BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]
Department of Computer Engineering

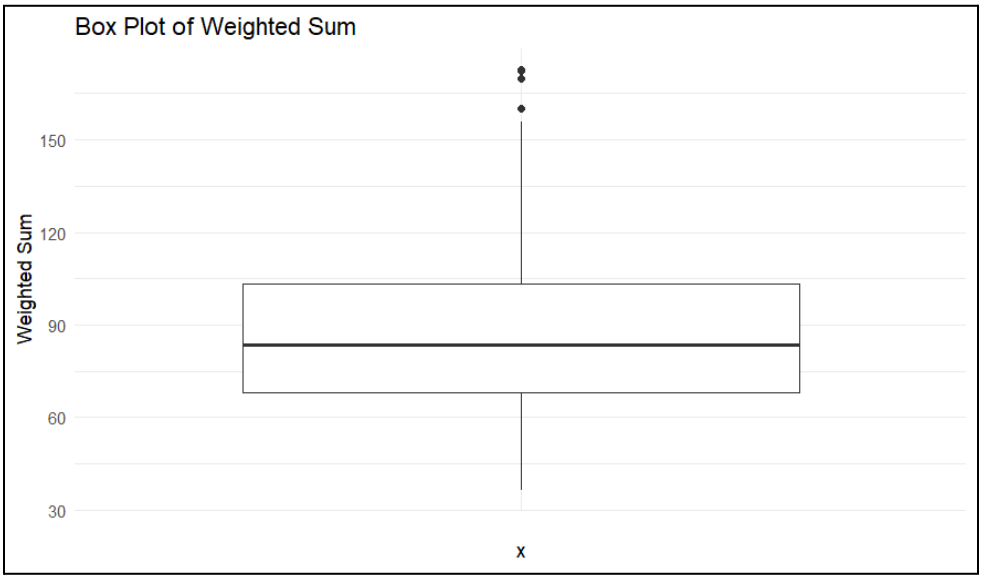
Name	Ms Neerja Doshi
UID	2021300029
DIV	BE COMPS [ADV -> BATCH F]`
ADV EXP 5	

AIM	To use Rstudio and do Linear and Logistic Regression on Housing Dataset
Dataset Particulars	**Name: Housing Dataset** Link: https://www.kaggle.com/datasets/ictinstitute/utrecht-housing-dataset/data **Column Details :** 1) id: a number between 0 and 100000 that is a unique identifier for each house. 2) zipcode: Each house has a zipcode corresponding to the area the house is in. The zipcode can be an indicator of build year or other properties. There are four different zip codes in use: 3520, 3525, 3525, 3800. 3) lot-len: the length in meters of the plot of land the house is built on. Each house is built on a square plot of land. It can be anything from 5.0 to 100.0 meters 4) lot-width: the width in meters of the plot of land the house is built on. It can be anything from 5.0 to 100.0 meters 5) lot-area: the total area of the plot of land the house is built on. You can probably compute this from lot-len and lot-width. house-area. The living area of the house in square meters. 30.0 square meters is a tiny house, 200.0 square meters would be a mansion. 6) garden-size: The size of the garden in square meters. Many people want to have a large garden. balcony: the number of balconies the house has. Common values are 0,1, or 3 balconies 7) x-coor: the x-coordinate describing the location of the house. It is an integer value between 2000 and 3000 8) y-coor: the y-coordinate describing the location of the house. It is an integer value between 5000 and 6000

	9) buildyear. The year that the house was built. Some of the oldest houses are from 1100,, but most houses were built in the 2oth and 21st century. 10) bathrooms: Most houses in Utrecht have one bathroom. Some houses have 2 or 3 bathrooms 11) taxvalue: The taxvalue of the house is a number between 50.000 and 1.000.000 that is a conservative value of the house. It is estimated (often based on real housing data) by the government to determine taxes. In a calm market, it is close to but often slightly lower than the retail value. It is rounded to the nearest 100 euros. 12) retailvalue: the market value of a house. It is a number between 50.000 and 1.000.000. It is rounded to the nearest 1000 euros. 13) energy-eff: This value is either 0 or 1. If it is 1 it means that the house is energy efficient. This is important for certain climate goals of the city of Utrecht 14) monument: Some houses in Utrecht, especially older houses, have monumental value since they have a unique architectural design. People have been trying to predict if this is the case from other data, since it is timeconsuming to visit each house and inspect the architecture.
Published Graphs on RPubs	**Click here :)** **https://rpubs.com/Neerja**
Analysis	**Linear Regression :** **1) Problem Statement :** a) To predict the Tax Value of house on basis of all relevant features. **2) Solution :** a) As there are many features ; i have taken weights sum of relevant columns and created a new column names “Weighted_Sum” b) Then the regression model is done on the independent variable == weighted_sum and dependent variable is == tax.value c) Heres the equation to generate Weighted_Sum column: ***Weighted_sum = house_area * 0.4 + garden_size * 0.3 + balcony * 0.1 + buildyear * 0.1 + bathrooms + 0.1***

1) Box Plot to figure out outliers :

```
> ggplot(my_data, aes(x = "", y = weighted_sum)) +
+     geom_boxplot() +
+     labs(title = "Box Plot of Weighted Sum",
+          y = "Weighted Sum") +
+     theme_minimal()
```

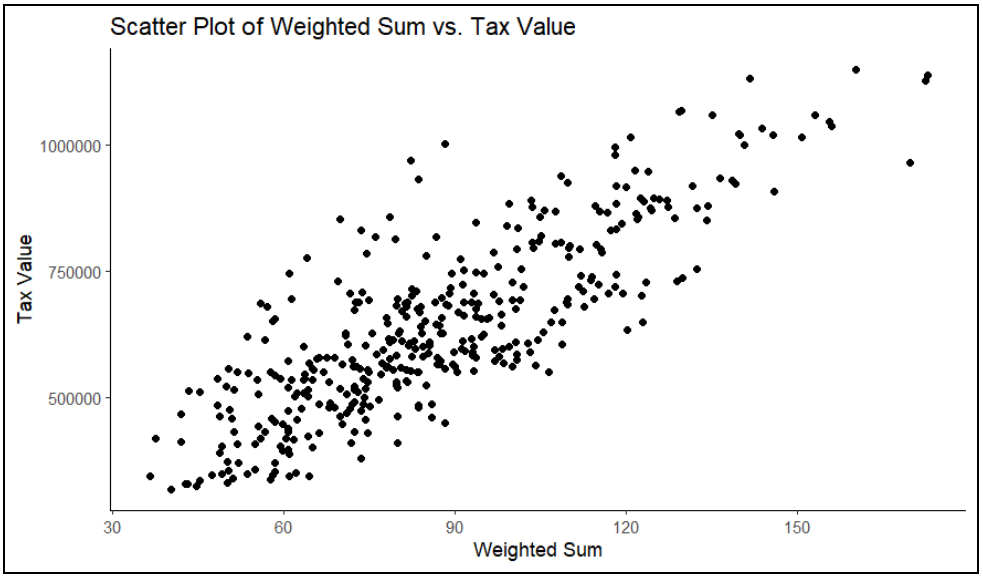


2) Making the model :

```
> weights <- c(house.area = 0.4, lot.area = 0.4, garden.size = 0.3, balcony = 0.3, energy.eff =
0.2, bathrooms = 0.2)
> my_data$weighted_sum <- rowSums(my_data[, c("house.area", "lot.area", "garden.size", "balcony",
"energy.eff", "bathrooms")] * weights)
>
> X <- my_data$weighted_sum
> y <- my_data$taxvalue
> model <- lm(taxvalue ~ weighted_sum, data = my_data)
> predictions <- predict(model, newdata = data.frame(weighted_sum = X))
> rmse <- sqrt(mean((my_data$taxvalue - predictions)^2))
> cat("RMSE:", rmse)
RMSE: 97840.59
```

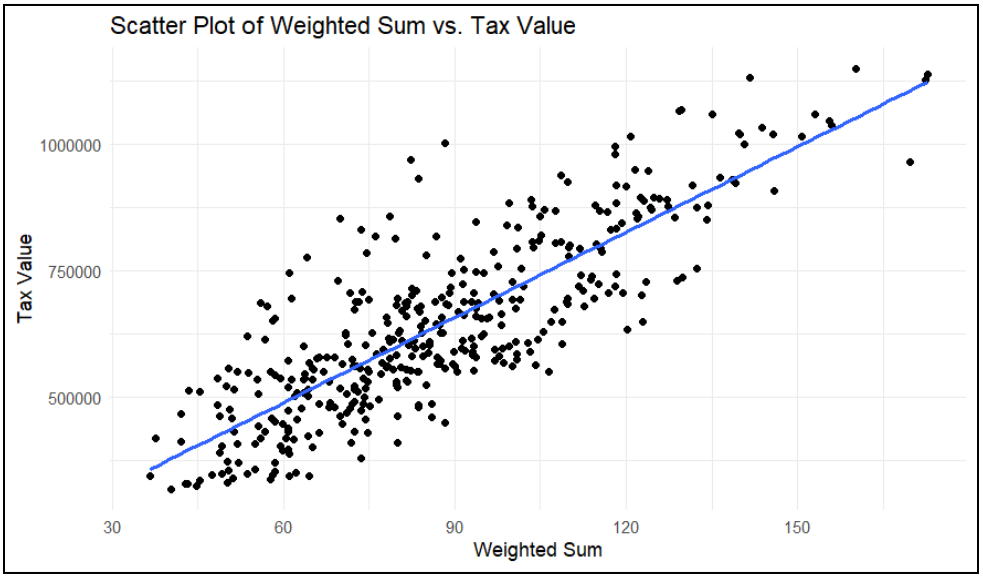
3) Scatter Plot

```
> ggplot(my_data, aes(x = weighted_sum, y = taxvalue)) +
+     geom_point() +
+     geom_smooth(method = "lm", se = FALSE) +
+     labs(title = "Scatter Plot of Weighted Sum vs. Tax Value",
+          x = "Weighted Sum",
+          y = "Tax Value") +
+     theme_minimal()
```



4) Scatter Plot with regression Line :

```
> ggplot(my_data, aes(x = weighted_sum, y = taxvalue)) +
+     geom_point() +
+     geom_line(aes(y = predictions), color = "red") +
+     labs(title = "Tax Value Prediction based on Weighted Features",
+          x = "Weighted Sum",
+          y = "Tax Value") +
+     theme_minimal()
>
```



5) Accuracy score of Model :

```
> mse <- mean((my_data$taxvalue - predictions)^2)
> cat("Mean Squared Error:", mse)
Mean Squared Error: 9572780854>
> rmse <- sqrt(mse)
> cat("Root Mean Squared Error:", rmse)
Root Mean Squared Error: 97840.59
> r_squared <- summary(model)$r.squared
> cat("R-squared:", r_squared)
R-squared: 0.6895374
>
```

Logistic Regression :

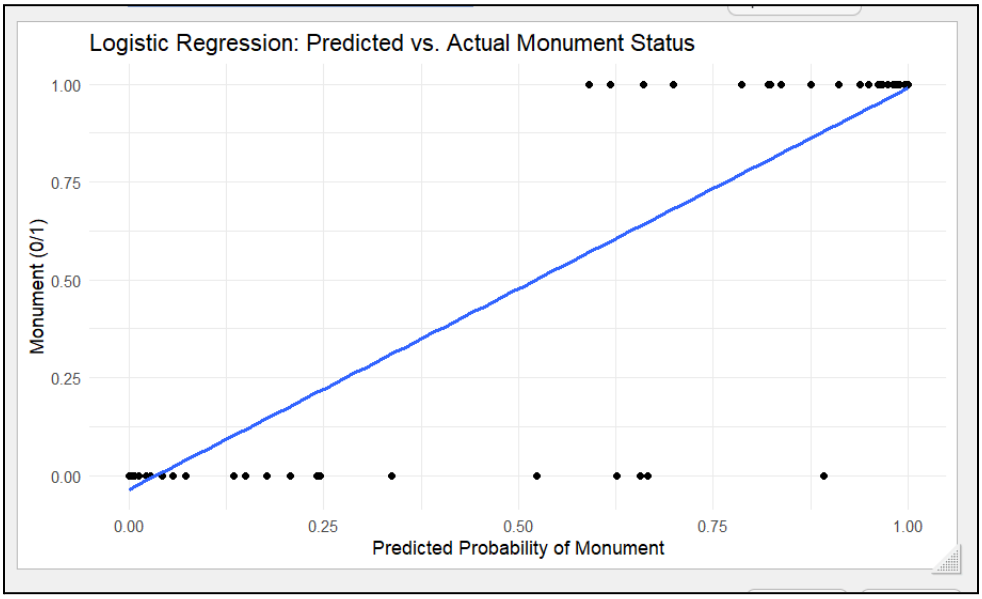
1) **Problem Statement :**
   a) To classify weather a house is a monument or not on basis of all relevant features.
2) **Solution :**
   a) The following columns are selected with the given logic to classify weather a house is a monument or not monument
      i) **house_area:** Larger houses might be more likely to be monuments.
      ii) **garden_size:** Historic houses often have larger gardens.
      iii) **buildyear:** Older houses are more likely to be monuments.
      iv) **balcony:** Historic houses might have unique balcony designs.
      v) **zipcode:** Certain areas might have more historic houses.

1) Model Making :

```
> set.seed(123)
> trainIndex <- createDataPartition(my_data$monument, p = 0.75, list = FALSE)
> trainData <- my_data[trainIndex, ]
> testData <- my_data[-trainIndex, ]
> model <- glm(monument ~ house.area + garden.size + buildyear + balcony + zipcode, data = train
ata, family = "binomial")
> predictions <- predict(model, newdata = testData, type = "response")
> predicted_classes <- ifelse(predictions > 0.5, 1, 0)
```

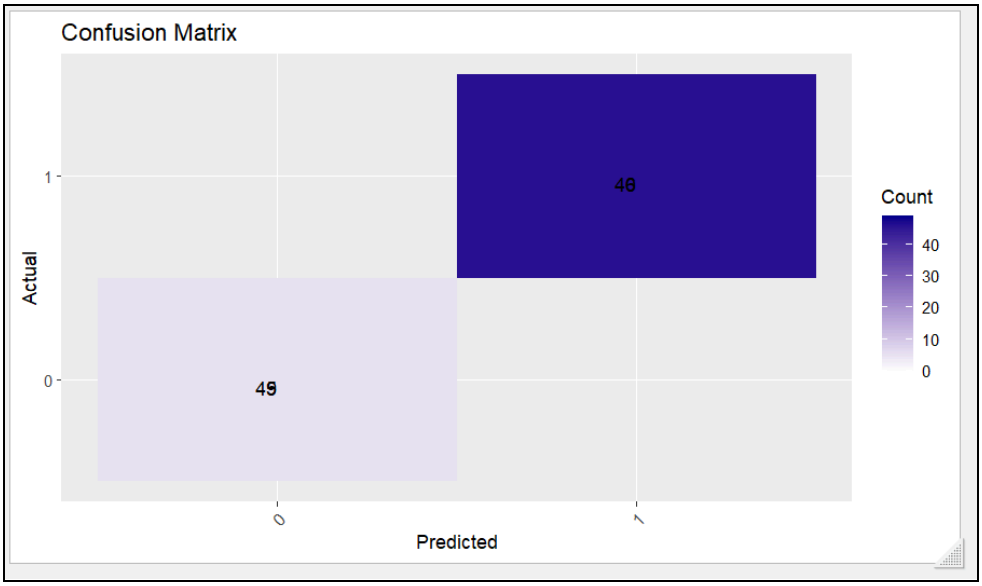
2) Logistic Regression Plot

```
> ggplot(plot_data, aes(x = predicted_prob, y = monument)) +
+     geom_point() +
+     geom_smooth(method = "glm", se = FALSE) +
+     labs(title = "Logistic Regression: Predicted vs. Actual Monument Status",
+          x = "Predicted Probability of Monument",
+          y = "Monument (0/1)") +
+     theme_minimal()
```



3) Confusion Matrix

```
> cm_values <- confusion_matrix$table
>
> # Create a data frame for plotting
> cm_data <- data.frame(
+     Actual = factor(rownames(cm_values), levels = c("0", "1")),
+     Predicted = factor(colnames(cm_values), levels = c("0", "1")),
+     Count = as.vector(cm_values)
+ )
>
> # Create the confusion matrix plot
> ggplot(cm_data, aes(x = Predicted, y = Actual, fill = Count)) +
+     geom_tile() +
+     geom_text(aes(label = Count), vjust = 1, hjust = 1) +
+     scale_fill_gradient(low = "white", high = "darkblue") +
+     labs(title = "Confusion Matrix", x = "Predicted", y = "Actual") +
+     theme(axis.text.x = element_text(angle = 45, hjust = 1))
>
```



4) All Classification Matrics :

```
> metrics <- confusionMatrix(confusion_matrix$table)
> print(metrics)
Confusion Matrix and Statistics

   predicted_classes
     0  1
  0 49  5
  1  0 46

               Accuracy : 0.95
                 95% CI : (0.8872, 0.9836)
    No Information Rate : 0.51
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9002

 Mcnemar's Test P-Value : 0.07364

            Sensitivity : 1.0000
            Specificity : 0.9020
         Pos Pred Value : 0.9074
         Neg Pred Value : 1.0000
             Prevalence : 0.4900
         Detection Rate : 0.4900
   Detection Prevalence : 0.5400
      Balanced Accuracy : 0.9510
```

Conclusion	By performing this experiment I learnt how to use lm and glm libraries of R Studio well along with the syntax of R language.