Predictive Modeling Project Report

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.

Dataset for Problem 1: [compactiv.xlsx]

DATA DICTIONARY:

System measures used:

lread - Reads (transfers per second) between system memory and user memory
lwrite - writes (transfers per second) between system memory and user memory
scall - Number of system calls of all types per second
sread - Number of system read calls per second .
swrite - Number of system write calls per second .
fork - Number of system fork calls per second.
exec - Number of system exec calls per second.
rchar - Number of characters transferred per second by system read calls
wchar - Number of characters transfreed per second by system write calls
pgout - Number of page out requests per second
ppgout - Number of pages, paged out per second
pgfree - Number of pages per second placed on the free list.
pgscan - Number of pages checked if they can be freed per second
atch - Number of page attaches (satisfying a page fault by reclaiming a page in memory) per second
pgin - Number of page-in requests per second
ppgin - Number of pages paged in per second
pflt - Number of page faults caused by protection errors (copy-on-writes).
vflt - Number of page faults caused by address translation .
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.)
freemem - Number of memory pages available to user processes
freeswap - Number of disk blocks available for page swapping.
----
usr - Portion of time (%) that cpus run in user mode

Problem 1: Linear Regression

Executive Summary

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.

Linear equation has to be designed 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.

Introduction

The purpose of this whole exercise is to explore the dataset and build a linear regression model. The data consists of various features of system attributes; this analysis is to build a effective linear regression model which predicts the usr feature. usr feature is the portion of time the CPUs run in user mode.

Data Description

lread - Reads (transfers per second) between system memory and user memory
lwrite - writes (transfers per second) between system memory and user memory
scall - Number of system calls of all types per second
sread - Number of system read calls per second .
swrite - Number of system write calls per second .
fork - Number of system fork calls per second.
exec - Number of system exec calls per second.
rchar - Number of characters transferred per second by system read calls
wchar - Number of characters transfreed per second by system write calls
pgout - Number of page out requests per second
ppgout - Number of pages, paged out per second
pgfree - Number of pages per second placed on the free list.
pgscan - Number of pages checked if they can be freed per second
atch - Number of page attaches (satisfying a page fault by reclaiming a page in memory) per second
pgin - Number of page-in requests per second
ppgin - Number of pages paged in per second
pflt - Number of page faults caused by protection errors (copy-on-writes).
vflt - Number of page faults caused by address translation .
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.)
freemem - Number of memory pages available to user processes

freeswap - Number of disk blocks available for page swapping.

----

usr - Portion of time (%) that cpus run in user mode

Sample of dataset:

```
   lread  lwrite  scall  sread  swrite  fork  exec    rchar    wchar  pgout  \
0      1       0   2147     79      68   0.2   0.2  40671.0  53995.0    0.0   
1      0       0    170     18      21   0.2   0.2    448.0   8385.0    0.0   
2     15       3   2162    159     119   2.0   2.4      NaN  31950.0    0.0   
3      0       0    160     12      16   0.2   0.2      NaN   8670.0    0.0   
4      5       1    330     39      38   0.4   0.4      NaN  12185.0    0.0   

   ...  pgscan  atch  pgin  ppgin    pflt    vflt         runqsz  freemem  \
0  ...     0.0   0.0   1.6    2.6   16.00   26.40      CPU_Bound     4670   
1  ...     0.0   0.0   0.0    0.0   15.63   16.83  Not_CPU_Bound     7278   
2  ...     0.0   1.2   6.0    9.4  150.20  220.20  Not_CPU_Bound      702   
3  ...     0.0   0.0   0.2    0.2   15.60   16.80  Not_CPU_Bound     7248   
4  ...     0.0   0.0   1.0    1.2   37.80   47.60  Not_CPU_Bound      633   

  freeswap  usr  
0  1730946   95  
1  1869002   97  
2  1021237   87  
3  1863704   98  
4  1760253   90  

[5 rows x 22 columns]
```

Exploratory Data Analysis (EDA)

Let us check the types of variables in the data frame and the null values

Column	Non-Null Count	Dtype
lread	8192	int64
lwrite	8192	int64
scall	8192	int64
sread	8192	int64
swrite	8192	int64
fork	8192	float64
exec	8192	float64
rchar	8088	float64
wchar	8177	float64
pgout	8192	float64
ppgout	8192	float64
pgfree	8192	float64
pgscan	8192	float64
atch	8192	float64
pgin	8192	float64
ppgin	8192	float64
pflt	8192	float64
vflt	8192	float64
runqsz	8192	object
freemem	8192	int64
freeswap	8192	int64
usr	8192	int64

• There are total 8192 rows and 22 columns in the dataset

• Out of 22 columns, o 1 column is object type o 8 columns are integer data type o 13 columns are float data type

• rchar and wchar columns has null values

5 Point Summary

5-Point Summary:

	lread	lwrite	scall	sread	swrite	\
count	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	
mean	19.559692	13.106201	2306.318237	210.479980	150.058228	
std	53.353799	29.891726	1633.617322	198.980146	160.478980	
min	0.000000	0.000000	109.000000	6.000000	7.000000	
25%	2.000000	0.000000	1012.000000	86.000000	63.000000	
50%	7.000000	1.000000	2051.500000	166.000000	117.000000	
75%	20.000000	10.000000	3317.250000	279.000000	185.000000	
max	1845.000000	575.000000	12493.000000	5318.000000	5456.000000	

	fork	exec	rchar	wchar	pgout	...	\
count	8192.000000	8192.000000	8.088000e+03	8.177000e+03	8192.000000	...	
mean	1.884554	2.791998	1.973857e+05	9.590299e+04	2.285317	...	
std	2.479493	5.212456	2.398375e+05	1.408417e+05	5.307038	...	
min	0.000000	0.000000	2.780000e+02	1.498000e+03	0.000000	...	
25%	0.400000	0.200000	3.409150e+04	2.291600e+04	0.000000	...	
50%	0.800000	1.200000	1.254735e+05	4.661900e+04	0.000000	...	
75%	2.200000	2.800000	2.678288e+05	1.061010e+05	2.400000	...	
max	20.120000	59.560000	2.526649e+06	1.801623e+06	81.440000	...	

	pgfree	pgscan	atch	pgin	ppgin	\
count	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	
mean	11.919712	21.526849	1.127505	8.277960	12.388586	
std	32.363520	71.141340	5.708347	13.874978	22.281318	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.600000	0.600000	
50%	0.000000	0.000000	0.000000	2.800000	3.800000	
75%	5.000000	0.000000	0.600000	9.765000	13.800000	
max	523.000000	1237.000000	211.580000	141.200000	292.610000	

	pflt	vflt	freemem	freeswap	usr
count	8192.000000	8192.000000	8192.000000	8.192000e+03	8192.000000
mean	109.793799	185.315796	1763.456299	1.328126e+06	83.968872
std	114.419221	191.000603	2482.104511	4.220194e+05	18.401905
min	0.000000	0.200000	55.000000	2.000000e+00	0.000000
25%	25.000000	45.400000	231.000000	1.042624e+06	81.000000
50%	63.800000	120.400000	579.000000	1.289290e+06	89.000000
75%	159.600000	251.800000	2002.250000	1.730380e+06	94.000000
max	899.800000	1365.000000	12027.000000	2.243187e+06	99.000000

[8 rows x 21 columns]

• All the numerical columns have numerical values alone.

• 75% of pgscan data are 0, it doesnt make value to the y variable usr. Therefore pgscan variable will be removed from the dataset

Univariate, Bivariate and Multivariate Analysis:

Figure 1 - Univariate Process Chart

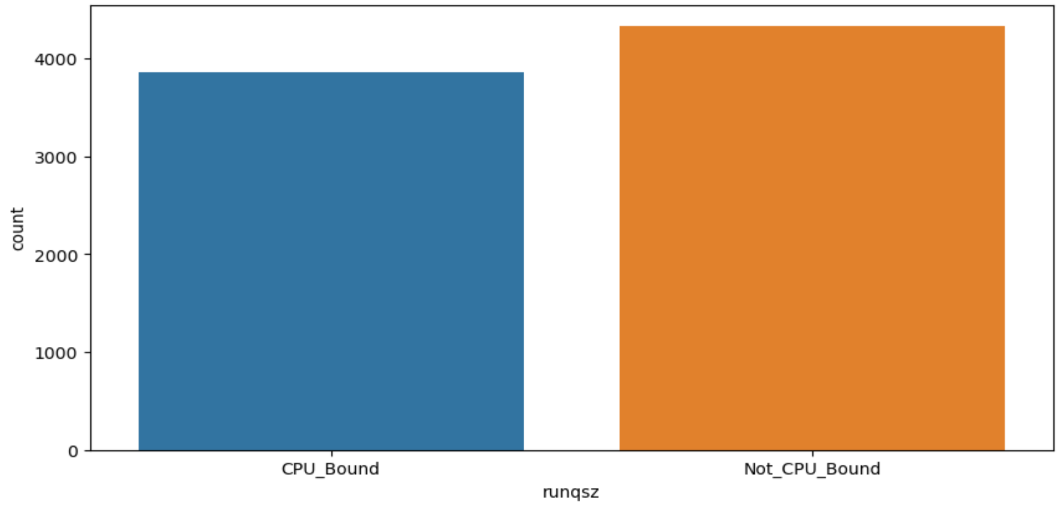


Figure 2 - Univariate User Mode Chart

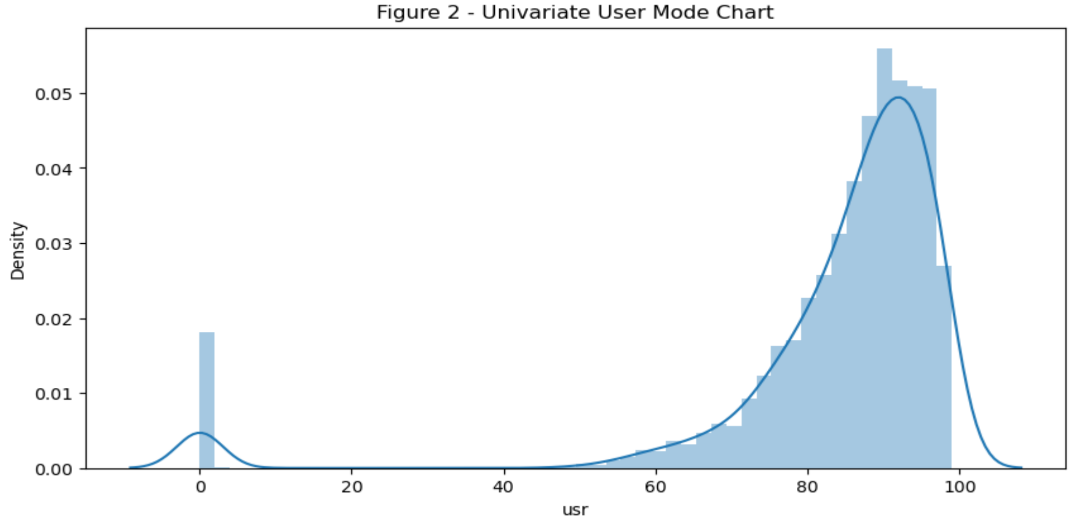
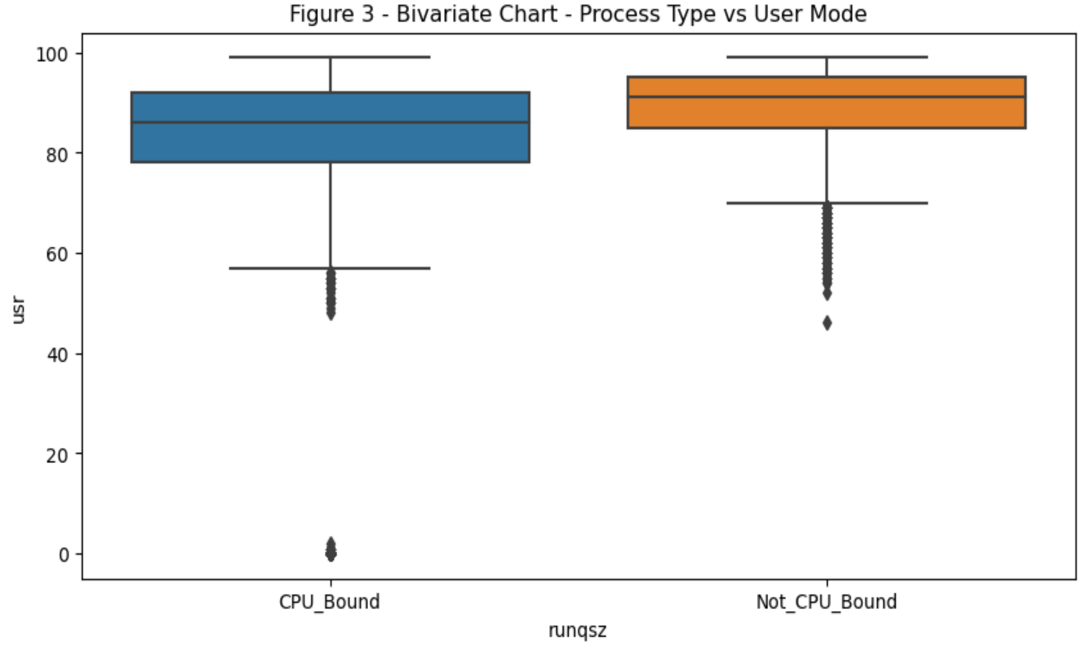


Figure 3 - Bivariate Chart - Process Type vs User Mode



Observation:

- runqsz has 2 unique values, “Not_CPU_Bound” has more count than CPU_Bound
- usr has less 0 values and more higher values. This shows systems runs more time in user mode

Let's replace the “runqsz” origin column values with their actual values. So that “runqsz” feature will

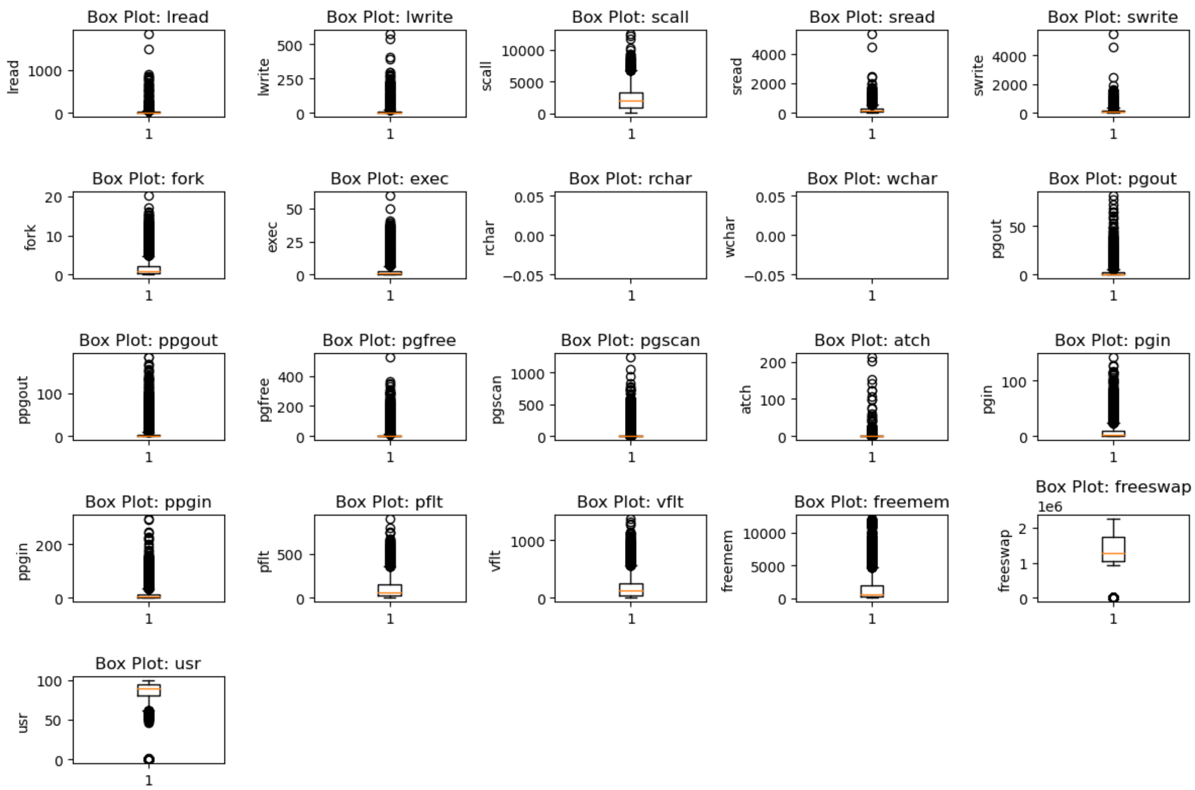
be used in linear regression model preparation.

runqsz values are imputed has shown below.

CPU_Bound to 1

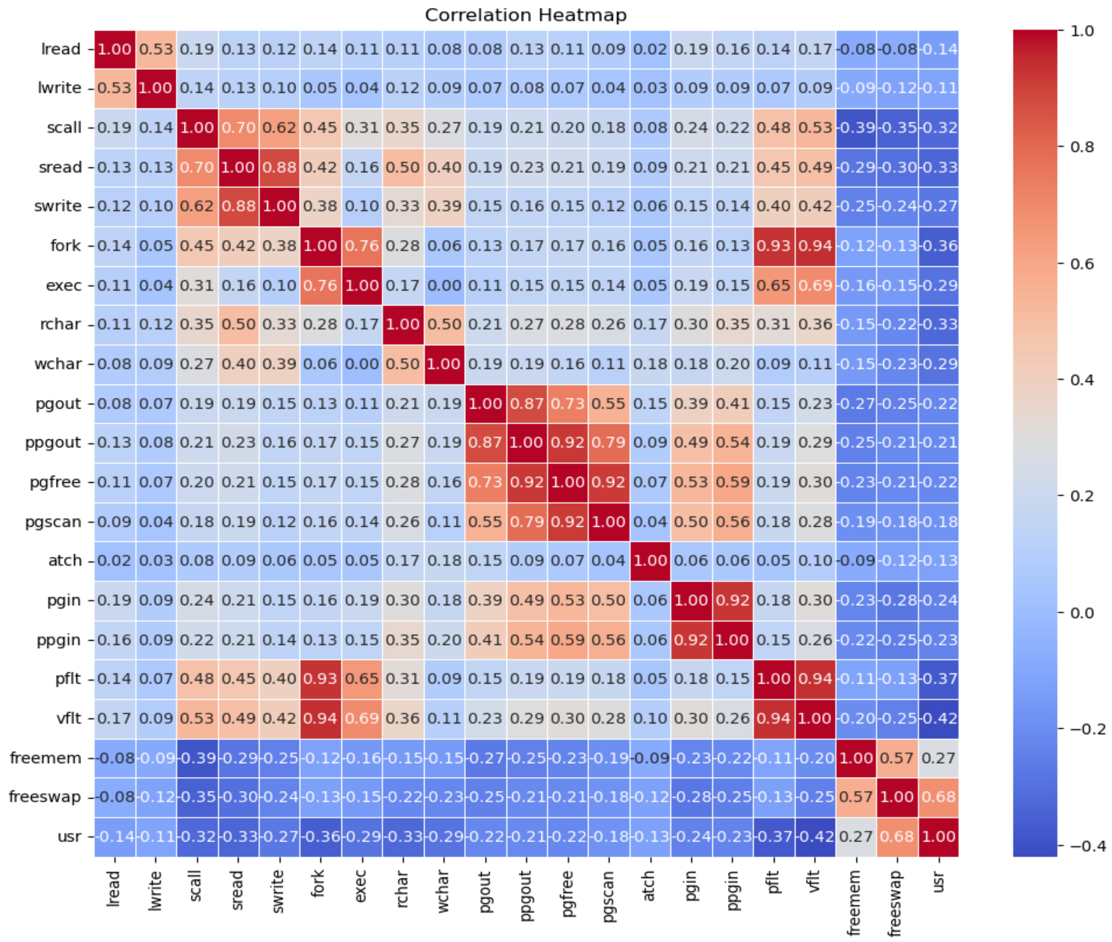
Not_CPU_Bound to 0

Box Plot



- From the box plot it is clearly visible all the features has outliers except runqsz

Heat Map



Observation:

- Above heat map brings out the correlation between the features.
- There is a high correlation (94%) between vflt and fork; vflt and pflt
- 93% correlation shown between pflt and fork
- 92% correlation shown between ppgin and pgin
- 92% correlation shown between pgfree and ppgout
- 88% correlation shown between swrite and sread
- 87% correlation shown between ppgout and pgout

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.

Null/Missing value treatment:

Two features has null values:

rchar 104

wchar 15

rchar and wchar null values are treated with mean value.

Duplicate Checks

- There is no duplicate rows present in the data set.

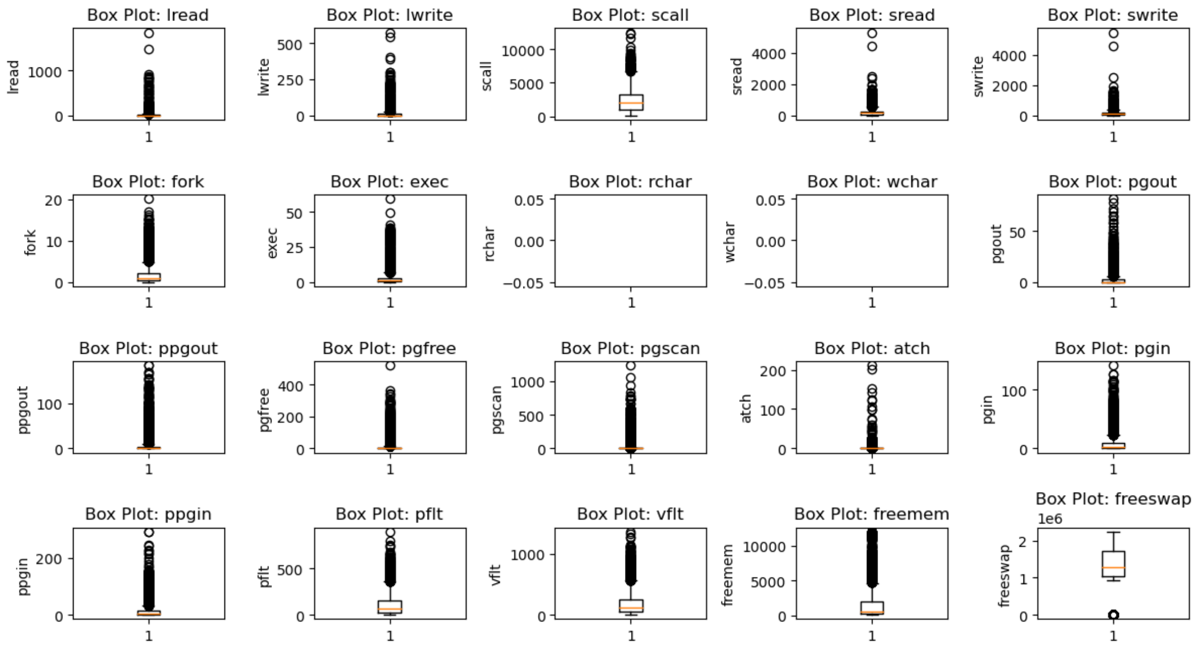
Zero value Check

```
Columns with Zero Values and Their Percentage:
lread       8.239746
lwrite     32.763672
fork        0.256348
exec        0.256348
pgout      59.545898
ppgout     59.545898
pgfree     59.436035
pgscan     78.710938
atch       55.847168
pgin       14.892578
ppgin      14.892578
pflt        0.036621
usr         3.454590
dtype: float64
```

- pgscan feature has more than 75 percentile 0 value, therefore pgscan will be dropped from the data frame
- Other features zero value are less than 60 percentile, therefore those features are not dropped from the data frame
- New feature is not needed for the compactiv data set

Outlier Checks

Box Plot



Except runqsz all features has outliers;

Liner model before treating outliers

lread int64

lwrite int64

scall int64

sread int64

swrite int64

fork float64

exec float64

rchar float64

wchar float64

pgout float64

ppgout float64

pgfree float64

atch float64

```
pgin       float64
ppgin      float64
pflt       float64
vflt       float64
runqsz      object
freemem      int64
freeswap     int64
dtype: object. The data was
0       95
1       97
2       87
3       98
4       90
        ..
8187    80
8188    90
8189    87
8190    83
8191    94
Name: usr, Length: 8192, dtype: int64
and
      const  lread  lwrite  scall  sread  swrite  fork  exec     rchar  \
0       1.0      1       0   2147     79      68   0.2  0.20   40671.0
1       1.0      0       0    170     18      21   0.2  0.20     448.0
2       1.0     15       3   2162    159     119   2.0  2.40       NaN
3       1.0      0       0    160     12      16   0.2  0.20       NaN
4       1.0      5       1    330     39      38   0.4  0.40       NaN
...     ...    ...     ...    ...    ...     ...   ...   ...       ...
8187    1.0     16      12   3009    360     244   1.6  5.81  405250.0
8188    1.0      4       0   1596    170     146   2.4  1.80   89489.0
8189    1.0     16       5   3116    289     190   0.6  0.60  325948.0
```

```
8190   1.0    32     45   5180    254     179   1.2  1.20   62571.0
8191   1.0     2      0    985     55      46   1.6  4.80  111111.0

        wchar  ...  ppgout  pgfree  atch   pgin  ppgin    pflt    vflt  \
0     53995.0  ...    0.00    0.00   0.0   1.60   2.60   16.00   26.40
1      8385.0  ...    0.00    0.00   0.0   0.00   0.00   15.63   16.83
2     31950.0  ...    0.00    0.00   1.2   6.00   9.40  150.20  220.20
3      8670.0  ...    0.00    0.00   0.0   0.20   0.20   15.60   16.80
4     12185.0  ...    0.00    0.00   0.0   1.00   1.20   37.80   47.60
...       ...  ...     ...     ...   ...    ...    ...     ...     ...
8187  85282.0  ...   20.64   43.69   0.6  35.87  47.90  139.28  270.74
8188  41764.0  ...    4.80    4.80   0.8   3.80   4.40  122.40  212.60
8189  52640.0  ...    0.60    0.60   0.4  28.40  45.20   60.20  219.80
8190  29505.0  ...    1.60   13.03   0.4  23.05  24.25   93.19  202.81
8191  22256.0  ...    0.00    0.00   0.2   3.40   6.20   91.80  110.00

            runqsz  freemem  freeswap
0        CPU_Bound     4670   1730946
1    Not_CPU_Bound     7278   1869002
2    Not_CPU_Bound      702   1021237
3    Not_CPU_Bound     7248   1863704
4    Not_CPU_Bound      633   1760253
...            ...      ...       ...
8187     CPU_Bound      387    986647
8188 Not_CPU_Bound      263   1055742
8189 Not_CPU_Bound      400    969106
8190     CPU_Bound      141   1022458
8191     CPU_Bound      659   1756514

[8192 rows x 21 columns]
```

before. After,

[95 97 87 ... 87 83 94]

[[1.0 1 0 ... 'CPU_Bound' 4670 1730946]

[1.0 0 0 ... 'Not_CPU_Bound' 7278 1869002]

[1.0 15 3 ... 'Not_CPU_Bound' 702 1021237]

...

[1.0 16 5 ... 'Not_CPU_Bound' 400 969106]

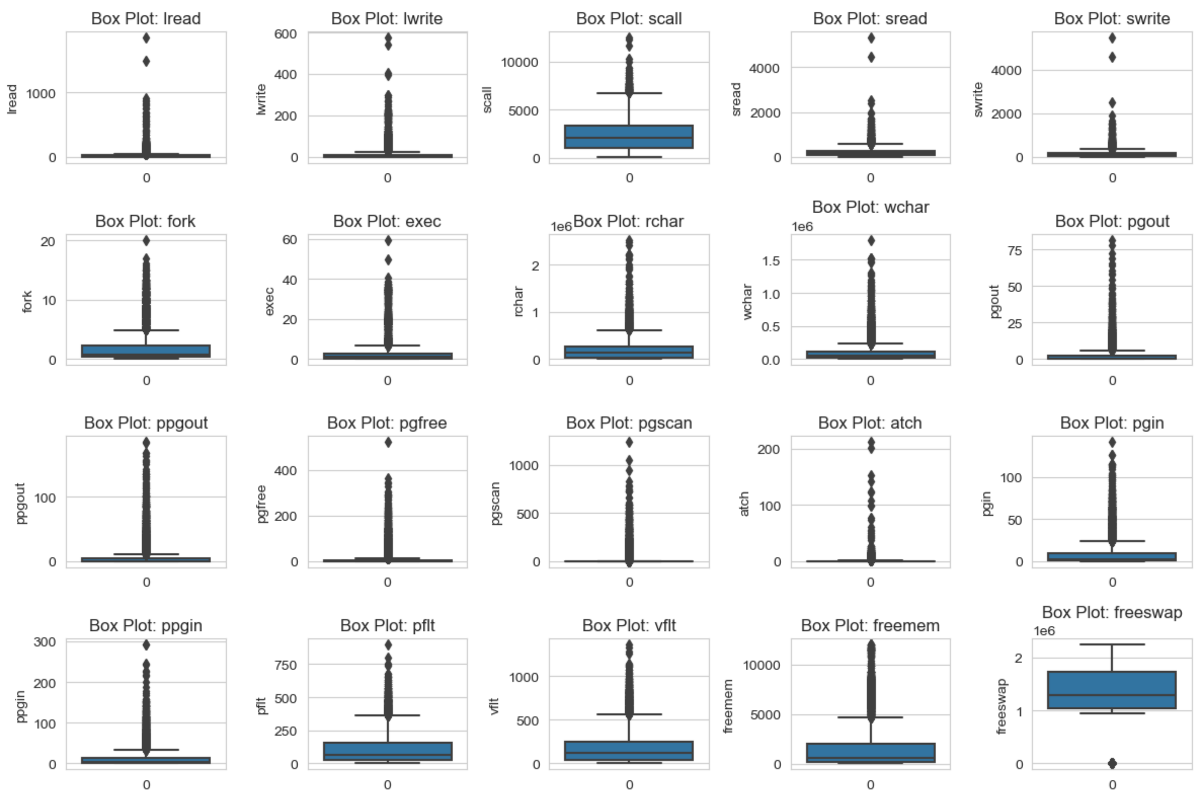
[1.0 32 45 ... 'CPU_Bound' 141 1022458]

[1.0 2 0 ... 'CPU_Bound' 659 1756514]].

Liner regression is sensitive on outliers, R-squared and Adjusted R-squared value are 64.3% and 64.1% respectively.

Box plot post to outlier treatment

Box Plot



Model building

Encoding string variable

In the given data set "runqsz" is the string variable, which is encoded manually. CPU_Bound is encoded as 1 and Not_CPU_Bound is encoded as 0. runqsz variable is type casted from object to integer.

Dummy Encoding is not necessary at this data set since the runqsz has only 2 category in it.

Split Data

usr variable has taken has a y variable (dependent variable) and all other variables are taken has x variable (independent variable).

The given data set is split into 70:30; 70% data are consider has training data and 30% of data are taken for testing the model.

X_train dataset for training the model; 21 columns with 5734 rows

X_train

```
First 5 rows of X_train:
      lread  lwrite  scall  sread  swrite  fork  exec     rchar     wchar  \
1310     26      36   5731    312     224  0.80  0.80  155004.0  264757.0
7365     15       3   1203     61      34  1.60  1.80  163076.0   33674.0
2284     39      16   5213    754     767  6.99  4.99  435848.0  314796.0
7076      2       0   2585    203     145  0.60  0.60  329604.0  126738.0
3114      2       1   1827     65      88  0.40  0.20    4487.0    8828.0

      pgout  ...  pgfree  pgscan  atch   pgin  ppgin   pflt    vflt  \
1310   0.00  ...    0.00    0.00  3.20   0.20   0.20   48.8  134.00
7365   0.00  ...    0.00    0.00  0.00   0.00   0.00  127.8  199.40
2284   6.19  ...   10.18    5.19  5.39  15.77  17.56  348.1  617.17
7076   1.00  ...    1.00    0.00  0.80  29.46  30.46   49.9  194.39
3114   0.00  ...    0.00    0.00  0.00   0.20   0.20   17.4   17.00

             runqsz  freemem  freeswap
1310      CPU_Bound      249   1383946
7365      CPU_Bound     2744   1542915
2284      CPU_Bound      236   1002172
7076  Not_CPU_Bound      451   1057294
3114  Not_CPU_Bound      689   1752789

[5 rows x 21 columns]
```

X_test dataset for testing the model; 21 columns with 2458 rows

X_test

```
First 5 rows of X_test:
      lread  lwrite  scall  sread  swrite  fork   exec      rchar     wchar  \
5670     14       7   1495    197     169  0.80   1.00    10304.0   24435.0
5369     10       8   3158    324     172  0.60   2.20  1037534.0  884253.0
2111      2       0    813    117     113  1.80   0.60    59903.0   24550.0
6659     48      68   3283    134     125  0.40   0.40    33832.0   23626.0
5227     12       2   2357    113      96  6.99  20.16    55137.0   36291.0

      pgout  ...  pgfree  pgscan  atch   pgin  ppgin    pflt    vflt  \
5670   7.98  ...   24.75   38.52   1.0   2.00   2.00   63.07  106.79
5369   0.00  ...    0.00    0.00   0.0  26.00  45.80   46.00   79.20
2111   0.60  ...    7.20   14.00   0.0   0.00   0.00   96.00  135.60
6659   4.20  ...    9.00    0.00   0.6   1.80   2.20   36.40   56.20
5227   0.00  ...    0.00    0.00   0.0   8.38  12.18  231.14  423.35

             runqsz  freemem  freeswap
5670      CPU_Bound      186    974392
5369      CPU_Bound      510   1032922
2111  Not_CPU_Bound      179   1761718
6659  Not_CPU_Bound      461   1129531
5227  Not_CPU_Bound      530   1077027

[5 rows x 21 columns]
```

Linear regression

Ordinary Least Squares Regression

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

dtype: object. The data was

0 95

1 97

2 87

3 98

4 90

..

8187 80

8188 90

```
8189    87
8190    83
8191    94
Name: usr, Length: 8192, dtype: int64
and
      const  lread  lwrite  scall  sread  swrite  fork  exec     rchar  \
0       1.0      1       0   2147     79      68   0.2  0.20   40671.0
1       1.0      0       0    170     18      21   0.2  0.20     448.0
2       1.0     15       3   2162    159     119   2.0  2.40       NaN
3       1.0      0       0    160     12      16   0.2  0.20       NaN
4       1.0      5       1    330     39      38   0.4  0.40       NaN
...     ...    ...     ...    ...    ...     ...   ...   ...       ...
8187    1.0     16      12   3009    360     244   1.6  5.81  405250.0
8188    1.0      4       0   1596    170     146   2.4  1.80   89489.0
8189    1.0     16       5   3116    289     190   0.6  0.60  325948.0
8190    1.0     32      45   5180    254     179   1.2  1.20   62571.0
8191    1.0      2       0    985     55      46   1.6  4.80  111111.0

        wchar  ...  pgfree  pgscan  atch   pgin  ppgin    pflt    vflt  \
0     53995.0  ...    0.00    0.00   0.0   1.60   2.60   16.00   26.40
1      8385.0  ...    0.00    0.00   0.0   0.00   0.00   15.63   16.83
2     31950.0  ...    0.00    0.00   1.2   6.00   9.40  150.20  220.20
3      8670.0  ...    0.00    0.00   0.0   0.20   0.20   15.60   16.80
4     12185.0  ...    0.00    0.00   0.0   1.00   1.20   37.80   47.60
...       ...  ...     ...     ...   ...    ...    ...     ...     ...
8187  85282.0  ...   43.69   55.11   0.6  35.87  47.90  139.28  270.74
8188  41764.0  ...    4.80    0.20   0.8   3.80   4.40  122.40  212.60
8189  52640.0  ...    0.60    0.00   0.4  28.40  45.20   60.20  219.80
8190  29505.0  ...   13.03   18.04   0.4  23.05  24.25   93.19  202.81
8191  22256.0  ...    0.00    0.00   0.2   3.40   6.20   91.80  110.00

          runqsz  freemem freeswap
```

```
0       CPU_Bound    4670  1730946
1    Not_CPU_Bound    7278  1869002
2    Not_CPU_Bound     702  1021237
3    Not_CPU_Bound    7248  1863704
4    Not_CPU_Bound     633  1760253
...           ...     ...      ...
8187     CPU_Bound     387   986647
8188  Not_CPU_Bound     263  1055742
8189  Not_CPU_Bound     400   969106
8190     CPU_Bound     141  1022458
8191     CPU_Bound     659  1756514

[8192 rows x 22 columns]
```

before. After,

```
[95 97 87 ... 87 83 94]
[[1.0 1 0 ... 'CPU_Bound' 4670 1730946]
 [1.0 0 0 ... 'Not_CPU_Bound' 7278 1869002]
 [1.0 15 3 ... 'Not_CPU_Bound' 702 1021237]
 ...
 [1.0 16 5 ... 'Not_CPU_Bound' 400 969106]
 [1.0 32 45 ... 'CPU_Bound' 141 1022458]
 [1.0 2 0 ... 'CPU_Bound' 659 1756514]].
```

Observation on initial Linear Regression:

- We have R-squared 0.796 and Adjusted R-squared 0.795
- F-statistic is 1116
- Coefficient of each feature for this initial linear regression model is mostly in negative. The coefficients show how a unit change in X has an effect on the y variable. A positive onegative sign on the coefficient denotes a positive or negative correlation, respectively.
- Few features has higher P value
- sread 0.737
- fork 0.822
- ppgout 0.318
- pgin 0.487

Multicollinearity check

Multicollinearity occurs when the predictor variables are correlated in the regression model. This correlation is a problem because predictors must be independent. If the variables are highly collinear, we may not be able to rely on the p-value to identify statistically significant independent variables. Variance Inflation factor technique is used to identify the multicollinearity between the variables.

VIF values:

```
     Variable
0       const
1       lread
2      lwrite
3       scall
4       sread
5      swrite
6        fork
7        exec
8       rchar
9       wchar
10      pgout
11     ppgout
12     pgfree
13     pgscan
14       atch
15       pgin
16      ppgin
17       pflt
18       vflt
19     runqsz
20    freemem
21   freeswap
```

The VIF values are sorted in descending order to uniquely identify the top variables with high collinearity between variables. ppgout is called out has highest collinearity variable with the value of 29.40. ppgout will be dropped from the training dataset and new regression model will be created.

Multiple models and check the performance of Predictions

Above Table 5 explains how the model is build step by step

Model1:

R-squared 0.796

Adj. R-squared 0.795

ppgout has the highest VIF value, it will be dropped to build Model 2

Model2:

Based on VIF value ppgout is dropped and new model is created

R-squared 0.796

Adj. R-squared 0.795

vflt has the highest VIF value, it will be dropped to build Model 3

Model3:

Based on VIF value vflt is dropped and new model is created

R-squared 0.796

Adj. R-squared 0.795

ppgin has highest VIF value, but R-squared value is lesser while comparing with pgin , therefore pgin will be dropped to build Model 4

Model4:

Based on VIF value pgin is dropped and new model is created

R-squared 0.796

Adj. R-squared 0.795

fork has highest VIF value, but R-squared value is lesser while comparing with sread, therefore sread will be dropped to build Model 5

Model5:

Based on VIF value sread is dropped and new model is created

R-squared 0.796

Adj. R-squared 0.795

fork has highest VIF value, but R-squared value is lesser while comparing with pgree, therefore pgfree will be dropped to build Model 6

Model6:

Based on VIF value pgfree is dropped and new model is created

R-squared 0.796

Adj. R-squared 0.795

fork has the highest VIF value, it will be dropped to build Model 7

Model7:

Based on VIF value fork is dropped and new model is created

R-squared 0.795

Adj. R-squared 0.795

lread has highest VIF value, but R-squared value is lesser while comparing with lwrite therefore lwrite will be dropped to build Model 8

Model8:

Based on VIF value lwrite is dropped and new model is created

R-squared 0.795

Adj. R-squared 0.794

pflt has highest VIF value, but R-squared value is lesser while comparing with swrite therefore swrite will be dropped to build Model 9

Model9:

Based on VIF value swrite is dropped and new model is created

R-squared 0.794

Adj. R-squared 0.793

pflt has highest VIF value, but R-squared value is lesser while comparing with exec therefore exec will be dropped to build Model 10

Model10:

Based on VIF value exec is dropped and new model is created

R-squared 0.792

Adj. R-squared 0.789

pgout is the only varibale which has VIF value greater than 2. Therefore it will be dropped to build Model 11

Model11:

Based on VIF value pgout is dropped and new model is created

R-squared 0.789

Adj. R-squared 0.789

atch has 0.530 P value, therefore it will be dropped to build Model 12

Model12:

Based on P value atch variable is dropped and new model is created

R-squared 0.789

Adj. R-squared 0.789

Post to Model 12 transformation technique is performed to fit the model very well in train and test dataset.

Assumptions of Linear Regression

These assumptions are essential conditions that should be met before we draw inferences regarding the

model estimates or use the model to make a prediction.

For Linear Regression, we need to check if the following assumptions hold:-

- Linearity
- Independence
- Homoscedasticity
- Normality of error terms
- No strong Multicollinearity

	Actual	Predicted	Residuals
0	95	85.198668	9.801332
1	97	92.035939	4.964061
2	87	84.770611	2.229389
3	98	92.070409	5.929591
4	90	91.351435	-1.351435
...	...	...	...
8187	80	81.649279	-1.649279
8188	90	87.030095	2.969905
8189	87	81.419780	5.580220
8190	83	73.147215	9.852785
8191	94	89.181427	4.818573

[8192 rows x 3 columns]

TEST FOR LINEARITY AND INDEPENDENCE

Why the test?

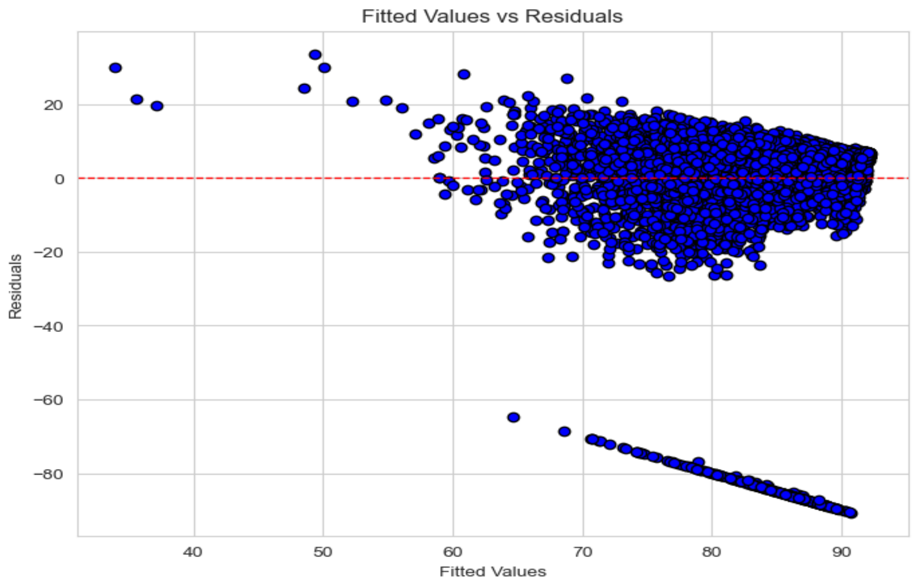
- Linearity describes a straight-line relationship between two variables, predictor variables must have
  a linear relation with the dependent variable.

How to check linearity?

- Make a plot of fitted values vs residuals. If they don't follow any pattern (the curve is a straight line),
  then we say the model is linear otherwise model is showing signs of non-linearity.

How to fix if this assumption is not followed?

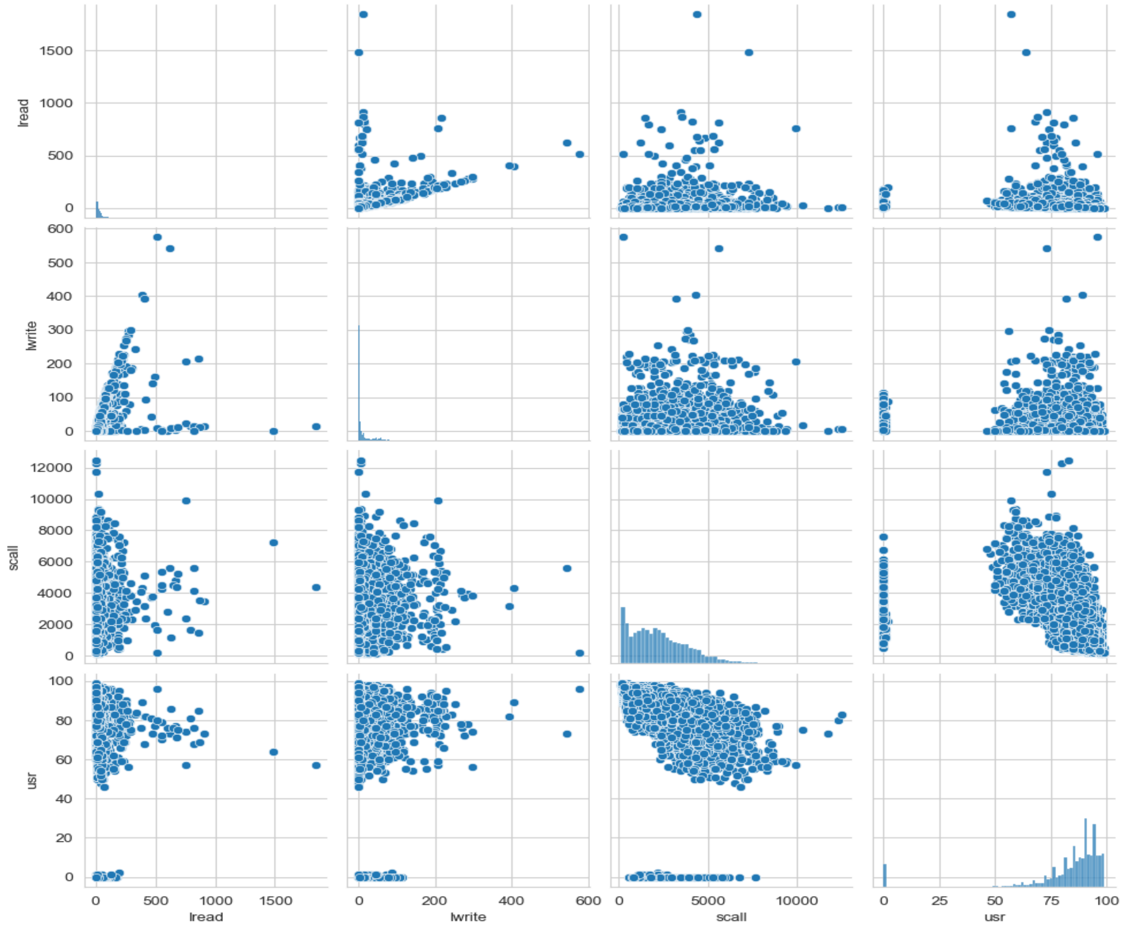
- We can try different transformations.
  Below plot shows the fitted and residual values of the regression model.



- We can observe a pattern in the residual vs fitted values, hence we will try to transform the continous variables in the data.

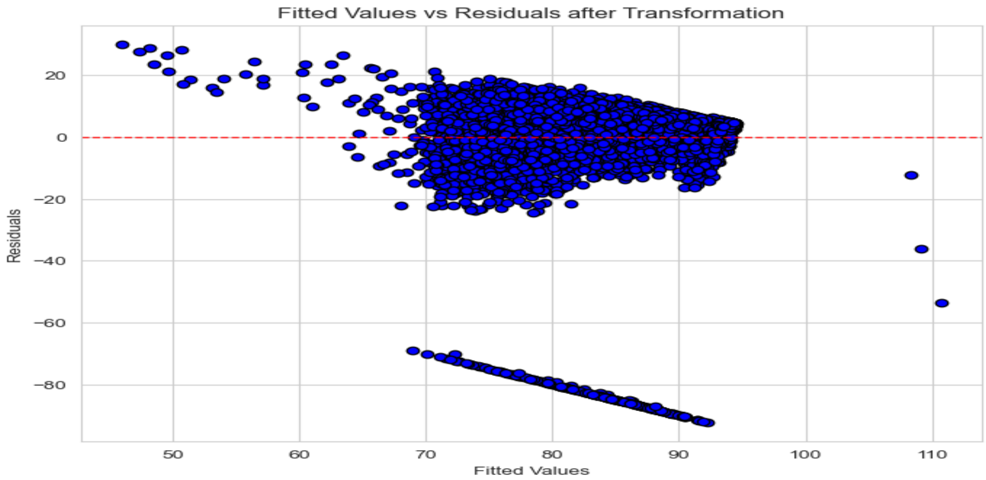
Pair plot to visualize the nonlinear relationship

Pair Plot



From the above Pair plot we can see 'scall, pflt and freeswap ' column has a slight nonlinear relationship with 'usr'. We can transform the scall, pflt and freeswap' variables by square the values and 3 new columns will be introduced to the dataset scall_sq, pflt_sq and freeswap_sq respectively.

Fitted vs residual after transformation



This transformation makes the model more effective which can be seen through R-squared: 0.890 and Adj R-squared 0.890.

TEST FOR NORMALITY

What is the test?

- Error terms/residuals should be normally distributed.
- If the error terms are not normally distributed, confidence intervals may become too wide or narrow. Once confidence interval becomes unstable, it leads to difficulty in estimating coefficients based on minimization of least squares.

What does non-normality indicate?

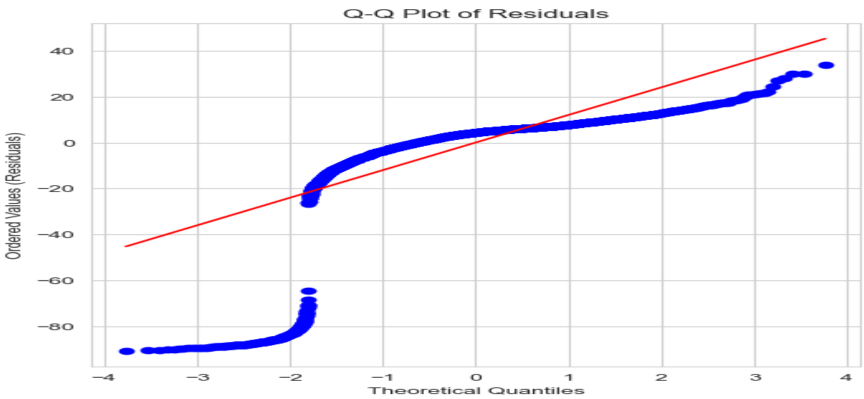
- It suggests that there are a few unusual data points which must be studied closely to make a better model.

How to check the Normality?

- It can be checked via QQ Plot - residuals following normal distribution will make a straight line plot, otherwise not.
- Another test to check for normality is the Shapiro-Wilk test.

How to Make residuals normal?

- We can apply transformations like log, exponential, arcsinh, etc as per our data.



- Major points are lying on the straight line in QQ plot

The Shapiro-Wilk test can also be used for checking the normality. The null and alternate hypotheses of the test are as follows:

- Null hypothesis - Data is normally distributed.
- Alternate hypothesis - Data is not normally distributed.

Shapiro-Wilk test result

Statistic = 0.9539812803268433
Pvalue = 3.8082710014370186e-39

- Since p-value < 0.05, the residuals are not normal as per shapiro test.
- Strictly speaking - the residuals are not normal. However, as an approximation, we might be willing to accept this distribution as close to being normal

Test For Homoscedasticity

- **Homoscedacity** - If the variance of the residuals are symmetrically distributed across the regression line , then the data is said to homoscedastic.
- **Heteroscedacity** - If the variance is unequal for the residuals across the regression line, then the data is said to be heteroscedastic. In this case the residuals can form an arrow shape or any other non symmetrical shape.

Why the test?

- The presence of non-constant variance in the error terms results in heteroscedasticity.
  Generally, non-constant variance arises in presence of outliers.

How to check if model has Heteroscedasticity?

- Can use the goldfeldquandt test. If we get p-value > 0.05 we can say that the residuals are homoscedastic, otherwise they are heteroscedastic.

How to deal with Heteroscedasticity?

- Can be fixed via adding other important features or making transformations.

The null and alternate hypotheses of the goldfeldquandt test are as follows:

- Null hypothesis : Residuals are homoscedastic
- Alternate hypothesis : Residuals have hetroscedasticity

HOMOSCEDASTICITY test result

F statistic = 1.0021079752518829
p-value = 0.4775741855501464

- Since p-value > 0.05 we can say that the residuals are homoscedastic.

Final model

All the assumptions of linear regression are now satisfied. Let's check the summary of our final model

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    usr   R-squared:                       0.112
Model:                            OLS   Adj. R-squared:                  0.111
Method:                 Least Squares   F-statistic:                     343.4
Date:                Sun, 05 Nov 2023   Prob (F-statistic):          4.55e-210
Time:                        19:35:57   Log-Likelihood:                -34997.
No. Observations:                8192   AIC:                         7.000e+04
Df Residuals:                    8188   BIC:                         7.003e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         92.6219      0.335    276.674      0.000      91.966      93.278
lread         -0.0226      0.004     -5.274      0.000      -0.031      -0.014
lwrite        -0.0199      0.008     -2.622      0.009      -0.035      -0.005
scall         -0.0034      0.000    -28.801      0.000      -0.004      -0.003
==============================================================================
Omnibus:                     6797.630   Durbin-Watson:                   2.014
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           125246.449
Skew:                          -4.111   Prob(JB):                         0.00
Kurtosis:                      20.302   Cond. No.                     4.94e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.94e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Observations

- R-squared of the model is 0.890 and adjusted R-squared is 0.890, which shows that the model is able to explain ~89% variance in the data. This is quite good.
- A unit increase in the freemem will result in a 0.0006 unit increase in the usr, all other variables remaining constant.
- The usr of a process of CPU_Bound will be -0.1209 units lesser than a process of Not_CPU_Bound, all other variables remaining constant.

Predictions

Model Parameters

```
Predictions:
0       85.198668
1       92.035939
2       84.770611
3       92.070409
4       91.351435
          ...
8187    81.649279
8188    87.030095
8189    81.419780
8190    73.147215
8191    89.181427
Length: 8192, dtype: float64
```

```
Model Parameters:
const     92.621924
lread     -0.022617
lwrite    -0.019904
scall     -0.003447
dtype: float64
```

Equation of linear regression

Equation of the linear regression line:

Y = (92.622)*const + (-0.023)*lread + (-0.02)*lwrite + (-0.003)*scall

Observations

- Freemem is the only positive feature which has a positive tendency towards usr (CPU runs in user mode). When Freemem unit increases chances of CPU runs in user mode increases.
- A unit increase in the freemem will result in a 0.0006 unit increase in the usr, all other variables remaining constant.
- The usr of a process of CPU_Bound will be -0.1209 units lesser than a process of Not_CPU_Bound, all other variables remaining constant.

Predictions on the test dataset

```
RMSE on the train data: 17.278
RMSE on the test data: 17.605
MAE on the train data: 8.555
MAE on the test data: 8.743
```

Observations

- We can see that RMSE on the train and test sets are comparable. So, our model is not suffering from overfitting.
- MAE indicates that our current model is able to predict usr within a mean error of 2.3 units on the test data.
- Hence, we can conclude the final model is good for prediction as well as inference purposes.

Inference

We constructed a number of models by removing variables one at a time in order to produce an effective model. By taking into account several aspects like R-squared, Adj R-squared, P value, and creating VIF, the variables are eliminated. On beforehand we have to clean up the data by handling the outliers and impute the missing values before moving on to the linear regression model. We have tried to build a Linear Regression without treating the outliers which gave us a very low R-squared value which shows the model is not efficient.

Linear Regression before Outlier treatment: R-squared and Adjusted R-squared value are 64.3% and 64.1% respectively. This value is considered has a very low score therefore we have moved over to build an effective liner regression model.

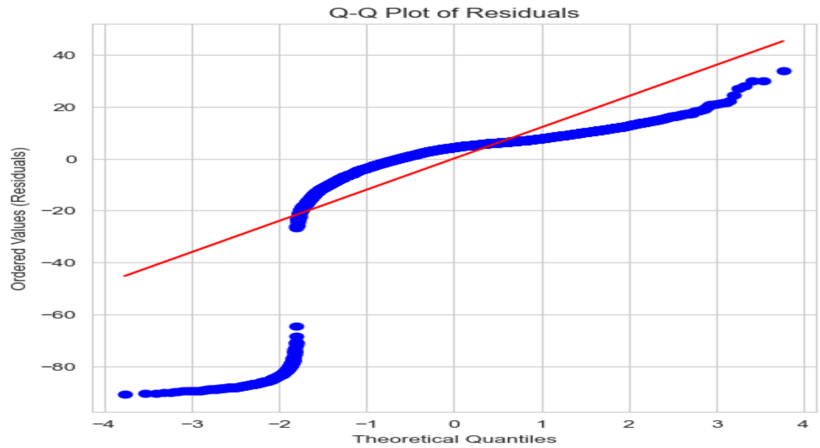
Below is the iteration we have gone to bring the linear regression model

Variables	R-squared	Adj R-squared
ppgout	0.796	0.795
vflt	0.796	0.795
pgin	0.796	0.795
sread	0.796	0.795
pgfree	0.796	0.795
fork	0.795	0.795
lwrite	0.795	0.794
swrite	0.794	0.793
exec	0.792	0.792
pgout	0.789	0.789
atch	0.789	0.789

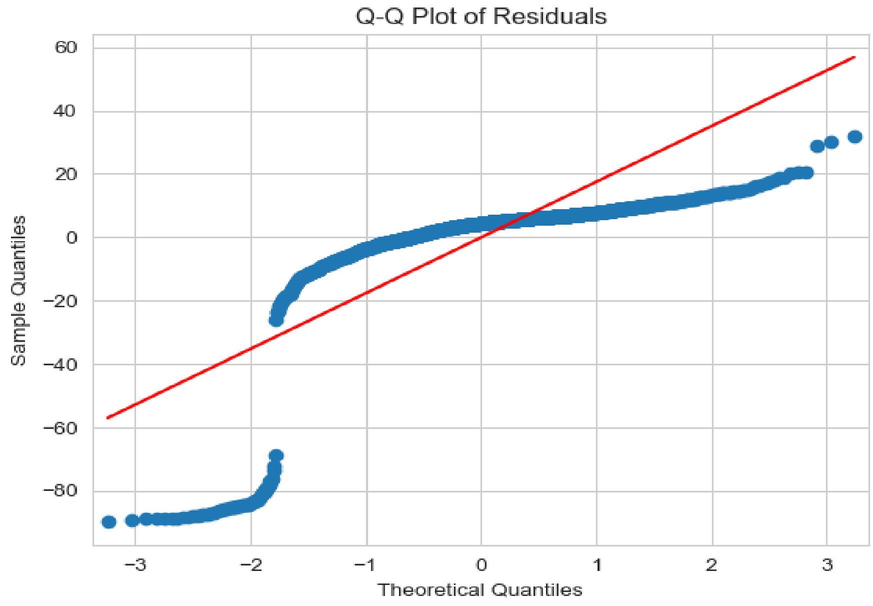
Variable column has the variables that we have dropped one by one, corresponding changes in R-squared and Adj R-squared are filled up beside to it. It shows we have started with 0.795 and ended with 0.789 even though it shows negative improvement in R-squared value, the reason we have choosen to drop the variables is they have Multicollinearity within the independent variables which effects the effective model therefore we have removed the variables which has Multicollinearity.

To improve the model we have transformed the scall, pflt and freeswap' variables by square the values and 3 new columns will be introduced to the dataset scall_sq, pflt_sq and freeswap_sq respectively. Which makes the model more effective and it can be measured by the R-squared: 0.890 and Adj R-squared 0.890 values.

For linear regression the residuals has to be in normal distributed, in our model the residuals are build up close to normal distribution form which make the model very effective. Shapiro test which helps to identify if the residuals are in normal distribution, p value (Pvalue = 3.8082710014370186e-39) on the Shapiro test is lesser than 0.05 therefore it is proved the residual is not normally distributed.



Homoscedasticity test is performed to check if the presence of non-constant variance in the error terms results in heteroscedasticity. The P-value on Homoscedasticity test is 0.4775741855501464 therefore the null hypothesis is rejected so that we can say that the residuals are homoscedastic.



QQ plot shows the majority of residuals are on the linear line. This is an evidence of an effective linear regression model.

Last but not the least, our model worked very well in both training and test data. This is tested using RMSE and MAE. RMSE on the train and test sets are comparable(Train data: 3.241 & Test data:3.298). Therefore, our model is not suffering from overfitting. MAE indicates that our current model is able to predict usr within a mean error of 2.3 units on the test data.

Recommendations:

usr – Portion of time (%) that CPUs run in user mode can be predicted using the below linear regression equation.

Recommendation:

usr = 1.500 + 0.300 * lread + 0.500 * lwrite + -0.100 * scall + 0.300 * lread + 0.500 * lwrite + -0.100 * scall

The dependent variable urs – Portion of time CPUs run in user mode rises as the following variables'

units fall

lread - Reads (transfers per second) between system memory and user memory

scall - Number of system calls of all types per second

rchar – Number of characters transferred per second by system read calls

wchar – Number of characters transferred per second by system write calls

ppgin – Number of pages paged in per second

pflt – Number of page faults caused by protection errors (copy on writes)

runqsz – Process run queue size

freemem – Number of memory pages available to user processes.

It has a non negative coefficient

freeswap – Number of disk blocks available for page swapping

Through this model, we advise that there is a greater likelihood of an increase in the amount of time CPUs are used in user mode when the aforementioned factors are used sparingly.

Problem 2: Logistic Regression and LDA (linear discriminant analysis) and CART

Executive Summary

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.

Introduction

The purpose of this whole exercise is to explore the dataset and predict do/don't they use acontraceptive method of choice based on their demographic and socio-economic

characteristics. The data consists of various features of Contraceptive Prevalence Survey

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 dataset:

```
   Wife_age Wife_ education Husband_education  No_of_children_born  \
0      24.0         Primary         Secondary                  3.0
1      45.0      Uneducated         Secondary                 10.0
2      43.0         Primary         Secondary                  7.0
3      42.0       Secondary           Primary                  9.0
4      36.0       Secondary         Secondary                  8.0

  Wife_religion Wife_Working  Husband_Occupation Standard_of_living_i
ndex  \
0   Scientology           No                   2
High
1   Scientology           No                   3                 Very
High
2   Scientology           No                   3                 Very
High
3   Scientology           No                   3
High
4   Scientology           No                   3
Low

  Media_exposure  Contraceptive_method_used
0        Exposed                         No
1        Exposed                         No
2        Exposed                         No
3        Exposed                         No
4        Exposed                         No
```

Exploratory Data Analysis (EDA)

Let us check the types of variables in the data frame and the null values

```
Variable Types:
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
```

5 Point Summary

5 Point Summary

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Wife_age	1393.0	NaN	NaN	NaN	32.55967	8.087315	16.0	26.0	32.0	38.0	49.0
Wife_ education	1393	4	Tertiary	515	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Husband_education	1393	4	Tertiary	827	NaN	NaN	NaN	NaN	NaN	NaN	NaN
No_of_children_born	1393.0	NaN	NaN	NaN	3.280931	2.345425	0.0	1.0	3.0	5.0	11.0
Wife_religion	1393	2	Scientology	1186	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Wife_Working	1393	2	No	1043	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Husband_Occupation	1393.0	NaN	NaN	NaN	2.174444	0.85459	1.0	1.0	2.0	3.0	4.0
Standard_of_living_index	1393	4	Very High	618	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Media_exposure	1393	2	Exposed	1284	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Contraceptive_method_used	1393	2	Yes	779	NaN	NaN	NaN	NaN	NaN	NaN	NaN

- All the numerical columns have numerical values alone.
- From the above 5 point summary we can observer there are 80 duplicate rows in the data set. Count for each feature is shown has 1393 wherein total record in the data set is 1473

Null value check

```
Null Values:
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
```

Wife_age and No_of_children_born has null values which will be treated with the mean value.

Post to Null treatment:

```
Wife_age                     0
Wife_ education              0
Husband_education            0
No_of_children_born          0
Wife_religion                0
Wife_Working                 0
Husband_Occupation           0
Standard_of_living_index     0
Media_exposure               0
Contraceptive_method_used    0
```

Duplicate Check

Our dataset has 80 duplicate rows. All the duplicate rows are removed from the dataset.

Getting unique counts of all Objects

Wife_ education

Tertiary 515

Secondary 398

Primary 330

Uneducated 150

Name: Wife_ education, dtype: int64

Husband_education

Tertiary 827

Secondary 347

Primary 175

Uneducated 44

Name: Husband_education, dtype: int64

Wife_religion

Scientology 1186

Non-Scientology 207

Name: Wife_religion, dtype: int64

Wife_Working

No 1043

Yes 350

Name: Wife_Working, dtype: int64

Standard_of_living_index

Very High 618

High 419

Low 227

Very Low 129

Name: Standard_of_living_index, dtype: int64

Media_exposure

Exposed 1284

Not-Exposed 109

Name: Media_exposure , dtype: int64

Contraceptive_method_used

Yes 779

No 614

Name: Contraceptive_method_used, dtype: int64

Outlier Check

Outlier check

Outlier check

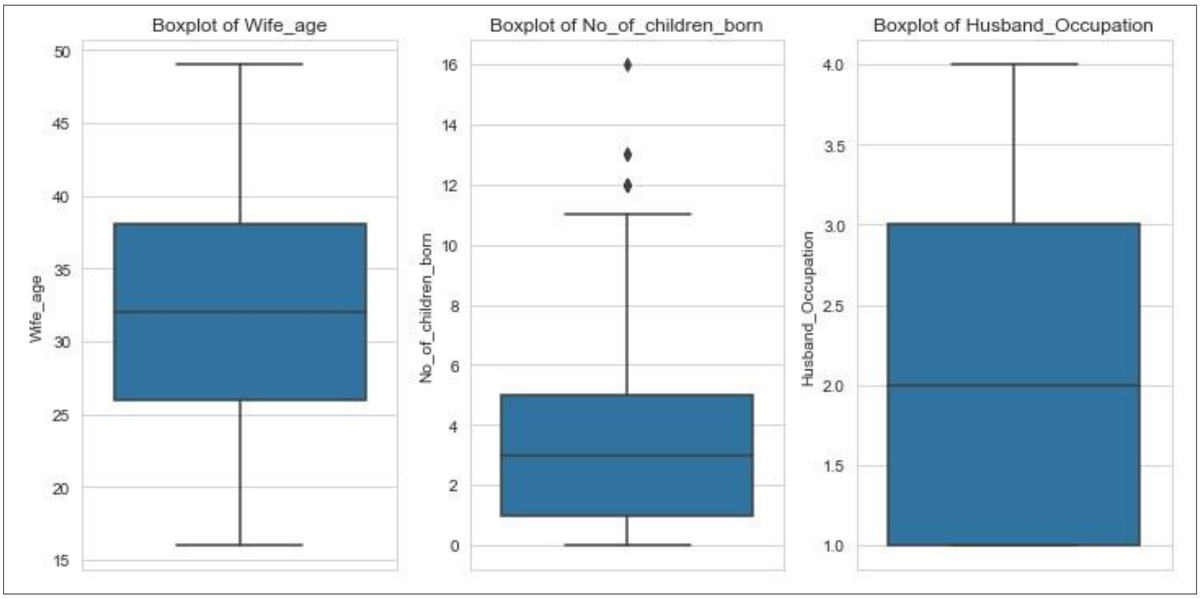
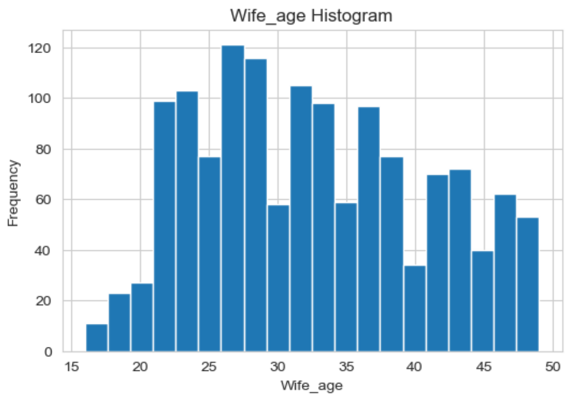


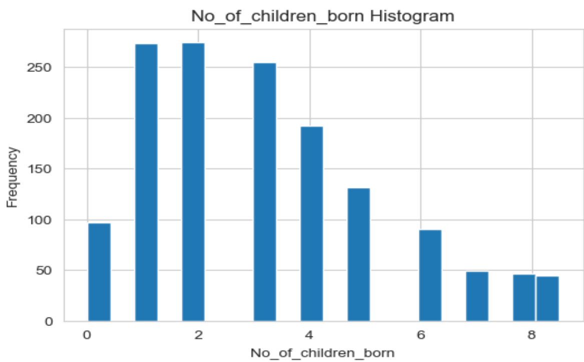
Figure 18 - Outlier check

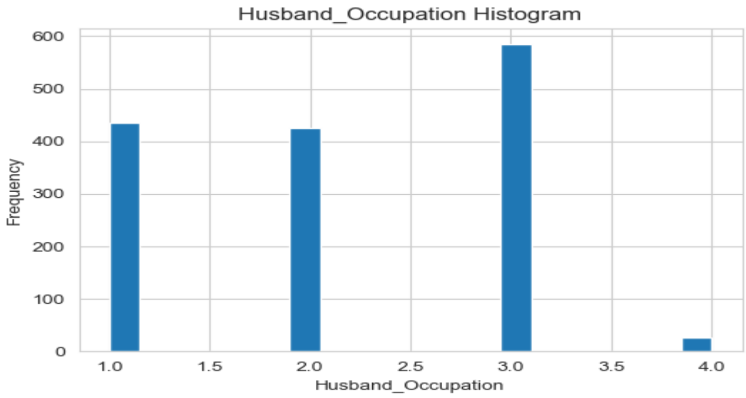
- It is evident that No_of_children_born feature has outliers
- Outliers will be treated using the IQR technique

Post to Outlier treatment

Post to Outlier check

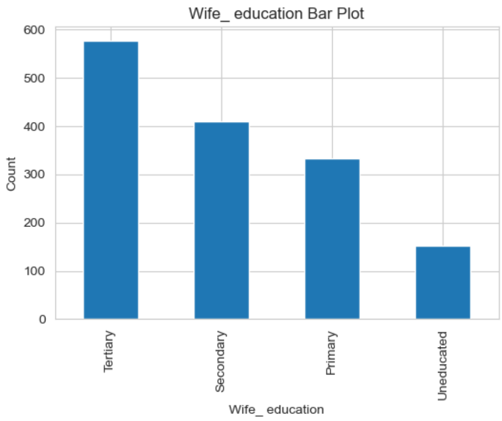


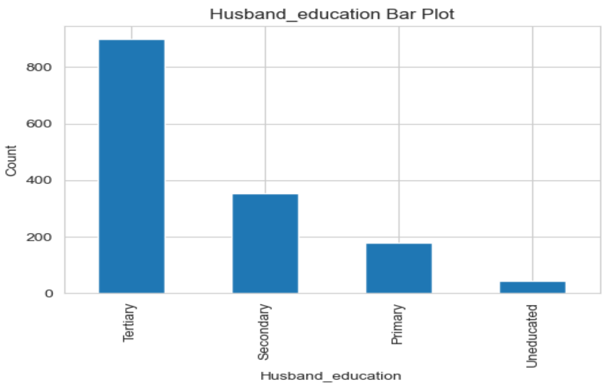


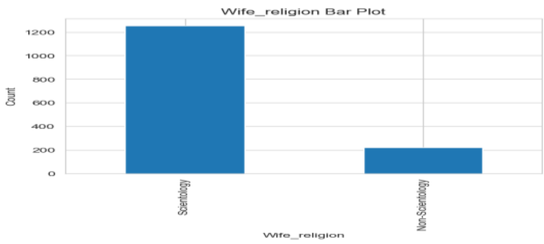


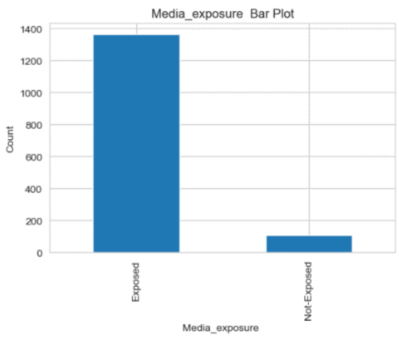
Univariate Analysis:

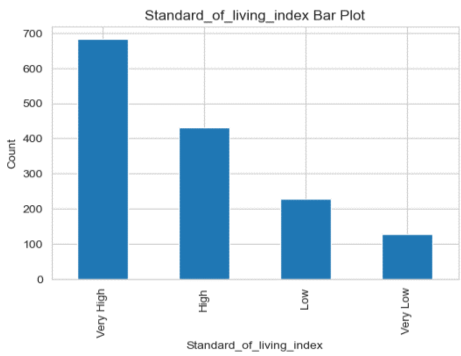
- All the variables are neatly distributed

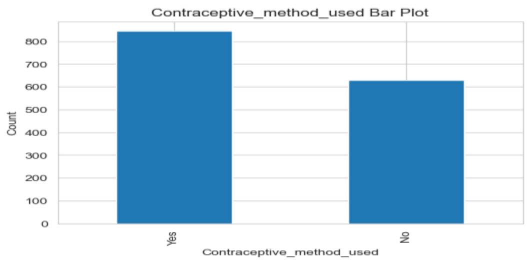






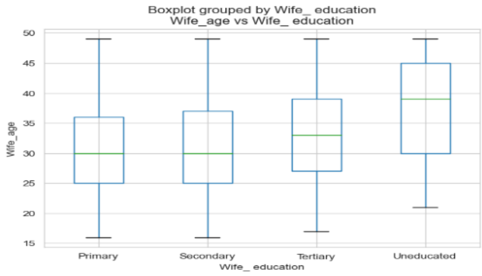


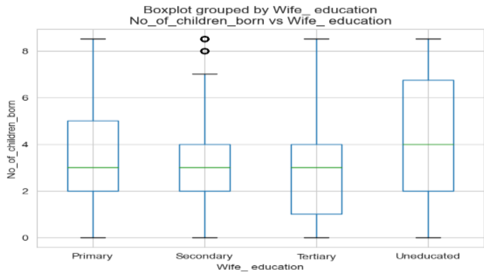




Bivariate Analysis

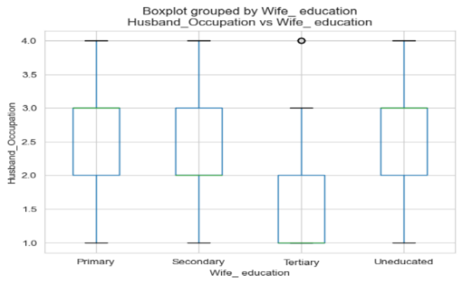
ure size 600x400 with 0 Axes>

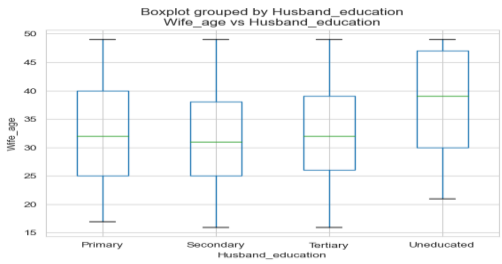




<Figure size 600x400 with 0 Axes>

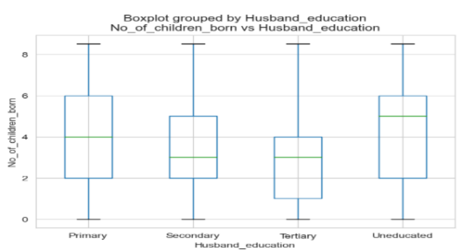
<Figure size 600x400 with 0 Axes>

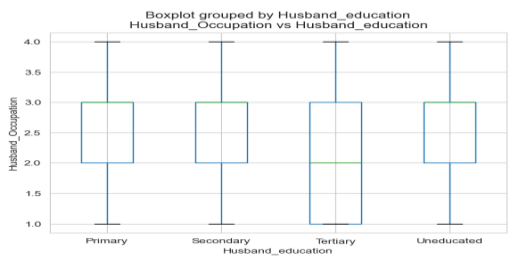




<Figure size 600x400 with 0 Axes>

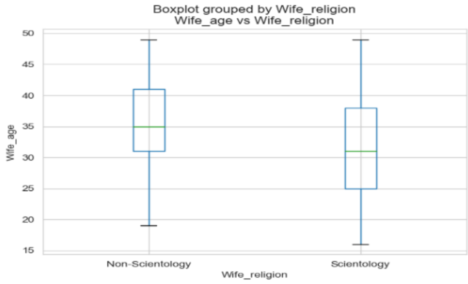
<Figure size 600x400 with 0 Axes>

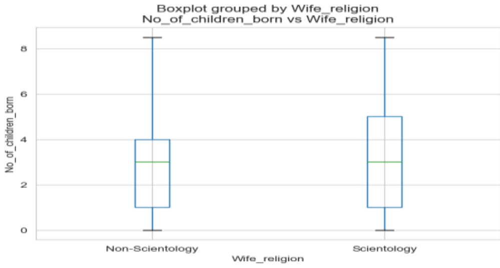




<Figure size 600x400 with 0 Axes>

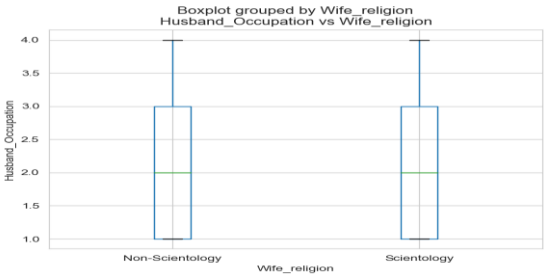
<Figure size 600x400 with 0 Axes>

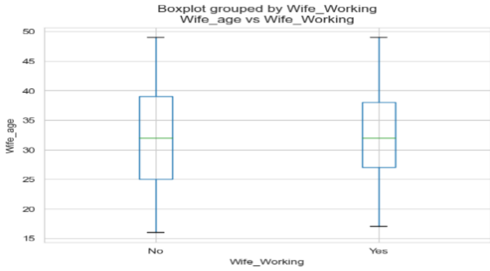




<Figure size 600x400 with 0 Axes>

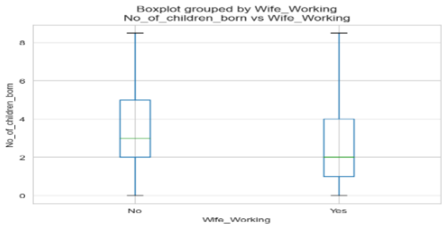
<Figure size 600x400 with 0 Axes>

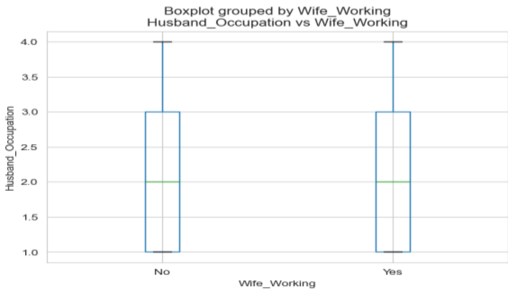




<Figure size 600x400 with 0 Axes>

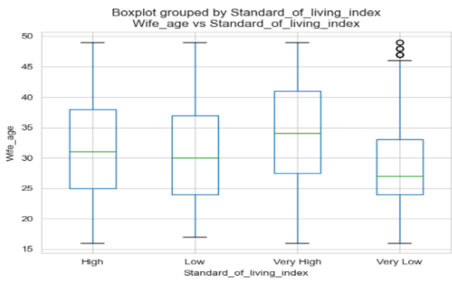
<Figure size 600x400 with 0 Axes>

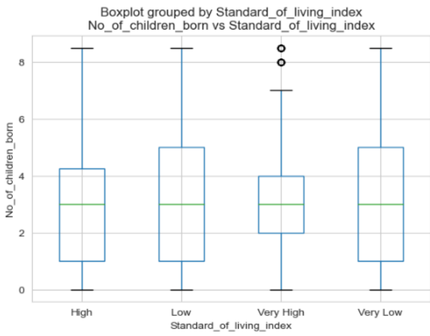




<Figure size 600x400 with 0 Axes>

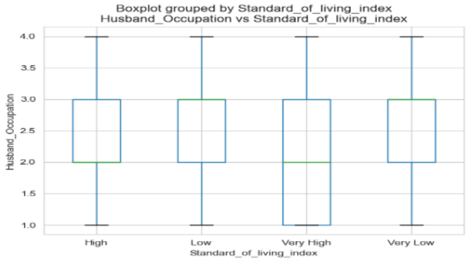
<Figure size 600x400 with 0 Axes>

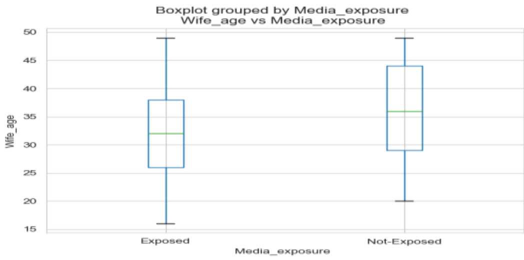




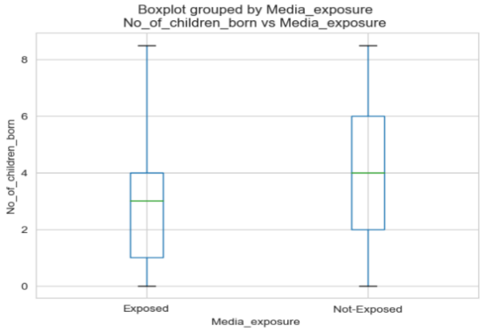
<Figure size 600x400 with 0 Axes>

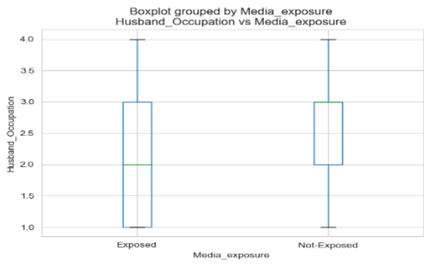
<Figure size 600x400 with 0 Axes>



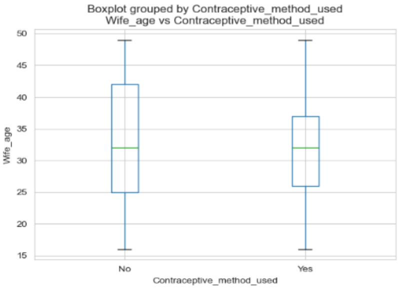


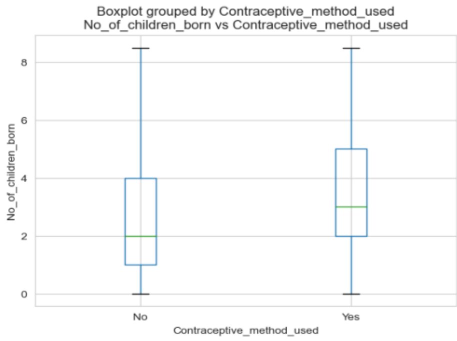

```
<Figure size 600x400 with 0 Axes>

<Figure size 600x400 with 0 Axes>
```



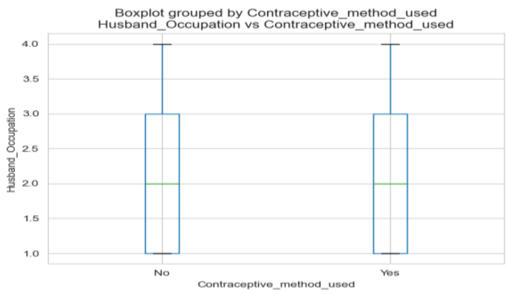



```
<Figure size 600x400 with 0 Axes>

<Figure size 600x400 with 0 Axes>
```






```
<Figure size 600x400 with 0 Axes>

<Figure size 600x400 with 0 Axes>
```



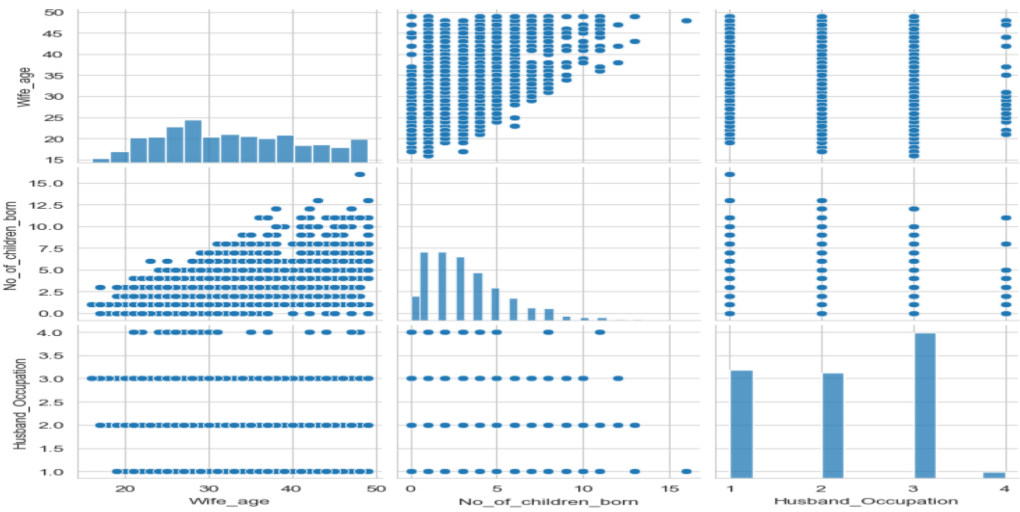
In [553]:

```python
import pandas as pd
```

- All the variables are neatly distributed

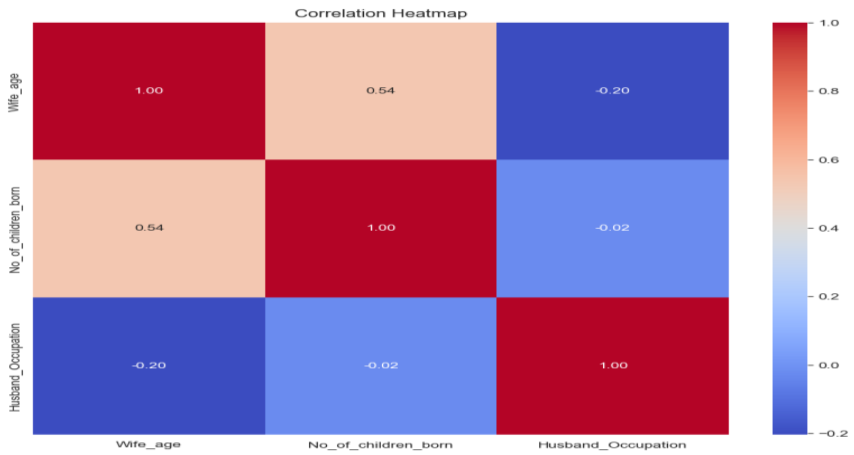
Multivariate Analysis

Pair Plot



- There is no variance in the depth of variables, scattered data will help models to perform well
- Each variable has equivalent contribution of Contraceptive_method_used dependent variable

Heat Map



- Wife_age vs No_of_children_born has correlation of 53%

Data Encode

Data has been encoded for the given dataset which enable us to use the data for different models like Logistic Regression, LDA and CART

Contraceptive_method_used has two unique values “Yes” and “No”, these values are encoded to “0”and “1” respectively.

Before Encoding:

```
Before Encoding:
   Wife_age Wife_ education Husband_education  No_of_children_born  \
0      24.0         Primary         Secondary                  3.0
1      45.0      Uneducated         Secondary                 10.0
2      43.0         Primary         Secondary                  7.0
3      42.0       Secondary           Primary                  9.0
4      36.0       Secondary         Secondary                  8.0

  Wife_religion Wife_Working  Husband_Occupation Standard_of_living_i
ndex  \
0   Scientology           No                   2
High
1   Scientology           No                   3               Very
High
2   Scientology           No                   3               Very
High
3   Scientology           No                   3
High
4   Scientology           No                   3
Low

  Media_exposure  Contraceptive_method_used
0        Exposed                        No
1        Exposed                        No
2        Exposed                        No
3        Exposed                        No
4        Exposed                        No
```

After Encoding:

```
After Encoding:
   Wife_age  Wife_ education  Husband_education  No_of_children_born
\
0      24.0                0                  1                  3.0
1      45.0                3                  1                 10.0
2      43.0                0                  1                  7.0
3      42.0                1                  0                  9.0
4      36.0                1                  1                  8.0

   Wife_religion  Wife_Working  Husband_Occupation  Standard_of_livin
g_index  \
0              1             0                   2
0
1              1             0                   3
2
2              1             0                   3
2
3              1             0                   3
0
4              1             0                   3
1

   Media_exposure  Contraceptive_method_used
0               0                          0
1               0                          0
2               0                          0
3               0                          0
4               0                          0
```

Split Data

Contraceptive_method_used variable has taken has a y variable (dependent variable) and all other variables are taken has x variable (independent variable).

The given data set is split into 70:30; 70% data are consider has training data and 30% of data are taken for testing the model.

X_train dataset for training the model; 8 columns with 975 rows

```
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Wife_age                            975 non-null    float64
 1   No_of_children_born                 975 non-null    float64
 2   Husband_Occupation                  975 non-null    float64
 3   Wife  education Secondary           975 non-null    uint8
 4   Wife_ education_Tertiary            975 non-null    uint8
 5   Wife_ education_Uneducated          975 non-null    uint8
 6   Husband_education_Secondary         975 non-null    uint8
 7   Husband_education_Tertiary          975 non-null    uint8
```

```
8   Husband_education_Uneducated      975 non-null    uint8
9   Wife_religion_Scientology         975 non-null    uint8
10  Wife_Working_Yes                  975 non-null    uint8
11  Standard_of_living_index_Low      975 non-null    uint8
12  Standard_of_living_index_Very High 975 non-null   uint8
13  Standard_of_living_index_Very Low 975 non-null    uint8
14  Media_exposure _Not-Exposed       975 non-null    uint8
```

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure
336	34.0	2	1	0.0	0	1	3.0	0	0
781	37.0	2	2	3.0	1	0	2.0	2	0
433	37.0	2	2	2.0	1	1	3.0	0	0
588	29.0	2	2	2.0	1	0	3.0	2	0
468	24.0	3	2	1.0	1	1	4.0	1	1

X_test dataset for testing the model; 8 columns with 418 rows

```
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Wife age                            418 non-null    float64
 1   No_of_children_born                 418 non-null    float64
 2   Husband_Occupation                  418 non-null    float64
 3   Wife_ education_Secondary           418 non-null    uint8
 4   Wife_ education_Tertiary            418 non-null    uint8
 5   Wife_ education_Uneducated          418 non-null    uint8
 6   Husband education Secondary         418 non-null    uint8
 7   Husband education Tertiary          418 non-null    uint8
 8   Husband_education_Uneducated        418 non-null    uint8
 9   Wife_religion_Scientology           418 non-null    uint8
 10  Wife_Working_Yes                    418 non-null    uint8
 11  Standard_of_living_index_Low        418 non-null    uint8
 12  Standard of living index Very High  418 non-null    uint8
 13  Standard of living index Very Low   418 non-null    uint8
 14  Media_exposure _Not-Exposed         418 non-null    uint8
```

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure
1012	29.000000	1	2	4.0	1	0	1.0	2	0
446	39.000000	2	2	3.0	1	0	1.0	2	0
909	31.000000	1	1	3.0	1	0	3.0	3	1
1400	32.606277	1	2	4.0	1	0	3.0	2	0
486	38.000000	2	2	6.0	1	1	3.0	0	0

Apply Models

Logistic Regression Model

Using logistic regression we are trying to predict the dependent variable, logistic regression is used in predicting the categorical dependent variable. To perform the regression model the data set has to be all numeric, to achieve this we have encoded all the object data in the dataset to numeric.

Logistic Regression Model Score: 0.6748717948717948

As shown above we have obtained 67.5 as a Logistic Regression Model Score

AUC on the training: 0.708

AUC on the test: 0.708

AUC chart Train vs Test

AUC chart Train vs Test

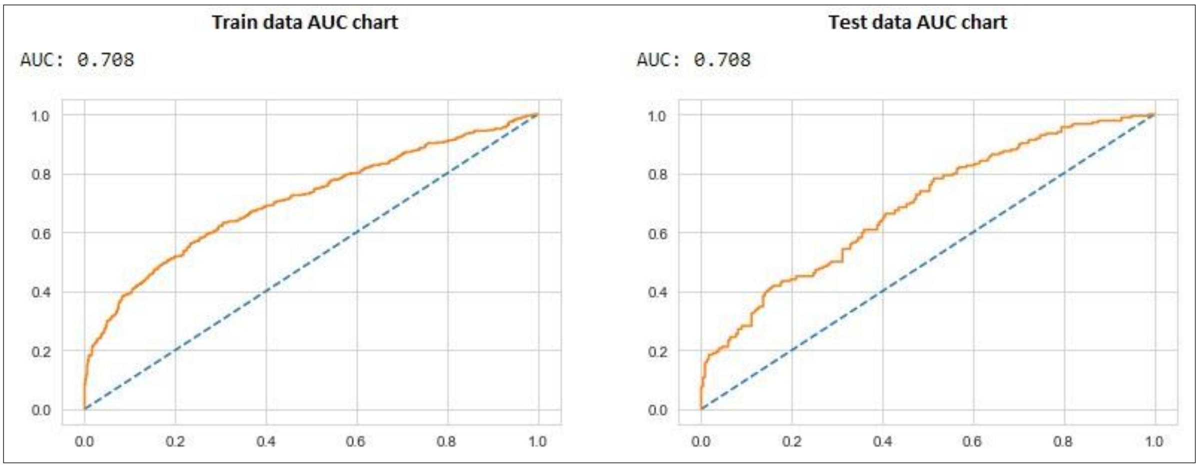


Figure 24 - Logistic AUC

From the Figure 24 we could clearly visualize Logistic regression model is performed well in both the Train and Test data

Confusion Matrix Train vs Test

Confusion Matrix Train vs Test

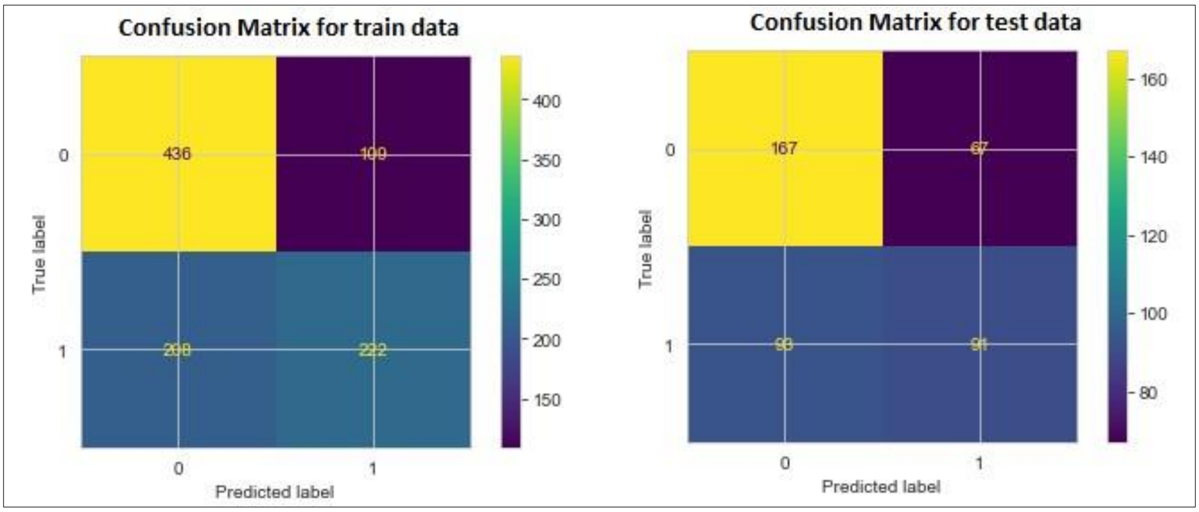


Figure 25 - Logistic Confusion Matrix

Observation:

Value 0 indicates Contraceptive_method_used=No

Value 1 indicates Contraceptive_method_used=Yes

Inference from Train data

- 436 is True Positive; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used No is predicted as No
- 222 is True Negative; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used Yes is predicted as Yes
- 200 is False Positive; this denotes cases where actual class was negative (0) but predicted as positive (1)
- 100 is False Negative; this denotes cases where actual class was positive (1) but predicted as negative (0)

Inference from Test data

- 167 is True Positive; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used No is predicted as No
- 91 is True Negative; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used Yes is predicted as Yes
- 90 is False Positive; this denotes cases where actual class was negative (0) but predicted as positive (1)
- 67 is False Negative; this denotes cases where actual class was positive (1) but predicted as negative (0)

Classification Report

Train Data set:

	precision	recall	f1-score	support
0	0.68	0.80	0.73	545
1	0.67	0.52	0.58	430
accuracy			0.67	975
macro avg	0.67	0.66	0.66	975
weighted avg	0.67	0.67	0.67	975

Test Data set:

	precision	recall	f1-score	support
0	0.64	0.71	0.68	234
1	0.58	0.49	0.53	184
accuracy			0.62	418
macro avg	0.61	0.60	0.60	418
weighted avg	0.61	0.62	0.61	418

LDA Model

Linear Discriminant Function

= -1.2982693 + (0.06 * Wife_age) + (-0.22 * No_of_children_born) + (-0.04 * Husband_Occupation)

+ (-0.43 * Wife_education_Secondary) + (-0.97 * Wife_education_Tertiary)

+ (-0.06 * Wife_education_Uneducated) + (0.1 * Husband_education_Secondary)

+ (0.09 * Husband_education_Tertiary) + (0.6 * Husband_education_Uneducated)

+ (0.12 * Wife_religion_Scientology) + (0.11 * Wife_Working_Yes)

+ (0.07 * Standard_of_living_index_Low) + (-0.28 * Standard_of_living_index_VeryHigh)

+ (0.46 * Standard_of_living_index_VeryLow) + (0.32 * Media_exposure_Not-Exposed)

+ (0.67 * Prediction)

With LDA model we came up with the above equation; equation starts with constant and all the variable

have their coefficient. Based on the value of coefficient the variable contributes on prediction of y

(dependent) variable.

T raining Data and Test Data Confusion Matrix Comparison

Confusion Matrix

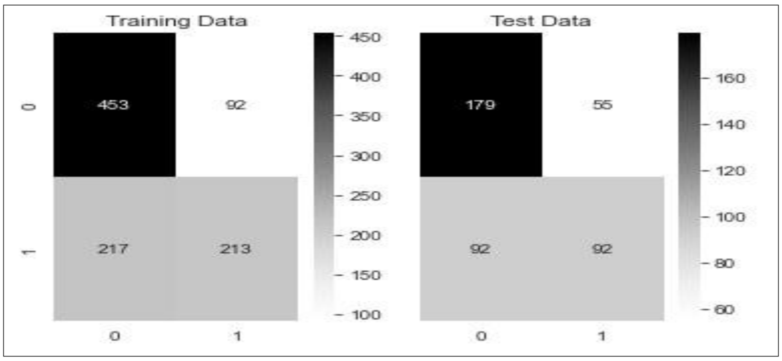


Figure 26 - Confusion Matrix LDA

Observation:

Value 0 indicates Contraceptive_method_used=No

Value 1 indicates Contraceptive_method_used=Yes

Inference from Train data

- 453 is True Positive; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used No is predicted as No
- 213 is True Negative; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used Yes is predicted as Yes
- 217 is False Positive; this denotes cases where actual class was negative (0) but predicted as positive (1)
- 92 is False Negative; this denotes cases where actual class was positive (1) but predicted as negative (0)

Inference from Test data

- 179 is True Positive; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used No is predicted as No
- 92 is True Negative; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used Yes is predicted as Yes
- 92 is False Positive; this denotes cases where actual class was negative (0) but predicted as positive (1)
- 55 is False Negative; this denotes cases where actual class was positive (1) but predicted as negative (0)

Classification Report

Classification Report of the training data:

	precision	recall	f1-score	support
0	0.68	0.83	0.75	545
1	0.70	0.50	0.58	430
accuracy			0.68	975
macro avg	0.69	0.66	0.66	975
weighted avg	0.69	0.68	0.67	975

Classification Report of the test data:

	precision	recall	f1-score	support
0	0.66	0.76	0.71	234
1	0.63	0.50	0.56	184
accuracy			0.65	418
macro avg	0.64	0.63	0.63	418
weighted avg	0.65	0.65	0.64	418

AUC chart

Figure 27 - LDA AUC Chart

From the Figure 24 we could clearly visualize LDA model is performed well in both the Train and Test data

CART Model

Performed CART model to predict the dependent variable, in our data set “Contraceptive method used”is the dependent variable, where other variables are used to predict “Contraceptive method used“

Features	Coffecient
Wife_age	0.324850
No of children born	0.249632
Husband Occupation	0.095042
Standard_of_living_index_Very High	0.061963
Wife_ education_Tertiary	0.052172
Wife_Working_Yes	0.040333
Wife_religion_Scientology	0.031388
Standard of living index Low	0.025066
Wife education Secondary	0.023623
Husband_education_Secondary	0.022700
Husband_education_Tertiary	0.021561
Wife_ education_Uneducated	0.018615
Standard_of_living_index_Very Low	0.017410
Media exposure Not-Exposed	0.008938
Husband_education_Uneducated	0.006707

In CART model we could clearly see the entire coefficients are positive. The unit increase in the independent variable likely truns to be a positive impact to dependent variable.

AUC chart Train vs Test

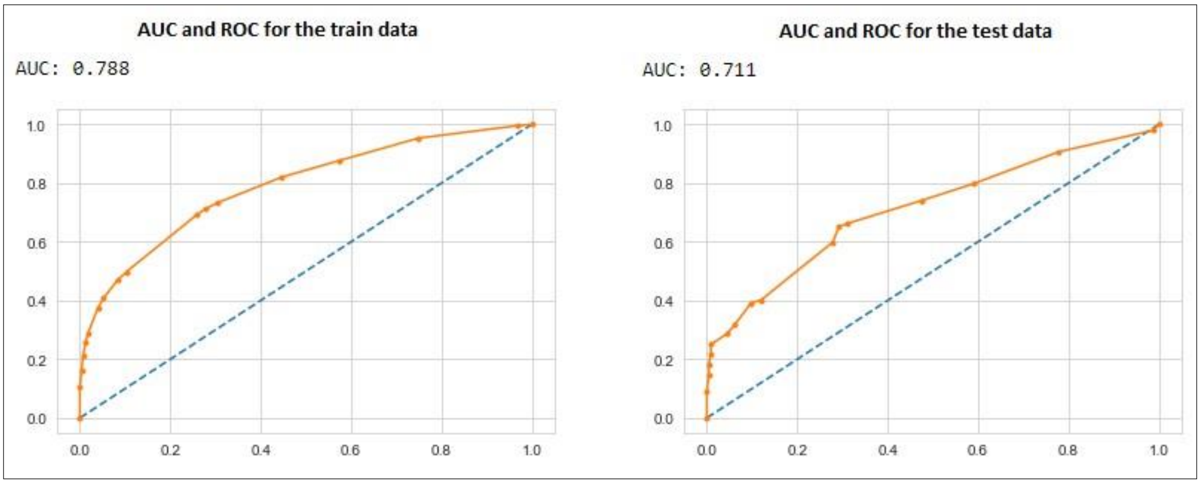


Figure 28 - AUC chart Train vs Test for CART

From the Figure 28 we could clearly visualize CART model is performed well in both the Train and Test data

Classification Report

Train Data set:

	precision	recall	f1-score	support
0	0.70	0.90	0.79	553
1	0.79	0.50	0.61	422
accuracy			0.72	975
macro avg	0.74	0.70	0.70	975
weighted avg	0.74	0.72	0.71	975

Accuracy of train data set: 0.72

Test Data set:

	precision	recall	f1-score	support
0	0.63	0.88	0.74	226
1	0.74	0.40	0.52	192
accuracy			0.66	418
macro avg	0.69	0.64	0.63	418
weighted avg	0.68	0.66	0.64	418

Accuracy of test data set: 0.66

Training Data and Test Data Confusion Matrix Comparison

Figure 29 - Confusion Matrix for CART

Observation:

Value 0 indicates Contraceptive_method_used=No

Value 1 indicates Contraceptive_method_used=Yes

Inference from Train data

- 391 is True Positive; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used No is predicted as No
- 113 is True Negative; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used Yes is predicted as Yes
- 317 is False Positive; this denotes cases where actual class was negative (0) but predicted as positive (1)
- 154 is False Negative; this denotes cases where actual class was positive (1) but predicted as negative (0)

Inference from Test data

- 179 is True Positive; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used No is predicted as No
- 92 is True Negative; this denotes cases where the actual calss of the data point and the predicted is same; Contraceptive method used Yes is predicted as Yes
- 92 is False Positive; this denotes cases where actual class was negative (0) but predicted as positive (1)
- 55 is False Negative; this denotes cases where actual class was positive (1) but predicted as negative (0)

Model Comparison

Logistic regression, LDA and CART models are thoroughly explained in the before sections. We are here to compare the all 3 models and identify which make more sense with respect you predicting dependent variable (Contraceptive method used).

Comparison Chart

	Logistic Regression		LDA		CART	
	Train	Test	Train	Test	Train	Test
AUC	0.708	0.708	0.708	0.685	0.788	0.711
Accuracy	0.67	0.62	0.68	0.65	0.72	0.66
precision 0	0.68	0.64	0.68	0.66	0.7	0.63
precision 1	0.67	0.58	0.7	0.63	0.79	0.74
recall 0	0.8	0.71	0.83	0.76	0.9	0.88
recall 1	0.52	0.49	0.5	0.5	0.5	0.4
f1-score 0	0.73	0.68	0.75	0.71	0.79	0.74
f1-score 1	0.58	0.53	0.58	0.56	0.61	0.52

Figure 30 helps us to understand how each models came out with the important component like AUC, Accuracy, precision, recall,f1-score. Logistic regression performed well on predicting the dependent variable, but when it is compared with LDA and CART model it shows lesser performance in both train and test data. LDA performed well than Logistic Regression, in precision 0 both Logistic and LDA model outcome are same. Accuracy of LDA is much better than Logistic Regression.

CART performed well than other models.

- CART has highest values in most of the criteria
- Highest Accuracy score 0.72
- Top score 0.79 in precision 1
- Top score 0.9 in recall 0
- Top score 0.79 in f1-score 0
- Performed well in both train and test data

Inference

We constructed three different models Logistic regression, LDA and CART models to predict Contraceptive method used dependent variable. By taking into account several aspects like coefficient, AUC, Accuracy, precision, recall, f1-score we were able to compare models between them. On beforehand we did the encoding so make sure the data are ready to build the Logistic regression, LDA and CART models. Outliers are treated and object variables are encoded to convert it to numeric variable.

As explained in the model comparison CART model performed well than the other models. This is evident by reviewing the Figure Below is the coffecient values from CART model

Features	Coffecient
Wife age	0.324850
No_of_children_born	0.249632
Husband_Occupation	0.095042
Standard_of_living_index_Very High	0.061963
Wife_ education_Tertiary	0.052172
Wife_Working_Yes	0.040333
Wife religion Scientology	0.031388
Standard_of_living_index_Low	0.025066
Wife_ education_Secondary	0.023623
Husband_education_Secondary	0.022700
Husband_education_Tertiary	0.021561
Wife_ education_Uneducated	0.018615
Standard of living index Very Low	0.017410
Media exposure Not-Exposed	0.008938
Husband_education_Uneducated	0.006707

Where we have highest Coffecient that variable is the main contributor in predicting dependent variable. In our case Contraceptive method used is the dependent variable all other variables are independent variable. All the variable has positive Coffecient, this shows where there is a unit increase in the independent variable, dependent variable has the impact of Coffecient times. For an example

- Wife age unit increase impact the Contraceptive method used by 0.33 times No of children bornage unit increase impact the Contraceptive method used by 0.25 times