

BASANTI DEVI COLLEGE
AFFILIATED TO THE UNIVERSITY OF CALCUTTA

CERTIFICATE

This serves to certify that the project paper titled "**Optimizing Wine Quality Prediction : Model Fitting and Cross Validation Technique**", submitted by **Anusua Paul** in partial fulfillment of the requirements for the **Bachelor's degree in Statistics (Honours)**, is grounded in the results of beneficial research work conducted by the investigator under my guidance and supervision. The findings presented by the investigator in this project paper have not previously been submitted for any degree or diploma.

DATE : 26/07/24

SIGNATURE : Ganesh Dutta

Dr. Ganesh Dutta
Assistant Professor
Department of Statistics
Basanti Devi College

DECLARATION

I, **Anusua Paul**, bearing Roll Number **213041-11-0040**, hereby solemnly declare that the work submitted for the module **DSE-B2** is entirely my original creation. I affirm that I have not replicated any content from the work of fellow students or any other sources, save where explicit reference or acknowledgment is provided. Furthermore, I assert that no portion of this work has been authored by any other individual.

SIGNATURE :

Anusua Paul
Department of Statistics
Basanti Devi College

ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to my teacher and mentor, **Dr. Ganesh Dutta**, for his unwavering support and guidance throughout this endeavor. Additionally, I wish to express my appreciation to my other professors, friends, and seniors whose assistance was invaluable in the completion of this project. Finally, I offer my heartfelt thanks to the University of Calcutta for providing me with this opportunity.



CONTENTS

TOPIC	PAGE NO
❖ CERTIFICATE	1
❖ DECLARATION	2
❖ ACKNOWLEDGEMENT	3
❖ ABSTRACT	5
❖ INTRODUCTION	6
❖ DATA DESCRIPTION	7
❖ DATASET	8
❖ EXPLORATORY DATA ANALYSIS	10
❖ REGRESSION ANALYSIS	13
❖ MODEL FITTING	18
❖ CROSS VALIDATION	19
❖ CONCLUSION	21
❖ REFERENCE	22
❖ APPENDIX	23

ABSTRACT

This study aims to develop a **predictive model** for wine quality ratings using a set of explanatory variables, followed by validating its accuracy through **cross-validation** techniques. The focus is on accurately forecasting the quality rating (y) based on the given predictors. Anticipated challenges include managing data complexities, addressing outliers, and dealing with **multicollinearity**. To overcome these issues, the research will delve into **regression diagnostics**, highlighting their crucial role in the **machine learning** process. This comprehensive approach ensures the robustness and reliability of the predictive model, contributing valuable insights into wine quality assessment.

INTRODUCTION

“You can have data without information but you can not have information without data”.....Daniel Keys Maran

Model fitting and cross-validation are essential components of **machine learning** and **regression analysis**. In today's era of artificial intelligence and data science, our focus is on the intricacies of model building in machine learning. Machine learning is categorized into three types: supervised learning, unsupervised learning, and reinforcement learning. The diagram below provides a clear overview of these types.

SUPERVISED LEARNING



UNSUPERVISED LEARNING



REINFORCEMENT LEARNING



It's crucial to evaluate whether a model is a good fit, bad fit, or overfit. This section delves into **cross-validation**, where we split the dataset into training and testing sets. We fit the model using the training data and assess its performance on the testing data. Our project will explore these concepts to enhance understanding and insights in this domain.

DATA DESCRIPTION

In this study, we are working with a wine dataset where the response variable is the wine quality rating. Our objective is to predict this rating based on various explanatory variables. The dataset includes the following variables:

y: quality rating (20 maximum)
x_1: wine varietal (0 - Cabernet Sauvignon, 1 - Shiraz)
x_2: pH
x_3: Total SO₂(ppm)
x_4: color density
x_5: wine color
x_6: polymeric pigment color
x_7: anthocyanin color
x_8: total anthocyanins (g/L)
x_9: degree of ionization of anthocyanins (percent)
x_10: ionized anthocyanins (percent)

The dataset comprises 11 columns and 32 rows. After dividing it into training and testing sets, we have 26 rows and 11 columns in the training data, and 6 rows and 11 columns in the testing data.

Initially, we attempted to fit a linear regression model using ordinary least squares (OLS) estimates. However, this approach was compromised by multicollinearity and the presence of outliers, causing repeated failures. Our primary challenge, therefore, is to identify a model that is robust against outliers and multicollinearity without discarding any data points, given the limited size of our dataset.

DATASET

TRAIN DATA

y	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_10
19.2	0	3.85	66	9.35	5.65	2.4	3.25	0.33	19	0.065
18.3	0	3.73	79	11.15	6.95	3.15	3.8	0.36	21	0.076
17.1	0	3.88	73	9.4	5.75	2.1	3.65	0.4	18	0.073
16.8	0	3.98	75	8.55	5.05	2.05	3	0.49	12	0.06
15.2	0	3.66	86	6.4	4	1.5	2.5	0.27	19	0.05
15.2	0	3.91	78	5.8	3.3	1.4	1.9	0.4	9	0.038
14	0	3.47	178	3.6	2.25	0.75	1.5	0.37	8	0.03
14	0	3.91	81	3.9	2.15	1	1.15	0.32	7	0.023
13.8	0	3.75	108	5.8	3.2	1.6	1.6	0.38	8	0.032
13.6	0	3.9	92	5.4	2.85	1.55	1.3	0.44	6	0.026
12.8	0	3.92	96	5	2.7	1.4	1.3	0.35	7	0.026
18.5	1	3.87	89	9.15	5.6	1.95	3.65	0.46	16	0.073
17.3	1	3.97	59	10.25	6.1	2.4	3.7	0.4	19	0.074
16.3	1	3.76	22	8.2	5	1.85	3.15	0.25	25	0.063
16	1	3.98	58	10.15	6	2.6	3.4	0.38	18	0.068
16	1	3.88	85	6.85	4.1	1.5	2.6	0.33	16	0.052
15.5	1	3.98	94	5.45	3.05	1.5	1.55	0.41	8	0.031
15.3	1	3.69	122	8	5.05	1.9	3.15	0.27	23	0.063
15.3	1	3.77	144	5.6	3.35	1.1	2.25	0.36	12	0.045
14.8	1	3.74	10	7.9	4.75	1.95	2.8	0.25	23	0.056
14.3	1	3.76	100	5.55	3.25	1.15	2.1	0.34	12	0.042
14.3	1	3.91	73	4.65	2.7	0.95	1.75	0.36	10	0.035
14.2	1	3.6	301	4.25	2.4	1.25	1.15	0.42	6	0.023
13.8	1	3.9	67	7.4	4.4	1.6	2.8	0.45	13	0.056
12.5	1	3.8	89	5.35	3.15	1.2	1.95	0.32	12	0.039
11.5	1	3.65	192	6.35	3.9	1.25	2.65	0.63	8	0.053

TEST DATA

y	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_10
17.3	0	3.86	99	12.85	7.7	3.9	3.8	0.35	22	0.076
16.5	0	3.85	61	10.3	6.2	2.5	3.7	0.38	20	0.074
15.8	0	3.93	66	4.9	2.75	1.2	1.55	0.29	11	0.031
16.3	1	3.76	77	8.35	5.05	1.9	3.15	0.37	17	0.063
15.7	1	3.75	120	8.8	5.5	1.85	3.65	0.39	19	0.073
14	1	3.76	104	8.7	5.1	2.25	2.85	0.34	17	0.057

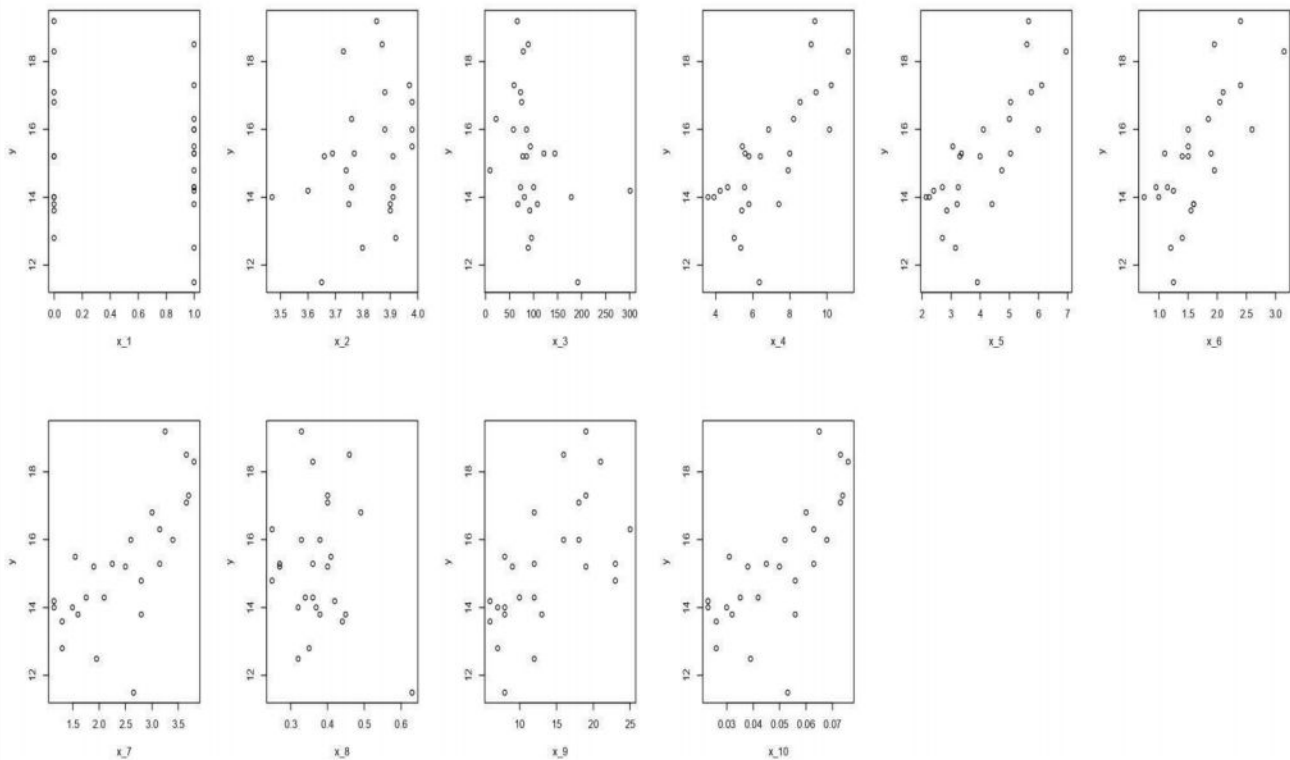
DATA SOURCE

Reference: <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>

EXPLORATORY DATA ANALYSIS (EDA)

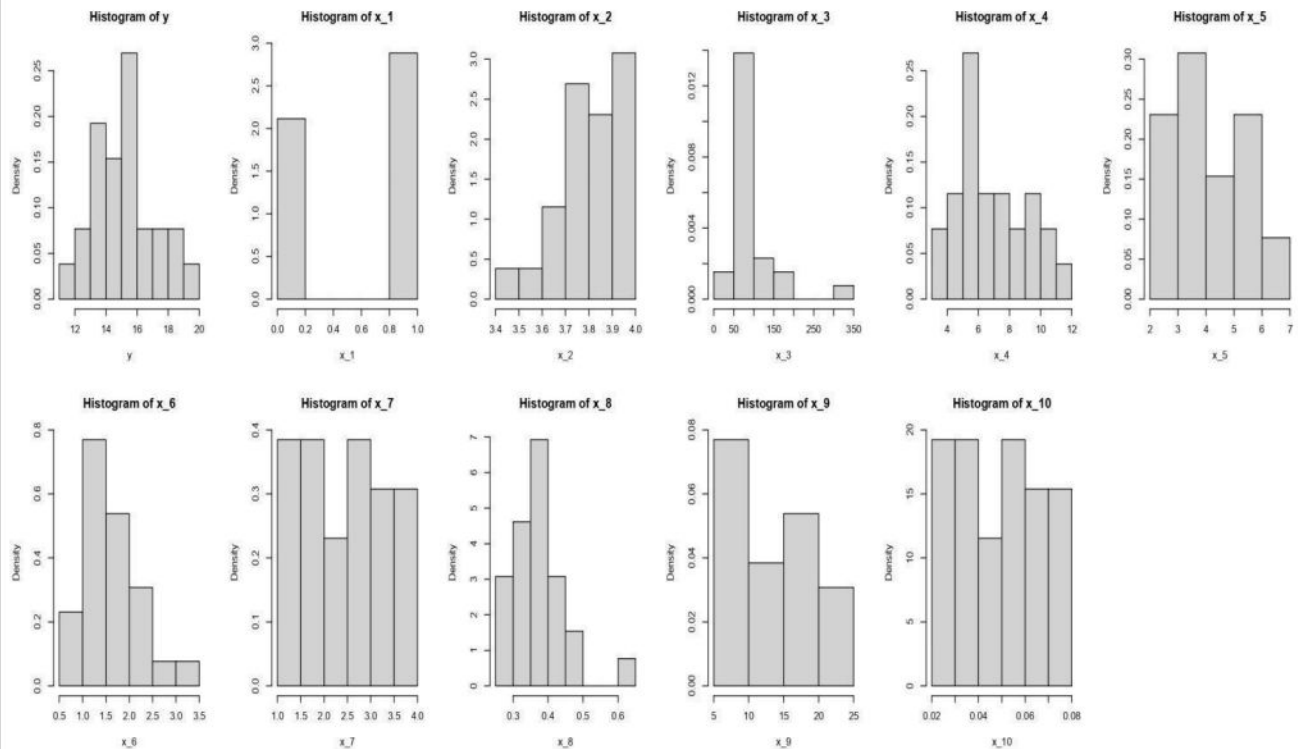
Exploratory data analysis (EDA) is the process of summarizing the main characteristics of a dataset to gain initial insights and detect patterns or anomalies. In this section, we will investigate the relationship between the response variable and the other explanatory variables. We will draw histograms to visualize their distributions and use boxplots to identify any outliers present in the dataset. Our EDA will be conducted solely on the training data as we consider it unnecessary to perform EDA on the test data.

PLOT OF Y VS X_i $i=1(1)10$



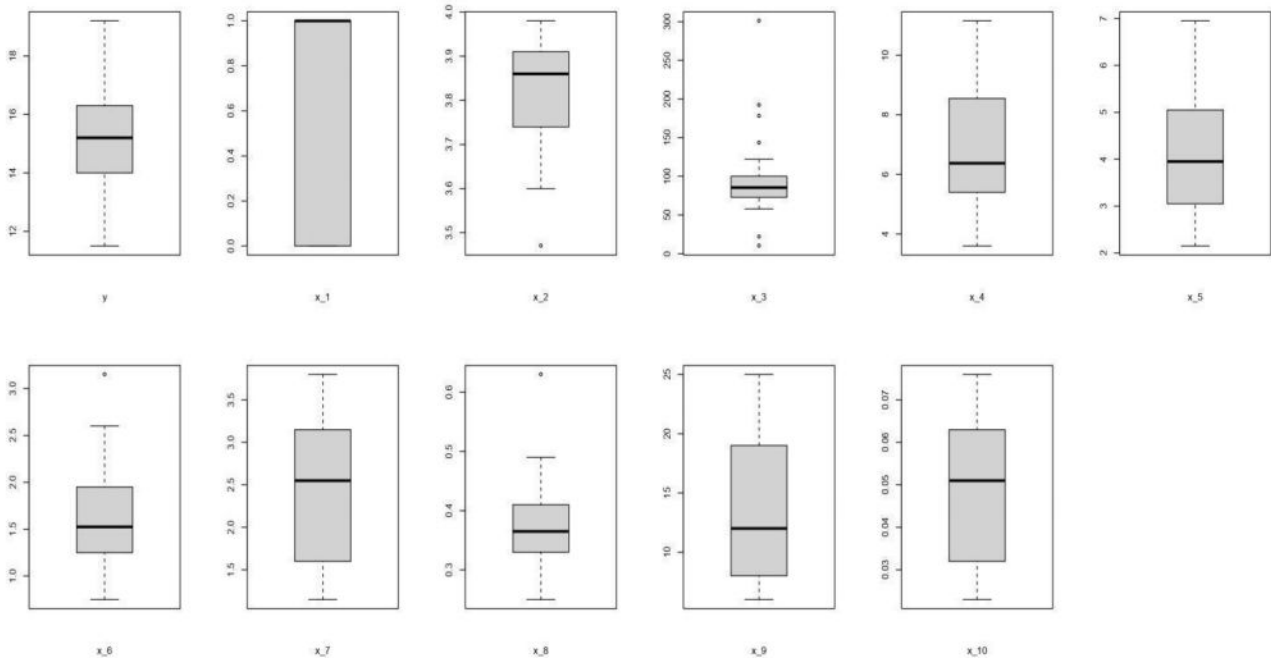
Based on the plot, it's evident that apart from x_1 and x_2 , there exists a linear relationship between the other explanatory variables and y . Therefore, a linear model would be appropriate in this case. However, we will include x_1 and x_2 along with the other variables in our analysis.

HISTOGRAM OF THE RESPONSE AND EXPLANATORY VARIABLES



The histogram provides a visual representation of the observations of both the response and explanatory variables, giving us a sense of their distributions.

BOXPLOTS OF RESPONSE VARIABLE AND EXPLANATORY VARIABLES



The boxplots indicate the presence of suspected outliers in the explanatory variables x_2 , x_3 , x_6 and x_8 . It's worth noting that some of these outliers are indeed significant. Therefore, it's evident that we need to consider the presence of outliers when fitting the regression model. It's essential to remain mindful of this aspect throughout our analysis.

This completes our required exploratory data analysis for this study.

REGRESSION ANALYSIS

In this section, we'll explore different data analysis methods and appropriately fit regression models. Additionally, we'll identify methods unsuitable for our dataset and understand why they're not being used.

Based on the descriptive analysis and the scatter plot, it's apparent that, apart from the explanatory variables x_1 and x_2 , the remaining variables exhibit a linear relationship with the response variable y . However, due to the presence of multiple explanatory variables, simple linear regression using any single explanatory variable is not appropriate or relevant.

Nevertheless, it appears that fitting a Multiple Linear Regression model could be both feasible and worthwhile in this scenario. This approach allows us to consider the combined influence of all explanatory variables on the response variable y , providing a more comprehensive understanding of the relationship between the variables.

❖ METHOD 1 : MULTIPLE LINEAR REGRESSION

Multiple regression involves analyzing the relationship between a dependent variable and two or more independent variables. Its assumptions include linearity, independence of errors, homoscedasticity (constant variance of errors), absence of multicollinearity (no strong correlations between independent variables), and normally distributed errors with a mean of zero.

In our dataset, some variables have a Variance Inflation Factor (VIF) greater than 10, indicating the presence of multicollinearity. Consequently, we cannot apply the Multiple Linear Regression method as one of our assumptions is violated.

Now, we'll proceed to our next method known as Stepwise Regression.

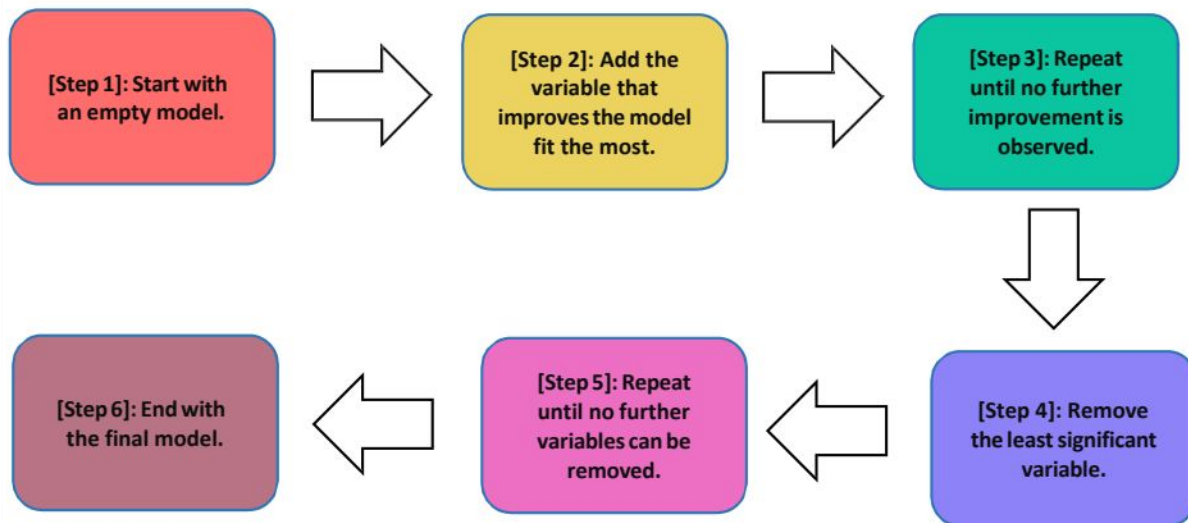
❖ METHOD 2 : STEPWISE REGRESSION

Stepwise regression is a statistical technique used to select the most significant variables for inclusion in a regression model. It iteratively adds or removes variables based on their statistical significance.

The assumptions of stepwise regression include:

1. Linearity: The relationship between the independent and dependent variables is linear.
2. Independence: Observations are independent of each other.
3. Homoscedasticity: The variance of the errors is constant across all levels of the independent variables.
4. Normality: The errors are normally distributed with a mean of zero.
5. No multicollinearity: There is no strong correlation between independent variables.

[Algorithm of Stepwise Regression]



After applying the stepwise regression method, we observe that the data required for the model is also affected by multicollinearity. Consequently, we are unable to proceed with this method as well.

Upon developing suspicions regarding the integrity of the data, we proceed to implement regularization techniques in the ensuing step.

❖ METHOD 3 : REGULARIZATION TECHNIQUE

Regularization in regression is a method employed to prevent overfitting by imposing a penalty on the size of the coefficients. This penalty encourages the model to select simpler, more generalizable solutions by shrinking the coefficients towards zero. Regularization techniques, such as Ridge and Lasso regression, strike a balance between fitting the data well and avoiding excessive complexity, ultimately enhancing the model's predictive performance on new data.

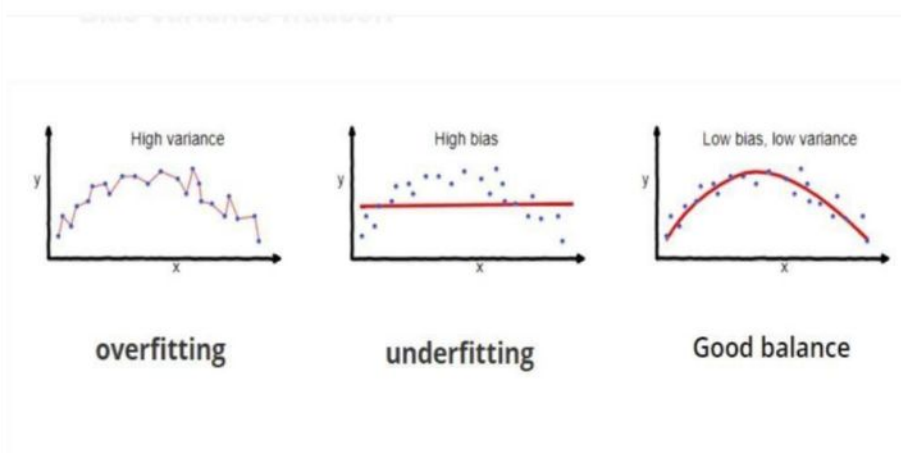


IMAGE : REGULARIZATION IN MACHINE LEARNING

Types of regularization techniques :

Ridge regression, a refined technique in regression analysis, fosters stability by constraining the magnitudes of coefficients, thus curbing overfitting. This method gracefully balances model complexity and accuracy, ensuring robust predictions in the face of multicollinearity and high-dimensional datasets.

Lasso regression, a beacon of elegance in regression analysis, artfully selects variables by shrinking some coefficients to zero, effectively performing variable selection and regularization simultaneously. This method gracefully navigates the intricate landscape of high-dimensional data, delivering parsimonious models without compromising predictive accuracy.

Regularization techniques like ridge and lasso can effectively mitigate multicollinearity, yet they come with constraints. They are not suitable when outliers are present in the dataset. Thus, it is imperative to assess whether outliers exist in both the response and predictor variables before employing these techniques.

Next, we create box plots for the data to examine the presence of outliers.

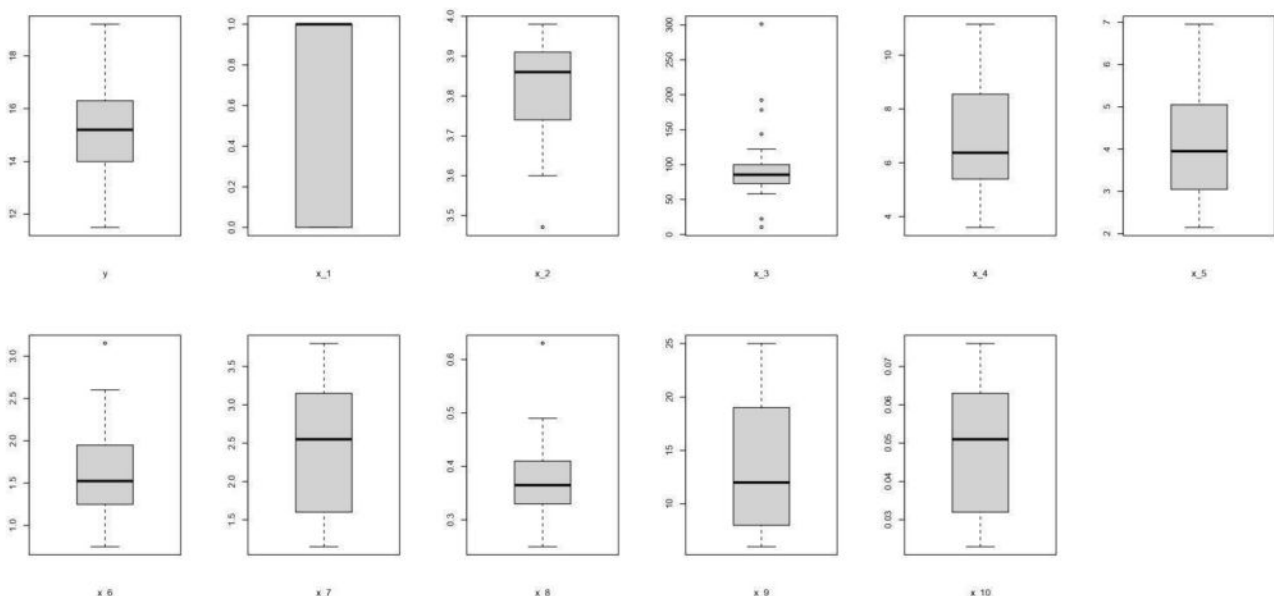


IMAGE : BOXPLOTS

The box plot analysis reveals the presence of outliers in the predictor variables x_2 , x_3 , x_6 , and x_8 . Given that ridge regression and lasso can yield misleading results in the presence of outliers, applying these regularization methods to fit the model becomes unfeasible once again.

Henceforth, we require a robust measure capable of handling heteroscedasticity, autocorrelation, multicollinearity, and non-normality simultaneously, while also exhibiting minimal sensitivity to outliers. A robust measure refers to a statistical method or technique that remains effective and reliable in the presence of outliers, non-normality, heteroscedasticity, multicollinearity, and other deviations from standard assumptions. Such measures are resilient and provide accurate results even when data does not conform to ideal statistical assumptions. We now proceed with our final approach using robust regression methodology.

❖ METHOD 4 : ROBUST REGRESSION

The ordinary least squares estimates for linear regression are optimal when all of the regression assumptions are valid. When some of these assumptions are invalid, least squares regression can perform poorly. Residual diagnostics can help guide you to where the breakdown in assumptions occurs but can be time-consuming and sometimes difficult for the untrained eye. **Robust regression** methods provide an alternative to least squares regression by requiring less restrictive assumptions. These methods attempt to dampen the influence of outlying cases in order to provide a better fit to the majority of the data.

Outliers have a tendency to pull the least squares fit too far in their direction by receiving much more "weight" than they deserve. Typically, you would expect that the weight attached to each observation would be on average $1/n$ in a data set with n observations. However, outliers may receive considerably more weight, leading to distorted estimates of the regression coefficients. This distortion results in outliers which are difficult to identify since their residuals are much smaller than they would otherwise be (if the distortion wasn't present). As we have seen, scatterplots may be used to assess outliers when a small number of predictors are present. However, the complexity added by additional predictor variables can hide the outliers from view in these scatterplots. Robust regression down-weights the influence of outliers, which makes their residuals larger and easier to identify.

For our first robust regression method, suppose we have a data set of size n such that

$$y_i = x_i^T \beta + \epsilon_i$$

$$\epsilon_i(\beta) = y_i - x_i^T \beta,$$

where $i = 1, \dots, n$. Here we have rewritten the error term as $\epsilon_i(\beta)$ to reflect the error term's dependency on the regression coefficients. Ordinary least squares are sometimes known as 2-norm regression since it is minimizing the 2-norm of the residuals (i.e., the squares of the residuals). Thus, observations with high residuals (and high squared residuals) will pull the least squares to fit more in that direction. An alternative is to use what is sometimes

known as least absolute deviation (or 1-norm regression), which minimizes the 1-norm of the residuals (i.e., the absolute value of the residuals). Formally defined, the least absolute deviation estimator is

$$\hat{\beta}_{LAD} = \arg \min_{\beta} \sum_{i=1}^n |\epsilon_i(\beta)|$$

which in turn minimizes the absolute value of the residuals (i.e., $|r_i|$).

Another quite common robust regression method falls into a class of estimators called M-estimators (and there are also other related classes such as R-estimators and S-estimators, whose properties we will not explore). M-estimators attempt to minimize the sum of a chosen function (\cdot) which is acting on the residuals. Formally defined, M-estimators are given by

$$\hat{\beta}_M = \arg \min_{\beta} \sum_{i=1}^n (\epsilon_i(\beta))$$

The M stands for "maximum likelihood" since (\cdot) is related to the likelihood function for a suitable assumed residual distribution. Notice that, if assuming normality, then $(\cdot) = 0.5 \cdot x^2$ results in the ordinary least squares estimate.

Some M-estimators are influenced by the scale of the residuals, so a scale-invariant version of the M-estimator is used:

$$\hat{\beta}_M = \arg \min_{\beta} \sum_{i=1}^n (\epsilon_i(\beta) / \rho)$$

where ρ is a measure of the scale. An estimate of ρ is given by

$$\hat{\rho} = (\text{med } |r_i - \tilde{r}|) / 0.6745$$

Where \tilde{r} is the median of the residuals. Minimization of the above is accomplished primarily in two steps:

1. Set $\rho = 0$ for each $i = 0, 1, \dots, p-1$, resulting in a set of p nonlinear equations $\sum_{i=1}^n \rho_i = 0$, where $\rho_i = \rho(\cdot)$. $\rho(\cdot)$ is called the influence function.
2. A numerical method called iteratively reweighted least squares (IRLS) is used to iteratively estimate the weighted least squares estimate until a stopping criterion is met.

The common method used in M estimation is given below:

Huber's Method

$$\rho(z) = \begin{cases} z^2, & \text{if } |z| < c; \\ |2z|c - c^2, & \text{if } |z| \geq c \end{cases}$$

$$\psi(z) = \begin{cases} z, & \text{if } |z| < c; \\ c[\text{sgn}(z)], & \text{if } |z| \geq c \end{cases}$$

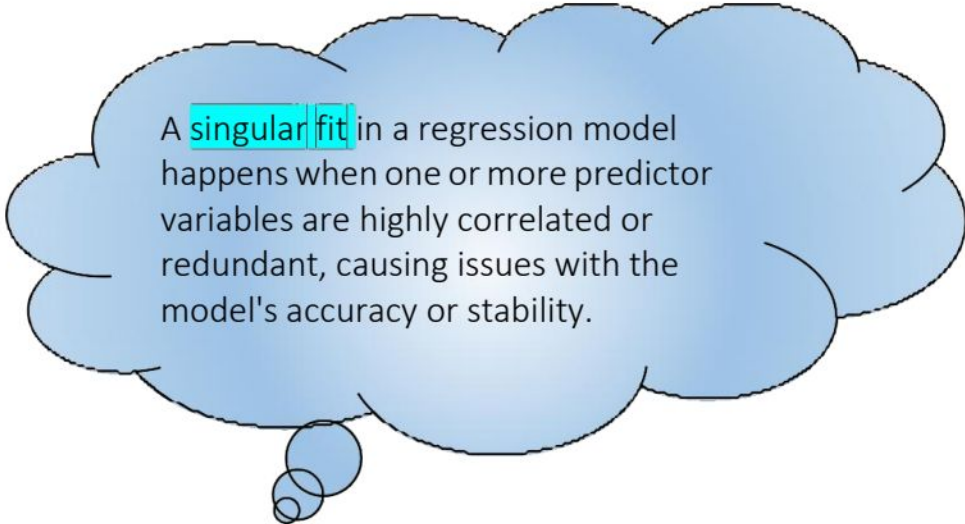
$$w(z) = \begin{cases} 1, & \text{if } |z| < c; \\ c/|z|, & \text{if } |z| \geq c, \end{cases}$$

where $c \approx 1.345$.

◆ MODEL FITTING

To conduct all the required analyses and calculations, we utilize R Studio. We begin by loading the dataset into R Studio. Subsequently, we proceed with the essential calculations and analyses as outlined below.

In order to prevent singular fits, we need to exclude two predictor variables, namely x_7 and x_10, from our analysis. Therefore, we'll proceed with fitting the robust regression model using the remaining predictor variables.



A **singular fit** in a regression model happens when one or more predictor variables are highly correlated or redundant, causing issues with the model's accuracy or stability.

After removing the unnecessary variables, the summary of the fitted robust regression model is as follows:

Residuals:

Min	1Q	Median	3Q	Max
-6.7837	-0.5009	-0.1042	0.2349	2.9251

Coefficients :

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-5.5726901	10.6276912	-0.524	0.6068
x_1	0.3357257	0.4438049	0.756	0.4597
x_2	4.0418534	2.6103963	1.548	0.1399
x_3	0.0004168	0.0061807	0.067	0.9470
x_4	-4.2431266	1.5660097	-2.710	0.0149 *
x_5	6.8012671	2.1284460	3.195	0.0053 **
x_6	0.8724919	1.2534564	0.696	0.4958
x_8	11.6848802	7.6697542	1.524	0.1460
x_9	0.0818414	0.1640936	0.499	0.6243

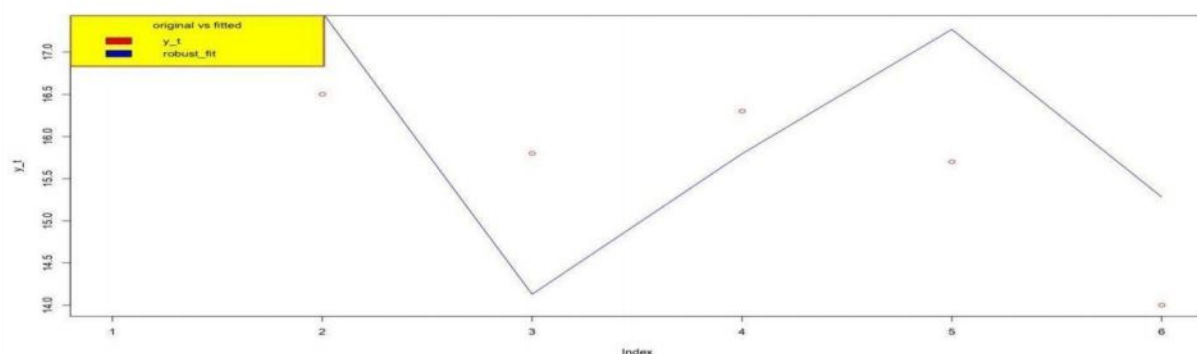
The fitted robust regression model is given by ,

$$y_{\text{fit}} = (-7.609163) + (-0.581110 * x_1) + (4.617232 * x_2) + (0.008216 * x_3) + (-2.086893 * x_4) + (4.093641 * x_5) + (-0.190132 * x_6) + (4.706333 * x_8) + (0.080280 * x_9)$$

Now this model is fitted on the basis of the train dataset . Now we use the concept of cross validation in order to check whether it is a good fitting over the test data also .

◆ CROSS VALIDATION

Cross-validation is a technique used to evaluate how well a machine learning model will perform on new, unseen data. It works by dividing the data into several parts, training the model on some parts, and testing it on the remaining parts. This process is repeated multiple times, and the results are averaged to get a more accurate measure of the model's performance. In our case due to as it is a very small dataset so we perform the process once only. We now apply this model to the test data and compare the observed and predicted values of y . The plot of the original versus fitted values is shown below.



From the plot, it is clear that although the overall fit is not very good, we have to be satisfied with this result since our other approaches have repeatedly failed. Thus, the model fitting and cross-validation are now complete.

Now we propose some measures to evaluate the efficiency of the model. Our first and foremost measure is the sum of squared errors (SSE), which is defined as follows:

The SSE (Error Sum of Squares) is given by,

$$SSE = \sum_{i=1}^n (y_i - y_{i,\text{fitted}})^2 = 9.317039$$

Note that this measure is calculated using the test data.

Here we introduce another measure that is adjusted R square which is defined by

R-squared, often written R^2 , is the proportion of the variance in the **response variable** that can be explained by the predictor variables in a **linear regression model**.

The value for R-squared can range from 0 to 1. A value of 0 indicates that the response variable cannot be explained by the predictor variable at all while a value of 1 indicates that the response variable can be perfectly explained without error by the predictor variables.

The **adjusted R-squared** is a modified version of R-squared that adjusts for the number of predictors in a regression model. It is calculated as:

$$\text{Adjusted } R^2 = 1 - [(1 - R^2) * (n - 1) / (n - k - 1)]$$

Robust residual standard error: 1.003

Multiple R-squared: 0.7278, Adjusted R-squared: 0.5997

Note that this measure is based on the training data that is the model .

As here we see that the adjusted r square measure is between 0.5 to 0.7 . So we conclude that overall this is an average fit .

CONCLUSION

The data analysis was approached using various methods. Initially, we attempted a multiple linear regression model, but the presence of multicollinearity among the predictor variables, as indicated by the Variance Inflation Factor (VIF) measure, prevented us from applying this method effectively.

Next, we explored stepwise regression to see if we could identify a suitable linear regression model by discarding some of the unimportant predictors that were causing multicollinearity. Despite trying both forward and backward stepwise regression, we could not eliminate the multicollinearity.

Following this, we considered ridge regression. However, ridge regression is sensitive to outliers, and our dataset contained numerous outliers, making this method unsuitable.

As our final approach, we chose robust regression using the Huber M-estimator. This method is less affected by multicollinearity and is more robust against outliers. Using robust regression, we obtained our fitted regression model.

From the fitted model, we calculated the Sum of Squared Errors (SSE) for the test data, which was 9.317039. Additionally, we calculated the adjusted R-squared value for the training data, which was 0.5997. Based on these observations and the corresponding plot, we concluded that while the model is not a very good fit, it is an average fit.

Further Procedures: In future research, we may consider discarding the outlier data points and then applying ridge regression. However, due to the small size of the dataset, we chose not to discard any data points to retain complete information. Additionally, we could employ various winsorization techniques before proceeding, but due to time constraints, we were unable to perform these experiments. These could be explored in future research.

REFERENCE

- <https://online.stat.psu.edu/stat501/lesson/topic-1-robust-regression>
- <https://www.geeksforgeeks.org/machine-learning/>
- <https://www.geeksforgeeks.org/cross-validation-machine-learning/>
- <https://www.statology.org/robust-regression-in-r/>
- <https://www.geeksforgeeks.org/stepwise-regression-in-r/>
- <https://www.geeksforgeeks.org/regularization-in-machine-learning/>

APPENDIX

FINAL R CODE

```
install.packages('readxl')
library('readxl')
mydata=read_excel(file.choose(),col_names=TRUE)
View(mydata)

set.seed(1)
sample=sample(c(TRUE,FALSE),nrow(mydata),replace=TRUE,prob=c(0.8,0.2))
train=mydata[sample,];train
test=mydata[!sample,];test
mydata=train;mydata
mydata_dash=test;mydata_dash
mydata1=data.frame(mydata);mydata1
mydata2=data.frame(mydata_dash);mydata2
summary(mydata)

y=mydata1[,1];y
x_1=mydata1[,2];x_1
x_2=mydata1[,3];x_2
x_3=mydata1[,4];x_3
x_4=mydata1[,5];x_4
x_5=mydata1[,6];x_5
x_6=mydata1[,7];x_6
x_7=mydata1[,8];x_7
x_8=mydata1[,9];x_8
x_9=mydata1[,10];x_9
x_10=mydata1[,11];x_10

y_t=mydata2[,1];y_t
x_1_t=mydata2[,2];x_1_t
x_2_t=mydata2[,3];x_2_t
x_3_t=mydata2[,4];x_3_t
x_4_t=mydata2[,5];x_4_t
x_5_t=mydata2[,6];x_5_t
x_6_t=mydata2[,7];x_6_t
x_7_t=mydata2[,8];x_7_t
x_8_t=mydata2[,9];x_8_t
x_9_t=mydata2[,10];x_9_t
x_10_t=mydata2[,11];x_10_t

par(mfrow=c(2,6))
```

```

plot(x_1,y,xlab="x_1")
plot(x_2,y,xlab="x_2")
plot(x_3,y,xlab="x_3")
plot(x_4,y,xlab="x_4")
plot(x_5,y,xlab="x_5")
plot(x_6,y,xlab="x_6")
plot(x_7,y,xlab="x_7")
plot(x_8,y,xlab="x_8")
plot(x_9,y,xlab="x_9")
plot(x_10,y,xlab="x_10")

par(mfrow=c(2,6))
hist(y,xlab="y",freq=FALSE)
hist(x_1,xlab="x_1",freq=FALSE)
hist(x_2,xlab="x_2",freq=FALSE)
hist(x_3,xlab="x_3",freq=FALSE)
hist(x_4,xlab="x_4",freq=FALSE)
hist(x_5,xlab="x_5",freq=FALSE)
hist(x_6,xlab="x_6",freq=FALSE)
hist(x_7,xlab="x_7",freq=FALSE)
hist(x_8,xlab="x_8",freq=FALSE)
hist(x_9,xlab="x_9",freq=FALSE)
hist(x_10,xlab="x_10",freq=FALSE)

par(mfrow=c(2,6))
boxplot(y,xlab="y")
boxplot(x_1,xlab="x_1")
boxplot(x_2,xlab="x_2")
boxplot(x_3,xlab="x_3")
boxplot(x_4,xlab="x_4")
boxplot(x_5,xlab="x_5")
boxplot(x_6,xlab="x_6")
boxplot(x_7,xlab="x_7")
boxplot(x_8,xlab="x_8")
boxplot(x_9,xlab="x_9")
boxplot(x_10,xlab="x_10")

model1=lm(y~x_1+x_2+x_3+x_4+x_5+x_6+x_8+x_9,data=mydata)
model1
library(car)
vif(model1)

# Initialize a model with all predictors
both_model <- lm(y ~ ., data = mydata)
both_model

```

```
# Both-direction stepwise regression
both_model <- step(both_model, direction = "both", trace = 0)
both_model
vif(both_model)

install.packages("robust")
library(robustbase)
robust=lmrob(y~x_1+x_2+x_3+x_4+x_5+x_6+x_8+x_9,data=mydata,method="MM");
robust

robust_fit=(-7.609163)+(-0.581110*x_1_t)+(4.617232*x_2_t)+(0.008216*x_3_t)+(-
2.086893*x_4_t)+(4.093641*x_5_t)+(-0.190132*x_6_t)+(4.706333*x_8_t)+(0.080280*x_9_t)
robust_fit

plot(y_t,col="red")
lines(robust_fit,col="blue",type="l")
legend(x = "topleft", box.col = "brown",bg ="yellow", box.lwd = 2,title="original vs
fitted",legend=c("y_t", "robust_fit"),fill = c("red","blue"))

summary(robust)

sse=sum((y_t-robust_fit)^2);sse
```