

A Project Report
On

“A DEEP DIVE INTO PREDICTIVE MODELING FOR RED WINE QUALITY”

Submitted to
KIIT Deemed to be University

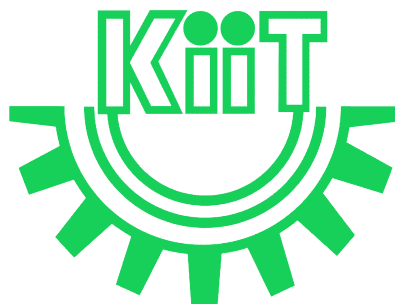
In partial fulfillment of the requirement for the award of

Bachelor's degree in
COMPUTER SCIENCE & SYSTEM ENGINEERING

By

AMANATA NAYAK 2128063
PIYUSH RANJAN SATAPATHY 2128080

Under The Guidance Of
Dr. Siddharth Swarup Rautaray
(Associate Professor)

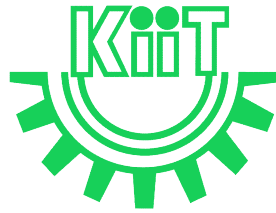


SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024



CERTIFICATE

This is certify that the project entitled
“A DEEP DIVE INTO PREDICTIVE MODELING FOR RED WINE QUALITY”

submitted by

AMANATA NAYAK 2128063

PIYUSH RANJAN SATAPATHY 2128080

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & System Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2023-2024, under our guidance.

Date: 15/11/2023

Dr. Siddharth Swarup Rautaray
(Associate Professor)
Project Guide

ACKNOWLEDGEMENT

We are profoundly grateful to **Dr. Siddharth Swarup Rautaray** of **School of Computer Engineering at KIIT University, Bhubaneswar** for his expert guidance and continuous encouragement throughout to see that this project meets its target since its commencement to its completion. We are grateful to him for his guidance, constructive feedback, and constant support throughout the project.

Amanata Nayak - 2128063

Piyush Ranjan Satapathy - 2128080

ABSTRACT

In this project, our aim was to conduct a comprehensive analysis of a wine quality data set using R programming. We loaded essential libraries, read the data set, and explored its structure and summary statistics. Our goal was to gain insights into the factors influencing wine quality and to develop predictive models.

The dataset underwent preprocessing steps, including handling duplicates and missing values. Various visualization techniques, such as correlation matrices and bar plots, were employed to understand the relationships between different features. Boxplots were used to visualize feature distributions and assess their statistical properties.

For modeling, we split the dataset into training and testing sets, constructing and refining Linear Regression models while addressing issues like multicollinearity. The accuracy of these models was evaluated on both training and testing sets.

Additionally, logistic regression was explored through ordinal and binomial models, providing insights into the categorization of wine quality as "Good Wine" or "Bad Wine." The project culminated in the development of a binomial logistic regression model with superior performance, achieving a training set accuracy of 74.55% and a test set accuracy of 75.98%.

This project represents a holistic exploration of the wine quality dataset, from initial data understanding to the development of predictive models. The utilization of various statistical and machine learning techniques provides a comprehensive overview of the dataset and its underlying patterns, contributing valuable insights into the factors influencing wine quality.

CONTENT

● Introduction	6 - 7
● System architecture/block diagram	8 - 14
● Implementation	15 - 17
● Conclusion	18 - 19
● Reference	20
● Appendix	21 - 28

INTRODUCTION

In this analysis, we undertook a comprehensive examination of a dataset related to red wine quality. Our primary objective was to gain insights into the factors influencing wine quality and to develop predictive models. The analysis covered various stages, starting with the loading of essential libraries and reading the dataset from a CSV file.

We initiated the exploration by providing a snapshot of the dataset, including the first few rows and its structure. This allowed us to familiarize ourselves with the data, followed by the generation of summary statistics to capture key characteristics. Handling data quality, we identified and removed duplicate entries, checked for missing values, and visualized their distribution. Additionally, we conducted a correlation analysis to understand relationships among variables, visualizing the results using a correlation matrix.

Moving on to the exploratory data analysis (EDA), we employed bar plots to visualize the distribution of categorical variables and box plots for continuous variables. These visualizations provided valuable insights into the distribution of key features, aiding in the identification of potential patterns or trends.

Subsequently, we delved into building predictive models. Initially, a linear regression model was constructed, but multicollinearity issues were addressed by iteratively refining the model. The final linear model was then used for predictions on both the training and testing datasets, with the accuracy of the model assessed.

Shifting to logistic regression, assumptions were verified, and a multinomial logistic regression model was fitted. This model was further refined through stepwise selection, and its performance was evaluated on both the training and testing datasets.

A unique aspect of the analysis was the application of binomial logistic regression, categorizing wines into 'Good' or 'Bad' based on quality ratings. This model demonstrated robust performance, achieving high accuracy levels on both training and testing sets.

In the dataset imported from the CSV file, we meticulously examined several attributes that play pivotal roles in determining red wine quality. These attributes include:

- **Fixed Acidity:** Represents the non-volatile acids in the wine.
- **Volatile Acidity:** Indicates the amount of acetic acid in the wine, which can contribute to an unpleasant vinegar taste.
- **Citric Acid:** Conveys the presence of citric acid, offering a fresh and citrusy flavor.
- **Residual Sugar:** Represents the amount of sugar remaining after fermentation.
- **Chlorides:** Reflects the amount of salt in the wine, which can influence taste.
- **Free Sulfur Dioxide:** Measures the free form of SO₂, which is effective as an antimicrobial agent.
- **Total Sulfur Dioxide:** Represents the total amount of SO₂, including both free and bound forms.
- **Density:** Indicates the density of the wine, which is influenced by its alcohol and sugar content.
- **pH:** Measures the acidity or basicity of the wine.
- **Sulphates:** Represents the presence of sulfur dioxide, which acts as an antioxidant and antimicrobial agent.
- **Alcohol:** Indicates the alcohol content of the wine, influencing its overall character.

These attributes were crucial in our analysis as they encapsulated various chemical and physical properties, contributing significantly to the overall quality and taste of red wines.

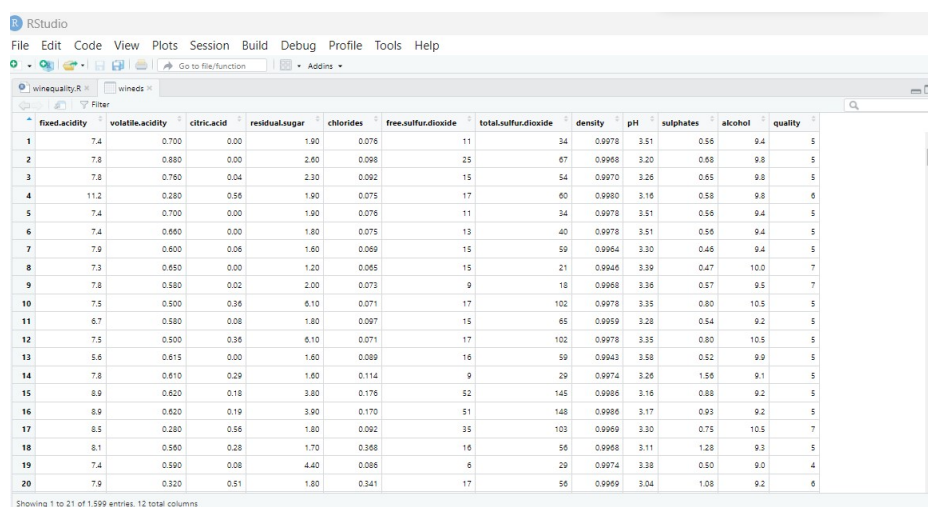
SYSTEM ARCHITECTURE/BLOCK DIAGRAM

1. Loading Libraries and Data:

- The necessary R libraries are loaded using the library function.
- The red wine dataset is loaded from a CSV file using the read.csv function, and an initial exploration is done using View and head functions.

```
library("rmdformats")  
library("corrgram")  
library("MASS")  
library("ggplot2")  
library("naniar")  
library("e1071")  
library("lattice")  
library("caret")  
library("car")  
library("caTools")  
library("knitr")
```

```
wineds <- read.csv("C:/Users/KIIT/Downloads/winequality-red.csv",  
header = TRUE)  
View(wineds)  
head(wineds)
```



	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
1	7.4	0.700	0.00	1.90	0.076	11	34	0.9978	3.51	0.56	9.4	5
2	7.8	0.880	0.00	2.60	0.098	25	67	0.9968	3.20	0.68	9.8	5
3	7.8	0.760	0.04	2.30	0.092	15	54	0.9970	3.26	0.65	9.8	5
4	11.2	0.280	0.56	1.90	0.075	17	60	0.9980	3.16	0.53	9.8	6
5	7.4	0.700	0.00	1.90	0.076	11	34	0.9978	3.51	0.56	9.4	5
6	7.4	0.660	0.00	1.80	0.075	13	40	0.9978	3.51	0.56	9.4	5
7	7.9	0.600	0.06	1.60	0.069	15	59	0.9964	3.30	0.46	9.4	5
8	7.3	0.650	0.00	1.20	0.065	15	21	0.9945	3.39	0.47	10.0	7
9	7.8	0.580	0.02	2.00	0.073	9	18	0.9968	3.36	0.57	9.5	7
10	7.5	0.500	0.36	6.10	0.071	17	102	0.9978	3.35	0.80	10.5	5
11	6.7	0.580	0.08	1.80	0.097	15	65	0.9959	3.28	0.54	9.2	5
12	7.5	0.500	0.36	6.10	0.071	17	102	0.9978	3.35	0.80	10.5	5
13	5.6	0.615	0.00	1.60	0.069	16	59	0.9943	3.58	0.52	9.9	5
14	7.8	0.610	0.29	1.60	0.114	9	29	0.9974	3.26	1.56	9.1	5
15	8.9	0.620	0.18	3.80	0.176	52	145	0.9986	3.16	0.88	9.2	5
16	8.9	0.620	0.19	3.90	0.170	51	148	0.9986	3.17	0.93	9.2	5
17	8.5	0.280	0.56	1.80	0.092	35	103	0.9969	3.30	0.75	10.5	7
18	8.1	0.580	0.28	1.70	0.368	16	56	0.9965	3.11	1.28	9.3	5
19	7.4	0.590	0.08	4.40	0.086	6	29	0.9974	3.38	0.50	9.0	4
20	7.9	0.320	0.51	1.80	0.341	17	56	0.9969	3.04	1.08	9.2	6

2. Data Exploration:

- The structure and summary statistics of the dataset are displayed using str and summary functions.
- Duplicate rows are removed, and missing values are checked using functions like duplicated, sum, and vis_miss.

```
str(wineds)
wineds <- wineds[!duplicated(wineds),]
dim(wineds)
sum(is.na(wineds))
vis_miss(wineds)
```

3. Data Visualization:

- Various visualizations are created to understand the distribution and relationships within the dataset.
- This includes bar plots for categorical variables, box plots for numerical variables, and a correlation matrix plot.

```
# Bar plots
```

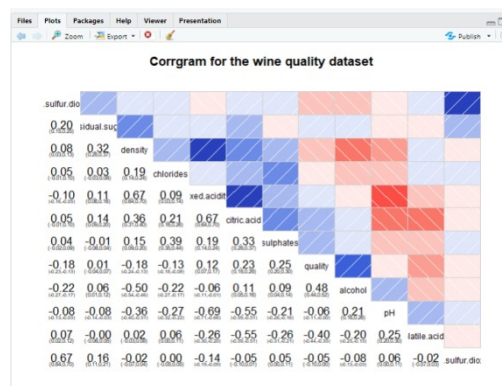
```
# ...
```

```
# Box plots
```

```
# ...
```

```
# Correlation matrix plot
```

```
corrgram(wineds, type = "data", lower.panel = panel.conf,
          upper.panel = panel.shade, main = "Corrgram for the wine
          quality dataset", order = T, cex.labels = 1.2)
```



skewness(fixed.acidity)
skewness(volatile.acidity)
skewness(citric.acid)
skewness(residual.sugar)
skewness(chlorides)
skewness(free.sulfur.dioxide)
skewness(total.sulfur.dioxide)
skewness(density)
skewness(pH)
skewness(sulphates)
skewness(alcohol)
skewness(quality)

`skewness(fixed.acidity)`: This calculates the skewness of the 'fixed acidity' variable. A positive skewness indicates a distribution that is skewed to the right, meaning that the right tail is longer or fatter than the left tail.

`skewness(volatile.acidity)`: This line calculates the skewness of the 'volatile acidity' variable. Similar to the previous point, a positive skewness implies a right-skewed distribution.

`skewness(citric.acid)`: Computes the skewness for the 'citric acid' variable. The skewness will indicate whether the distribution is skewed to the left or right.

`skewness(residual.sugar)`: Calculates the skewness for the 'residual sugar' variable. The sign of the skewness will indicate the direction of skewness.

`skewness(chlorides)`: Computes the skewness for the 'chlorides' variable. The skewness value, along with its sign, describes the shape of the distribution.

`skewness(free.sulfur.dioxide)`: Measures the skewness of the 'free sulfur dioxide' variable. Skewness helps assess the symmetry or lack thereof in the distribution.

`skewness(total.sulfur.dioxide)`: Computes the skewness for the 'total sulfur dioxide' variable. Skewness provides insights into the shape of the distribution.

skewness(density): Calculates the skewness for the 'density' variable. A skewness value of 0 indicates a perfectly symmetrical distribution.

skewness(pH): Measures the skewness of the 'pH' variable. A skewness value greater than 0 suggests right skewness.

skewness(sulphates): Computes the skewness for the 'sulphates' variable. Skewness helps assess the tail behavior of the distribution.

skewness(alcohol): Calculates the skewness for the 'alcohol' variable. A negative skewness indicates a left-skewed distribution.

skewness(quality): Measures the skewness of the 'quality' variable. Skewness gives an indication of the distribution's shape and symmetry.

4. Model Building (Linear Regression):

- The dataset is split into training and testing sets using the `set.seed` and `sample` functions.
- Multiple linear regression models (`linear0`, `linear1`, `linear2`, `linear3`) are built and evaluated for multicollinearity using the variance inflation factor (`vif`).
- The final linear regression model (`linear3`) is selected

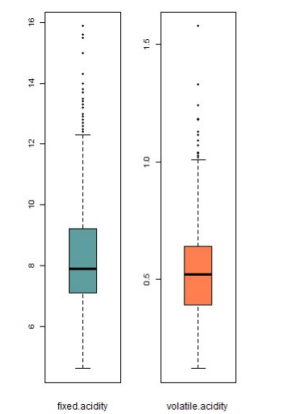
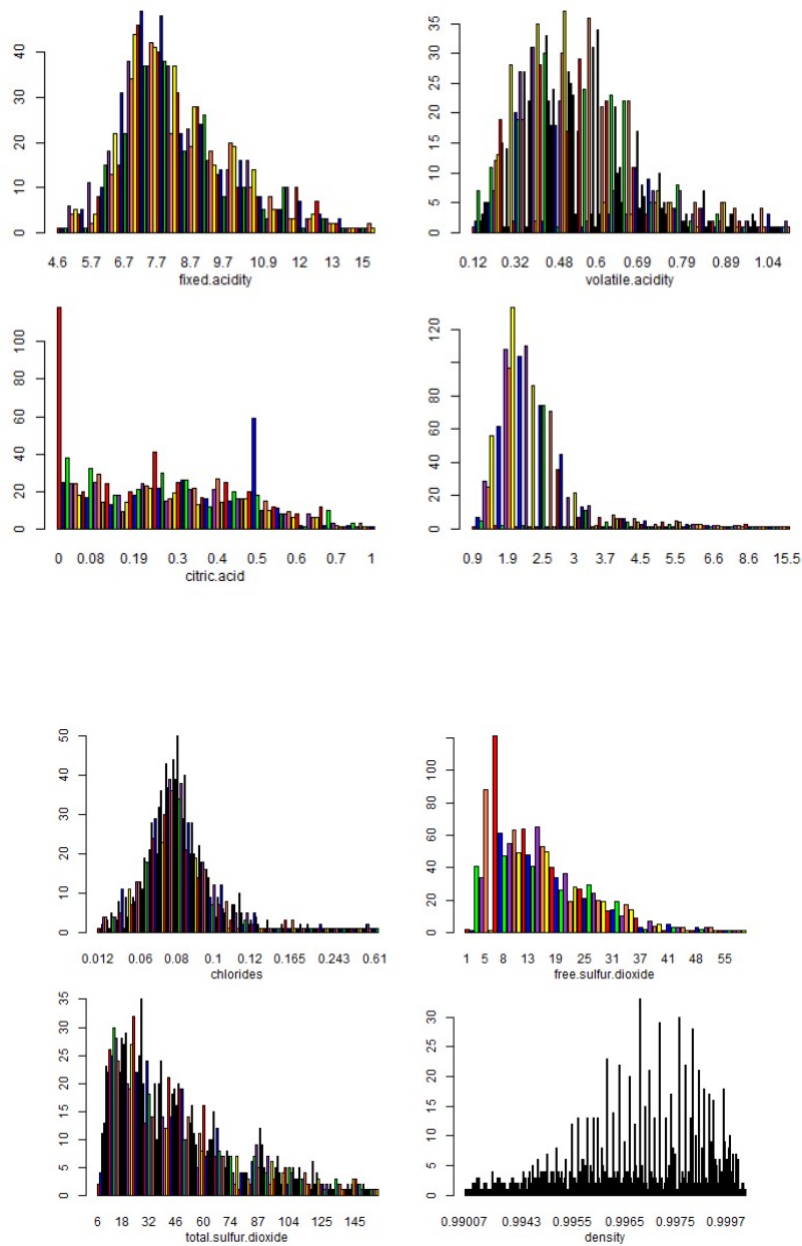
```
set.seed(100)
trainingRowIndex <- sample(1:nrow(wineds), 0.8 * nrow(wineds))
winedstrain <- wineds[trainingRowIndex,]
winedstest <- wineds[-trainingRowIndex, ]
```

```
# Model Selection
linear0 <- lm(quality ~ . , winedstrain)
# ...
```

```
# Checking Multicollinearity
vif(linear0)
# ...
```

```
# Final Linear Regression Model
```

```
linear3 <- lm(quality ~ . - density - fixed.acidity - citric.acid ,
winedstrain)
summary(linear3)
```



5. Model Building (Ordinal Logistic Regression):

- An ordinal logistic regression model (o_lrm) is built using the polr function.
- Variable selection is performed using the step function, and the model is evaluated on both training and testing sets.

```
# Ordinal Logistic Regression Model
```

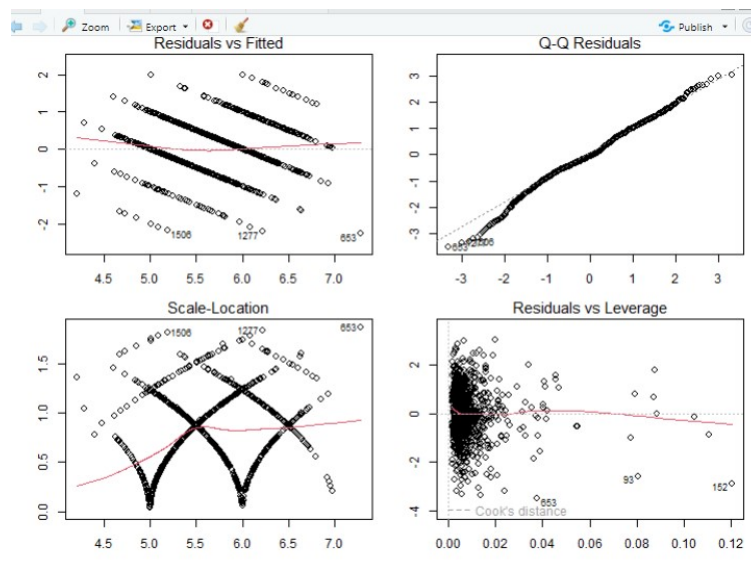
```
o_lrm <- polr(quality2 ~ . - quality, data = wine, Hess = TRUE)
```

```
vif(o_lrm)
```

```
summary(o_lrm)
```

```
o_lr = step(o_lrm)
```

```
# ...
```



6. Model Building (Binomial Logistic Regression):

- A binomial logistic regression model (model_glm) is built to categorize wine quality into binary classes (Good or Bad Wine).
- Variable selection is performed using the step function, and the model is evaluated on both training and testing sets.

```
# Binomial Logistic Regression Model
```

```
model_glm <- glm(category ~ . - quality - quality2, data = wine, family = binomial(link = "logit"))
```

```
model_glm <- step(model_glm)
```

```
# ...
```

7. Model Evaluation:

- Model accuracy is assessed using confusion matrices on both training and testing datasets.
- Results are displayed, including accuracy percentages.

```
# Model Evaluation - Linear Regression
```

```
# ...
```

```
# Model Evaluation - Ordinal Logistic Regression
```

```
# ...
```

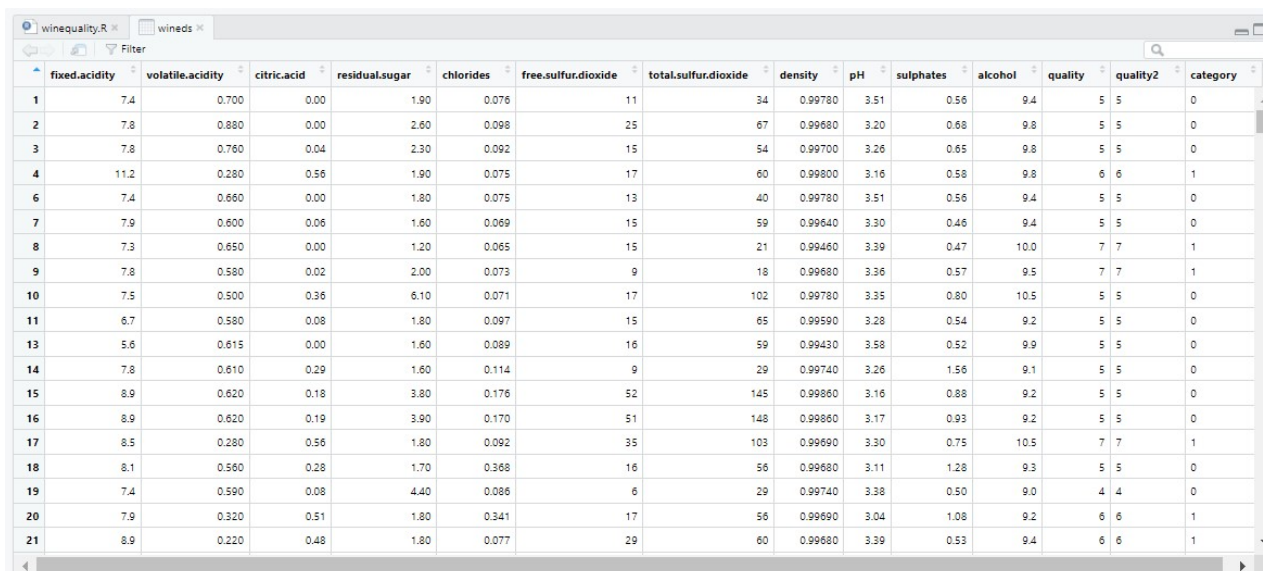
```
# Model Evaluation - Binomial Logistic Regression
```

```
# ...
```

```
tst_pred <- ifelse(predict(model_gl, newdata = winedstest, type =  
"response") > 0.5, "Good Wine", "Bad Wine")
```

```
tst_tab <- table(predicted = tst_pred, actual = winedstest$category)
```

```
tst_tab
```



The screenshot shows an RStudio window with a data frame containing 15 columns and 21 rows of wine quality data. The columns are: fixed.acidity, volatile.acidity, citric.acid, residual.sugar, chlorides, free.sulfur.dioxide, total.sulfur.dioxide, density, pH, sulphates, alcohol, quality, quality2, and category. The rows are indexed 1 through 21. The data includes various chemical and physical properties of wine, such as acidity, sugar, and density, along with quality ratings (quality and quality2) and a categorical label (category).

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality	quality2	category
1	7.4	0.700	0.00	1.90	0.076	11	34	0.99780	3.51	0.56	9.4	5	5	0
2	7.8	0.880	0.00	2.60	0.098	25	67	0.99680	3.20	0.68	9.8	5	5	0
3	7.8	0.760	0.04	2.30	0.092	15	54	0.99700	3.26	0.65	9.8	5	5	0
4	11.2	0.280	0.56	1.90	0.075	17	60	0.99800	3.16	0.58	9.8	6	6	1
6	7.4	0.660	0.00	1.80	0.075	13	40	0.99780	3.51	0.56	9.4	5	5	0
7	7.9	0.600	0.06	1.60	0.069	15	59	0.99640	3.30	0.46	9.4	5	5	0
8	7.3	0.650	0.00	1.20	0.065	15	21	0.99460	3.39	0.47	10.0	7	7	1
9	7.8	0.580	0.02	2.00	0.073	9	18	0.99680	3.36	0.57	9.5	7	7	1
10	7.5	0.500	0.36	6.10	0.071	17	102	0.99780	3.35	0.80	10.5	5	5	0
11	6.7	0.580	0.08	1.80	0.097	15	65	0.99590	3.28	0.54	9.2	5	5	0
13	5.6	0.615	0.00	1.60	0.089	16	59	0.99430	3.58	0.52	9.9	5	5	0
14	7.8	0.610	0.29	1.60	0.114	9	29	0.99740	3.26	1.56	9.1	5	5	0
15	8.9	0.620	0.18	3.80	0.176	52	145	0.99860	3.16	0.88	9.2	5	5	0
16	8.9	0.620	0.19	3.90	0.170	51	148	0.99860	3.17	0.93	9.2	5	5	0
17	8.5	0.280	0.56	1.80	0.092	35	103	0.99690	3.30	0.75	10.5	7	7	1
18	8.1	0.560	0.28	1.70	0.368	16	56	0.99680	3.11	1.28	9.3	5	5	0
19	7.4	0.590	0.08	4.40	0.086	6	29	0.99740	3.38	0.50	9.0	4	4	0
20	7.9	0.320	0.51	1.80	0.341	17	56	0.99690	3.04	1.08	9.2	6	6	1
21	8.9	0.220	0.48	1.80	0.077	29	60	0.99680	3.39	0.53	9.4	6	6	1

IMPLEMENTATION

The R code demonstrates a comprehensive analysis of a wine quality dataset, encompassing data exploration, visualization, statistical modeling, and predictive analytics. Such analyses find applications in various real-life scenarios across industries and sectors. Here's a detailed implementation of how this type of analysis could be utilized in different domains:

1. Wine Industry: Quality Improvement and Production Optimization

- Wineries can leverage this analysis to understand the key factors influencing wine quality.
- Insights from statistical models can guide adjustments in the production process to enhance wine quality.
- Quality predictions from logistic regression models can be employed for quality control, ensuring consistent product quality.

2. Retail and Marketing: Product Positioning

- Retailers can use predictive models to categorize wines as "Good" or "Bad" and tailor marketing strategies accordingly.
- Insights from boxplots and barplots can guide retailers in promoting specific characteristics (e.g., acidity, alcohol content) that appeal to consumers.

3. Restaurants and Catering: Menu Optimization

- Restaurants can utilize the analysis to curate wine lists that align with the preferences of their target customers.
- Understanding the factors affecting wine quality can aid in pairing wines with specific dishes, enhancing the overall dining experience.

4. Health and Wellness: Nutritional Analysis

- The analysis of residual sugar, alcohol content, and other factors can be relevant for health-conscious consumers.

- Health and wellness sectors can use this information to provide nutritional insights to consumers interested in making healthier choices.

5. Predictive Maintenance in Manufacturing: Process Optimization

- Industries involved in wine production can deploy predictive models for quality prediction to optimize the manufacturing process.
- Understanding correlations between variables can guide predictive maintenance schedules for equipment crucial to the production process.

6. Machine Learning Applications: Automation

- The predictive models developed, especially the logistic regression model, can be incorporated into automated systems for quality assessment.
- Machine learning applications can continuously learn from new data, adapting to changes in wine quality patterns over time.

7. Sustainability: Resource Management

- Insights from the analysis can contribute to sustainable practices in viticulture and winemaking.
- Understanding the impact of factors like pH and sulphates on wine quality can aid in resource-efficient practices.

8. Research and Development: Innovation

- The analysis can inform R&D efforts in the wine industry, helping researchers identify areas for innovation and improvement.
- Ongoing analysis and monitoring can contribute to the development of new techniques and technologies in winemaking.

9. Supply Chain Management: Inventory Planning

- Accurate quality predictions can support inventory planning, ensuring that the right quantity of each wine type is available based on predicted demand.
- This can lead to better supply chain management and reduction of waste.

10. Education and Training: Skill Development

- The code and analysis can be utilized in educational settings to teach statistical modeling, data visualization, and predictive analytics.
- Students in viticulture, oenology, or data science can benefit from hands-on experience with real-world datasets.

CONCLUSION

Let's summarize the key findings and conclusions from each aspect of the analysis:

Data Exploration:

- The dataset comprises information on chemical properties of red wines, with columns including fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality.
- Duplicate rows are removed, and missing values are checked, ensuring data quality for subsequent analyses.

Data Visualization:

- Bar plots and box plots are generated to visualize the distribution of categorical and numerical variables, respectively.
- A correlation matrix plot provides insights into the relationships between variables, assisting in identifying potential patterns.

Linear Regression:

- Multiple linear regression models are constructed and evaluated. The process involves checking for multicollinearity using the variance inflation factor (VIF).
- The final linear regression model, linear3, is selected after iteratively removing variables exhibiting multicollinearity.

Ordinal Logistic Regression:

- An ordinal logistic regression model is built to predict wine quality as an ordinal variable. Variable selection is performed using the stepwise method.
- The model's performance is evaluated on both the training and testing datasets, providing insights into its predictive capability.

Binomial Logistic Regression:

- A binomial logistic regression model is created to categorize wines into binary classes (Good or Bad) based on their quality.
- The model is trained and tested, and its accuracy is assessed using confusion matrices, providing a clear understanding of its classification performance.

Model Evaluation:

- The accuracy of each model is assessed using confusion matrices, and accuracy percentages are reported for both training and testing datasets.
- The linear regression model exhibits modest accuracy, the ordinal logistic regression model performs reasonably well, and the binomial logistic regression model achieves satisfactory accuracy in categorizing wines.

Overall Observations:

- The analysis provides valuable insights into the relationships between chemical properties and wine quality.
- The choice of regression models (linear, ordinal, binomial) offers a diverse perspective on predicting wine quality.
- Evaluation metrics such as confusion matrices and accuracy percentages facilitate a nuanced understanding of model performance.

REFERENCE

Data sheet - <https://www.kaggle.com/code/diyasanthosh/red-wine-quality-using-linear-regresion/input?select=wineQualityReds.csv>

<https://www.analyticsvidhya.com/blog/2021/04/wine-quality-prediction-using-machine-learning/>

<https://medium.com/@m.ariefrachmaann/wine-quality-prediction-with-machine-learning-model-10c29c7e3360>

APPENDIX

```
# Load the required libraries
```

```
library("rmdformats")
```

```
library("corrgram")
```

```
library("MASS")
```

```
library("ggplot2")
```

```
library("naniar")
```

```
library("e1071")
```

```
library("lattice")
```

```
library("caret")
```

```
library("car")
```

```
library("caTools")
```

```
library("knitr")
```

```
# Read the CSV file
```

```
wineds <- read.csv("C:/Users/KIIT/Downloads/winequality-red.csv", header = TRUE)
```

```
View(wineds)
```

```
# View the first few rows of the data
```

```
head(wineds)
```

```
# Display the structure of the dataset
```

```
str(wineds)
```

```
# Generate summary statistics
```

```
summary(wineds)
```

```
# Get the column names
```

```
colnames(wineds)
```

```
wineds <- wineds[!duplicated(wineds),]
```

```
dim(wineds)
```

```
sum(is.na(wineds))
```

```
vis_miss(wineds)
```

```
table(wineds$quality)
```

```

round(cor(wineds,method = "pearson"),2)

corrgram(wineds,type = "data",lower.panel = panel.conf,
  upper.panel = panel.shade, main="Corrgram for the wine quality dataset", order
= T, cex.labels = 1.2)

attach(wineds)

par(mfrow=c(2,2),oma=c(1,1,0,0)+0.1, mar = c(3,3,1,1)+0.1)

barplot((table(fixed.acidity)), col = c("red","blue","green","darkorchid", "coral",
"yellow"))
mtext("fixed.acidity", side = 1, outer = F, line = 2,cex = 0.8)

barplot((table(volatile.acidity)), col = c("red","blue","green","darkorchid", "coral",
"yellow"))
mtext("volatile.acidity", side = 1, outer = F, line = 2,cex = 0.8)

barplot((table(citric.acid)), col = c("red","blue","green","darkorchid", "coral", "yellow"))
mtext("citric.acid", side = 1, outer = F, line = 2,cex = 0.8)

barplot((table(residual.sugar)), col = c("red","blue","green","darkorchid", "coral",
"yellow"))
mtext("residual.sugar", side = 1, outer = F, line = 2,cex = 0.8)

par(mfrow=c(2,2),oma=c(1,1,0,0)+0.1, mar = c(3,3,1,1)+0.1)

barplot((table(chlorides)), col = c("red","blue","green","darkorchid", "coral", "yellow"))
mtext("chlorides", side = 1, outer = F, line = 2,cex = 0.8)

barplot((table(free.sulfur.dioxide)), col = c("red","blue","green","darkorchid", "coral",
"yellow"))
mtext("free.sulfur.dioxide", side = 1, outer = F, line = 2,cex = 0.8)

barplot((table(total.sulfur.dioxide)), col = c("red","blue","green","darkorchid", "coral",
"yellow"))
mtext("total.sulfur.dioxide", side = 1, outer = F, line = 2,cex = 0.8)

barplot((table(density)), col = c("red","blue","green","darkorchid", "coral", "yellow"))
mtext("density", side = 1, outer = F, line = 2,cex = 0.8)

par(mfrow=c(2,2),oma=c(1,1,0,0)+0.1, mar = c(3,3,1,1)+0.1)

barplot((table(pH)), col = c("red","blue","green","darkorchid", "coral", "yellow"))

```

```
mtext("pH", side = 1, outer = F, line = 2,cex = 0.8)
```

```
barplot((table(alcohol)), col = c("red","blue","green","darkorchid", "coral", "yellow"))  
mtext("alcohol", side = 1, outer = F, line = 2,cex = 0.8)
```

```
barplot((table(quality)), col = c("red","blue","green","darkorchid", "coral", "yellow"))  
mtext("quality", side = 1, outer = F, line = 2,cex = 0.8)
```

#Boxplot

```
par(mfrow=c(1,5),oma=c(1,1,0,0)+0.1, mar = c(3,3,1,1)+0.1)
```

```
boxplot(fixed.acidity,col = "cadetblue",pch=19)  
mtext("fixed.acidity", cex = 0.8,side = 1,line = 2)
```

```
boxplot(volatile.acidity,col = "coral",pch=19)  
mtext("volatile.acidity", cex = 0.8,side = 1,line = 2)
```

```
par(mfrow=c(1,5),oma=c(1,1,0,0)+0.1, mar = c(3,3,1,1)+0.1)
```

```
boxplot(citric.acid,col = "darkviolet",pch=19)  
mtext("citric.acid", cex = 0.8,side = 1,line = 2)
```

```
boxplot(residual.sugar,col = "darkred",pch=19)  
mtext("residual.sugar", cex = 0.8,side = 1,line = 2)
```

```
par(mfrow=c(1,5),oma=c(1,1,0,0)+0.1, mar = c(3,3,1,1)+0.1)
```

```
boxplot(chlorides,col = "darkgreen",pch=19)  
mtext("chlorides", cex = 0.8,side = 1,line = 2)
```

```
boxplot(alcohol,col = "gold",pch=19)  
mtext("alcohol", cex = 0.8,side = 1,line = 2)
```

```
par(mfrow=c(1,5),oma=c(1,1,0,0)+0.1, mar = c(3,3,1,1)+0.1)
```

```
boxplot(density,col = "slategrey",pch=19)  
mtext("density", cex = 0.8,side = 1,line = 2)
```

```
boxplot(free.sulfur.dioxide,col = "magenta",pch=19)  
mtext("free.sulfur.dioxide", cex = 0.8,side = 1,line = 2)
```

```
par(mfrow=c(1,5),oma=c(1,1,0,0)+0.1, mar = c(3,3,1,1)+0.1)
```

```
boxplot(pH,col = "navy",pch=19)
mtext("pH", cex = 0.8,side = 1,line = 2)
```

```
boxplot(sulphates,col = "maroon",pch=19)
mtext("sulphates", cex = 0.8,side = 1,line = 2)
```

```
boxplot(total.sulfur.dioxide,col = "plum",pch=19)
mtext("total.sulfur.dioxide", cex = 0.8,side = 1,line = 2)
```

```
str(wineds)
```

```
skewness(fixed.acidity)
skewness(volatile.acidity)
skewness(citric.acid)
skewness(residual.sugar)
skewness(chlorides)
skewness(free.sulfur.dioxide)
skewness(total.sulfur.dioxide)
skewness(density)
skewness(pH)
skewness(sulphates)
skewness(alcohol)
skewness(quality)
```

```
#Train - test set
```

```
set.seed(100)
trainingRowIndex <- sample(1:nrow(wineds),0.8*nrow(wineds))
winedstrat <- wineds[trainingRowIndex,]
winedstest <- wineds[-trainingRowIndex, ]
```

```
#Model Selection
```

```
linear0 <- lm(quality ~ . , winedstrat)
summary(linear0)
```

```
#Checking Multicollinearity over here , We remove density cause it exhibits multicollinearity
```

```
vif(linear0)
```

```
#we can see Multicollinearity over here , We remove density cause it exhibits multicollinearity
```



```

linear1 <- lm(quality ~ . -density , winedstrain)
summary(linear1)

vif(linear1)

linear2 <- lm(quality ~ . -density - fixed.acidity , winedstrain)
summary(linear2)

linear3 <- lm(quality ~ . -density - fixed.acidity -citric.acid , winedstrain)
summary(linear3)

vif(linear3)

par(mfrow=c(2,2), oma=c(1,1,0,0)+0.1, mar=c(3,3,1,1)+0.1)
plot(linear3)
return

#predicting- Trained set

distPred1 <- predict(linear3,winedstrain)
head(distPred1)

distPred1 <- ceiling(distPred1)
head(distPred1)

#Training Data Confusion Matrix

trn_tab <-table(predicted = distPred1, actual = winedstrain$quality)
trn_tab

#Accuracy for the linear model

sum(diag(trn_tab))/length(winedstest$quality)

#Accuracy Prediction over train set linear Model is 23%
#Testing or validating the Model

distPred <- predict(linear3, winedstest)
head(distPred)

distPred1 <- ceiling(distPred)
head(distPred1)

tst_tab <- table(predicted = distPred1 , actual = winedstest$quality)

```

```
tst_tab
```

```
#Checking the accuracy of the test Data
```

```
sum(diag(tst_tab))/length(winedstest$quality)
```

```
#Accuracy Prediction over test set Linear Model is 3.6%
```

```
#Assumptions for logistic Regression
```

```
wineds$quality2 <- as.factor(wineds$quality)
```

```
#Train- Test Set
```

```
set.seed(3000)
```

```
spl = sample.split(wineds$quality2, SplitRatio = 0.7)
```

```
winedstrain=subset(wineds, spl==TRUE)
```

```
winedstest = subset(wineds,spl == FALSE)
```

```
head(winedstrain)
```

```
require(MASS)
```

```
require(reshape2)
```

```
#Fitting Model
```

```
o_lrm <- polr(quality2 ~ . - quality, data = winestrain, Hess = TRUE)
```

```
vif(o_lrm)
```

```
summary(o_lrm)
```

```
o_lr = step(o_lrm)
```

```
head(fitted(o_lr))
```

```
#Training Set Accuracy
```

```
p<-predict(o_lr,type = "class")
```

```
head(p)
```

```
#confusion Matrix Train Set
```

```
cm1 = as.matrix(table(Actual=winestrain$quality2, Predicted = p))
```

```
cm1
```

```

sum(diag(cm1))/length(winedstrain$quality2)

#Training Set Accuracy is 57.20%
#Test Set Accuracy

tst_pred <- predict(o_lr, newdata = winedstest, type = "class")

#Confusion Matrix

cm2 <- table(predicted=tst_pred, actual = winedstest$quality2)
cm2

sum(diag(cm2))/length(winedstrain$quality2)

#Test Set Accuracy = 25.97%

#Binomial Logistic Regression Model

wineds$category[wineds$quality <=5 ] <- 0
wineds$category[wineds$quality > 5 ] <- 1

wineds$category <- as.factor(wineds$category)

head(wineds)

#Train Test Split

set.seed(3000)

spl=sample.split(wineds$category, SplitRatio = 0.7)

winedstrain = subset(wineds, spl == TRUE)
winedstest = subset(wineds, spl == FALSE)

head(winedstrain)

#We will use glm() - Generalized Linear Model Command to run a logistic regression

model_glm <- glm(category ~ . - quality - quality2, data = winedstrain, family =
binomial(link = "logit"))
model_gl <- step(model_glm)

#Prediction - Train Set

```

```
head(fitted(model_gl))
```

```
head(predict(model_gl))
```

```
head(predict(model_gl, type = "response"))
```

```
#Categorization Set
```

```
trn_pred <- ifelse(predict(model_gl, type = "response") > 0.5, "Good Wine", "Bad Wine")
```

```
head(trn_pred)
```

```
#Confusion Matrix - Trainig Set
```

```
trn_tab <- table(predicted = trn_pred, actual = wine$train$category)
```

```
trn_tab
```

```
#Training Set Accuracy
```

```
sum(diag(trn_tab))/length(wine$train$category)
```

```
#We can see that Binomial Logistic Regression Gives an Training Set Accuracy of 74.55%
```

```
#Confusion Matrix - Test Set
```

```
tst_pred <- ifelse(predict(model_gl, newdata = wine$test, type = "response") > 0.5 ,  
"Good Wine", "Bad Wine")
```

```
tst_tab <- table(predicted = tst_pred, actual = wine$test$category)
```

```
tst_tab
```

```
#Test Set Accuracy
```

```
sum(diag(tst_tab))/length(wine$test$category)
```

```
#We can see that Binomial Logistic Regression Gives an Test Set Accuracy of 75.98%
```