Statistical Learning Lab

Assignment – 1

Linear Regression Assignment

Name – Semanti Ghosh

Roll No – 22IM10036

1. Loading the dataset – "manufacturing.csv"

Code for loading the data

```r
# Loading the data
getwd() #directory check
setwd("C:/Study/Semester_6/Statistical_Learning_Lab")
getwd()
data <- read.csv("manufacturing.csv")
head(data) #since number is specified, I took the default,
```

Data printed

```
> head(data) #since number is specified, I took the default,
  Temperature...C. Pressure..kPa. Temperature.x.Pressure Material.Fusion.Metric Material.Transformation.Metric Quality.Rating
1         209.7627       8.050855               1688.769               44522.22                        9229576       99.99997
2         243.0379      15.812068               3842.931               63020.76                       14355367       99.98570
3         220.5527       7.843130               1729.823               49125.95                       10728389       99.99976
4         208.9766      23.786089               4970.737               57128.88                        9125702       99.99997
5         184.7310      15.797812               2918.345               38068.20                        6303792      100.00000
6         229.1788       8.498306               1947.632               53136.69                       12037072       99.99879
> 
```

2. Matrix plot and correlation analysis

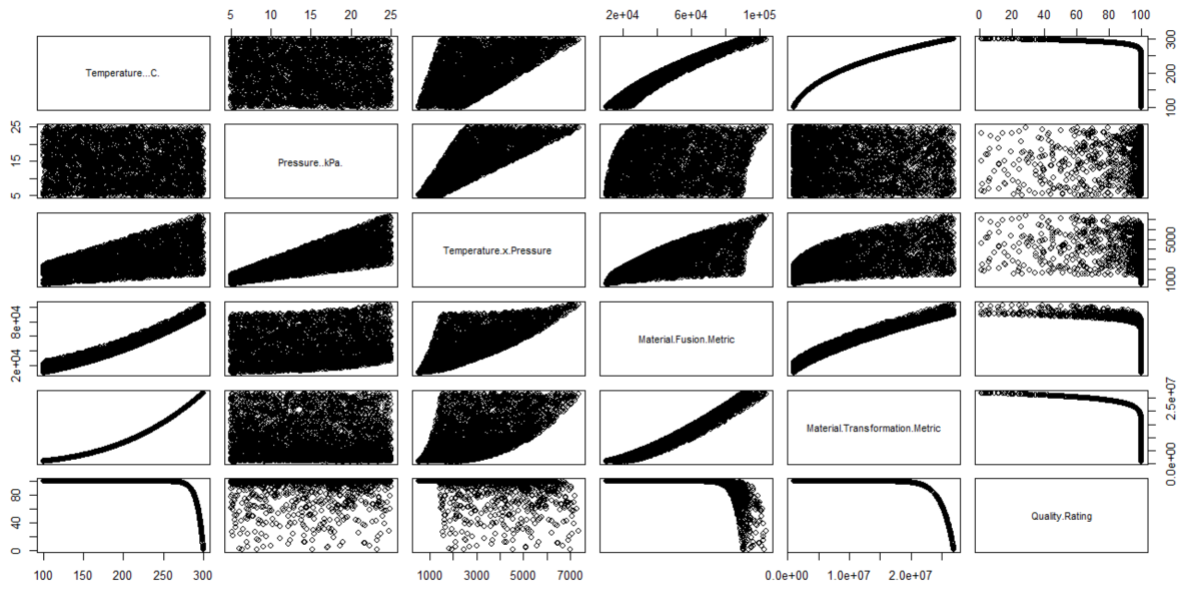
Code snippet

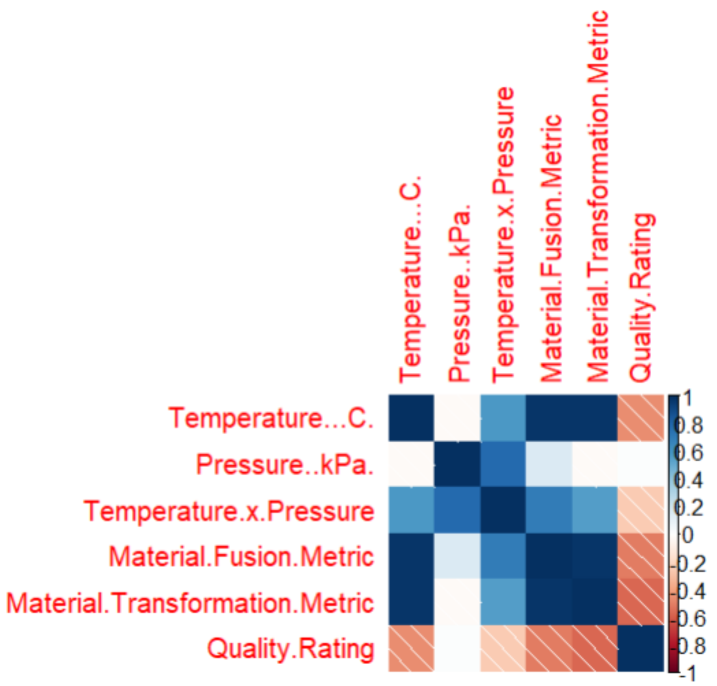
```r
# matrix plot and correlation analysis
pairs(data)
corr_mat <- cor(data)
print(corr_mat)
```

```r
install.packages("corrplot")
library(corrplot)
corrplot(corr_mat, method="shade")
```

Output

```
> # matrix plot and correlation analysis
> pairs(data)
> corr_mat <- cor(data)
> print(corr_mat)
                           Temperature...C. Pressure..kPa. Temperature.x.Pressure Material.Fusion.Metric Material.Transformation.Metric
Temperature...C.                1.00000000    -0.02475416              0.5717431              0.9749007                      0.9712102
Pressure..kPa.                 -0.02475416     1.00000000              0.7735724              0.1510952                     -0.0228617
Temperature.x.Pressure          0.57174309     0.77357240              1.0000000              0.6947331                      0.5555792
Material.Fusion.Metric          0.97490068     0.15109524              0.6947331              1.0000000                      0.9767082
Material.Transformation.Metric  0.97121016    -0.02286170              0.5555792              0.9767082                      1.0000000
Quality.Rating                 -0.46127851     0.01312935             -0.2584739             -0.5119715                     -0.5757561
                           Quality.Rating
Temperature...C.               -0.46127851
Pressure..kPa.                  0.01312935
Temperature.x.Pressure         -0.25847388
Material.Fusion.Metric         -0.51197151
Material.Transformation.Metric -0.57575605
Quality.Rating                  1.00000000
>
```





From the above plots, the factors that are correlated are:

- Temperature and material fusion metric
- Temperature and material transformation metric

Also, we can see there are factors even negatively correlated with each other (E.g. – quality rating and material transformation metric). There are some factors that are independent (E.g. – pressure and temperature).

3. Fitting a linear regression model excluding the interaction term

Code snippet

```
# Fitting the actual linear regression model
model <- lm(Quality.Rating ~ Temperature...C. + Pressure..kPa. + Material.Fusion.Metric + Material.Transformation.Metric, data = data)
summary(model)
```

Model summary

```
> # Fitting the actual linear regression model
> model <- lm(Quality.Rating ~ Temperature...C. + Pressure..kPa. + Material.Fusion.Metric + Material.Transformation.Metric, data = data)
> summary(model)

Call:
lm(formula = Quality.Rating ~ Temperature...C. + Pressure..kPa. +
    Material.Fusion.Metric + Material.Transformation.Metric,
    data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-69.416  -3.559  -0.563   4.746  14.728

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    6.978e+01  2.326e+00  29.999  < 2e-16 ***
Temperature...C.               2.522e-01  2.294e-02  10.993  < 2e-16 ***
Pressure..kPa.                -4.879e-01  8.021e-02  -6.082 1.30e-09 ***
Material.Fusion.Metric         6.905e-04  1.057e-04   6.535 7.17e-11 ***
Material.Transformation.Metric -4.980e-06  1.908e-07 -26.103  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.139 on 3952 degrees of freedom
Multiple R-squared:  0.5057,	Adjusted R-squared:  0.5052
F-statistic:  1011 on 4 and 3952 DF,  p-value: < 2.2e-16

>
```

From the values obtained from linear regression, we can see that the p values for the four parameters used in linear regression are less than 0.001. So, we can say that all four parameters are highly significant.

4. Interpreting R^2 and R^2 adjusted

Over here, we got **Multiple R-squared = 0.5057** and **Adjusted R-squared = 0.5052**

This is acceptable because the value of R-squared is above 0.3. A value of R^2 closer to 1 would have been preferred. R^2 value of 0.5057 means that approximately 50.57% of the variation in the output (Quality.Rating) can be explained by the model.

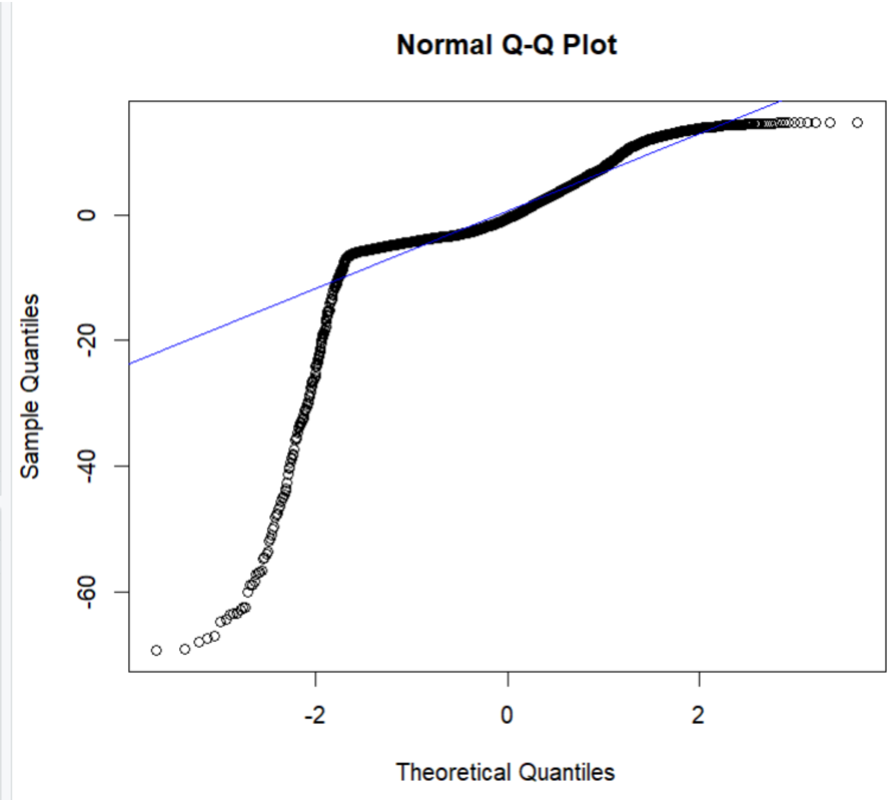
Also, the value of R^2 adjusted is very slightly less than the R^2 value. Since R^2 adjusted accounts for the number of predictors in the model, this may suggest that some predictors do not contribute significantly. However, this shouldn't be a problem since the difference is very less.

5. **Residual analysis and normal probability plot of residuals**

Code Snippet

```
# Error analysis and NPP
residuals <- model$residuals
qqnorm(residuals)
qqline(residuals, col="blue")
```

Normal Probability Plot



The basic assumption was that the error will follow normal distribution, however on performing NPP, we see that the errors (residuals) do not actually follow normal distribution.

Since our model was based on the above assumption, the model is not adequate.

6. Dividing into training and testing sets, performing linear regression then calculating the RMSE

Code snippets

Dividing into training and test sets

```
# Splitting into test and train sets
set.seed(97)
test_ind <- sample(1:nrow(data), size=0.2*nrow(data))
test_data <- data[test_ind, ]
train_data <- data[-test_ind, ]
```

Training the linear regression model

```
# Training using the train set and then predicting using the test set
train_model <- lm(Quality.Rating ~ Temperature...C. + Pressure..kPa. + Material.Fusion.Metric + Material.Transformation.Metric, data=train_data)
summary(train_model)
predictions <- predict(train_model, newdata = test_data)
summary(predictions)
```

Summary of predictions obtained

```
> # Training using the train set and then predicting using the test set
> train_model <- lm(Quality.Rating ~ Temperature...C. + Pressure..kPa. + Material.Fusion.Metric + Material.Transformation.Metric, data=train_data)
> summary(train_model)

Call:
lm(formula = Quality.Rating ~ Temperature...C. + Pressure..kPa. +
    Material.Fusion.Metric + Material.Transformation.Metric,
    data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-69.306  -3.505  -0.668   4.627  14.506

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    6.941e+01  2.517e+00  27.577  < 2e-16 ***
Temperature...C.               2.554e-01  2.488e-02  10.268  < 2e-16 ***
Pressure..kPa.                -4.289e-01  8.737e-02  -4.909 9.61e-07 ***
Material.Fusion.Metric         6.316e-04  1.153e-04   5.476 4.69e-08 ***
Material.Transformation.Metric -4.814e-06  2.096e-07 -22.963  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.948 on 3161 degrees of freedom
Multiple R-squared:  0.5077,    Adjusted R-squared:  0.5071
F-statistic: 814.9 on 4 and 3161 DF,  p-value: < 2.2e-16

> predictions <- predict(train_model, newdata = test_data)
> summary(predictions)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  69.21   92.68   98.87   95.84  102.79  106.74
> 
```

Calculating RMSE of test data

Code snippet

```
# Calculating the RMSE
true_values <- test_data$Quality.Rating
rmse <- sqrt(mean((predictions - true_values)^2))
print(paste("RMSE: ",rmse))
```

Output

```
> # Calculating the RMSE
> true_values <- test_data$Quality.Rating
> rmse <- sqrt(mean((predictions - true_values)^2))
> print(paste("RMSE: ",rmse))
[1] "RMSE:  9.87268244158578"
> 
```

Root Mean Square Error on test data = 9.87268244158578