# Predictive Modelling

# PROJECT REPORT

DSBA

**Submitted by : Renjith K P**

**Course : Post Graduate Program in Data Science and Business Analytics**

# Contents

## List of Figures

## List of Tables

## Problem 1:

### Linear Regression

The comp-activ databases is a collection of a computer systems activity measures . The data was collected from a Sun Sparcstation 20/712 with 128 Mbytes of memory running in a multi-user university department. Users would typically be doing a large variety of tasks ranging from accessing the internet, editing files or running very cpu-bound programs.

As you are a budding data scientist you thought to find out a linear equation to build a model to predict 'usr'(Portion of time (%) that cpus run in user mode) and to find out how each attribute affects the system to be in 'usr' mode using a list of system attributes.

### 1.1 Read the data and do exploratory data analysis. Describe the data briefly. (Check the Data types, shape, EDA, 5 point summary). Perform Univariate, Bivariate Analysis, Multivariate Analysis.

**Executive Summary**

Here the database have collection of information about a computer systems activity measures. Data consists of information about usage of computer for various tasks by university department users. In this problem we will understand the data in depth then build a linear regression model to predict the user mode percentage time.

**Introduction**

Purpose of this whole exercise is to build a model that can be used to predict the user mode time(percentage) of system. Data consist of information about 8192 usages of the system, all the users are from various departments in university. All the usage information's have 22 variable parameters which contains the information about the usage. This assignment will help in improving the understanding of the student in exploring summary statistics, exploratory data analysis, data cleaning, linear regression model developing, prediction of response variable using linear regression and checking the accuracy of the linear regression model.

**Data Description**

1. lread - Reads (transfers per second ) between system memory and user memory
2. lwrite - writes (transfers per second) between system memory and user memory
3. scall - Number of system calls of all types per second
4. sread - Number of system read calls per second
5. swrite - Number of system write calls per second
6. fork - Number of system fork calls per second
7. exec - Number of system exec calls per second
8. rchar - Number of characters transferred per second by system read calls

9. wchar - Number of characters transfreed per second by system write calls
10. pgout - Number of page out requests per second
11. ppgout - Number of pages, paged out per second
12. pgfree - Number of pages per second placed on the free list.
13. pgscan - Number of pages checked if they can be freed per second
14. atch - Number of page attaches (satisfying a page fault by reclaiming a page in memory) per second
15. pgin - Number of page-in requests per second
16. ppgin - Number of pages paged in per second
17. pflt - Number of page faults caused by protection errors (copy-on-writes)
18. vflt - Number of page faults caused by address translation
19. runqsz - Process run queue size (The number of kernel threads in memory that are waiting for a CPU to run. Typically, this value should be less than 2. Consistently higher values mean that the system might be CPU-bound.)
20. freemem - Number of memory pages available to user processes
21. freeswap - Number of disk blocks available for page swapping
22. usr - Portion of time (%) that cpus run in user mode

**Sample of the Dataset**

| lread | lwrite | scall | sread | swrite | fork | exec | rchar | wchar | pgout | ... | pgscan | atch | pgin | ppgin | pflt | vflt | runqsz | freemem | freeswap | usr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 2147 | 79 | 68 | 0.2 | 0.2 | 40671.0 | 53995.0 | 0.0 | ... | 0.0 | 0.0 | 1.6 | 2.6 | 16.00 | 26.40 | CPU_Bound | 4670 | 1730946 | 95 |
| 0 | 0 | 170 | 18 | 21 | 0.2 | 0.2 | 448.0 | 8385.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 15.63 | 16.83 | Not_CPU_Bound | 7278 | 1869002 | 97 |
| 15 | 3 | 2162 | 159 | 119 | 2.0 | 2.4 | NaN | 31950.0 | 0.0 | ... | 0.0 | 1.2 | 6.0 | 9.4 | 150.20 | 220.20 | Not_CPU_Bound | 702 | 1021237 | 87 |
| 0 | 0 | 160 | 12 | 16 | 0.2 | 0.2 | NaN | 8670.0 | 0.0 | ... | 0.0 | 0.0 | 0.2 | 0.2 | 15.60 | 16.80 | Not_CPU_Bound | 7248 | 1863704 | 98 |
| 5 | 1 | 330 | 39 | 38 | 0.4 | 0.4 | NaN | 12185.0 | 0.0 | ... | 0.0 | 0.0 | 1.0 | 1.2 | 37.80 | 47.60 | Not_CPU_Bound | 633 | 1760253 | 90 |

Table 1 : Dataset Sample

Data set having 22 variables for 8192 usage information's.

**Check the types of variables in the data frame.**

```
lread         int64
lwrite        int64
scall         int64
sread         int64
swrite        int64
fork        float64
exec        float64
rchar       float64
wchar       float64
pgout       float64
ppgout      float64
pgfree      float64
pgscan      float64
atch        float64
pgin        float64
ppgin       float64
pflt        float64
vflt        float64
runqsz       object
freemem       int64
freeswap      int64
usr           int64
dtype: object
```

Table 2 : Datatypes

Out of the 22 variables, 8 variables are of integer data type (int64), 13 are of float data type(float64) and one is of object data type

**Five Point Summary**

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **lread** | 8192.0 | 1.955969e+01 | 53.353799 | 0.0 | 2.0 | 7.0 | 20.000 | 1845.00 |
| **lwrite** | 8192.0 | 1.310620e+01 | 29.891726 | 0.0 | 0.0 | 1.0 | 10.000 | 575.00 |
| **scall** | 8192.0 | 2.306318e+03 | 1633.617322 | 109.0 | 1012.0 | 2051.5 | 3317.250 | 12493.00 |
| **sread** | 8192.0 | 2.104800e+02 | 198.980146 | 6.0 | 86.0 | 166.0 | 279.000 | 5318.00 |
| **swrite** | 8192.0 | 1.500582e+02 | 160.478980 | 7.0 | 63.0 | 117.0 | 185.000 | 5456.00 |
| **fork** | 8192.0 | 1.884554e+00 | 2.479493 | 0.0 | 0.4 | 0.8 | 2.200 | 20.12 |
| **exec** | 8192.0 | 2.791998e+00 | 5.212456 | 0.0 | 0.2 | 1.2 | 2.800 | 59.56 |
| **rchar** | 8088.0 | 1.973857e+05 | 239837.493526 | 278.0 | 34091.5 | 125473.5 | 267828.750 | 2526649.00 |
| **wchar** | 8177.0 | 9.590299e+04 | 140841.707911 | 1498.0 | 22916.0 | 46619.0 | 106101.000 | 1801623.00 |
| **pgout** | 8192.0 | 2.285317e+00 | 5.307038 | 0.0 | 0.0 | 0.0 | 2.400 | 81.44 |
| **ppgout** | 8192.0 | 5.977229e+00 | 15.214590 | 0.0 | 0.0 | 0.0 | 4.200 | 184.20 |
| **pgfree** | 8192.0 | 1.191971e+01 | 32.363520 | 0.0 | 0.0 | 0.0 | 5.000 | 523.00 |
| **pgscan** | 8192.0 | 2.152685e+01 | 71.141340 | 0.0 | 0.0 | 0.0 | 0.000 | 1237.00 |
| **atch** | 8192.0 | 1.127505e+00 | 5.708347 | 0.0 | 0.0 | 0.0 | 0.600 | 211.58 |
| **pgin** | 8192.0 | 8.277960e+00 | 13.874978 | 0.0 | 0.6 | 2.8 | 9.765 | 141.20 |
| **ppgin** | 8192.0 | 1.238859e+01 | 22.281318 | 0.0 | 0.6 | 3.8 | 13.800 | 292.61 |
| **pflt** | 8192.0 | 1.097938e+02 | 114.419221 | 0.0 | 25.0 | 63.8 | 159.600 | 899.80 |
| **vflt** | 8192.0 | 1.853158e+02 | 191.000603 | 0.2 | 45.4 | 120.4 | 251.800 | 1365.00 |
| **freemem** | 8192.0 | 1.763456e+03 | 2482.104511 | 55.0 | 231.0 | 579.0 | 2002.250 | 12027.00 |
| **freeswap** | 8192.0 | 1.328126e+06 | 422019.426957 | 2.0 | 1042623.5 | 1289289.5 | 1730379.500 | 2243187.00 |
| **usr** | 8192.0 | 8.396887e+01 | 18.401905 | 0.0 | 81.0 | 89.0 | 94.000 | 99.00 |

Table 3 : Five Point Summary

From 5 point summary, it is visible that many variables have values as 0. Some variables are having more than 50% values as 0. These entries must be treated.

**Check for any null values or duplicates**

```
lread         0
lwrite        0
scall         0
sread         0
swrite        0
fork          0
exec          0
rchar       104
wchar        15
pgout         0
ppgout        0
pgfree        0
pgscan        0
atch          0
pgin          0
ppgin         0
pflt          0
vflt          0
runqsz        0
freemem       0
freeswap      0
usr           0
dtype: int64
```

Table 4 : Null values

Null values are present in rchar and wchar.

There are no duplicated values in this dataset.

**Exploratory Data Analysis**

**Univariate Analysis**

For the continuous variable histograms were plotted to understand the distribution. Also, boxplots were studied to understand more about the data and about outliers present in it.

From the box plot, it is clear that all the variables are having outliers in them. When we check the distribution except for 'freeswap' and 'usr', all other continuous variables are right skewed which means the mean is greater than the median for all these. 'usr' is left-skewed, and here mean is less than the median. There is no skewness observed in 'freeswap'.

For the categorical variables, bar plot is plotted.

- When the process run queue size is CPU bound user mode time is less compared to the process run queue size is not CPU bound
- Heatmap and pair plot shows there are multicollinearity present in the data

Plots were shown below.

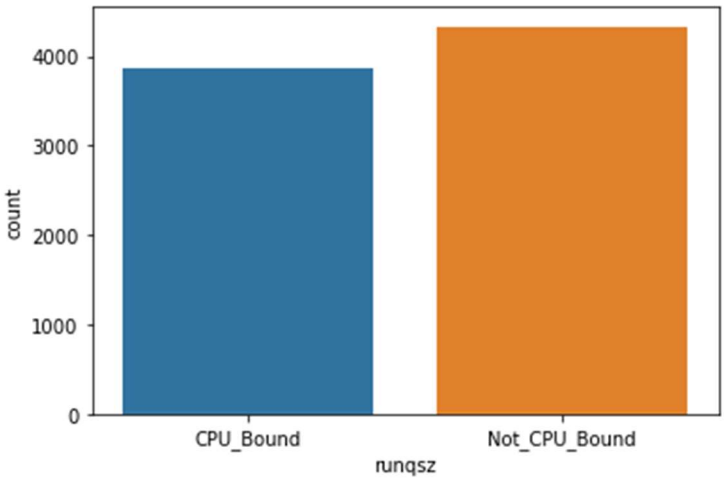


Figure 1 : runqsz count plot

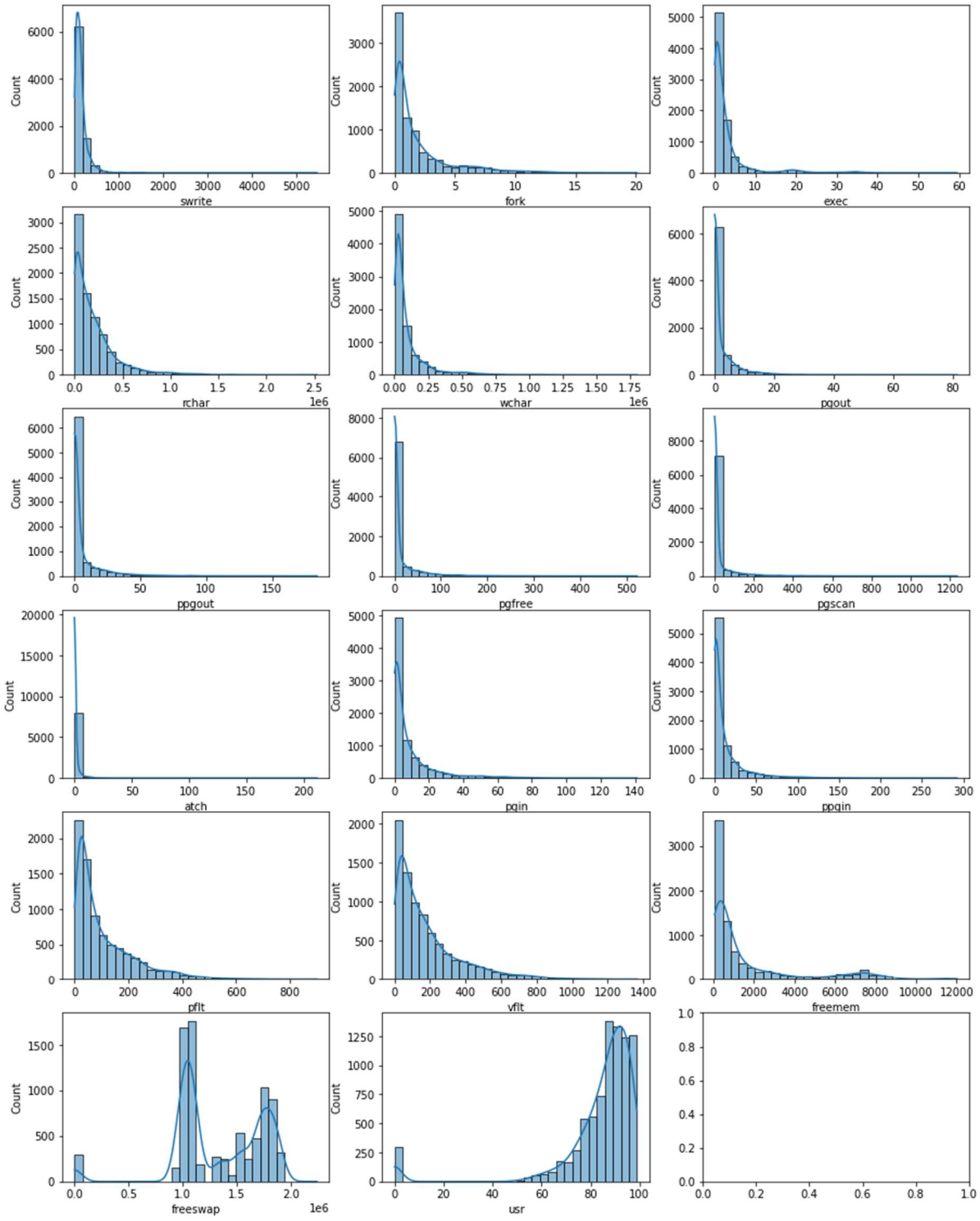


Figure 2 : Histograms for continuous variables

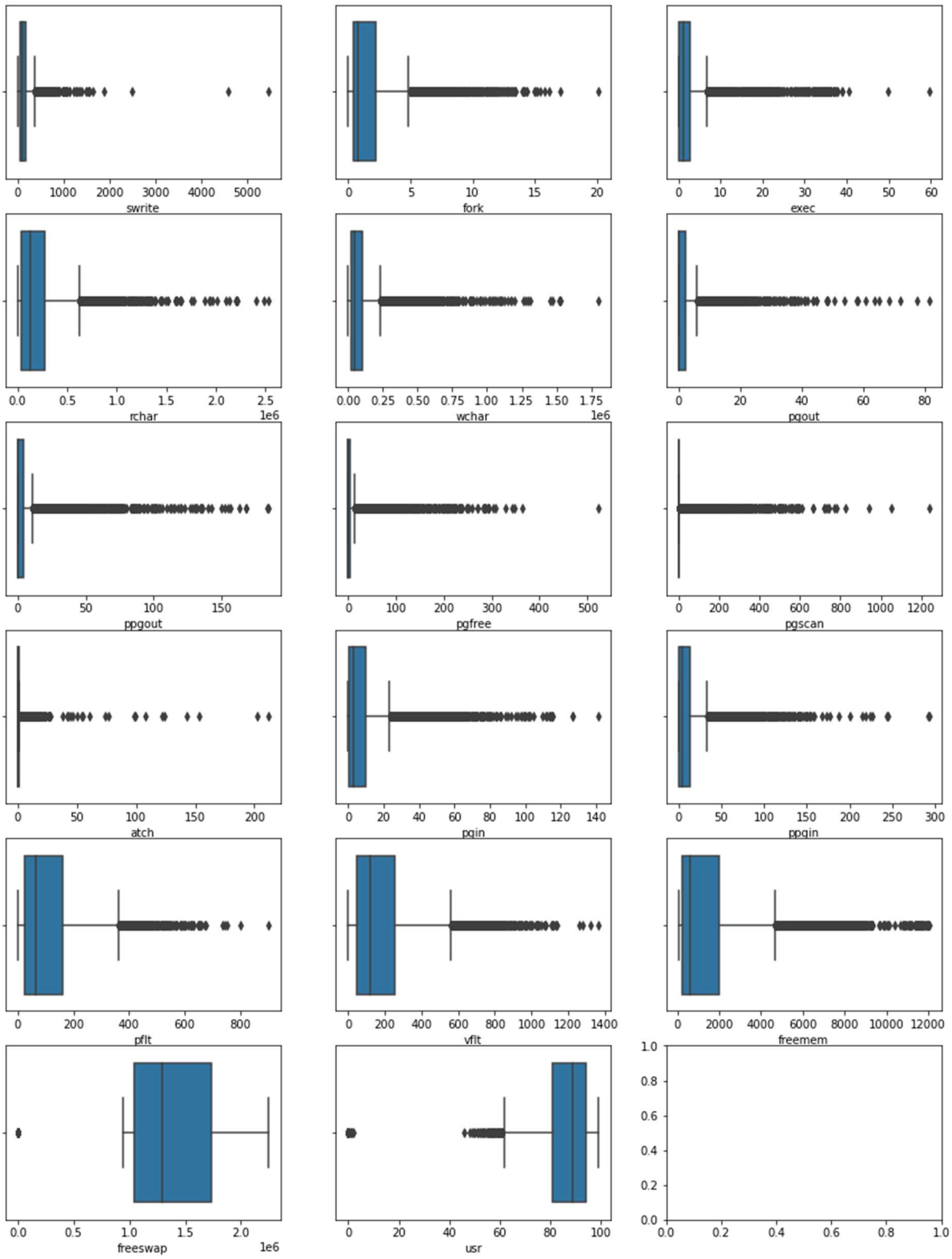


**Figure 3 : Boxplots for continuous variables**

## Bivariate & Multivariate Analysis

## Heatmap

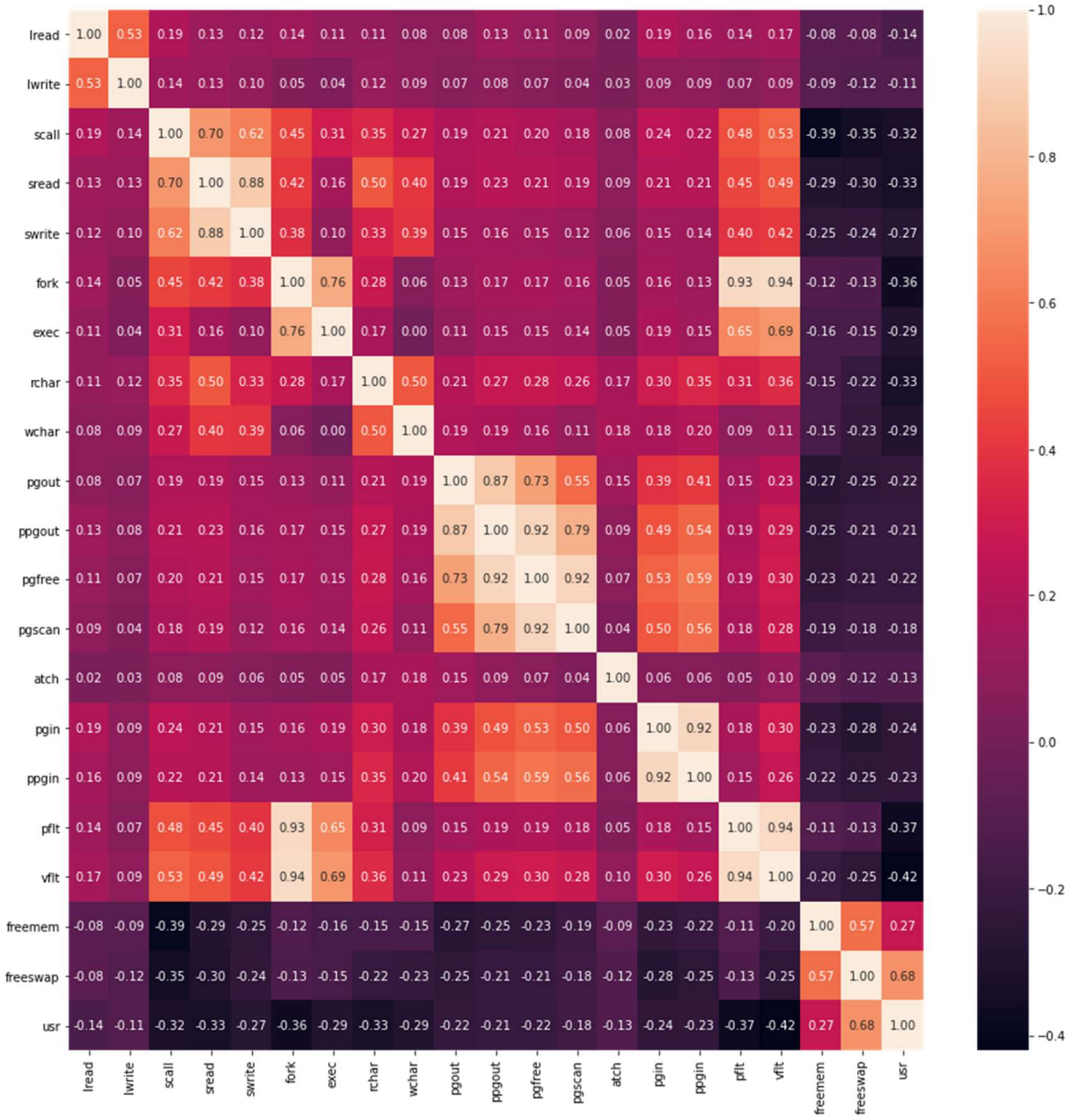


**Figure 4 : Heatmap**

## Pair plot

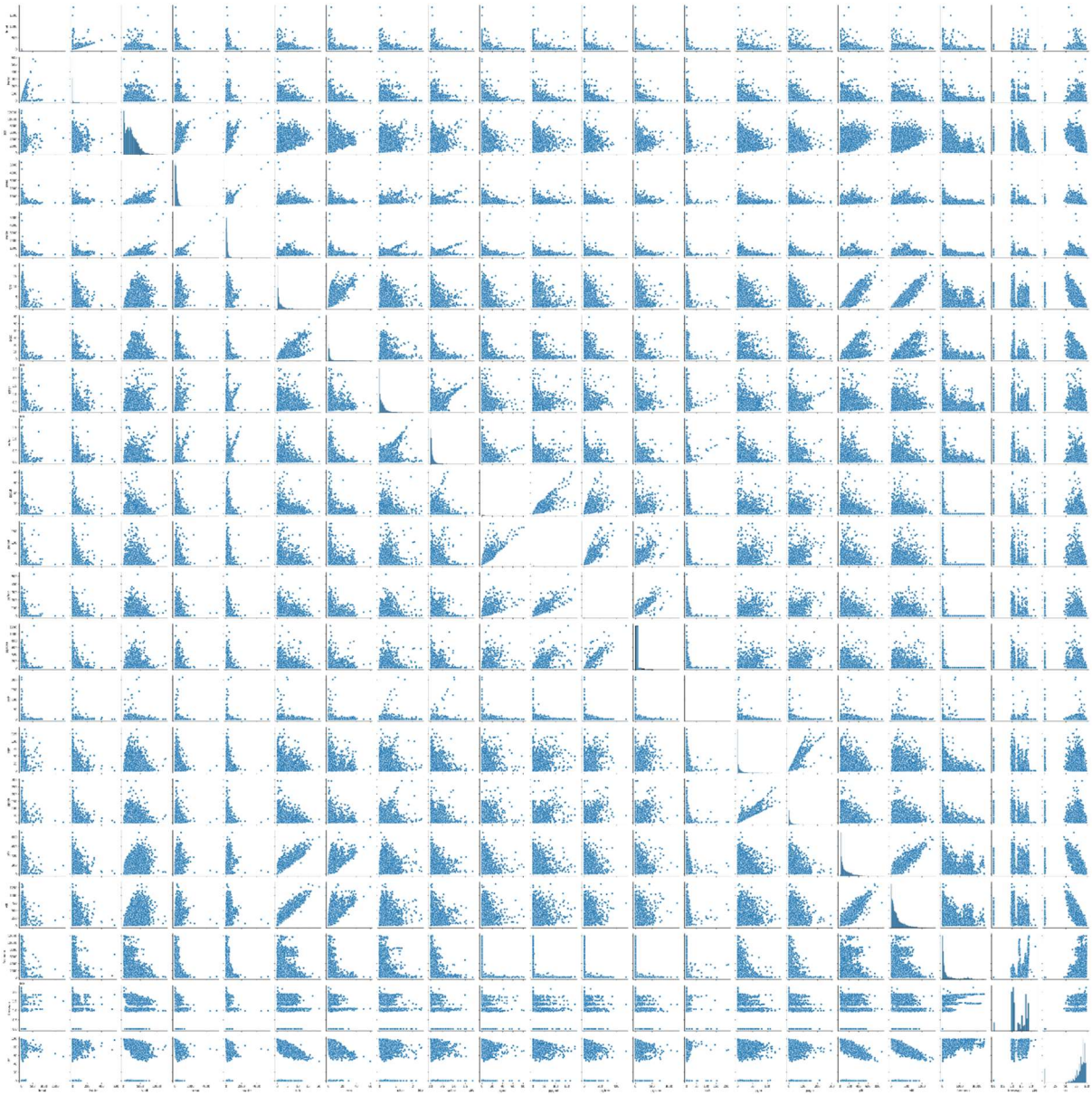

**Figure 5 : Pair plot**

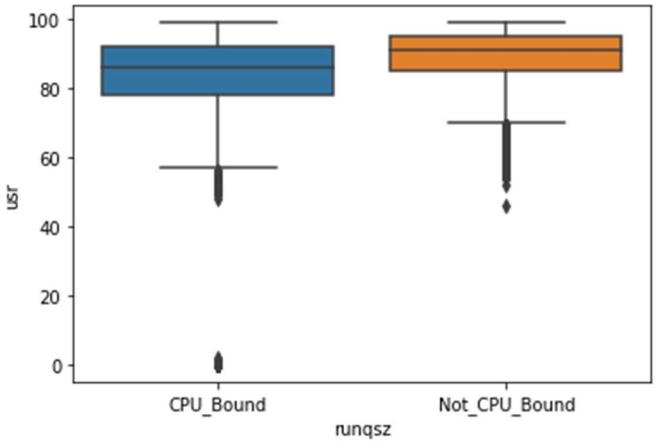


**Figure 6 : Boxplot of runqsz vs user time**

**1.2 Impute null values if present, also check for the values which are equal to zero. Do they have any meaning or do we need to change them or drop them? Check for the possibility of creating new features if required. Also check for outliers and duplicates if there.**

```
lread         0
lwrite        0
scall         0
sread         0
swrite        0
fork          0
exec          0
rchar       104
wchar        15
pgout         0
ppgout        0
pgfree        0
pgscan        0
atch          0
pgin          0
ppgin         0
pflt          0
vflt          0
runqsz        0
freemem       0
freeswap      0
usr           0
dtype: int64
```

**Figure 7 : Null value check**

Null values are present in rchar and wchar.

rchar and wchar represent the number of characters transferred per second, also the distribution of both were right skewed. When the distribution is right skewed median will be the better option to impute null values.

Null values replaced with median values

```
lread       0
lwrite      0
scall       0
sread       0
swrite      0
fork        0
exec        0
rchar       0
wchar       0
pgout       0
ppgout      0
pgfree      0
pgscan      0
atch        0
pgin        0
ppgin       0
pflt        0
vflt        0
runqsz      0
freemem     0
freeswap    0
usr         0
dtype: int64
```

**Figure 8 : Null value check after filling null values**

Below are the values with zero

lread – 675

lwrite – 2684

fork – 21

excec-21

pgout – 4878

ppgout - 4878

pgfree – 4869

pgscan – 6448

atch – 4575

pgin – 1220

ppgin – 1220

pflt – 3

usr – 283

'lread' and 'lwrite' represents the character transfer per second, When a system is being used there will be character transfer. Entries as zero might be a mistake. Lets replace zeroes by median values. Number of system fork and exec calls with values as 0 are same and these entries might be true. We are not replacing these. 'pgout','pgfree' ,'pgscan' and 'atch' is having more than 50% of values zero hence better we drop these variables, without droping this may affect our model. Lets confirm the impact of these variables when we build our model. For this we will try one model without droping these entries and considering these values as true entries. 'pgin' & 'ppgin' is having same number of 0 entries and it might be true entries . Pflt is having only 3 entries as 0, we are not doing anything here also. Usr is having 283 entries a 0, since usr is our response variable better we drop these entries as replacing them may alter our model. To check multiple model performance measures here we will create a dataset without droping 'pgout','pgfree' ,'pgscan' and 'atch' .

We checked whether any duplicated entries are there and found that there is no any.

When checked the outliers, found that many of the variables having outliers.

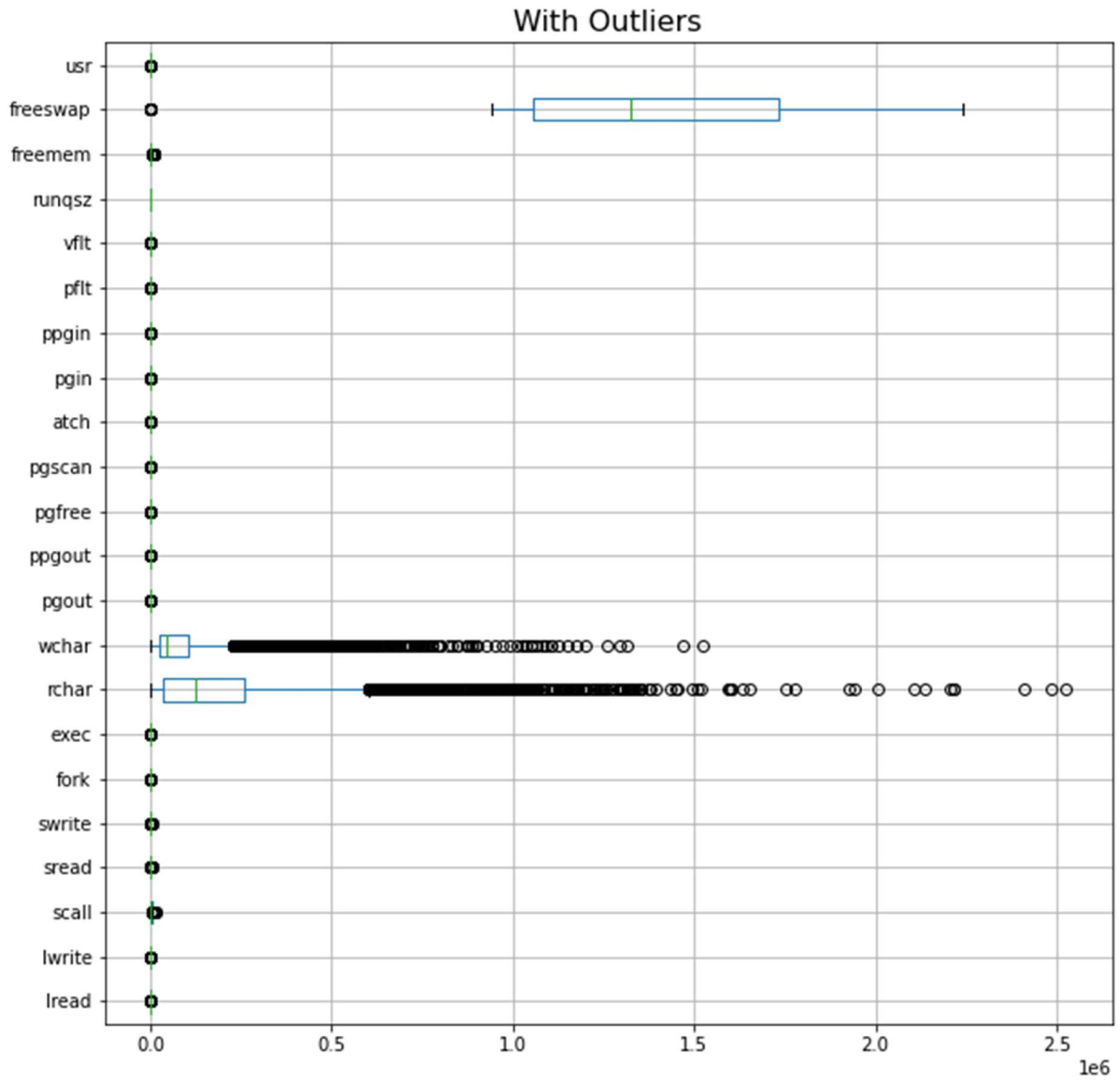


**Figure 9 : Boxplot for outlier check**

Here we are creating two datasets to understand the effect of outliers in linear regression model performance going forward. One dataset after treating outliers and the other without treating outliers.

After treating outliers, the same is checked and there were no any outliers present.

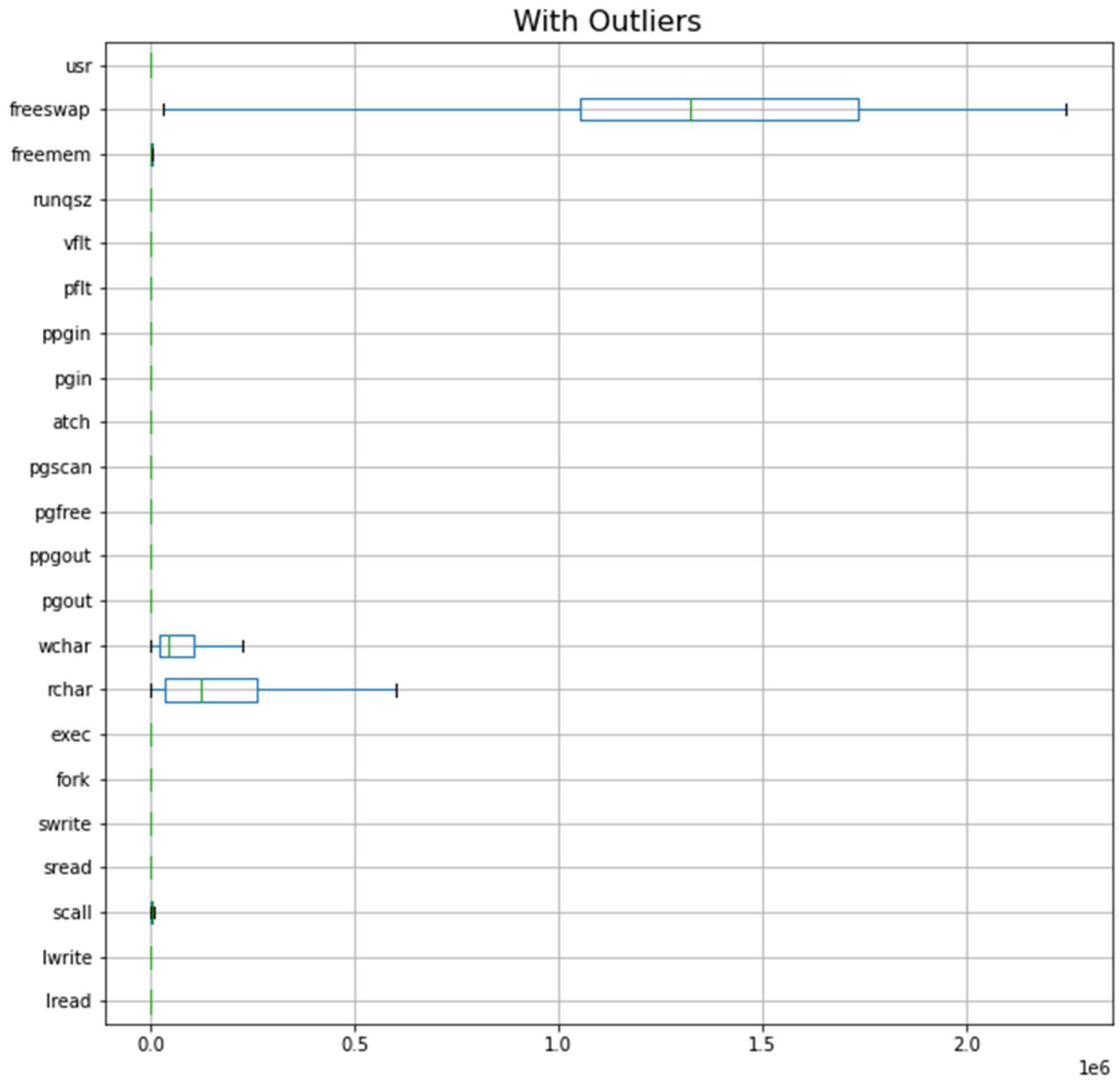


**Figure 10 : Boxplot after outlier treatment**

### 1.3 Encode the data (having string values) for Modelling. Split the data into train and test (70:30). Apply Linear regression using scikit learn. Perform checks for significant variables using appropriate method from statsmodel. Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare. Compare these models and select the best one with appropriate reasoning.

Data with string values encoded.

We made two models using scikit learn and statsmodel with and without droping the variables 'pgout','pgfree' ,'pgscan' and 'atch', then we compared model performance measures between these .

| Model No | Scikit Learn/Statsmodels | R2 Train Data | R2 Test Data | Adjusted R2 Train Data | Adjusted R2 Test Data | RMSE Train Data | RMSE Test Data |
|---|---|---|---|---|---|---|---|
| Model 1 | Scikit Learn | 0.805 | 0.856 | | | 4.218 | 3.508 |
| Model 2 | Statsmodel | 0.805 | 0.861 | 0.804 | 0.86 | 4.218 | 3.508 |
| Model 3 | Scikit Learn | 0.806 | 0.857 | | | 4.2 | 3.488 |
| Model 4 | Statsmodel | 0.806 | 0.863 | 0.806 | 0.862 | 4.2 | 3.488 |

Table 5 : Model comparison

From the comparison we could clearly see that adding these four variables increased R2 by a very minor value, also there was no significant impact on RMSE. Hence we could proceed further to make models with dropping these variables.

Regression model created from dataset created by treating outliers from the original dataset and dropping 'pgout','pgfree' ,'pgscan' and 'atch'. Performance measures compared against other models created so far.

| Model No | Scikit Learn/Statsmodels | R2 Train Data | R2 Test Data | Adjusted R2 Train Data | Adjusted R2 Test Data | RMSE Train Data | RMSE Test Data |
|---|---|---|---|---|---|---|---|
| Model 1 | Scikit Learn | 0.805 | 0.856 | | | 4.218 | 3.508 |
| Model 2 | Statsmodel | 0.805 | 0.861 | 0.804 | 0.86 | 4.218 | 3.508 |
| Model 3 | Scikit Learn | 0.806 | 0.857 | | | 4.2 | 3.488 |
| Model 4 | Statsmodel | 0.806 | 0.863 | 0.806 | 0.862 | 4.2 | 3.488 |
| Model 5 | Scikit Learn | 0.885 | 0.88 | | | 2.825 | 2.872 |
| Model 6 | Statsmodel | 0.885 | 0.882 | 0.885 | 0.881 | 2.825 | 2.872 |

Table 6 : Model comparison

We could observe that R2 has improved significantly here, also RMSE reduced by a big margin.

Now lets check the impact of variables using statsmodel

Out[816]: OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | usr | R-squared: | 0.885 |
| Model: | OLS | Adj. R-squared: | 0.885 |
| Method: | Least Squares | F-statistic: | 2666. |
| Date: | Sat, 07 Jan 2023 | Prob (F-statistic): | 0.00 |
| Time: | 16:29:34 | Log-Likelihood: | -13606. |
| No. Observations: | 5536 | AIC: | 2.725e+04 |
| Df Residuals: | 5519 | BIC: | 2.736e+04 |
| Df Model: | 16 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 99.2688 | 0.272 | 364.463 | 0.000 | 98.735 | 99.803 |
| lread | -0.0539 | 0.006 | -9.148 | 0.000 | -0.065 | -0.042 |
| lwrite | 0.0396 | 0.009 | 4.586 | 0.000 | 0.023 | 0.057 |
| scall | -0.0013 | 4.21e-05 | -31.312 | 0.000 | -0.001 | -0.001 |
| sread | -0.0002 | 0.001 | -0.277 | 0.782 | -0.002 | 0.001 |
| swrite | -0.0054 | 0.001 | -5.602 | 0.000 | -0.007 | -0.003 |
| fork | -0.1279 | 0.082 | -1.567 | 0.117 | -0.288 | 0.032 |
| exec | -0.3606 | 0.033 | -10.985 | 0.000 | -0.425 | -0.296 |
| rchar | -3.019e-06 | 3.21e-07 | -9.405 | 0.000 | -3.65e-06 | -2.39e-06 |
| wchar | -6.919e-06 | 6.9e-07 | -10.024 | 0.000 | -8.27e-06 | -5.57e-06 |
| pgin | -0.0449 | 0.019 | -2.378 | 0.017 | -0.082 | -0.008 |
| ppgin | -0.0704 | 0.013 | -5.458 | 0.000 | -0.096 | -0.045 |
| pflt | -0.0200 | 0.001 | -15.783 | 0.000 | -0.022 | -0.017 |
| vflt | -0.0124 | 0.001 | -13.314 | 0.000 | -0.014 | -0.011 |
| runqsz | -0.4957 | 0.084 | -5.914 | 0.000 | -0.660 | -0.331 |
| freemem | 0.0003 | 3.07e-05 | 11.205 | 0.000 | 0.000 | 0.000 |
| freeswap | -6.07e-07 | 1.66e-07 | -3.647 | 0.000 | -9.33e-07 | -2.81e-07 |

| | | | |
|---|---|---|---|
| Omnibus: | 1255.024 | Durbin-Watson: | 2.028 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 4761.149 |
| Skew: | -1.088 | Prob(JB): | 0.00 |
| Kurtosis: | 6.989 | Cond. No. | 1.03e+07 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.03e+07. This might indicate that there are strong multicollinearity or other numerical problems.

**Figure 11 : Model Summary**

From the above summary of the model, we could see that some variables are having very high p values ('sread','fork') which indicate that these variables doesn't have significant contributions to the model. Also many variables' coefficient is very low.

We could write the linear model as below

usr = (99.27) * const + (-0.05) * lread + (0.04) * lwrite + (-0.0) * scall + (-0.0) * sread + (-0.01) * swrite + (-0.13) * fork + (-0.36) * exec + (-0.0) * rchar + (-0.0) * wchar + (-0.04) * pgin + (-0.07) * ppgin + (-0.02) * pflt + (-0.01) * vflt + (-0.5) * runqsz + (0.0) * freemem + (-0.0) * freeswap +

Before dropping non significant variables lets check the VIF

```
VIF values:

const       51.282729
lread        4.648116
lwrite       3.878619
scall        3.124708
sread        6.925564
swrite       6.131479
fork        13.668159
exec         3.121266
rchar        1.996154
wchar        1.553213
pgin        14.286403
ppgin       14.114773
pflt        11.643730
vflt        16.096923
runqsz       1.201713
freemem      1.863069
freeswap     2.290922
dtype: float64
```

Table 6 : VIF

Some variables have a very high VIF value, which means multicollinearity is present in the data. Lets check how dropping the variable with the highest VIF factor affects R2

After dropping 'vflt' R2 reduced to 0.882 and adj R2 reduced to 881, which is not a significant reduction. Let's drop 'vflt' comfortably since it does not contribute much to the model. After dropping 'vflt', again VIF checked and we could still see some variables with high VIF.

Same exercise repeated until all the variables are having VIF factor less than 5

```
VIF values:

const       49.089784
lread        4.560463
lwrite       3.826458
scall        2.790098
swrite       3.027834
exec         2.752947
rchar        1.588142
wchar        1.506386
pgin         1.453488
pflt         3.356728
runqsz       1.199388
freemem      1.857915
freeswap     2.172315
dtype: float64
```

Table 7 : VIF

Now let's create our next model after dropping variables with higher VIF .

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | usr | **R-squared:** | 0.880 |
| **Model:** | OLS | **Adj. R-squared:** | 0.879 |
| **Method:** | Least Squares | **F-statistic:** | 3360. |
| **Date:** | Sat, 07 Jan 2023 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 17:33:56 | **Log-Likelihood:** | -13746. |
| **No. Observations:** | 5536 | **AIC:** | 2.752e+04 |
| **Df Residuals:** | 5523 | **BIC:** | 2.760e+04 |
| **Df Model:** | 12 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | 98.6108 | 0.271 | 363.442 | 0.000 | 98.079 | 99.143 |
| **lread** | -0.0629 | 0.006 | -10.480 | 0.000 | -0.075 | -0.051 |
| **lwrite** | 0.0494 | 0.009 | 5.591 | 0.000 | 0.032 | 0.067 |
| **scall** | -0.0013 | 4.08e-05 | -31.907 | 0.000 | -0.001 | -0.001 |
| **swrite** | -0.0079 | 0.001 | -11.360 | 0.000 | -0.009 | -0.007 |
| **exec** | -0.4882 | 0.032 | -15.464 | 0.000 | -0.550 | -0.426 |
| **rchar** | -3.535e-06 | 2.96e-07 | -11.953 | 0.000 | -4.11e-06 | -2.95e-06 |
| **wchar** | -5.419e-06 | 6.97e-07 | -7.771 | 0.000 | -6.79e-06 | -4.05e-06 |
| **ppgin** | -0.1169 | 0.004 | -27.558 | 0.000 | -0.125 | -0.109 |
| **pflt** | -0.0364 | 0.001 | -52.204 | 0.000 | -0.038 | -0.035 |
| **runqsz** | -0.4745 | 0.086 | -5.527 | 0.000 | -0.643 | -0.306 |
| **freemem** | 0.0003 | 3.15e-05 | 10.316 | 0.000 | 0.000 | 0.000 |
| **freeswap** | -8.584e-08 | 1.65e-07 | -0.520 | 0.603 | -4.1e-07 | 2.38e-07 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 1005.118 | **Durbin-Watson:** | 2.043 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 3357.608 |
| **Skew:** | -0.907 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 6.356 | **Cond. No.** | 1.00e+07 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1e+07. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 12 : Model Summary

In the new model, VIF looks good (<5) for all variables.

```
VIF values:

const       48.415054
lread        4.589144
lwrite       3.836408
scall        2.789850
swrite       3.027731
exec         2.746950
rchar        1.611105
wchar        1.508252
ppgin        1.450030
pflt         3.360200
runqsz       1.199626
freemem      1.859217
freeswap     2.146683
dtype: float64
```

Table 8 : VIF

But in this model 'freeswap' is having a very high p value, we can drop 'freeswap' since it is not significant in predicting user time. We can create the next model after dropping this.

The final model can be written as

usr = (98.48) * const + (-0.06) * lread + (0.05) * lwrite + (-0.0) * scall + (-0.01) * swrite + (-0.49) * exec + (-0.0) * rchar + (-0.0) * wchar + (-0.12) * ppgin + (-0.04) * pflt + (-0.46) * runqsz + (0.0) * freemem

Final model is having following performance measures.

| Model No | Scikit Learn/Statsmodels | R2 Train Data | R2 Test Data | Adjusted R2 Train Data | Adjusted R2 Test Data | RMSE Train Data | RMSE Test Data |
|---|---|---|---|---|---|---|---|
| Model 1 | Scikit Learn | 0.805 | 0.856 | | | 4.218 | 3.508 |
| Model 2 | Statsmodel | 0.805 | 0.861 | 0.804 | 0.86 | 4.218 | 3.508 |
| Model 3 | Scikit Learn | 0.806 | 0.857 | | | 4.2 | 3.488 |
| Model 4 | Statsmodel | 0.806 | 0.863 | 0.806 | 0.862 | 4.2 | 3.488 |
| Model 5 | Scikit Learn | 0.885 | 0.88 | | | 2.825 | 2.872 |
| Model 6 | Statsmodel | 0.885 | 0.882 | 0.885 | 0.881 | 2.825 | 2.872 |
| Model 7 | Scikit Learn | 0.879 | 0.873 | | | 2.9 | 2.952 |
| Model 8 | Statsmodel | 0.88 | | 0.879 | | | |
| Model 9 | Statsmodel | 0.88 | 0.874 | 0.879 | 0.874 | 2.9 | 2.953 |
| Model 10 | Scikit Learn | 0.88 | 0.873 | | | 2.9 | 2.953 |

Table 9 : Model Comparison

Now let's create one more model after scaling the data also to understand the effect of scaling. After removing high VIF variables, the scaled model is having same R2 value of model 10. Hence we could finalize model 10 as the best model.

Our final model is the one built after dropping variables having more than 50% data as 0, then outlier treated, followed by VIF check and multicollinearity is removed by removing variables having high VIF.

Now lets check the assumptions of linear regression for our final model.

1. Linearity
2. Independence
3. Homocedacity
4. Normality of Error Terms
5. No strong multi collinearity

To check Linearity and Independence, residuals plotted. There are no patterns present. Lets conclude the assumptions of Linearity and Independence are meeting in our model

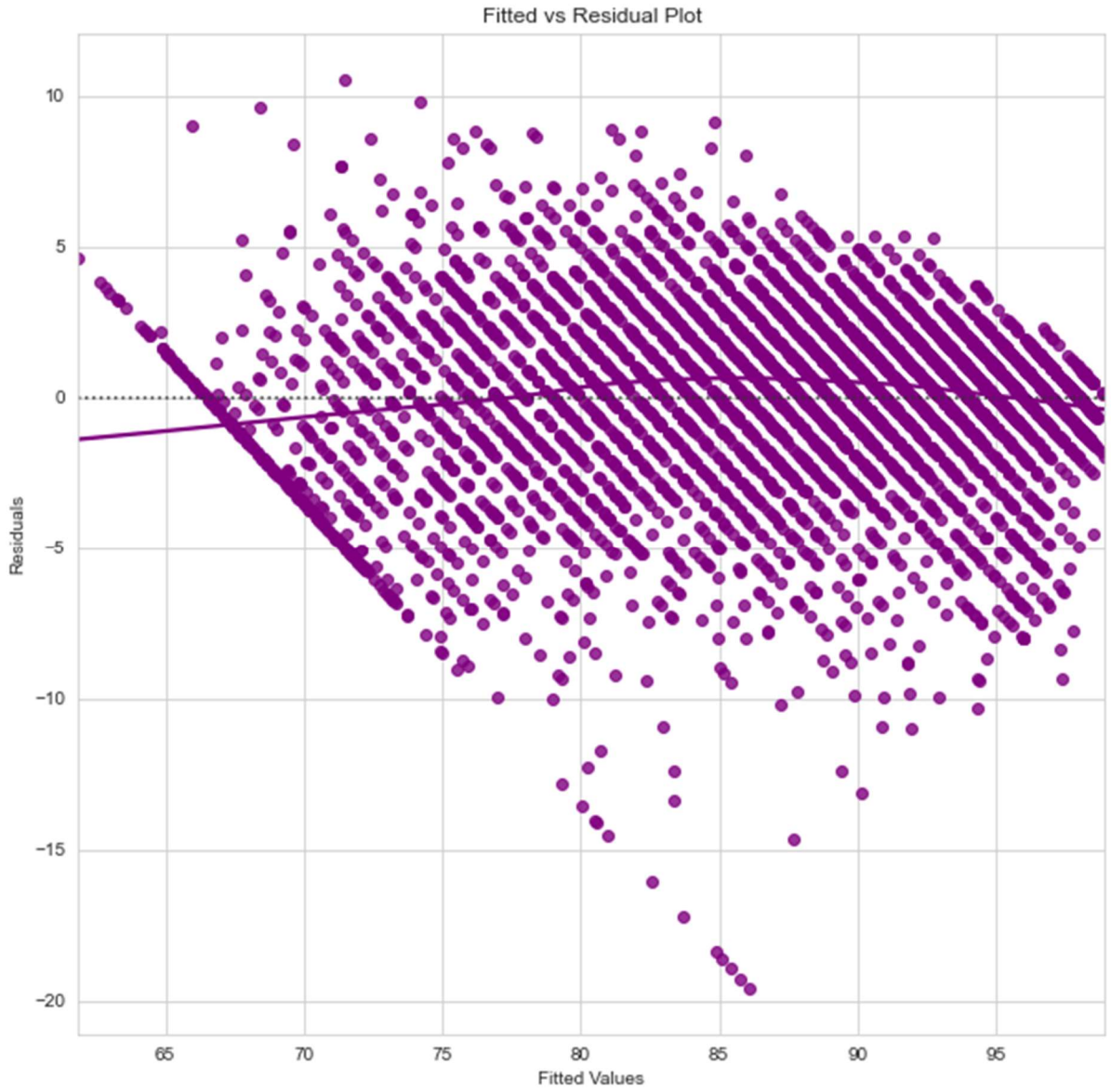


Figure 13 : Fitted vs Residual Plot

Let’s check the normality of the residuals

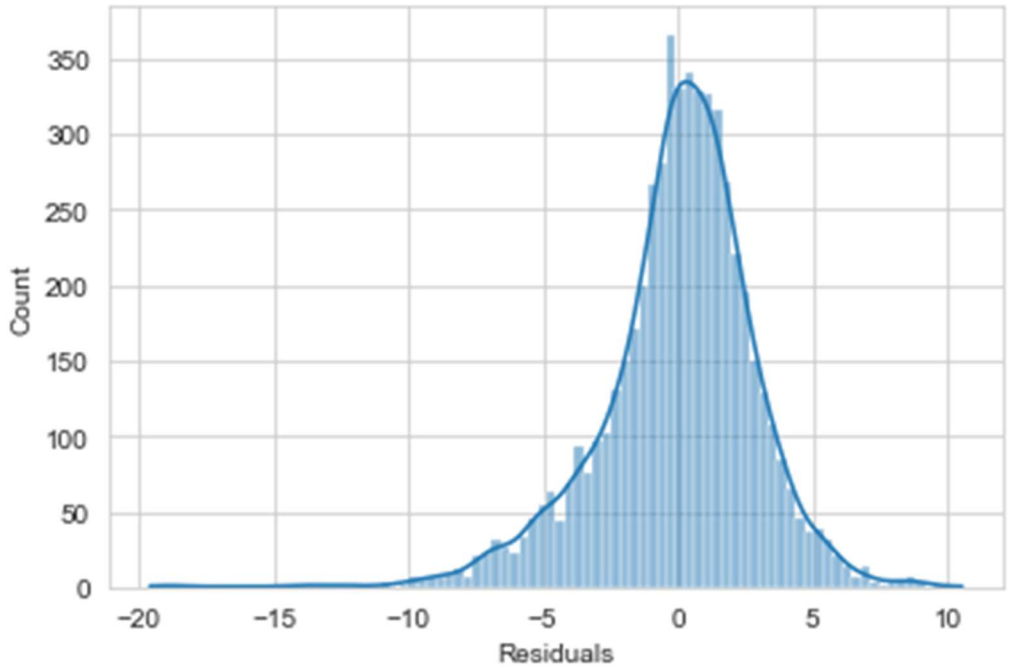


Figure 15 : Histogram Residuals

Residuals are normally distributed

Also Shapiro test was conducted to confirm normality.

Also tested for homocedacity and Since p value > 0.05 , we could say residuals are homoscedastic

No strong multicollinearity present in the data, confirmed the same with pair plot.

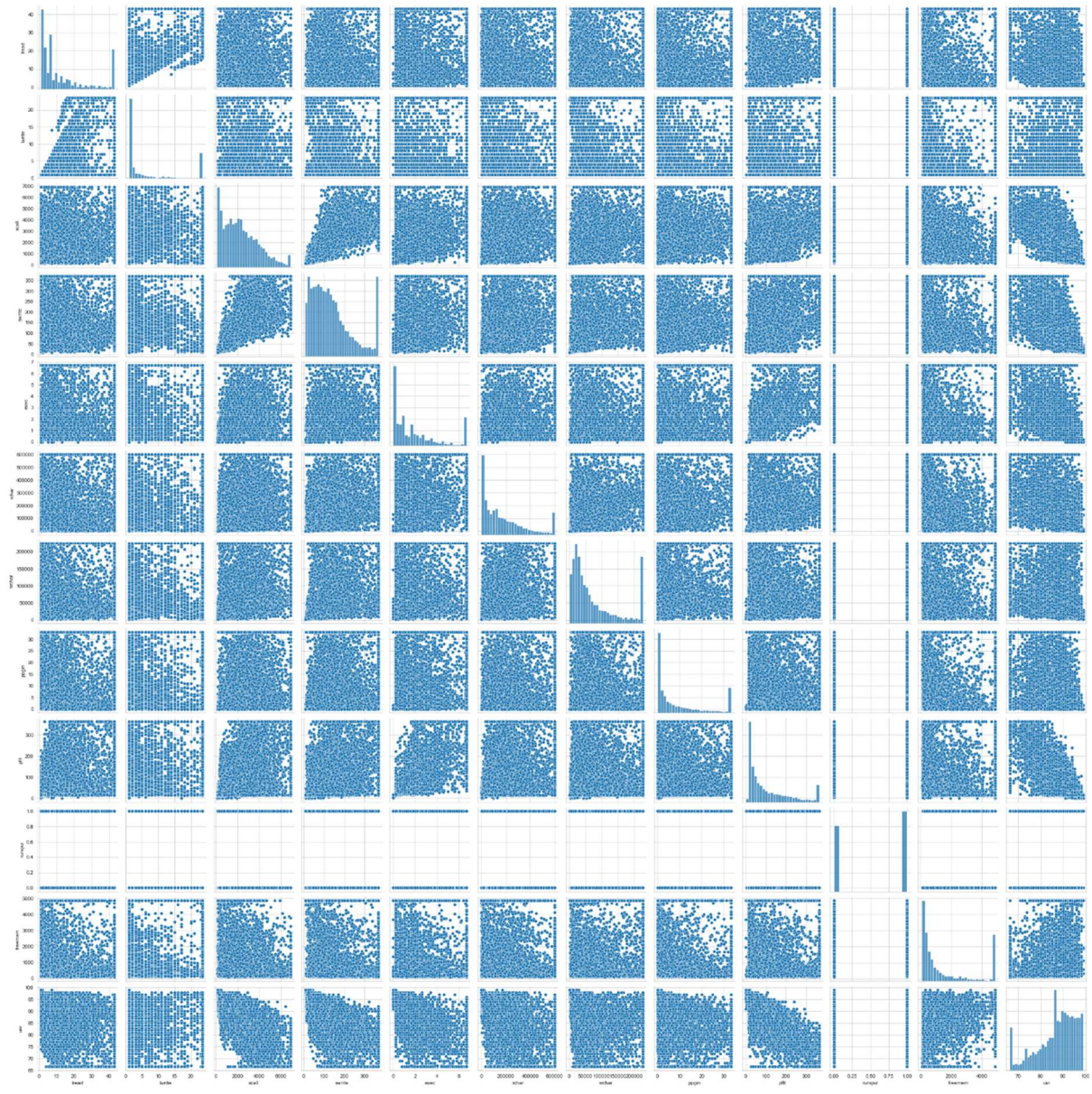

Figure 16 : Pairplot

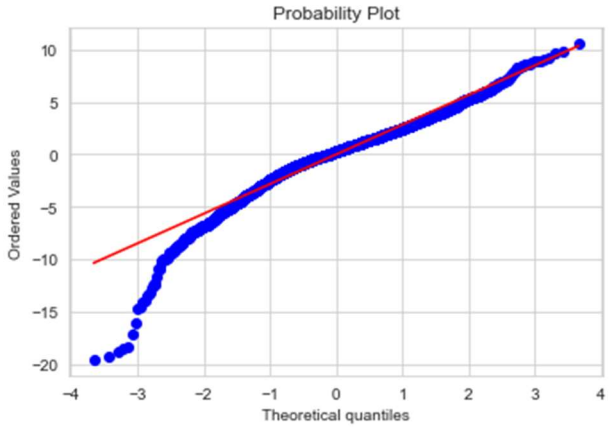


Figure 17 : Probability Plot

### 1.4 Inference: Basis on these predictions, what are the business insights and recommendations.

We started with understanding the data given first. It is very important to understand the data before analyzing and creating models with it. We checked whether the data loaded correctly by checking the first and last 5 rows. Later performed EDA to check the data types of variables, 5 point summary of the data, null values and duplicates present in the data. Also, we checked the info about the data. After checking these clearly understood how to clean the data to make a foundation for our model. Based on the understanding from the above checks decided how to treat the null values and anomalies present in the data. Then we visualised the data, for all continuous variables histograms and boxplots were plotted, From these plots, we got a clear idea about the distribution of variables and the presence of outliers. For categorical variables count plot is checked and understood counts of different entries. Then correlation plot and pair plot were checked to understand the collinearity and relationships. There is multicollinearity present in this data and noted to address this later on the process of building the model.

- Not_CPU_Bound is higher than CPU_Bound when we check the whole data which indicates that system is in user mode for more time compared to CPU mode

During cleaning the data we dropped some variables, which is having more than 50% of values as 0, and we confirmed these variables doesn't have any significant impact on linear regression model. We have treated the outliers, null values and some entries having 0 in the cleaning process. Then Different models were created (eg: with and without outliers, after scaling, etc..) and we compared the performance. Since we observed multicollinearity in the data checked the VIF factor of variables and removed variables with higher VIF factors and very minimum contribution to R2 and RMSE. Finally, after comparing many models we concluded the following model.

usr = (98.48) * const + (-0.06) * lread + (0.05) * lwrite + (-0.0) * scall + (-0.01) * swrite + (-0.49) * exec + (-0.0) * rchar + (-0.0) * wchar + (-0.12) * ppgin + (-0.04) * pflt + (-0.46) * runqsz + (0.0) * freemem

- When 'swrite' increased by 1 unit, user time percentage decreases by 0.49 units.
- When pflt increased by 1 unit, user time percentage decreases by 0.46 units.
- When lread increases by 1 unit,user time percentage increases by 0.05 units.

After building the final model, all the assumptions of linear regression is checked and found that model is meeting all assumptions.

Some more Insights

- When the system write calls per second increases user time percentage reduction observed
- It is observed that when page faults increase user time percentage is reducing

- User time percentage is reducing when more characters were transferred by system write calls
- When reads between system memory and user memory increase, user percentage ti me also increases
- Not_CPU_Bound is higher than CPU_Bound when we check the whole data which indicates that the system is in user mode for more time compared to CPU mode

## Problem 2:

## Logistic Regression, LDA and CART

You are a statistician at the Republic of Indonesia Ministry of Health and you are provided with a data of 1473 females collected from a Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of the survey.

The problem is to predict do/don't they use a contraceptive method of choice based on their demographic and socio-economic characteristics.

### 2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, check for duplicates and outliers and write an inference on it. Perform Univariate and Bivariate Analysis and Multivariate Analysis.Data Description

**Executive Summary**

The dataset has information about 473 females collected from a Contraceptive Prevalence Survey. In this assignment will understand the data in depth and then build logistic regression and CART models to predict do/don't they use a contraceptive method of choice based on their demographic and socio-economic characteristics.

**Introduction**

The purpose of this whole exercise is to build a model that predicts whether couples should use a contraceptive method or not based on their demographic and socio-economic characteristics. We will build logical regression, LDA, and CART models for the same.

**Data Description**

1. Wife's age (numerical)
2. Wife's education (categorical) 1=uneducated, 2, 3, 4=tertiary
3. Husband's education (categorical) 1=uneducated, 2, 3, 4=tertiary
4. Number of children ever born (numerical)
5. Wife's religion (binary) Non-Scientology, Scientology
6. Wife's now working? (binary) Yes, No
7. Husband's occupation (categorical) 1, 2, 3, 4(random)
8. Standard-of-living index (categorical) 1=verlow, 2, 3, 4=high
9. Media exposure (binary) Good, Not good
10. Contraceptive method used (class attribute) No,Yes

### Sample of the Dataset

| Wife_age | Wife_ education | Husband_education | No_of_children_born | Wife_religion | Wife_Working | Husband_Occupation | Standard_of_living_index | Media_exposure |
|---|---|---|---|---|---|---|---|---|
| 24.0 | Primary | Secondary | 3.0 | Scientology | No | 2 | High | Exposed |
| 45.0 | Uneducated | Secondary | 10.0 | Scientology | No | 3 | Very High | Exposed |
| 43.0 | Primary | Secondary | 7.0 | Scientology | No | 3 | Very High | Exposed |
| 42.0 | Secondary | Primary | 9.0 | Scientology | No | 3 | High | Exposed |
| 36.0 | Secondary | Secondary | 8.0 | Scientology | No | 3 | Low | Exposed |

**Table 10: Dataset Sample**

The data set having 10 variables for 1473 female information

### Check the types of variables in the data frame

```
Wife_age                  float64
Wife_ education             object
Husband_education           object
No_of_children_born       float64
Wife_religion               object
Wife_Working                object
Husband_Occupation           int64
Standard_of_living_index    object
Media_exposure              object
Contraceptive_method_used   object
dtype: object
```

**Table 11: Datatypes**

Out of the 10 variables, 7 variables are of object data type , 2 are of float data type(float64) and one is of integer(int64) data type

### Five Point Summary (descriptive statistics)

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Wife_age** | 1402.0 | 32.606277 | 8.274927 | 16.0 | 26.0 | 32.0 | 39.0 | 49.0 |
| **No_of_children_born** | 1452.0 | 3.254132 | 2.365212 | 0.0 | 1.0 | 3.0 | 4.0 | 16.0 |
| **Husband_Occupation** | 1473.0 | 2.137814 | 0.864857 | 1.0 | 1.0 | 2.0 | 3.0 | 4.0 |

**Table 12 : Five Point Summary**

From 5 point summary, it is visible that many variables have values as 0.

- Wife's age ranges from 16 to 49, the legally minimum age for marriage is 18 years, hence age less than 16 may be entered by mistake
- Number of children born is from 0 to 16, this data also may have some wrong entries since having 16 children is not practical

11 entries with ages less than 18 were removed.

**Check for any null values or duplicates**

```
Wife_age                    71
Wife_ education              0
Husband_education            0
No_of_children_born         21
Wife_religion                0
Wife_Working                 0
Husband_Occupation           0
Standard_of_living_index     0
Media_exposure               0
Contraceptive_method_used    0
dtype: int64
```

**Table 13: Null Values**

Null values are present in wife_age and wife_education. Null values were filled with the median values.

There are 80 duplicated entries were there. All 80 duplicated entries were removed.

**Outlier Treatment**

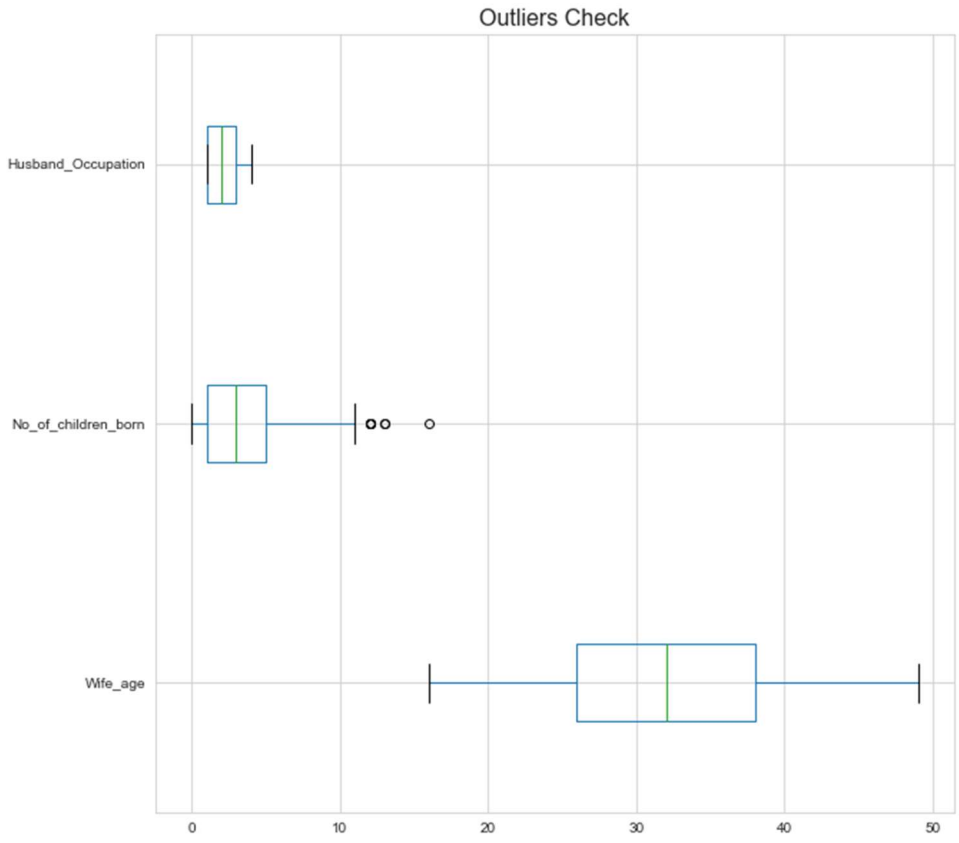


**Figure 18 : Box Plot Outlier Checking**

Outliers are present in number of children born, let's treat this since the higher number does not look practical.

Boxplot after treating outliers

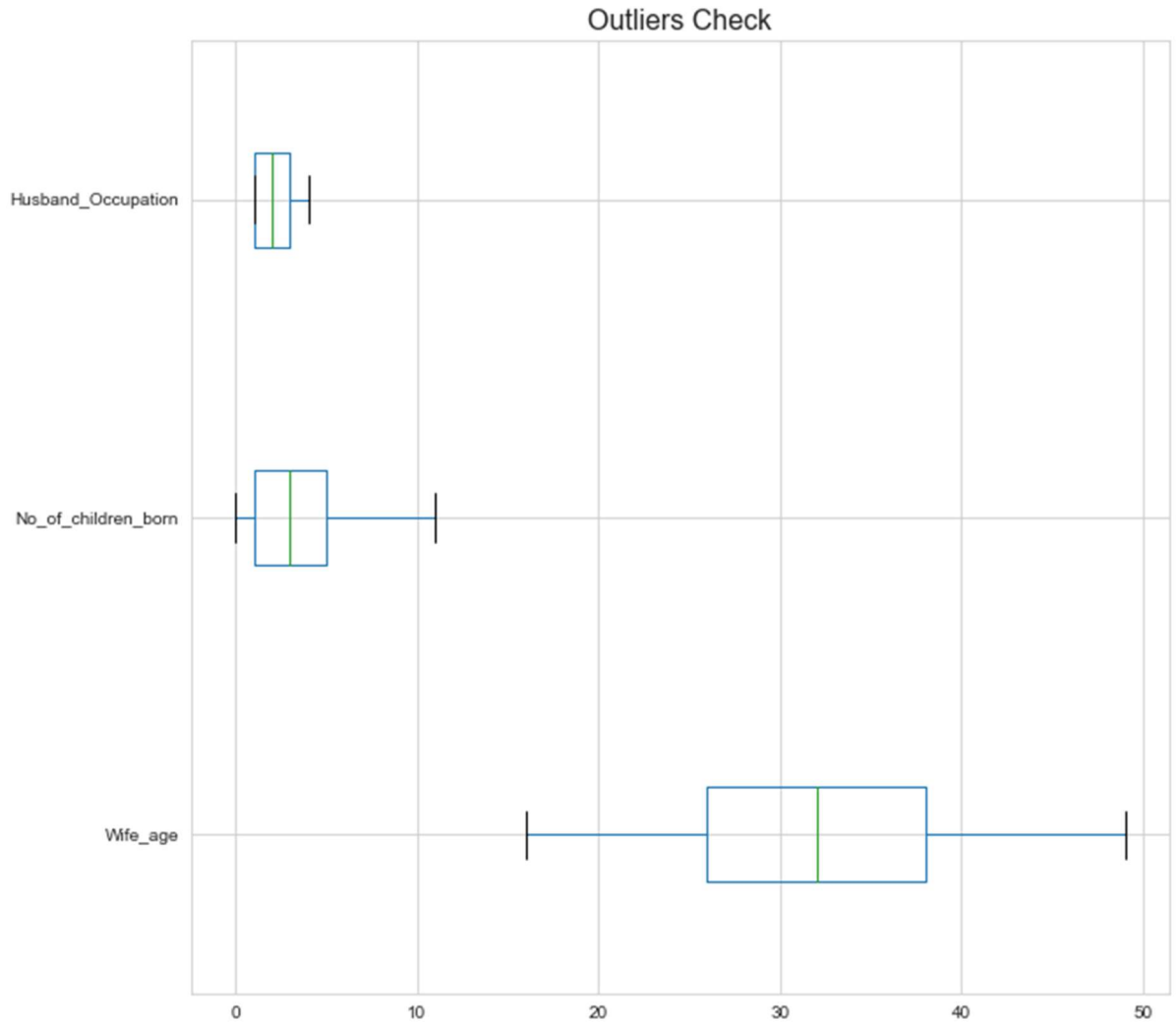


**Figure 19 : Box Plot After Outlier Treatment**

## Exploratory Data Analysis

## Univariate Analysis

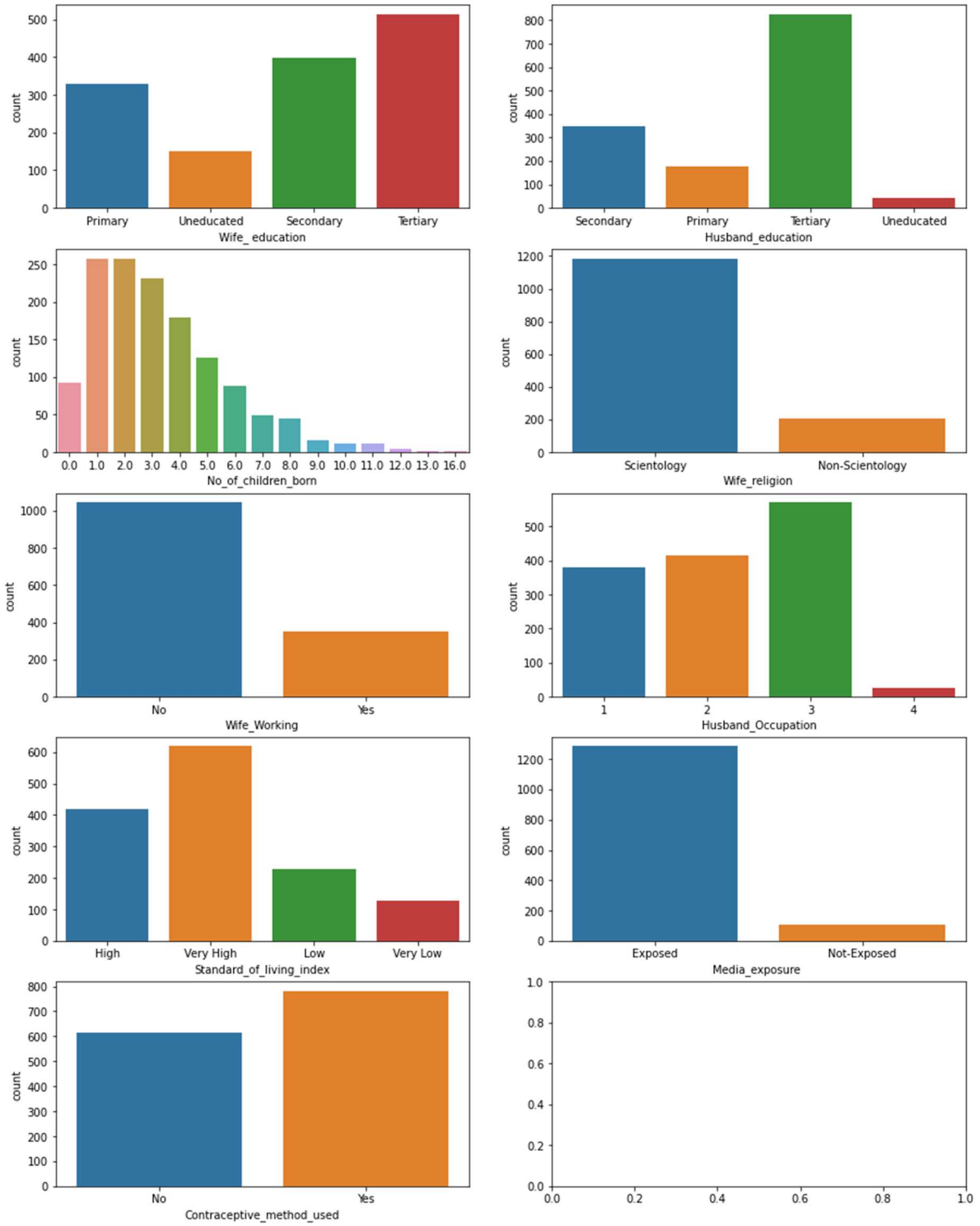


**Figure 20 : Count plots for categorical variables**

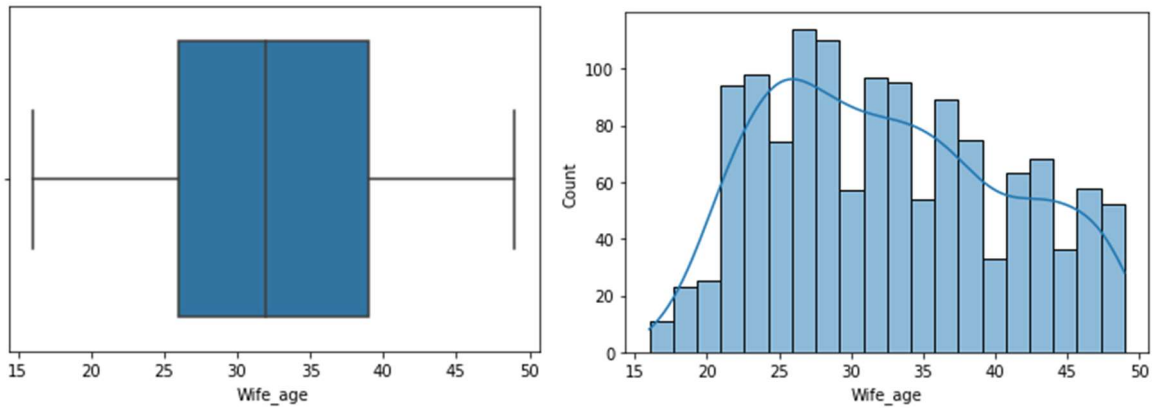


**Figure 21 : Boxplot and Histogram for Wife_age**

## Bivariate & Multivariate Analysis

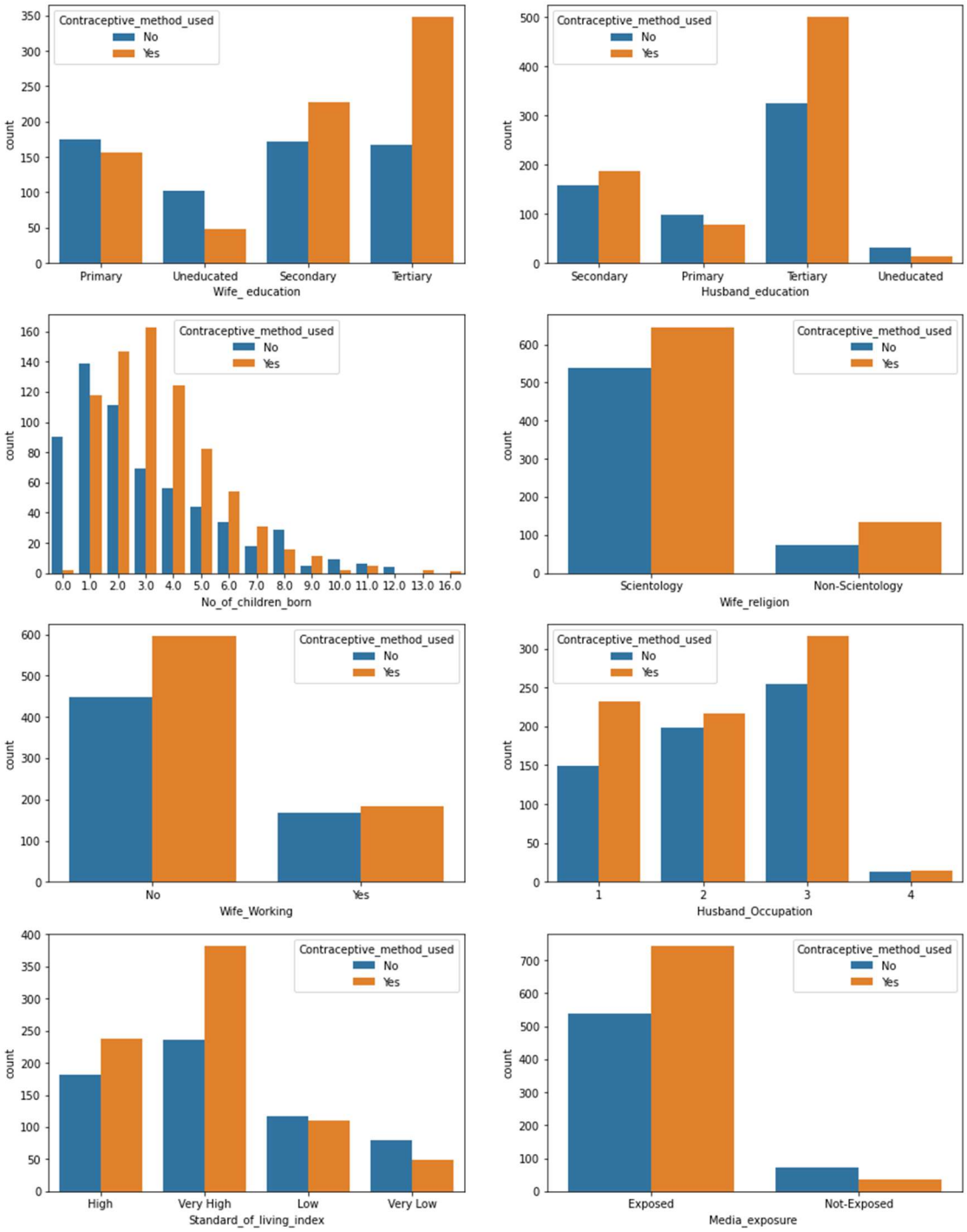


**Figure 22 : Bivariate Plots**

**Pair plot**

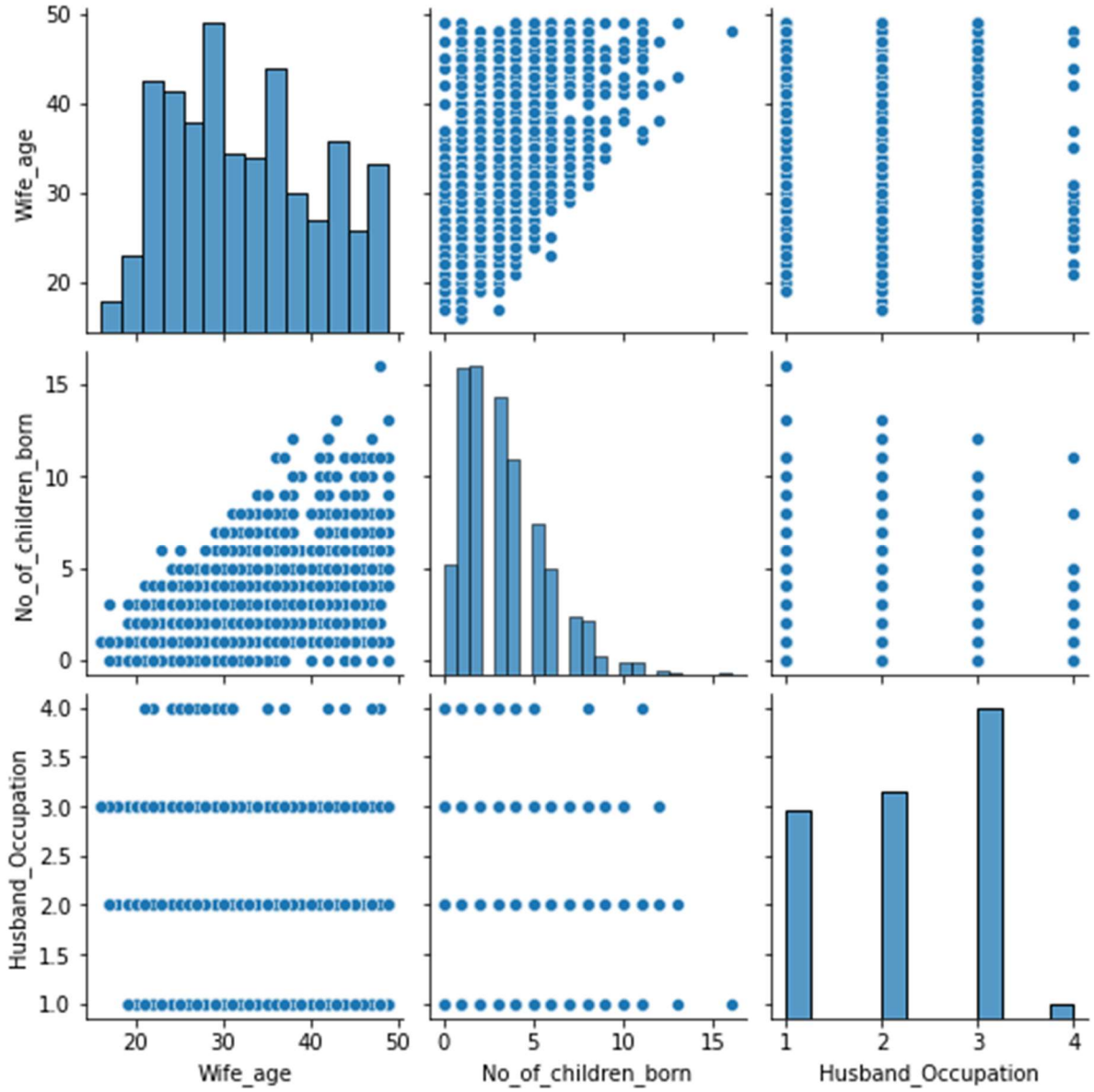


**Figure 23 : Pairplot**

**Heatmap**

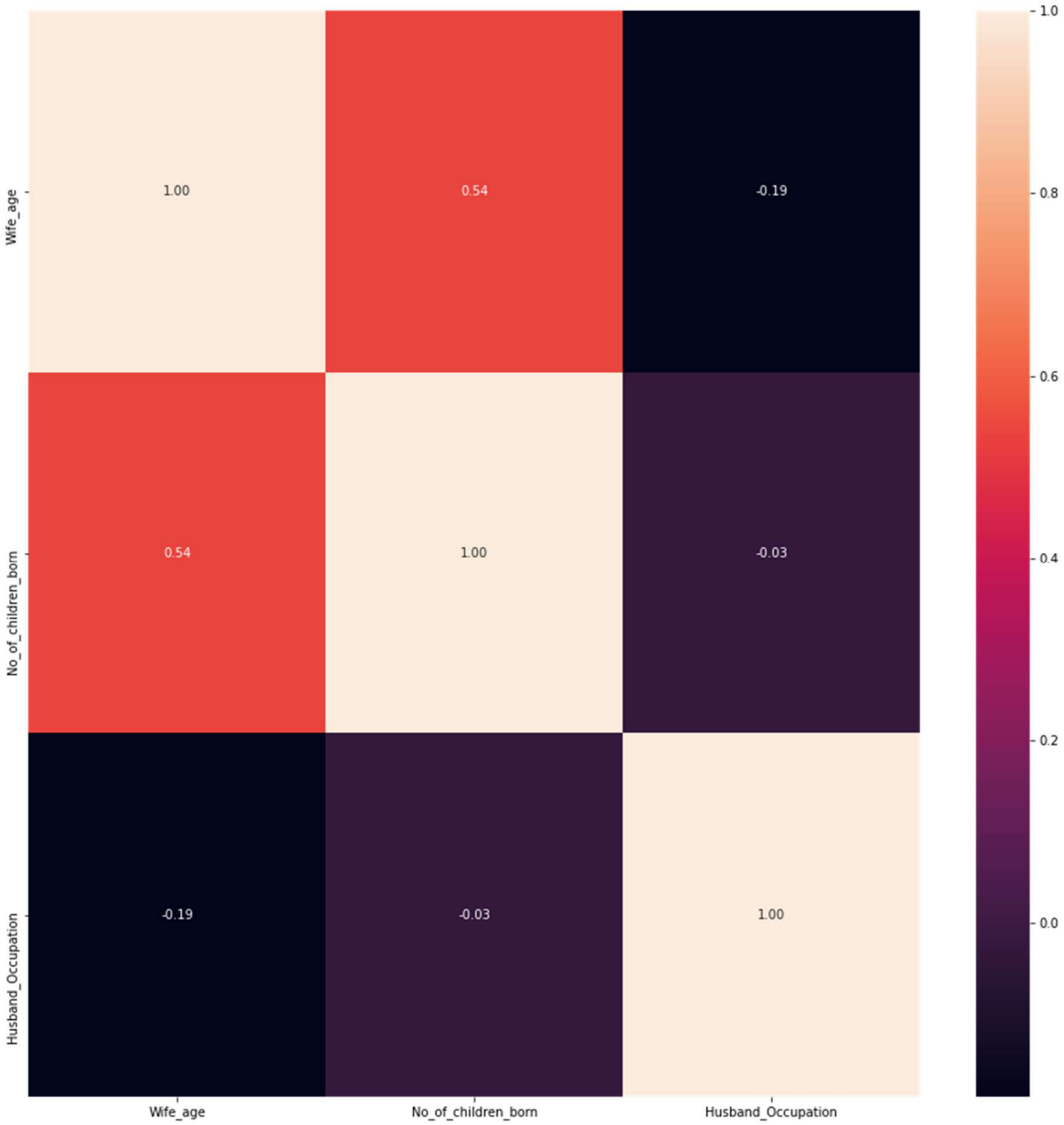


**Figure 24 : Heatmap**

**Following Insights Observed from EDA**

- Most of the women and their husbands have tertiary education
- Higher number of couples are having 1,2 or 3 children
- Very high number of women's religion is Scientology
- Compared to working women, not working women are more

- Most women are exposed to media
- Most of couples have a very high living index
- Wife's age ranges from 16 to 49
- Mostly contraceptive methods were used by women having 1 to 4 children
- Media exposed women uses contraceptive method more compared to Not-exposed women
- There are no strong correlations observed in the data
- Null values are present in the data

### 2.2 Do not scale the data. Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Logistic Regression and LDA (linear discriminant analysis) and CART.

Response variable encoded as follows

Contraceptive method used : 1

Contraceptive method not used : 0

Data is split into train and test (70:30) and applied Logistic Regression and LDA (linear discriminant analysis) and CART.

**Insights from Logistic Regression**

For predicting women's not using a contraceptive method

- 71% of the women predicted are actually not using contraceptive methods, out of all the women predicted not using a contraceptive method
- 46% of the women not using a contraceptive method predicted correctly

For predicting women's using a contraceptive method

- 67% of the women predicted are actually using contraceptive methods, out of all the women's predicted using a contraceptive method
- 85% of the women using a contraceptive method predicted correctly

The overall accuracy of the model is 68%. Accuracy, AUC, Precision, and Recall for test data is almost in line with training data. This proves no overfitting or underfitting has happened.

**Insights from LDA**

For predicting women's not using a contraceptive method

- 71% of the women predicted are actually not using contraceptive methods, out of all the women predicted not using a contraceptive method
- 45% of the women's not using a contraceptive method predicted correctly

For predicting women's using a contraceptive method

- 66% of the women predicted are actually using  contraceptive methods, out of all the women's predicted using a contraceptive method
- 86% of the women using a contraceptive method predicted correctly

The overall accuracy of the model is 68%. Accuracy, AUC, Precision, and Recall for test data is almost inline with training data. This proves no overfitting or underfitting has happened.

Insights from CART

A part of CART decision tree

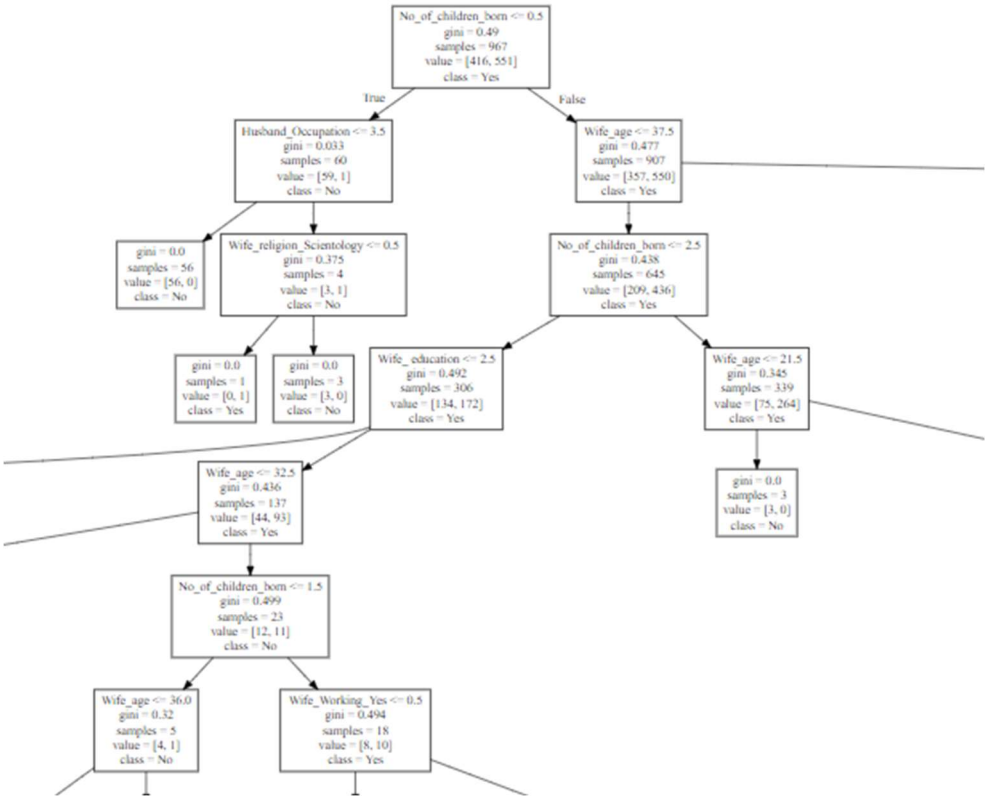


Figure 25 : Decision Tree

For predicting women's not using a contraceptive method

- 73% of the women predicted are actually not using  contraceptive methods, out of all the women's predicted not using a contraceptive method
- 54% of the women not using a contraceptive method predicted correctly

For predicting women's using a contraceptive method

- 67% of the women predicted are actually using  contraceptive methods, out of all the women's predicted using a contraceptive method
- 82% of the women's using a contraceptive method predicted correctly

The overall accuracy of the model is 69%. Accuracy, AUC, Precision and Recall for test data is lower than for training data. This proves overfitting has happened.

### 2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model Final Model: Compare Both the models and write inference which model is best/optimized.

**Logistic Regression Model**

Classification Report

Classification Report of the training data:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.65 | 0.50 | 0.57 | 427 |
| 1 | 0.67 | 0.79 | 0.72 | 540 |
| accuracy | | | 0.66 | 967 |
| macro avg | 0.66 | 0.65 | 0.65 | 967 |
| weighted avg | 0.66 | 0.66 | 0.65 | 967 |

Classification Report of the test data:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.71 | 0.46 | 0.56 | 183 |
| 1 | 0.67 | 0.85 | 0.75 | 232 |
| accuracy | | | 0.68 | 415 |
| macro avg | 0.69 | 0.66 | 0.65 | 415 |
| weighted avg | 0.69 | 0.68 | 0.67 | 415 |

Table 14 : Logistic Regression Classification Report

Confusion Matrix

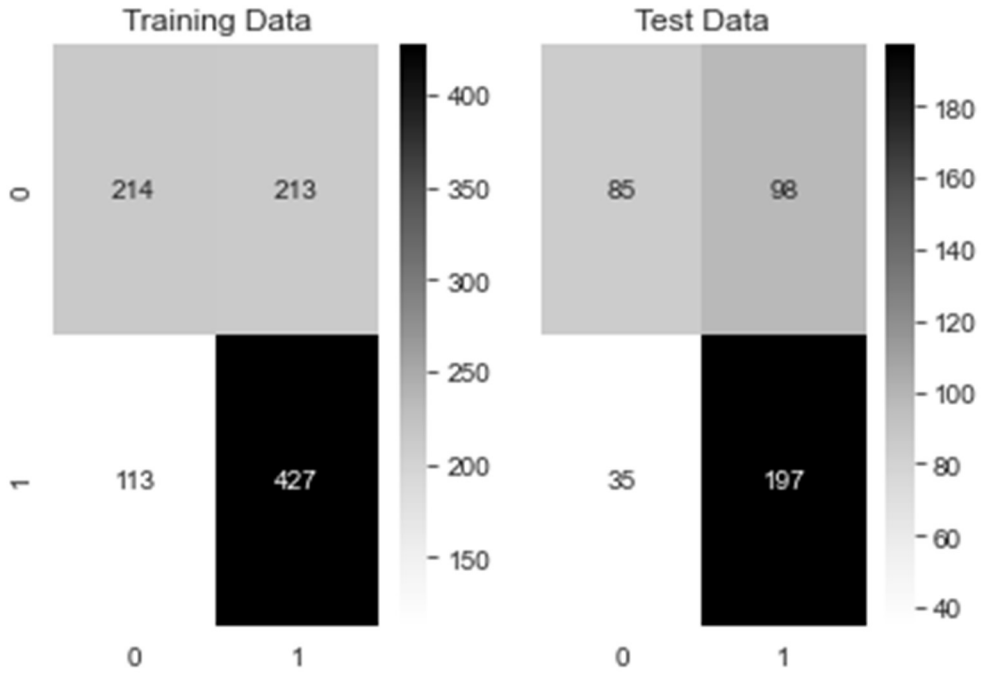


Figure 26 : Confusion Matrix

ROC Curve and ROC_AUC Scores

```
AUC for the Training Data: 0.701
AUC for the Test Data: 0.714
```

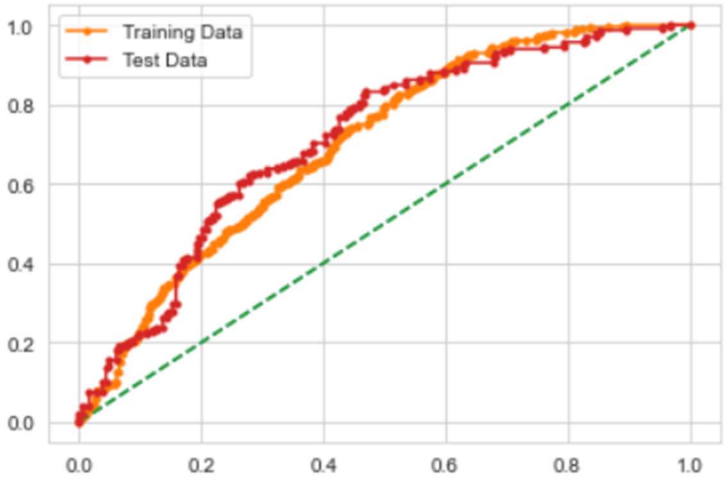


Figure 27 : Logistic Regression ROC Curve

**Linear Discriminant Analysis (LDA)**

Classification Report

Classification Report of the training data:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.66 | 0.49 | 0.56 | 427 |
| 1 | 0.67 | 0.80 | 0.73 | 540 |
| accuracy | | | 0.66 | 967 |
| macro avg | 0.66 | 0.65 | 0.65 | 967 |
| weighted avg | 0.66 | 0.66 | 0.66 | 967 |

Classification Report of the test data:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.71 | 0.45 | 0.55 | 183 |
| 1 | 0.66 | 0.86 | 0.75 | 232 |
| accuracy | | | 0.68 | 415 |
| macro avg | 0.69 | 0.65 | 0.65 | 415 |
| weighted avg | 0.69 | 0.68 | 0.66 | 415 |

Table 15 : LDA Classification Report

Confusion Matrix

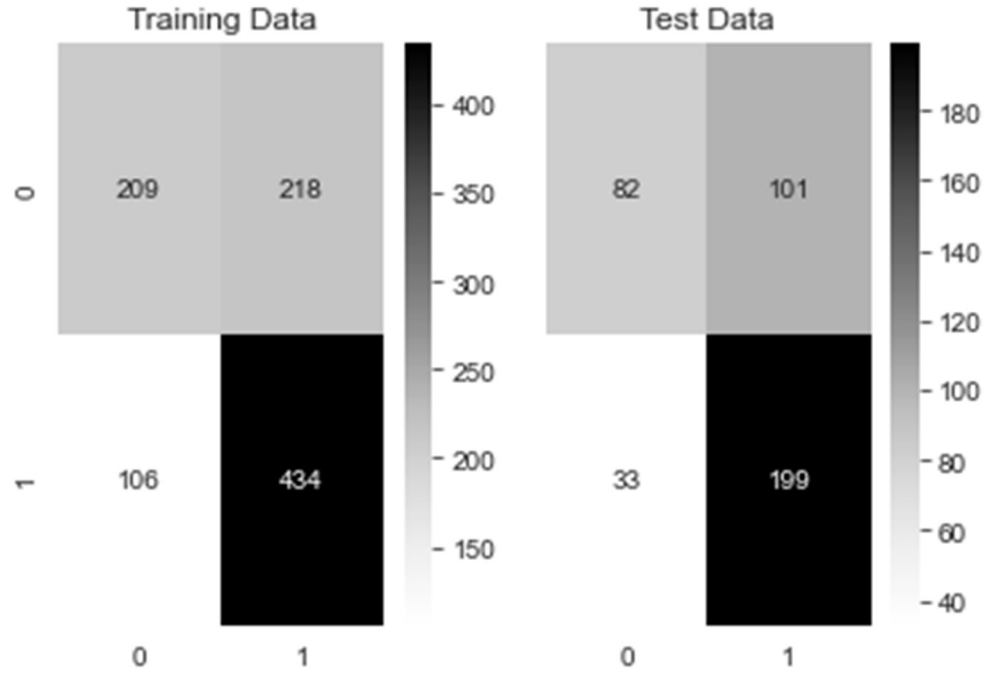


**Figure 28 : Confusion Matrix LDA**

ROC Curve and ROC_AUC Scores

```
AUC for the Training Data: 0.701
AUC for the Test Data: 0.714
```

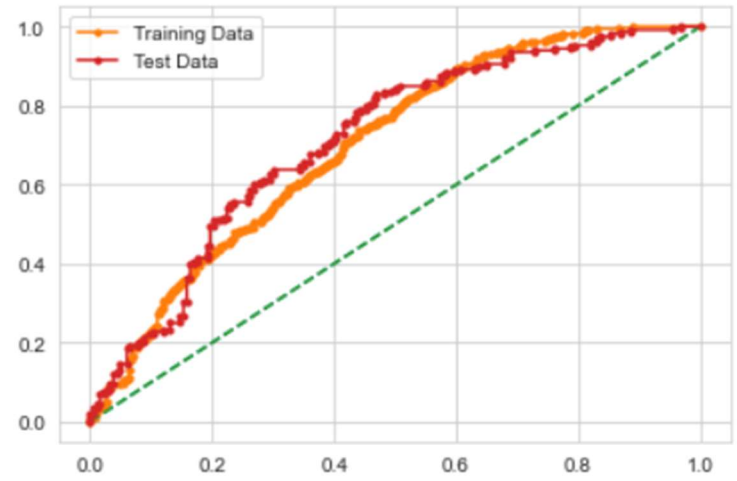


**Figure 29 : ROC Curve LDA**

**CART**

Classification Report

Classification Report of the training data:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.75 | 0.61 | 0.67 | 416 |
| 1 | 0.74 | 0.85 | 0.79 | 551 |
| accuracy | | | 0.75 | 967 |
| macro avg | 0.75 | 0.73 | 0.73 | 967 |
| weighted avg | 0.75 | 0.75 | 0.74 | 967 |

Classification Report of the test data:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.73 | 0.54 | 0.62 | 194 |
| 1 | 0.67 | 0.82 | 0.74 | 221 |
| accuracy | | | 0.69 | 415 |
| macro avg | 0.70 | 0.68 | 0.68 | 415 |
| weighted avg | 0.70 | 0.69 | 0.68 | 415 |

**Table 16 : CART Classification Report**

Confusion Matrix

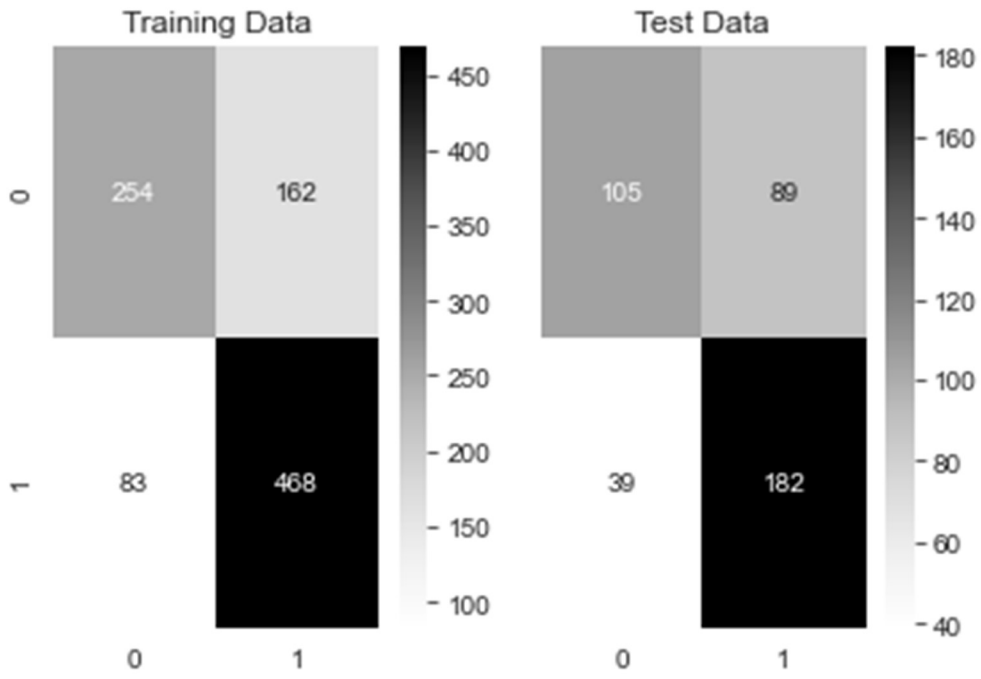


**Figure 30 : Confusion Matrix CART**

ROC Curve and ROC_AUC Scores

```
AUC for the Training Data: 0.831
AUC for the Test Data: 0.733
```

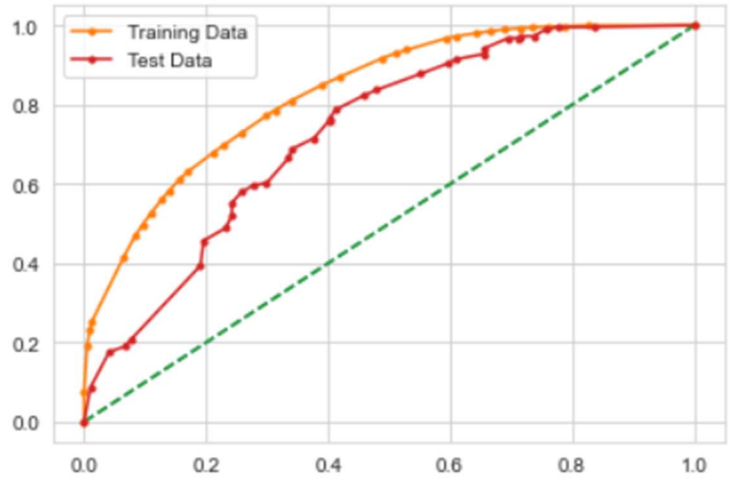


**Figure 31 : ROC Curve CART**

Below is the model comparison for Logistic Regression, LDA and CART

| | | Precision | | Recall | | Accuracy | | True Positives/Negatives | | False Positives/Negatives | | AUC Score | | Model Score | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Train Data | Test Data | Train Data | Test Data | Train Data | Test Data | Train Data | Test Data | Train Data | Test Data | Train Data | Test Data | Train Data | Test Data |
| **Logistic Regressio** | 0 | 0.65 | 0.71 | 0.5 | 0.46 | 0.66 | 0.68 | 214 | 85 | 213 | 98 | 0.7 | 0.71 | 0.68 | 0.66 |
| | 1 | 0.67 | 0.67 | 0.7 | 0.85 | | | 427 | 197 | 113 | 35 | | | | |
| **LDA** | 0 | 0.66 | 0.71 | 0.49 | 0.45 | 0.66 | 0.68 | 209 | 82 | 218 | 101 | 0.7 | 0.71 | 0.67 | 0.68 |
| | 1 | 0.67 | 0.66 | 0.8 | 0.86 | | | 434 | 199 | 106 | 33 | | | | |
| **CART** | 0 | 0.75 | 0.73 | 0.61 | 0.54 | 0.75 | 0.69 | 254 | 105 | 162 | 89 | 0.83 | 0.73 | 0.75 | 0.69 |
| | 1 | 0.74 | 0.67 | 0.85 | 0.82 | | | 468 | 182 | 83 | 39 | | | | |

**Table 17 : Model Comparison Logistic Regression,LDA & CART**

From the above comparison, CART is having the highest accuracy among the three models in train data and test data. Also when we compare the confusion matrix it's clear that CART identifies more accurately women's who should use a contraceptive method in the train data, but when we look into the test data CART identified less compared to the other two methods. Also CART model is overfitted. Accuracy, Recall & Precision reduces in test data compared to train data. In this assignment predicting the true positives correctly is important hence when

we check model performance in test data, LDA is having highest recall for label 1. There is no overfitting or underfitting in the LDA model. I choose LDA as the final best/optimized model.

### 2.4 Inference: Basis on these predictions, what are the insights and recommendations.

We started by loading the dataset, first attempt was to understand more about the dataset. We checked the dataset by printing the first and last five rows of the data. EDA is done to understand more about the data. From EDA we got some insights

- Most of the women and their husbands have tertiary education
- Higher number of couples are having 1,2 or 3 children
- Very high number of women's religion is Scientology
- Compared to working women, not working women's are more
- Most women are exposed to media
- Most of couples have a very high living index
- Wife's age ranges from 16 to 49
- Mostly contraceptive methods were used by women having 1 to 4 children
- Media exposed women uses contraceptive method more compared to Not-exposed women
- There are no strong correlations observed in the data
- Null values are present in the data

Duplicate entries and null values were observed during EDA. Duplicate entries is removed and null values are filled using the median to make our model better. Also, outliers treated on one variable. All the categorical variables are coded to build a model since all the models require values as numeric. Followed by this data is split into train and test (70:30). Applied Logistic Regression and LDA (linear discriminant analysis) and CART. Model comparison is done above and we concluded the best model is LDA. LDA can be used to predict do/don't they use a contraceptive method of choice based on their demographic and socio-economic characteristics. CART is overfitting and Logistic regression is having low model score compared to LDA.

Also following insights we get from LDA

- Coeff of Wife_Education is having highest in magnitude thus it helps in discriminating the target the best
- Coeff of Media exposure is having lowest in magnitude thus it helps in discriminating the target the least