**AMERICAN INTERNATIONAL**A close up of a sign

Description automatically generated

**UNIVERSITY-BANGLADESH**

**Faculty of Science and Technology**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assignment Title: | Mid term project | | | |
| Assignment No: 1 |  | | Date of Submission: | 9 September 2023 |
| Course Title: | Introduction to Data Science | | | |
| Course Code: | 10666 | | Section: | B |
| Semester: | Summer | 2023-24 | Course Teacher: | Tohedul Islam |

**Declaration and Statement of Authorship:**

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student’s work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaborationhas been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand thatPlagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

*\* Student(s) must complete all details except the faculty use part.*

\*\* Please submit all assignments to your course teacher or the office of the concerned teacher.

|  |  |
| --- | --- |
| Group Name/No.: | 14 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **ID** | **Program** | **Signature** |
| 1 | NAFIUR RAHMAN NILOY | 22-46459-1 | BSc [CSE] | Niloy |
| 2 | SUDIPTA BISWAS | 21-45981-3 | BSc [CSE] | Sudipto |

|  |  |  |
| --- | --- | --- |
| ***Faculty use only*** | | |
| FACULTY COMMENTS | **Marks Obtained** |  |
|  |
|  |
|  | **Total Marks** |  |
|  |
|  |

Dataset: **Introduction**

The Titanic dataset is a renowned collection of data that provides detailed information about the passengers aboard the RMS Titanic, which sank tragically during its maiden voyage in April 1912. This dataset comprises 106 individual records with 10 attributes: Gender, Age, SibSp, Parch, Fare, Embarked, Class, Who, Alone, and Survived. It includes key details such as the passenger's gender and age, the number of siblings/spouses (SibSp) and parents/children (Parch) on board, the fare paid, and the port of embarkation (Cherbourg, Queenstown, or Southampton). Additionally, it classifies passengers by their travel class, gender or age group (male, female, or child), and indicates whether they were traveling alone. The "Survived" attribute denotes whether a passenger survived the disaster. However, the dataset is not ideal in its current state, as it contains missing and noisy values, and some attributes have outliers. This makes it a valuable resource for developing data cleaning and preprocessing skills.

Dataset: **About data**

* **Library Use:**

library(dplyr)

* **Read Data**

mydata <- read.csv ("D:/Study Metarials/8th Semester/Data Science/Project/Dataset\_midterm.csv", header = TRUE, sep = ",")

View(mydata)

str(mydata)

summary(mydata)

#Number of row and column

num\_instances <- nrow(mydata)

num\_attributes <- ncol(mydata)

print(paste("Number of instances (rows):", num\_instances))

print(paste("Number of Columns:", num\_attributes))

#missing values

colSums(is.na(mydata))

missing\_values\_indices <- lapply(mydata, function(x) {

if (is.integer(x) | is.character(x)) {

return(which(is.na(x) | x == ""))

} else {

return(NULL)

}

})

print(missing\_values\_indices)

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

A close-up of a computer screen

Description automatically generated

A computer screen shot of a code

Description automatically generated

**Description:** Load Dataset to mydata. Found the total number of row and column using ncol() and nrow() function.Then we found all the missing values with the help of is.interger() and is.character() function. Used print(paste(“ ”)) functions to show the output in one line.

Dataset: **Data Preparation & Exploration**

**Remove unnecessary rows:**

**Code:**

#Remove unnecessary rows

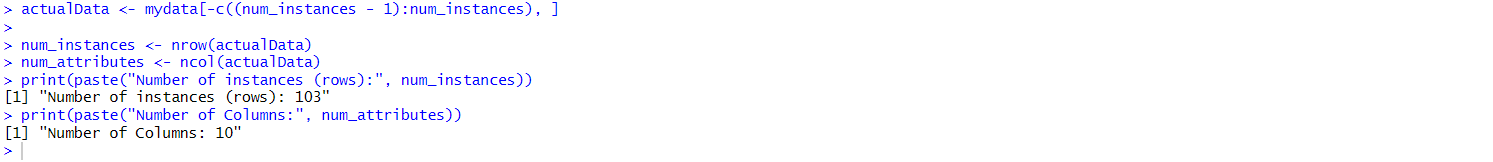
actualData <- mydata[-c((num\_instances - 1):num\_instances), ]

num\_instances <- nrow(actualData)

num\_attributes <- ncol(actualData)

print(paste("Number of instances (rows):", num\_instances))

print(paste("Number of Columns:", num\_attributes))

**Output:**

**Description:** Last two rows were unnecessary because those rows did not have enough data, so we needed to remove them.

**Column: Gender**

* **Detecting and Recovering Noisy Values: There is no Noisy values**

unique\_values <- unique(mydata$Gender)

print(unique\_values)

**output:**

A screenshot of a computer

Description automatically generated

* **Data conversion: Converting categorical attributes to numeric(Gender is a categorical data)**

mydata$Gender <- factor (mydata$Gender,

levels = c("male", "female"),

labels = c(1, 2))

View(mydata)

**Output:** A screenshot of a computer

Description automatically generated

* **Missing Value Imputation: Replacing NA Values with Mode**

mode\_gender <- names(which.max(table(actualData$Gender)))

actualData$Gender[is.na(actualData$Gender)] <- mode\_gende

View(actualData)

**Output:**

A screenshot of a computer

Description automatically generated

**Description:** At first we found the unique values using unique() then we have converted categorical attributes to numeric with the help of factor function.

**Column: age**

* **Missing Value Imputation: Replacing NA Values with Median**

age\_median <- round(median(mydata$age, na.rm = TRUE))

mydata$age[is.na(mydata$age)] <- age\_median

* **Outlier Detection and Removal with Interquartile Range (IQR) Method**

Q1 <- quantile(actualData$age, 0.25)

Q3 <- quantile(actualData$age, 0.75)

IQR\_value <- Q3 - Q1

threshold <- 1.5

outlier\_condition <- (actualData$age < (Q1 - threshold \* IQR\_value)) | (actualData$age > (Q3 + threshold \* IQR\_value))

* **Serial Update After Removing Outliers**

actualData <- actualData %>%

filter(!outlier\_condition) %>%

mutate(row\_number = row\_number())

View(actualData)

**Output:** **A screenshot of a computer

Description automatically generated**

**Description:** First, we replaced the na values with the median. Then with the help of IQR method we have found the Outliers then removed those Outliers.To update the rows after removing outliers we have used pipe operator (%>%)

**Column: sibsp**

* **The ‘sibsp’ column exhibits optimal data quality with no missing or invalid values.**

missing\_values\_sibsp <- sum(is.na(actualData$sibsp))

print(missing\_values\_sibsp)

**Output:**

A white background with blue text

Description automatically generated

**Column: parch**

* **The ‘parch’ column exhibits optimal data quality with no missing or invalid values.**

missing\_values\_sibsp <- sum(is.na(actualData$sibsp))

print(missing\_values\_sibsp)

**Output:** A blue text on a white background

Description automatically generated

**Column: embarked**

* **Detecting and Recovering Noisy Values: There is no Noisy values**

unique\_embarked <- unique(actualData$embarked)

print(unique\_values)

* **Data conversion: Converting categorical attributes to numeric**

actualData$embarked <- factor(actualData$embarked,

levels = c("S","Q","C"),

labels = c(1, 2,3))

* **Missing Value Imputation: Replacing NA Values with Mode**

missing\_values\_embarked <- sum(is.na(actualData$embarked))

print(missing\_values\_embarked)

print(paste("Total Missing Values:",missing\_values\_embarked))

**Output**A computer code with blue and white text

Description automatically generated

A screenshot of a computer

Description automatically generated

**Description:** At first we found the unique values using unique() then we have converted categorical attributes to numeric with the help of factor() function.

**Column: class**

* **Detecting and Recovering Noisy Values: There is no Noisy values**

unique\_class <- unique(actualData$class)

print(unique\_class)

* **Data conversion: Converting categorical attributes to numeric**

actualData$class<- factor(actualData$class,

levels=c("First","Second","Third"),

labels=c(1,2,3))

* **Missing Value Imputation: Replacing NA Values with Mode**

missing\_values\_class<-sum(is.na(actualData$class))

print(paste("Total Missing Values:",missing\_values\_class))

mode\_class <- names(which.max(table(actualData$class)))

actualData$class[is.na(actualData$class)] <- mode\_class

View(actualData)

* **Output:** A computer code with blue and white text

  Description automatically generated

**Description:** At first we found the unique values using unique() then we have converted categorical attributes to numeric with the help of factor function. Class has no missing or noisy values.

**A screenshot of a computer

Description automatically generated**

**Column: who**

* **Detecting and Recovering Noisy Values: There is no Noisy values**

unique\_values <- unique(actualData$who)

print(unique\_values)

* **Data conversion: Converting categorical attributes to numeric and replaces all instances of "mannn" with "man"**

actualData$who <- gsub("mannn", "man", actualData$who)

actualData$who<- factor(actualData$who,

levels=c("man","woman","child"),

labels=c(1,2,3))

* **Missing Value Imputation: Replacing NA Values with Mode**

missing\_values\_who <- sum(is.na(actualData$who))

print(paste("Total Missing Values:",missing\_values\_class))

View(actualData)

**Output:**

A white background with blue text

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**Description:** At first we found the unique values using unique() and recovered the noisy vale. There is no missing values. Then we have converted categorical attributes to numeric with the help of factor function.

**Column: alone**

* **Detecting and Recovering Noisy Values: There is no Noisy values**

unique\_values<-unique(actualData$alone)

print(unique\_values)

* **Data conversion: Converting categorical attributes to numeric**

actualData$alone<- factor(actualData$alone,

levels=c("TRUE","FALSE"),

labels=c(1,2))

* **Missing Value Imputation: Replacing NA Values with Mode**

missing\_values\_alone <- sum(is.na(actualData$alone))

print(paste("Total Missing Values:",missing\_values\_alone))

View(actualData)

**Output:**

A computer code with blue text

Description automatically generated

A screenshot of a computer

Description automatically generated

**Description:** At first we found the unique values using unique() then we have converted categorical attributes to numeric with the help of factor function.

**Column: survived**

* **The ‘survived’ column exhibits optimal data quality with no missing, invalid, or noisy values.**

missing\_values\_survived <- sum(is.na(actualData$survived))

noisy\_values\_survived <- sum(actualData$survived < 0 | actualData$survived > 1 )

print(missing\_values\_survived)

print(noisy\_values\_survived)

A screenshot of a computer

Description automatically generated

**Column: fare**

* **Converting fare char to numeric value**

actualData$fare <- as.numeric(actualData$fare)

View(actualData)

* **For the missing valse find median and replace with NAs**

fare\_median <- round(median(actualData$fare, na.rm = TRUE))

actualData$fare[is.na(actualData$fare)] <- fare\_median

**Output:**

A screenshot of a computer

Description automatically generated

**Date Visualization**

* **Find Missing Values:**

**Code:**

na\_counts <- colSums(is.na(actualData))

print(na\_counts)

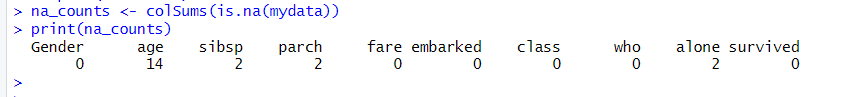
**Visualization:**

barplot(na\_counts, names.arg = names(na\_counts),

ylab = "Number of Missing Values", col = "red",cex.names = 0.9,

main = "Missing Values per Attribute", las =2)

**Output:**



**A graph with red squares and black text

Description automatically generated**

**Description:** Using colSums() and is.na() we found the number of missing values for each attribute.

* **Find Outliers:**

**A graph with a bar

Description automatically generated with medium confidence**

**Description:** Using boxplot, we found outliers on Age and we no need to find other attributes outliers.

After removing outliers:

A graph of a graph

Description automatically generated

* **Mean-Median-Mode Graph:**

**Code:**

getMode <- function(v) {

tabulated <- table(v)

mode\_value <- names(sort(tabulated, decreasing = TRUE))[1]

return(as.numeric(mode\_value))

}

means <- sapply(actualData, function(x) if(is.numeric(x)) mean(x, na.rm = TRUE) else NA)

medians <- sapply(actualData, function(x) if(is.numeric(x)) median(x, na.rm = TRUE) else NA)

modes <- sapply(actualData, function(x) if(is.numeric(x) || is.factor(x) || is.character(x)) getMode(x)

else NA)

stat\_values <- rbind(means, medians, modes)

row\_names <- c("Mean", "Median", "Mode")

rownames(stat\_values) <- row\_names

barplot(stat\_values, beside = TRUE,

col = c("green", "orange", "brown"),

legend.text = row\_names,

args.legend = list(x = "topright", cex = 0.9),

cex.names = 0.9,

ylim = c(0, 50)

**Vizualization:**

**A graph with red and green squares

Description automatically generated**

**Description:** Initially mean, mode and were found using sapply() function, then the values were combined in stat\_values matrix using rblind(). A Barplot has been drawn to visualize.

**Final Data Set:** **A screenshot of a computer

Description automatically generated**

This is the final outcome of the data set after cleaning all the data.

THE END