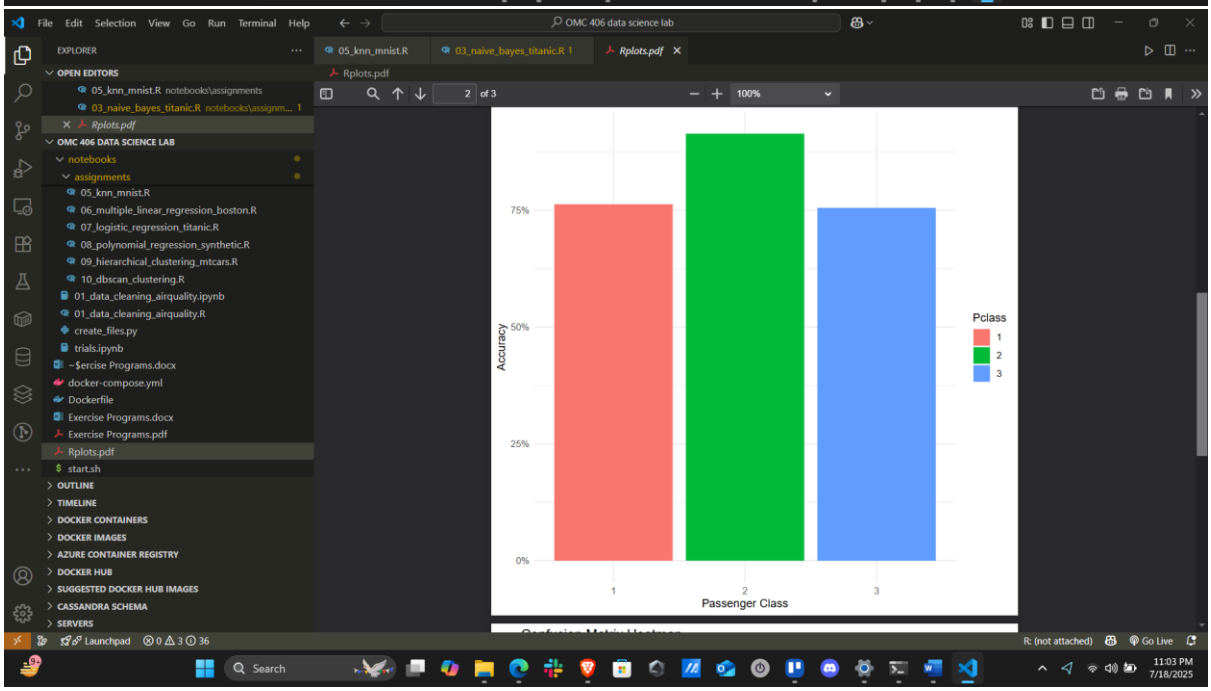
```r
# Confusion Matrix
conf_matrix <- table(Predicted = pred, Actual = test$Survived)
print(conf_matrix)

# Accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
cat("Accuracy:", round(accuracy * 100, 2), "%\n")

# Predict class probabilities
prob_pred <- predict(model, test, type = "raw")  # returns
matrix with prob for 0 and 1
test$prob_survived <- prob_pred[, "1"]
test$Sex <- factor(test$Sex)

# Plot: Probability of Survival by Age and Sex
library(ggplot2)

ggplot(test, aes(x = Age, y = prob_survived, color = Sex)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Predicted Probability of Survival by Age and
Sex",
       x = "Age", y = "Predicted Probability (Survived = 1)") +
  theme_minimal()

test$pred_class <- pred

# Compute accuracy by Pclass
library(dplyr)

accuracy_by_pclass <- test %>%
  group_by(Pclass) %>%
  summarise(accuracy = mean(pred_class == Survived))

# Plot
ggplot(accuracy_by_pclass, aes(x = Pclass, y = accuracy, fill =
Pclass)) +
  geom_col() +
```

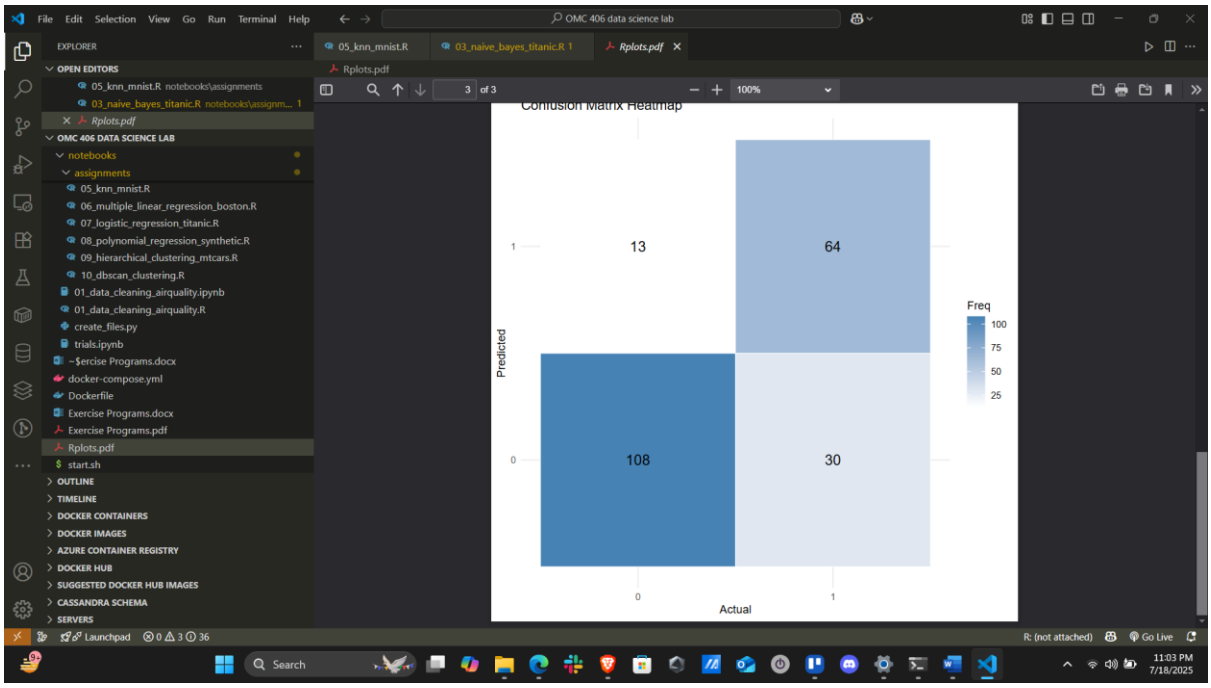
```r
  labs(title = "Model Accuracy by Passenger Class",
       x = "Passenger Class", y = "Accuracy") +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal()

# Convert confusion matrix to data frame
conf_df <- as.data.frame(conf_matrix)
colnames(conf_df) <- c("Predicted", "Actual", "Freq")

ggplot(conf_df, aes(x = Actual, y = Predicted, fill = Freq)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Freq), color = "black", size = 5) +
  scale_fill_gradient(low = "white", high = "steelblue") +
  labs(title = "Confusion Matrix Heatmap") +
  theme_minimal()
```

OUTPUT




Predicted Probability of Survival by Age and Sex
Predicted Probability (Survived = 1)
0.75
0.50
0.25
0
20
40
60
Age
Sex
female
male


Accuracy
75%
50%
25%
0%
1
2
3
Passenger Class
Pclass
1
2
3


Confusion Matrix Heatmap
13
64
108
30
Predicted
Actual
Freq
100
75
50
25
0
1
0
1

Exercise 4: Support Vector Machine (SVM) (mtcars)

Experiment No.: 4

Date:

Problem Definition:

Train SVM model to classify cars based on automatic/manual transmission.

Theory Background:

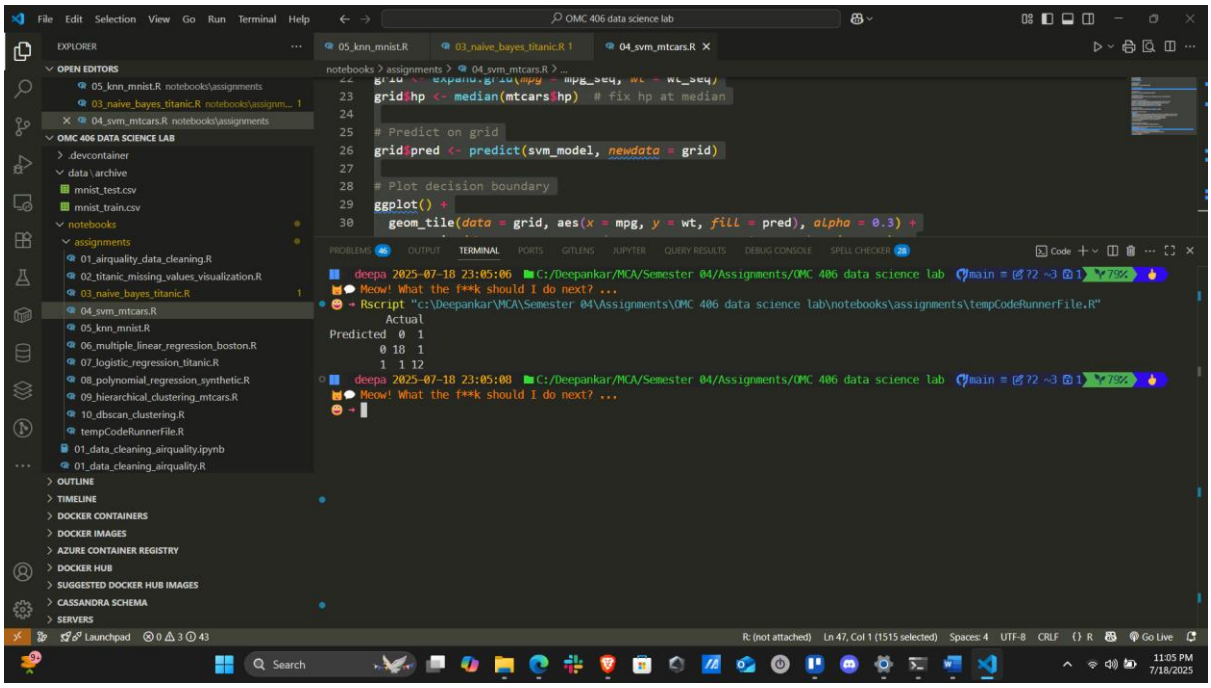
- SVM theory.
- e1071 package implementation.

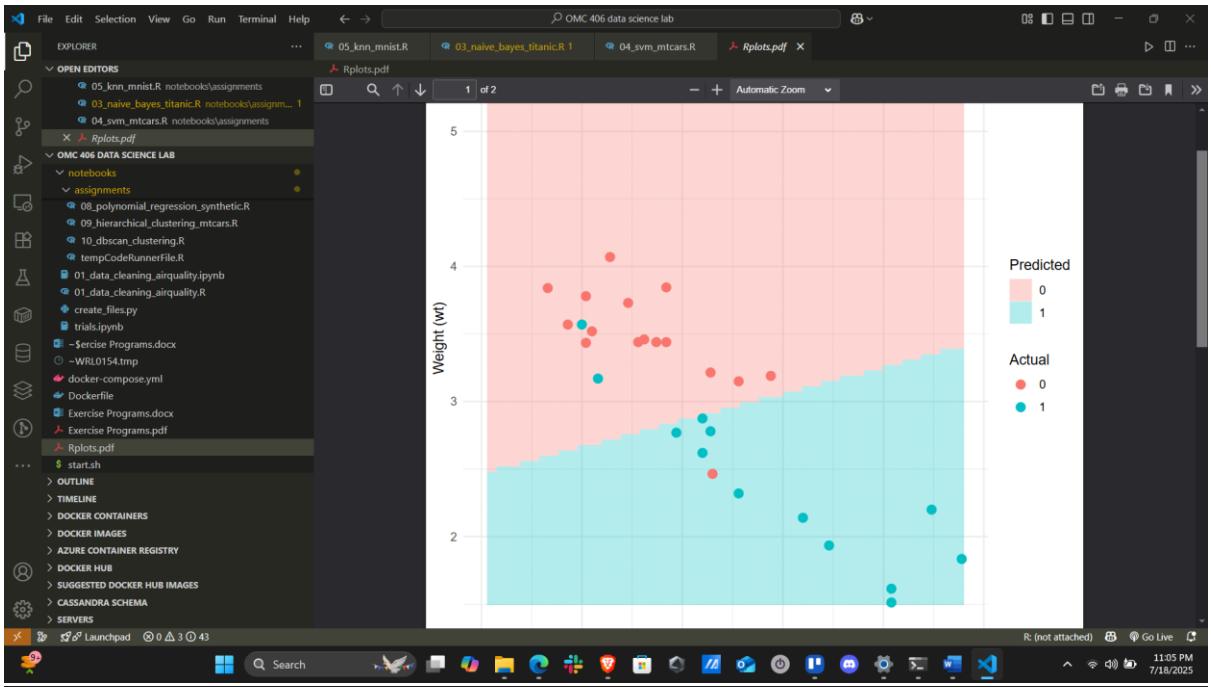
R Program:

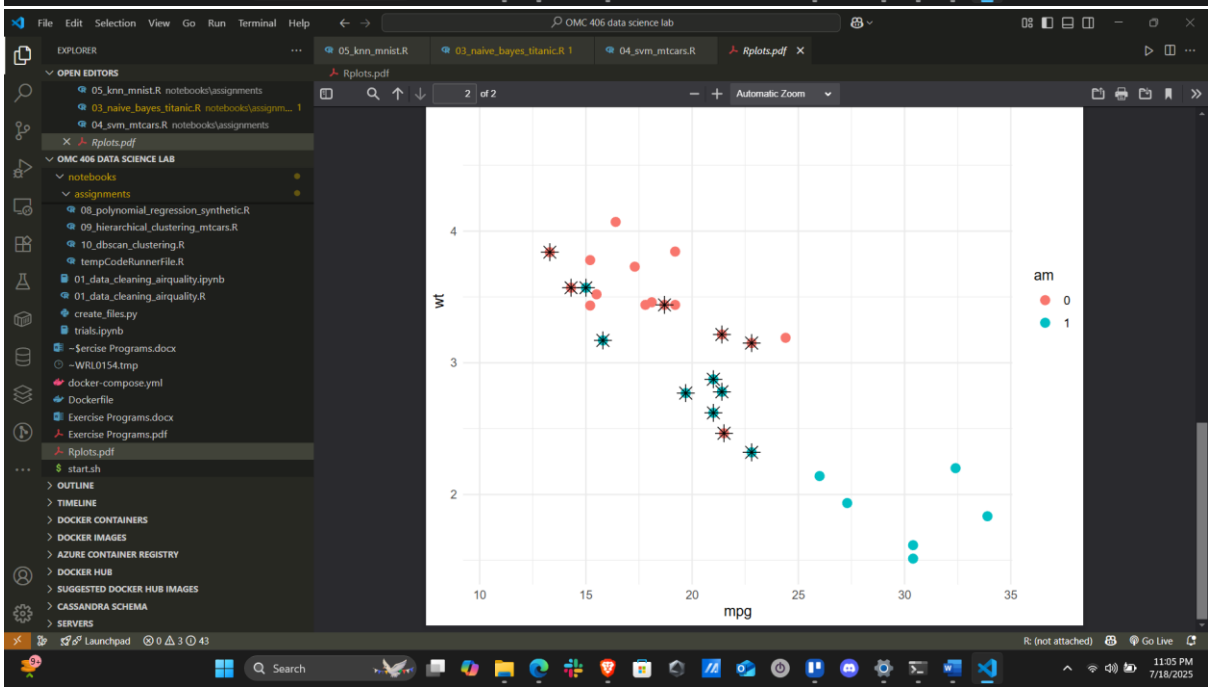
```
# Load packages
library(e1071)
library(ggplot2)

# Prepare data
data(mtcars)
mtcars$am <- factor(mtcars$am)

# Train SVM model
svm_model <- svm(am ~ mpg + hp + wt, data = mtcars, kernel =
"linear")  # using linear kernel for interpretability

# Predictions
pred_svm <- predict(svm_model, mtcars)

# Confusion matrix
conf_matrix <- table(Predicted = pred_svm, Actual = mtcars$am)
print(conf_matrix)

# Create prediction grid
mpg_seq <- seq(min(mtcars$mpg), max(mtcars$mpg), length = 100)
wt_seq <- seq(min(mtcars$wt), max(mtcars$wt), length = 100)
grid <- expand.grid(mpg = mpg_seq, wt = wt_seq)
grid$hp <- median(mtcars$hp)  # fix hp at median

# Predict on grid
```

```r
grid$pred <- predict(svm_model, newdata = grid)

# Plot decision boundary
ggplot() +
  geom_tile(data = grid, aes(x = mpg, y = wt, fill = pred),
alpha = 0.3) +
  geom_point(data = mtcars, aes(x = mpg, y = wt, color = am),
size = 3) +
  labs(title = "SVM Classification: Transmission (am)",
       x = "Miles per Gallon (mpg)", y = "Weight (wt)",
       fill = "Predicted", color = "Actual") +
  theme_minimal()

# Extract support vectors
support_vectors <- mtcars[svm_model$index, ]

ggplot(mtcars, aes(x = mpg, y = wt, color = am)) +
  geom_point(size = 3) +
  geom_point(data = support_vectors, aes(x = mpg, y = wt),
shape = 8, size = 4, color = "black") +
  labs(title = "Support Vectors Highlighted",
       subtitle = "Black stars are support vectors",
       x = "mpg", y = "wt") +
  theme_minimal()
```

OUTPUT


grid$hp <- median(mtcars$hp) # fix hp at median
Predict on grid
grid$pred <- predict(svm_model, newdata = grid)
Plot decision boundary
ggplot() +
geom_tile(data = grid, aes(x = mpg, y = wt, fill = pred), alpha = 0.3) +
deepa 2025-07-18 23:05:06 C:/Deepankar/MCA/Semester 04/Assignments/OMC 406 data science lab main
Meow! What the f**k should I do next? ...
Rscript "c:\Deepankar\MCA\Semester 04\Assignments\OMC 406 data science lab\notebooks\assignments\tempCodeRunnerFile.R"
Actual
Predicted 0 1
0 18 1
1 1 12
deepa 2025-07-18 23:05:08 C:/Deepankar/MCA/Semester 04/Assignments/OMC 406 data science lab main
Meow! What the f**k should I do next? ...


Weight (wt)
Predicted
0
1
Actual
0
1


wt
mpg
am
0
1

Exercise 5: k-Nearest Neighbors (MNIST)

Experiment No.: 5

Date:

Problem Definition:

Classify handwritten digits using k-NN classifier.

Theory Background:

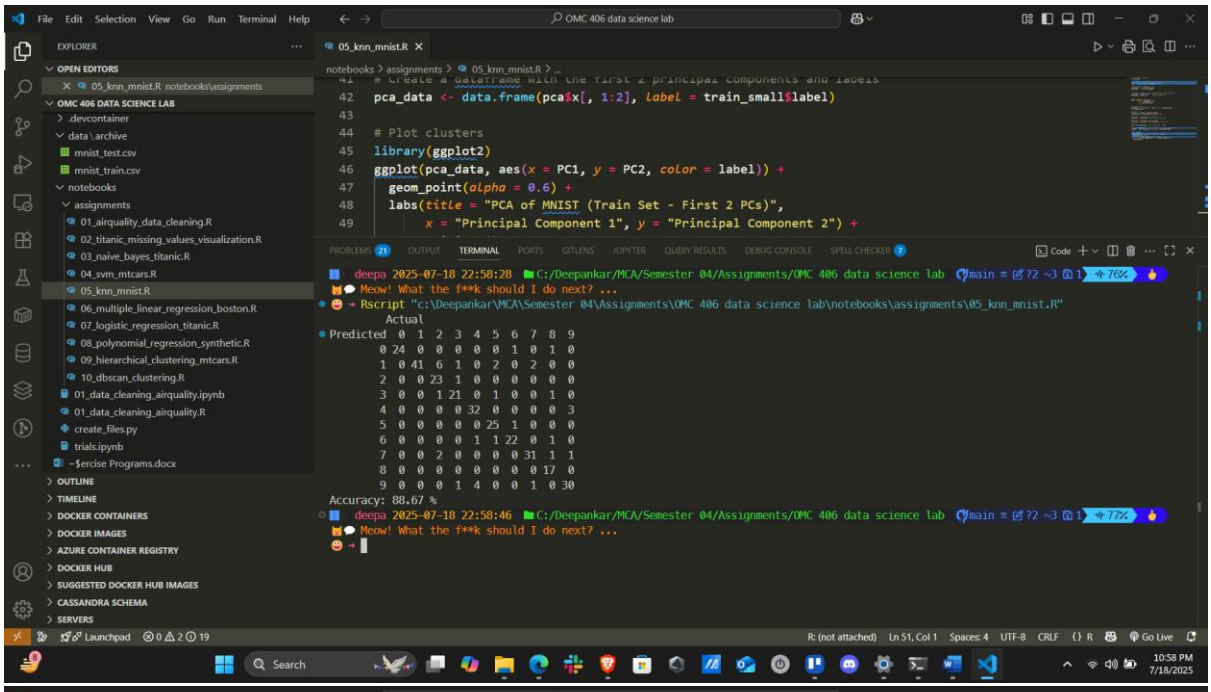
- k-NN algorithm.
- class package.

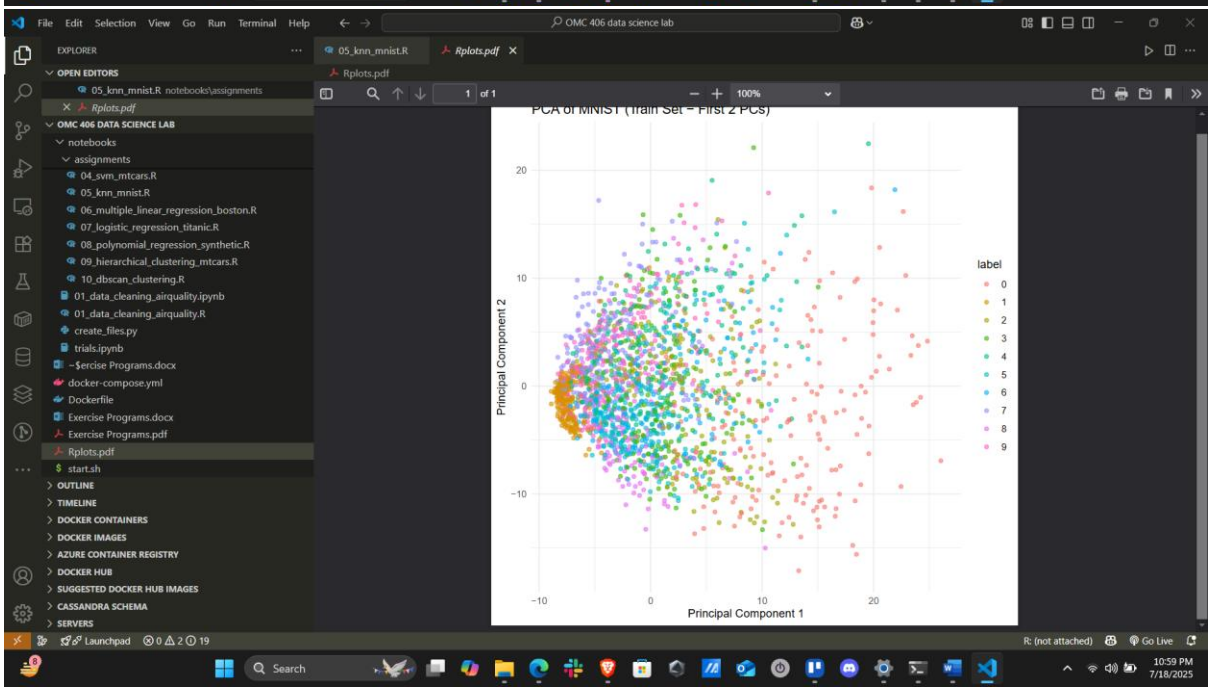
R Program:

```
# Load necessary libraries
library(class)

# Load MNIST data (make sure files are in your working
directory)
train <- read.csv("data/archive/mnist_train.csv")
test <- read.csv("data/archive/mnist_test.csv")

# Convert labels to factors
train$label <- as.factor(train$label)
test$label <- as.factor(test$label)


# Use smaller subsets for faster testing (k-NN is slow on large
data)
train_small <- train[1:2000, ]   # 2,000 training samples
test_small <- test[1:300, ]      # 300 test samples


# Apply k-NN with k = 5
pred <- knn(train = train_small[, -1],
            test = test_small[, -1],
            cl = train_small$label,
            k = 5)

# Confusion Matrix
conf_matrix <- table(Predicted = pred, Actual =
test_small$label)
```

```r
print(conf_matrix)

# Accuracy
accuracy <- mean(pred == test_small$label)
cat("Accuracy:", round(accuracy * 100, 2), "%\n")

# Remove zero-variance columns before PCA
pixel_data <- train_small[, -1]  # exclude label column

# Keep only columns with non-zero variance
pixel_data <- pixel_data[, apply(pixel_data, 2, var) != 0]

# Apply PCA
pca <- prcomp(pixel_data, center = TRUE, scale. = TRUE)

# Create a dataframe with the first 2 principal components and
labels
pca_data <- data.frame(pca$x[, 1:2], label = train_small$label)

# Plot clusters
library(ggplot2)
ggplot(pca_data, aes(x = PC1, y = PC2, color = label)) +
  geom_point(alpha = 0.6) +
  labs(title = "PCA of MNIST (Train Set - First 2 PCs)",
       x = "Principal Component 1", y = "Principal Component
2") +
  theme_minimal()
```

OUTPUT:


05_knn_mnist.R
pca_data <- data.frame(pca$x[, 1:2], label = train_small$label)
# Plot clusters
library(ggplot2)
ggplot(pca_data, aes(x = PC1, y = PC2, color = label)) +
geom_point(alpha = 0.6) +
labs(title = "PCA of MNIST (Train Set - First 2 PCs)",
x = "Principal Component 1", y = "Principal Component 2") +
deepa 2025-07-18 22:58:28 C:/Deepankar/MCA/Semester 04/Assignments/OMC 406 data science lab
Meow! What the f**k should I do next? ...
Rscript "c:\Deepankar\MCA\Semester 04\Assignments\OMC 406 data science lab\notebooks\assignments\05_knn_mnist.R"
Actual
Predicted 0 1 2 3 4 5 6 7 8 9
0 24 0 0 0 0 0 1 0 1 0
1 0 41 6 1 0 2 0 2 0 0
2 0 0 23 1 0 0 0 0 0 0
3 0 0 1 21 0 1 0 0 1 0
4 0 0 0 0 32 0 0 0 0 3
5 0 0 0 0 0 25 1 0 0 0
6 0 0 0 0 1 1 22 0 1 0
7 0 0 2 0 0 0 0 31 1 1
8 0 0 0 0 0 0 0 0 17 0
9 0 0 0 1 4 0 0 1 0 30
Accuracy: 88.67 %
deepa 2025-07-18 22:58:46 C:/Deepankar/MCA/Semester 04/Assignments/OMC 406 data science lab
Meow! What the f**k should I do next? ...


Rplots.pdf
PCA of MNIST (Train Set - First 2 PCs)
Principal Component 2
Principal Component 1
label
0
1
2
3
4
5
6
7
8
9

Exercise 6: Multiple Linear Regression (Boston Housing)

Experiment No.: 6

Date:

Problem Definition:

Predict house prices using multiple linear regression.

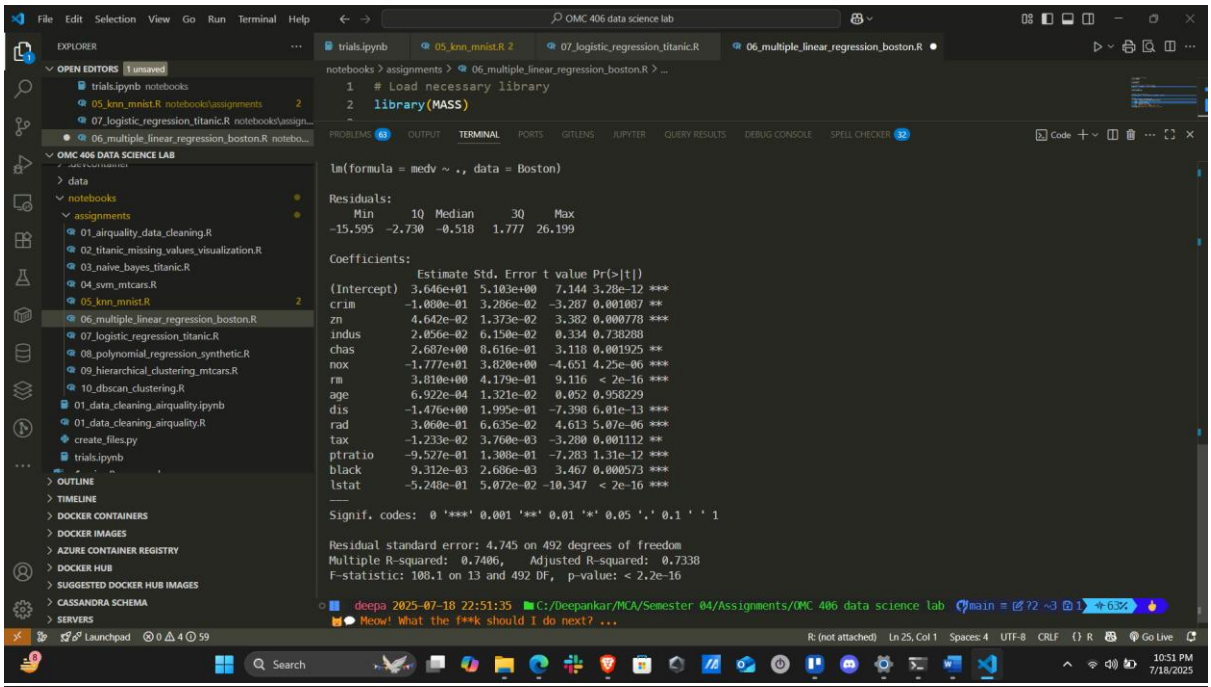
Theory Background:

- Multiple regression theory.
- lm() function.

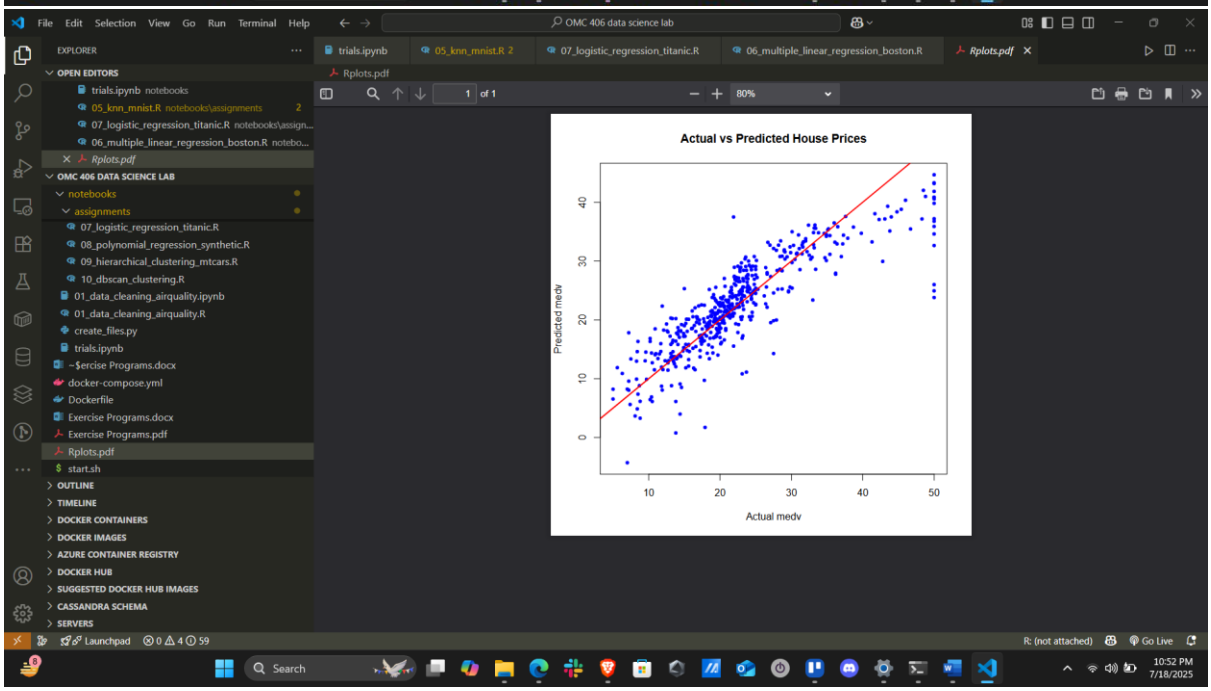
R Program:

```
# Load necessary library
library(MASS)

# Load dataset
data(Boston)

# Fit multiple linear regression model
model_lm <- lm(medv ~ ., data = Boston)

# Model summary
summary(model_lm)


# Predict on training data
predicted_medv <- predict(model_lm, newdata = Boston)

# Plot
plot(Boston$medv, predicted_medv,
     main = "Actual vs Predicted House Prices",
     xlab = "Actual medv",
     ylab = "Predicted medv",
     col = "blue", pch = 20)
abline(a = 0, b = 1, col = "red", lwd = 2)
plot(model_lm, which = 1)  # Residuals vs Fitted
```

OUTPUT :


File Edit Selection View Go Run Terminal Help
OMC 406 data science lab
EXPLORER
OPEN EDITORS 1 unsaved
trials.ipynb notebooks
05_knn_mnist.R notebooks\assignments 2
07_logistic_regression_titanic.R notebooks\assign...
06_multiple_linear_regression_boston.R notebo...
OMC 406 DATA SCIENCE LAB
data
notebooks
assignments
01_airquality_data_cleaning.R
02_titanic_missing_values_visualization.R
03_naive_bayes_titanic.R
04_svm_mtcars.R
05_knn_mnist.R 2
06_multiple_linear_regression_boston.R
07_logistic_regression_titanic.R
08_polynomial_regression_synthetic.R
09_hierarchical_clustering_mtcars.R
10_dbscan_clustering.R
01_data_cleaning_airquality.ipynb
01_data_cleaning_airquality.R
create_files.py
trials.ipynb
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
SUGGESTED DOCKER HUB IMAGES
CASSANDRA SCHEMA
SERVERS
trials.ipynb
05_knn_mnist.R 2
07_logistic_regression_titanic.R
06_multiple_linear_regression_boston.R
notebooks > assignments > 06_multiple_linear_regression_boston.R > ...
1 # Load necessary library
2 library(MASS)
PROBLEMS 63 OUTPUT TERMINAL PORTS GITLENS JUPYTER QUERY RESULTS DEBUG CONSOLE SPELL CHECKER 32
lm(formula = medv ~ ., data = Boston)
Residuals:
Min 1Q Median 3Q Max
-15.595 -2.730 -0.518 1.777 26.199
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.646e+01 5.103e+00 7.144 3.28e-12 ***
crim -1.080e-01 3.286e-02 -3.287 0.001087 **
zn 4.642e-02 1.373e-02 3.382 0.000778 ***
indus 2.056e-02 6.150e-02 0.334 0.738288
chas 2.687e+00 8.616e-01 3.118 0.001925 **
nox -1.777e+01 3.820e+00 -4.651 4.25e-06 ***
rm 3.810e+00 4.179e-01 9.116 < 2e-16 ***
age 6.922e-04 1.321e-02 0.052 0.958229
dis -1.476e+00 1.995e-01 -7.398 6.01e-13 ***
rad 3.060e-01 6.635e-02 4.613 5.07e-06 ***
tax -1.233e-02 3.760e-03 -3.280 0.001112 **
ptratio -9.527e-01 1.308e-01 -7.283 1.31e-12 ***
black 9.312e-03 2.686e-03 3.467 0.000573 ***
lstat -5.248e-01 5.072e-02 -10.347 < 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared: 0.7406, Adjusted R-squared: 0.7338
F-statistic: 108.1 on 13 and 492 DF, p-value: < 2.2e-16
deepa 2025-07-18 22:51:35 C:/Deepankar/MCA/Semester 04/Assignments/OMC 406 data science lab main
Meow! What the f**k should I do next? ...
Launchpad 0 4 59
R: (not attached) Ln 25, Col 1 Spaces: 4 UTF-8 CRLF R Go Live
Search
10:51 PM
7/18/2025


File Edit Selection View Go Run Terminal Help
OMC 406 data science lab
EXPLORER
OPEN EDITORS
trials.ipynb notebooks
05_knn_mnist.R notebooks\assignments 2
07_logistic_regression_titanic.R notebooks\assign...
06_multiple_linear_regression_boston.R notebo...
Rplots.pdf
OMC 406 DATA SCIENCE LAB
notebooks
assignments
07_logistic_regression_titanic.R
08_polynomial_regression_synthetic.R
09_hierarchical_clustering_mtcars.R
10_dbscan_clustering.R
01_data_cleaning_airquality.ipynb
01_data_cleaning_airquality.R
create_files.py
trials.ipynb
~$ercise Programs.docx
docker-compose.yml
Dockerfile
Exercise Programs.docx
Exercise Programs.pdf
Rplots.pdf
start.sh
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
SUGGESTED DOCKER HUB IMAGES
CASSANDRA SCHEMA
SERVERS
trials.ipynb
05_knn_mnist.R 2
07_logistic_regression_titanic.R
06_multiple_linear_regression_boston.R
Rplots.pdf
1 of 1
80%
Actual vs Predicted House Prices
Predicted medv
Actual medv
0
10
20
30
40
50
Launchpad 0 4 59
R: (not attached) Go Live
Search
10:52 PM
7/18/2025

Exercise 7: Logistic Regression (Titanic)

Experiment No.: 7

Date:

Problem Definition:

Predict survival using logistic regression.

Theory Background:

- Logistic regression theory.

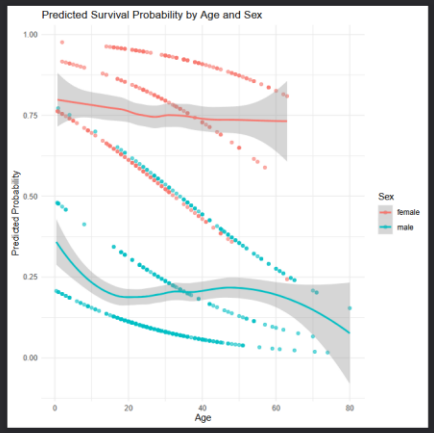
R Program:

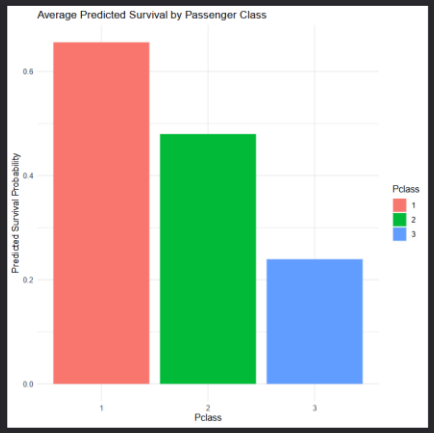
```
# Load necessary package
library(titanic)
library(ggplot2)
library(dplyr)

# Load dataset
data <- titanic_train

# Data Cleaning: Remove NAs
data <- na.omit(data)

# Ensure proper types
data$Survived <- factor(data$Survived)
data$Sex <- factor(data$Sex)
data$Pclass <- factor(data$Pclass)

# Train logistic regression model
model <- glm(Survived ~ Pclass + Sex + Age, data = data, family
= binomial)

# Model Summary
summary(model)

# Predict probabilities
data$predicted_prob <- predict(model, type = "response")

# Classify as 0 or 1 using 0.5 threshold
data$predicted_class <- ifelse(data$predicted_prob > 0.5, 1, 0)
```

```r
# Confusion Matrix
conf_matrix <- table(Predicted = data$predicted_class, Actual =
data$Survived)
print(conf_matrix)

# Accuracy
accuracy <- mean(data$predicted_class ==
as.numeric(as.character(data$Survived)))
cat("Accuracy:", round(accuracy * 100, 2), "%\n")


ggplot(data, aes(x = Age, y = predicted_prob, color = Sex)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "loess") +
  labs(title = "Predicted Survival Probability by Age and Sex",
       x = "Age", y = "Predicted Probability") +
  theme_minimal()

avg_pred <- data %>%
  group_by(Pclass) %>%
  summarise(Average_Predicted_Survival = mean(predicted_prob))

ggplot(avg_pred, aes(x = Pclass, y =
Average_Predicted_Survival, fill = Pclass)) +
  geom_col() +
  labs(title = "Average Predicted Survival by Passenger Class",
       x = "Pclass", y = "Predicted Survival Probability") +
  theme_minimal()
```

OUTPUT :

```
Call:
glm(formula = Survived ~ Pclass + Sex + Age, family = binomial,
    data = data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.777013   0.401123   9.416  < 2e-16 ***
Pclass2     -1.309799   0.278066  -4.710 2.47e-06 ***
Pclass3     -2.580625   0.281442  -9.169  < 2e-16 ***
Sexmale     -2.522781   0.207391 -12.164  < 2e-16 ***
Age         -0.036985   0.007656  -4.831 1.36e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 964.52  on 713  degrees of freedom
Residual deviance: 647.28  on 709  degrees of freedom
AIC: 657.28

Number of Fisher Scoring iterations: 5

         Actual
Predicted   0   1
        0 356  83
        1  68 207
Accuracy: 78.85 %
`geom_smooth()` using formula = 'y ~ x'
```





Exercise 8: Polynomial Regression (Synthetic Data)

Experiment No.: 8

Date:

Problem Definition:

Fit polynomial regression on synthetic nonlinear data.

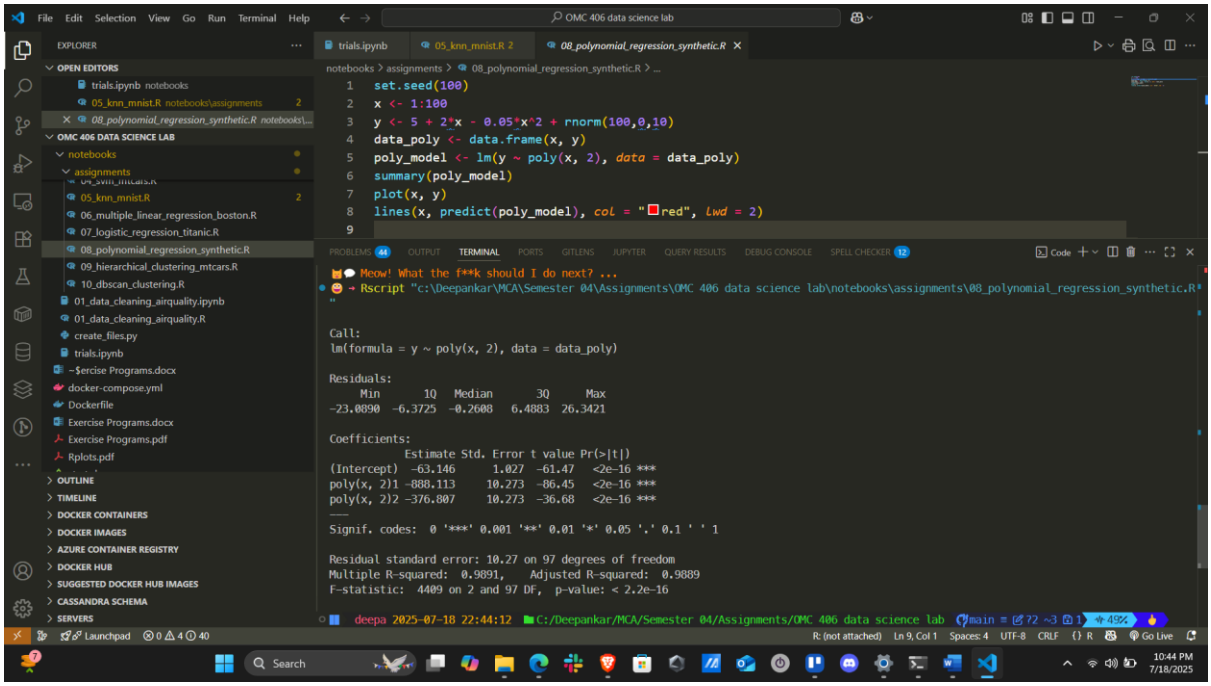
Theory Background:

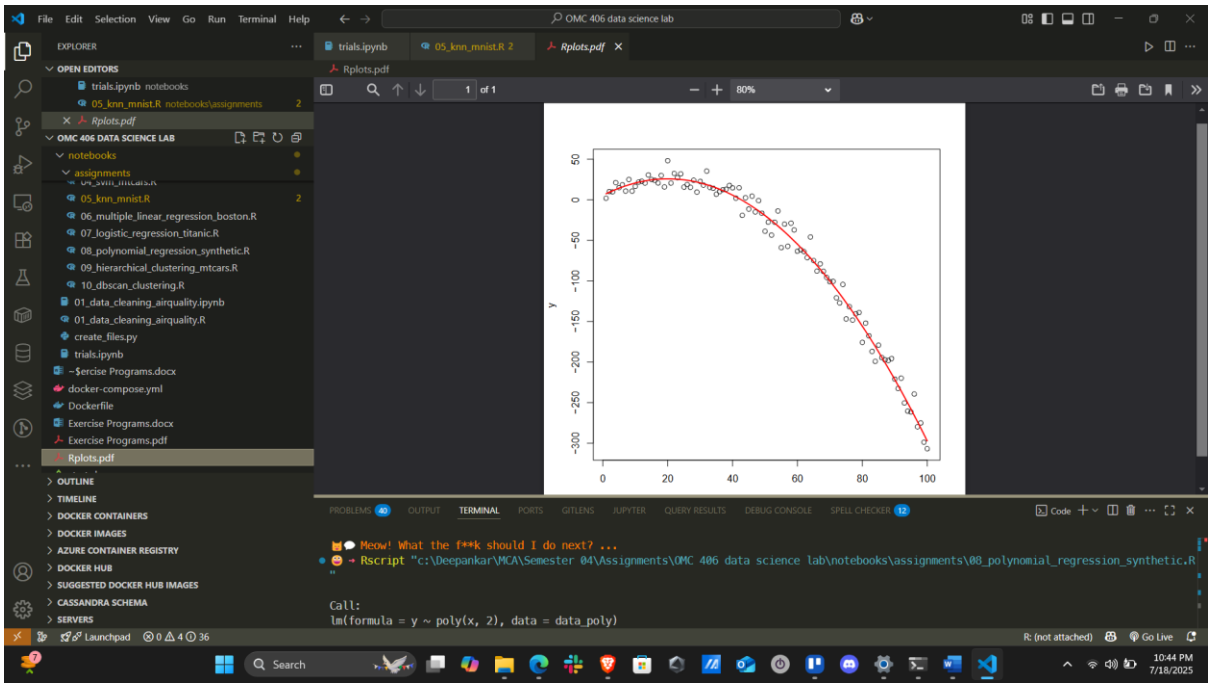
- Polynomial regression theory.

R Program:

```
set.seed(100)
x <- 1:100
y <- 5 + 2*x - 0.05*x^2 + rnorm(100,0,10)
data_poly <- data.frame(x, y)
poly_model <- lm(y ~ poly(x, 2), data = data_poly)
summary(poly_model)
plot(x, y)
lines(x, predict(poly_model), col = "red", lwd = 2)
```

OUTPUT (Screenshots from RStudio):




File Edit Selection View Go Run Terminal Help
OMC 406 data science lab
EXPLORER
OPEN EDITORS
trials.ipynb notebooks
05_knn_mnist.R notebooks\assignments
Rplots.pdf
OMC 406 DATA SCIENCE LAB
notebooks
assignments
05_knn_mnist.R
06_multiple_linear_regression_boston.R
07_logistic_regression_titanic.R
08_polynomial_regression_synthetic.R
09_hierarchical_clustering_mtcars.R
10_dbscan_clustering.R
01_data_cleaning_airquality.ipynb
01_data_cleaning_airquality.R
create_files.py
trials.ipynb
~$ercise Programs.docx
docker-compose.yml
Dockerfile
Exercise Programs.docx
Exercise Programs.pdf
Rplots.pdf
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
DOCKER HUB
SUGGESTED DOCKER HUB IMAGES
CASSANDRA SCHEMA
SERVERS
trials.ipynb
05_knn_mnist.R 2
Rplots.pdf
1 of 1
80%
y
PROBLEMS 40
OUTPUT
TERMINAL
PORTS
GITLENS
JUPYTER
QUERY RESULTS
DEBUG CONSOLE
SPELL CHECKER 12
Meow! What the f**k should I do next? ...
→ Rscript "c:\Deepankar\MCA\Semester 04\Assignments\OMC 406 data science lab\notebooks\assignments\08_polynomial_regression_synthetic.R"
Call:
lm(formula = y ~ poly(x, 2), data = data_poly)
Launchpad
R: (not attached)
Go Live
Search
10:44 PM
7/18/2025

Exercise 9: Hierarchical Clustering (mtcars)

Experiment No.: 9

Date:

Problem Definition:

Perform hierarchical clustering on mtcars dataset.

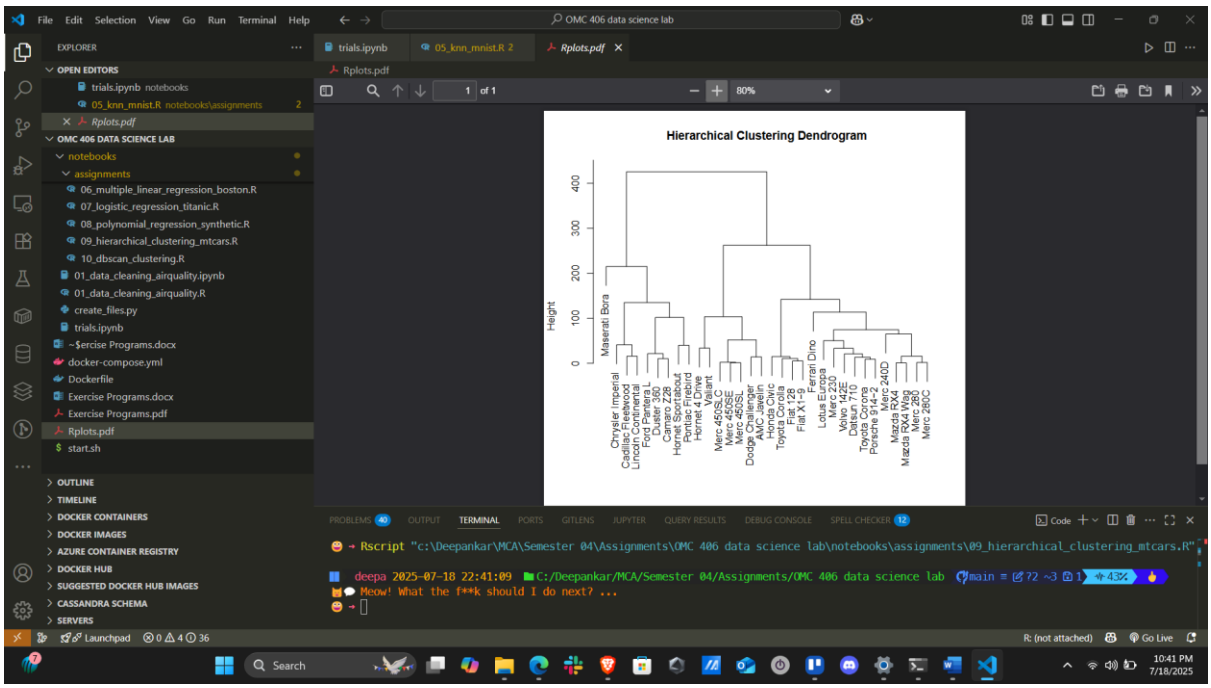
Theory Background:

- Clustering theory.
- Dendrograms.

R Program:

```
data(mtcars)
d <- dist(mtcars)
hc <- hclust(d)
plot(hc, main = "Hierarchical Clustering Dendrogram")
```

OUTPUT:



Exercise 10: DBSCAN Clustering (Noisy Data)

Experiment No.: 10

Date:

Problem Definition:

Cluster data using DBSCAN algorithm.

Theory Background:

- Density-based clustering.

R Program:

```r
library(dbscan)
set.seed(123)
data <- matrix(rnorm(200), ncol=2)
data[51:100, ] <- data[51:100, ] + 3
db <- dbscan(data, eps = 0.5, minPts = 5)
plot(data, col = db$cluster + 1L, main = "DBSCAN Clustering")
```

OUTPUT :