PREDICTIVE MODELLING PROJECT

JAYA PREETHI R M

04 JUNE 2023

TABLE OF CONTENT

TABLE OF CONTENT - DIAGRAMS & 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.

Data Dictionary:

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.)

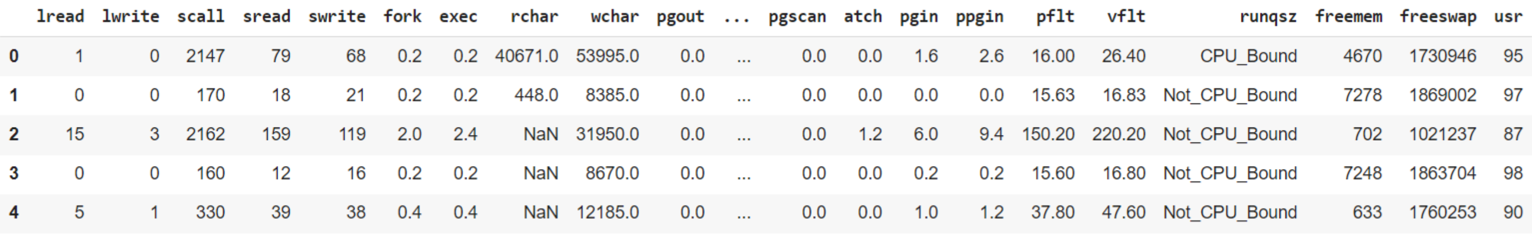
freemem - Number of memory pages available to user processes

freeswap - Number of disk blocks available for page swapping.

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

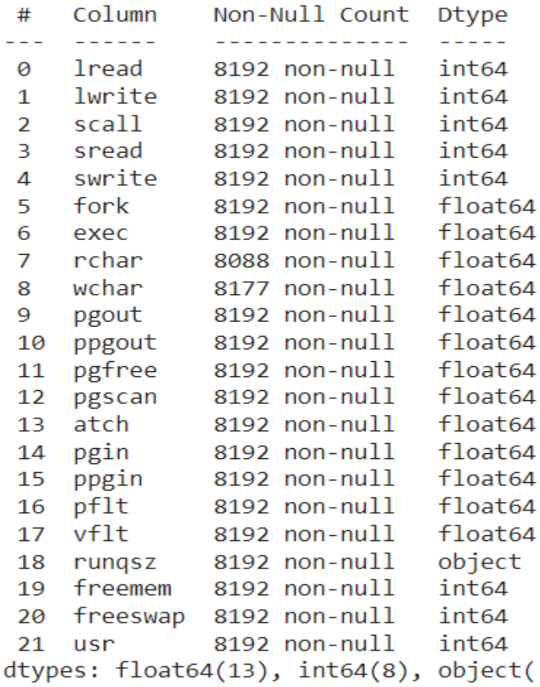
1.1 Read the data and do exploratory data analysis. Describe the data briefly. (Check the Data types, shape, EDA, 5 point summary). Perform Univariate, Bivariate Analysis, Multivariate Analysis.

- There are 22 variables / 22 columns and 8192 entries / rows in total. Our main variable or dependent variable is usr (Portion of time that cpus run in user mode in percentages).
- Below given is the head of the dataset:



	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
0	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
1	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
2	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
3	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
4	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

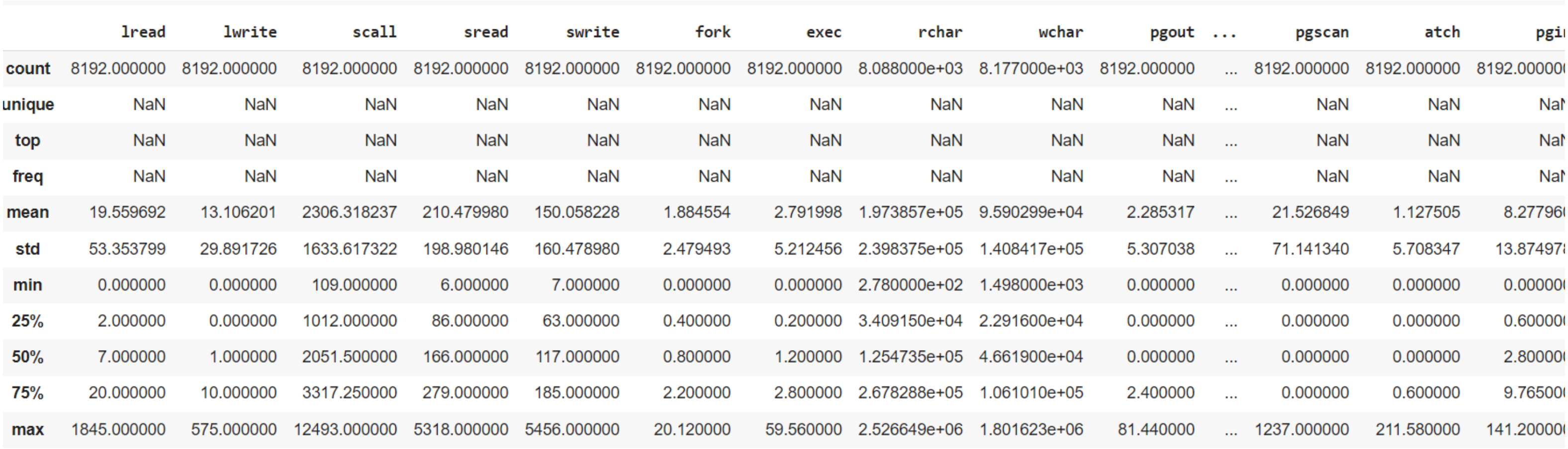
- Below given is the information table:



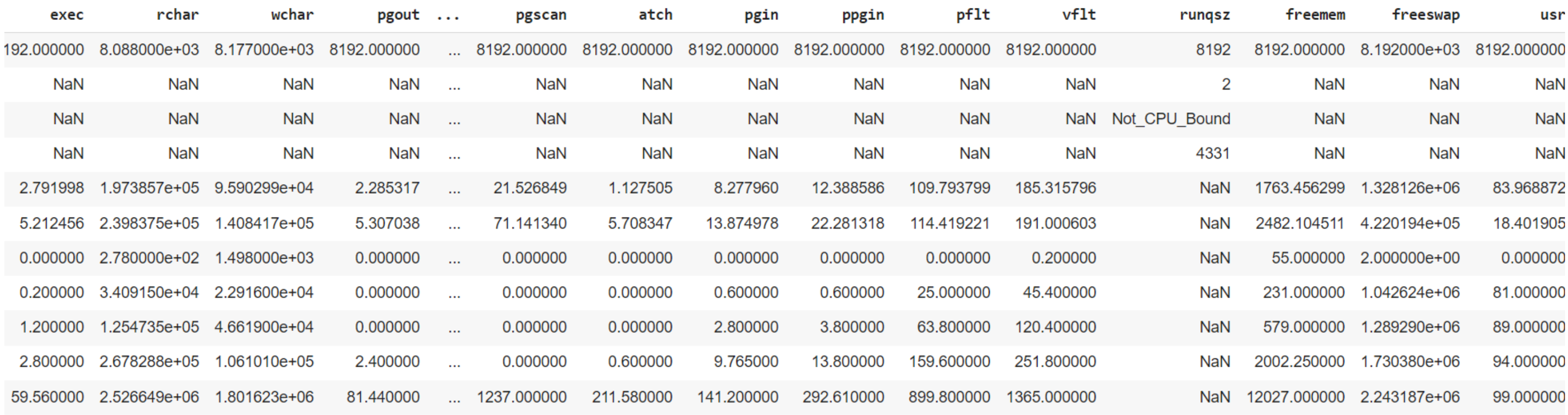
```
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   lread     8192 non-null   int64  
 1   lwrite    8192 non-null   int64  
 2   scall     8192 non-null   int64  
 3   sread     8192 non-null   int64  
 4   swrite    8192 non-null   int64  
 5   fork      8192 non-null   float64
 6   exec      8192 non-null   float64
 7   rchar     8088 non-null   float64
 8   wchar     8177 non-null   float64
 9   pgout     8192 non-null   float64
 10  ppgout    8192 non-null   float64
 11  pgfree    8192 non-null   float64
 12  pgscan    8192 non-null   float64
 13  atch      8192 non-null   float64
 14  pgin      8192 non-null   float64
 15  ppgin     8192 non-null   float64
 16  pflt      8192 non-null   float64
 17  vflt      8192 non-null   float64
 18  runqsz    8192 non-null   object 
 19  freemem   8192 non-null   int64  
 20  freeswap  8192 non-null   int64  
 21  usr       8192 non-null   int64  
dtypes: float64(13), int64(8), object(
```

From the table we can see that most of the variables are numerical and continuous. Except for runqsz variable which is an object type categorical variable.

- Below given is the descriptive details of the dataset in a table:

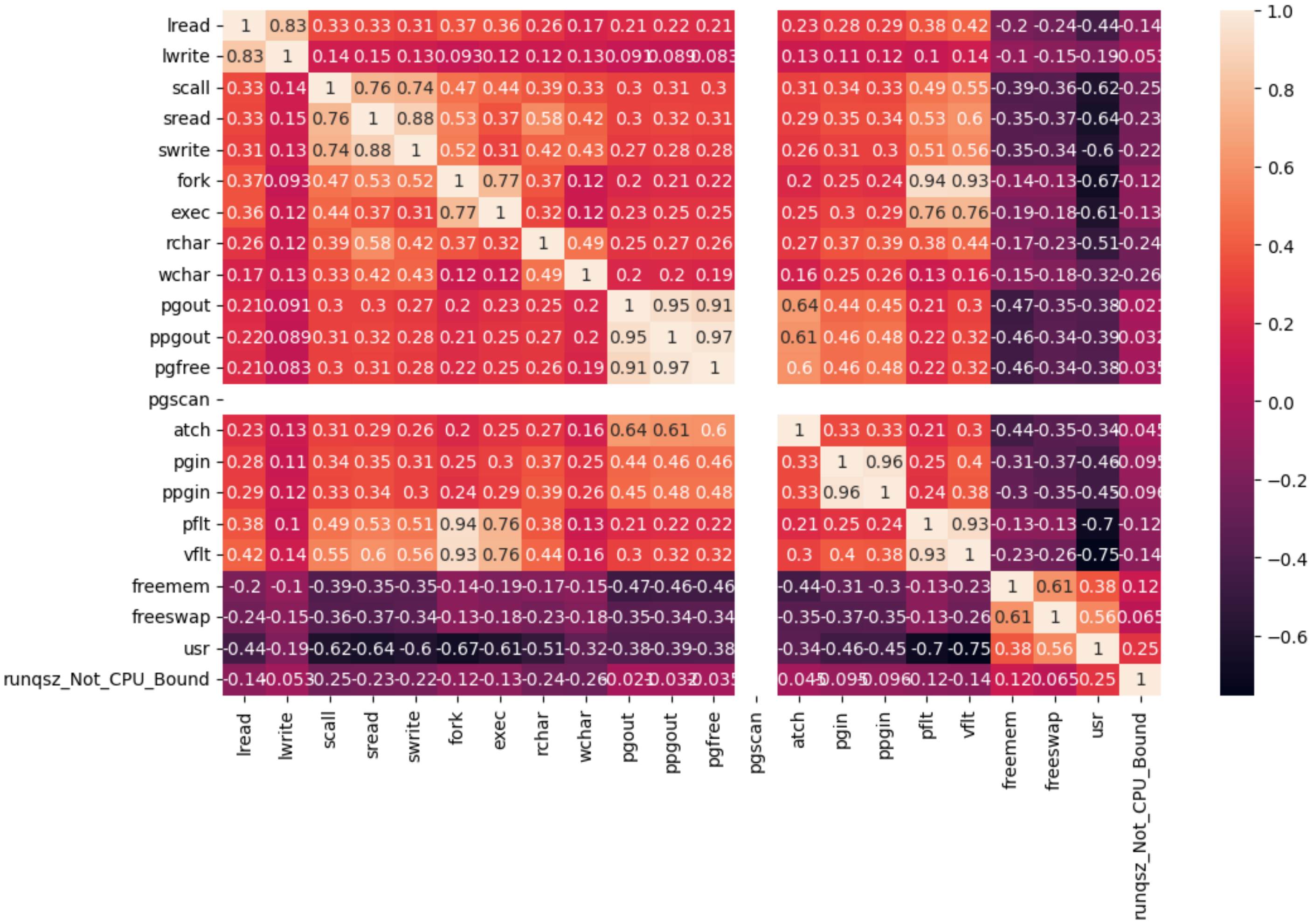


	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgi
count	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8.088000e+03	8.177000e+03	8192.000000	...	8192.000000	8192.000000	8192.00000
unique	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	Na
top	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	Na
freq	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	Na
mean	19.559692	13.106201	2306.318237	210.479980	150.058228	1.884554	2.791998	1.973857e+05	9.590299e+04	2.285317	...	21.526849	1.127505	8.27796
std	53.353799	29.891726	1633.617322	198.980146	160.478980	2.479493	5.212456	2.398375e+05	1.408417e+05	5.307038	...	71.141340	5.708347	13.87497
min	0.000000	0.000000	109.000000	6.000000	7.000000	0.000000	0.000000	2.780000e+02	1.498000e+03	0.000000	...	0.000000	0.000000	0.00000
25%	2.000000	0.000000	1012.000000	86.000000	63.000000	0.400000	0.200000	3.409150e+04	2.291600e+04	0.000000	...	0.000000	0.000000	0.60000
50%	7.000000	1.000000	2051.500000	166.000000	117.000000	0.800000	1.200000	1.254735e+05	4.661900e+04	0.000000	...	0.000000	0.000000	2.80000
75%	20.000000	10.000000	3317.250000	279.000000	185.000000	2.200000	2.800000	2.678288e+05	1.061010e+05	2.400000	...	0.000000	0.600000	9.76500
max	1845.000000	575.000000	12493.000000	5318.000000	5456.000000	20.120000	59.560000	2.526649e+06	1.801623e+06	81.440000	...	1237.000000	211.580000	141.20000

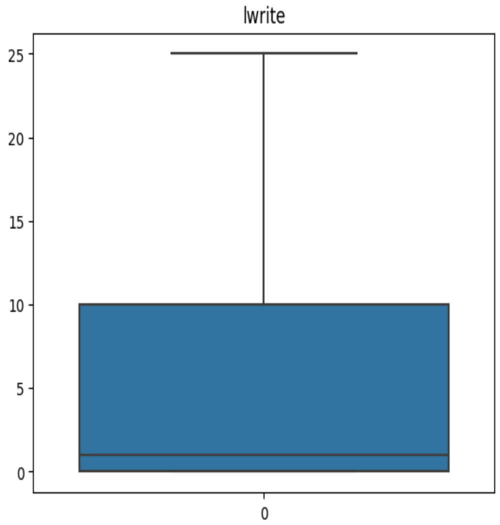


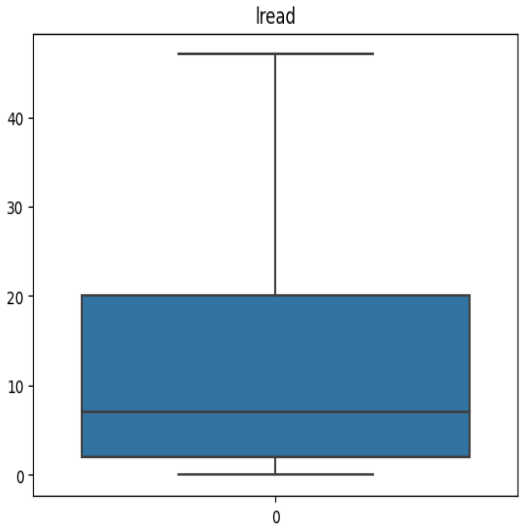
exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
192.000000	8.088000e+03	8.177000e+03	8192.000000	...	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8192	8192.000000	8.192000e+03	8192.000000
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	2	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	Not_CPU_Bound	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	4331	NaN	NaN	NaN
2.791998	1.973857e+05	9.590299e+04	2.285317	...	21.526849	1.127505	8.277960	12.388586	109.793799	185.315796	NaN	1763.456299	1.328126e+06	83.968872
5.212456	2.398375e+05	1.408417e+05	5.307038	...	71.141340	5.708347	13.874978	22.281318	114.419221	191.000603	NaN	2482.104511	4.220194e+05	18.401905
0.000000	2.780000e+02	1.498000e+03	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.200000	NaN	55.000000	2.000000e+00	0.000000
0.200000	3.409150e+04	2.291600e+04	0.000000	...	0.000000	0.000000	0.600000	0.600000	25.000000	45.400000	NaN	231.000000	1.042624e+06	81.000000
1.200000	1.254735e+05	4.661900e+04	0.000000	...	0.000000	0.000000	2.800000	3.800000	63.800000	120.400000	NaN	579.000000	1.289290e+06	89.000000
2.800000	2.678288e+05	1.061010e+05	2.400000	...	0.000000	0.600000	9.765000	13.800000	159.600000	251.800000	NaN	2002.250000	1.730380e+06	94.000000
59.560000	2.526649e+06	1.801623e+06	81.440000	...	1237.000000	211.580000	141.200000	292.610000	899.800000	1365.000000	NaN	12027.000000	2.243187e+06	99.000000

- From the table we can see that runqsz variable has 2 unique categories out of which Not_CPU_Bound has the highest frequency.
- Below given is the heatmap for all the variables:

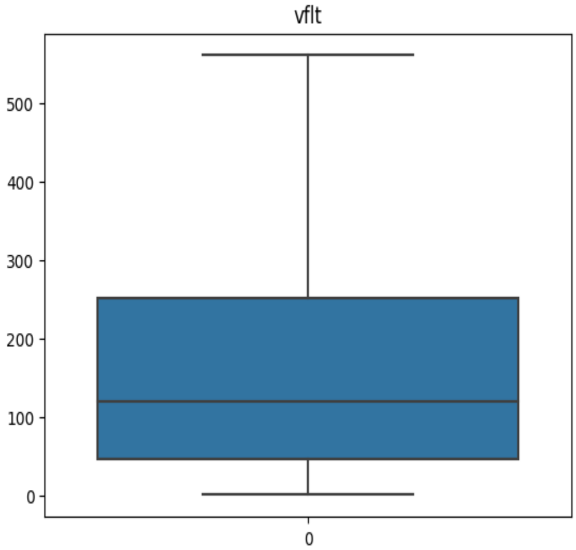


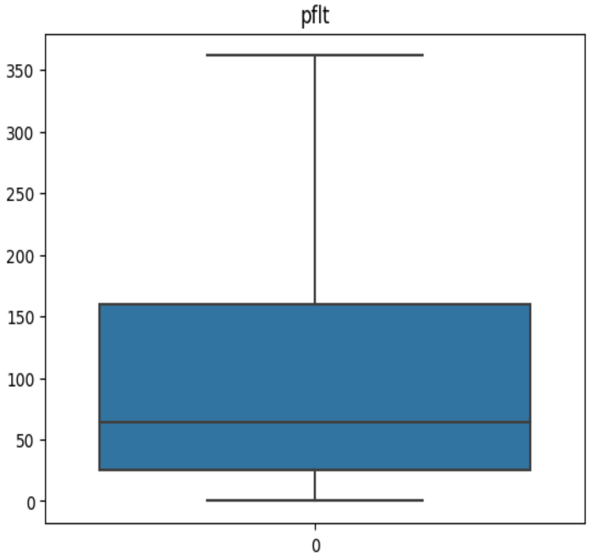
- We can notice that there is a strong correlation among lwrite and lread variables ; pgin and ppgin variables ; vflt and pflt variables, pgfree and ppgout variables and swrite and sread variables. Thus, we can certainly say the performance indicator variables that are related to one specific action or occurrence have high positive correlation.
- Let look at the boxplot for some of the correlated variables:



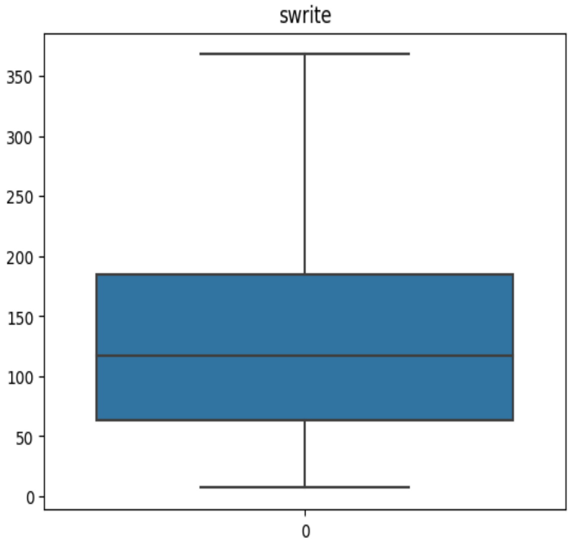


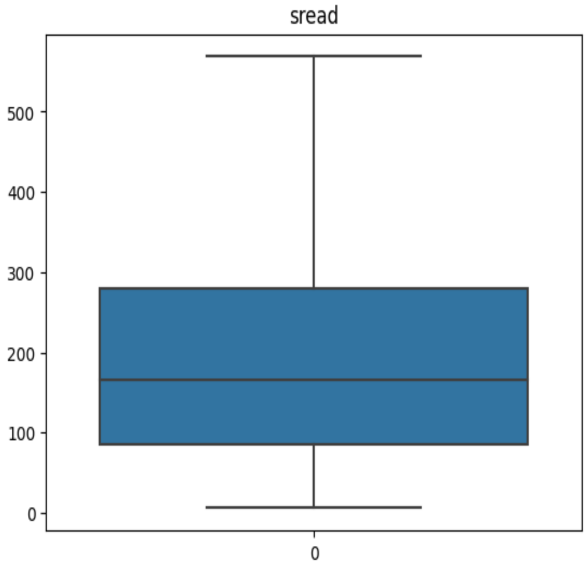
- We can notice that both the variables are highly positively skewed and both the variables have median closer to the minimum values. Except only lwrite has the minimum value of zero.



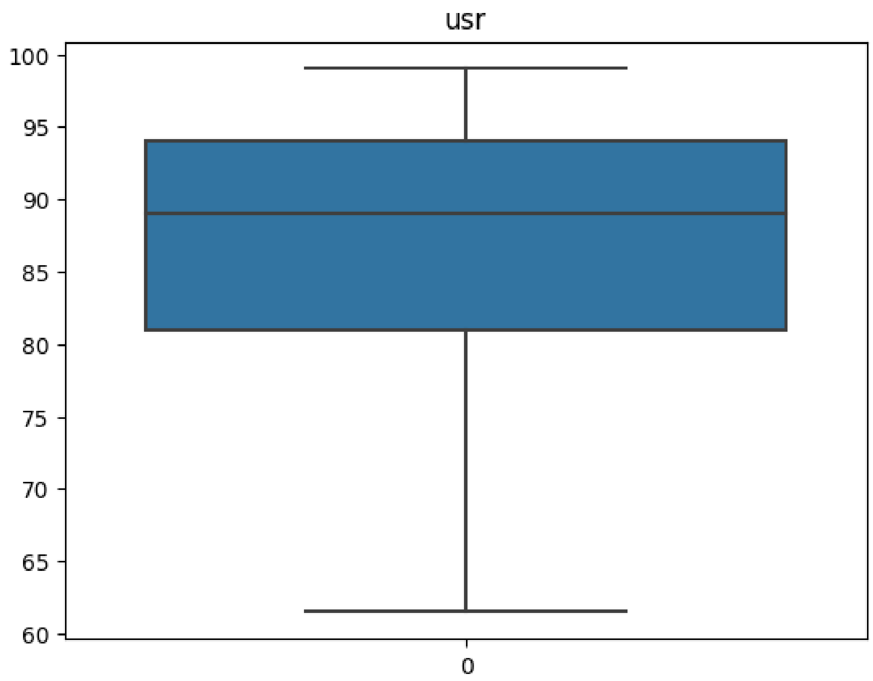


- Both the boxplot are positively skewed and the median for both the variables are closer to the minimum value. Interestingly, both the boxplots are almost identical.





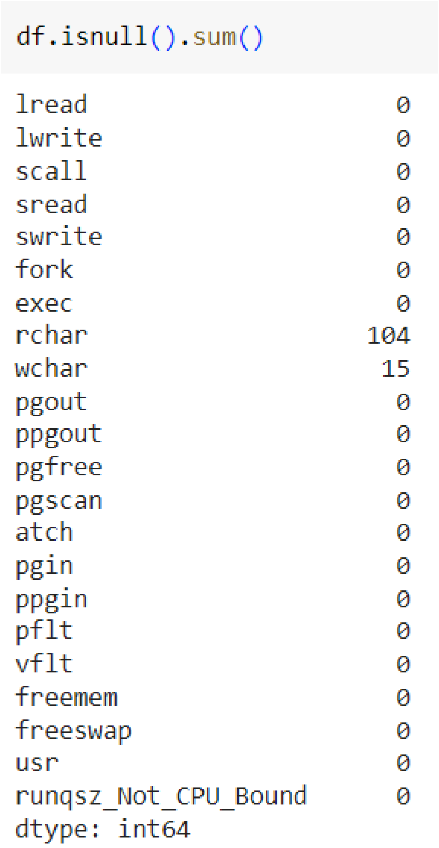
- Similar to previous boxplots, the correlated variables are positively skewed and the median for both the variables lay at similar range.



- Above the given boxplot is of the dependent variable we can see that the data is negatively skewed with median ranging near maximum value.

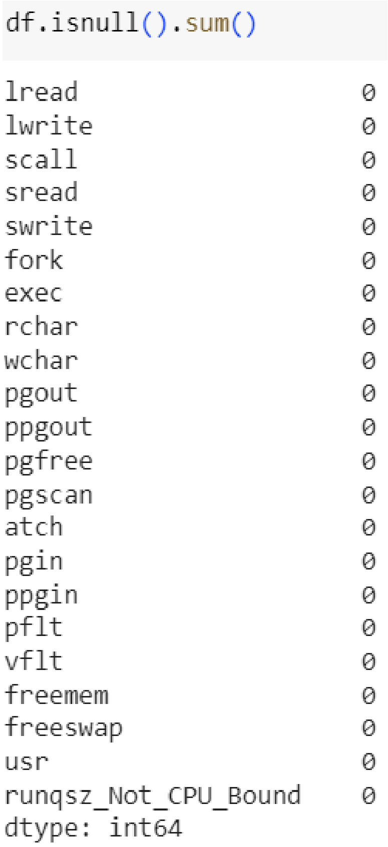
1.2 Impute null values if present, also check for the values which are equal to zero. Do they have any meaning or do we need to change them or drop them? Check for the possibility of creating new features if required. Also check for outliers and duplicates if there.

- One of the most important information from the descriptive table in page number 6 is that most variables have the minimum value of zero. This is because most of the variables in the dataset are performance indicators which tend to count the total number of a specific occurrence. For example, the variable pgin measures the number of pages in request at a time. Thus, in certain circumstances, when there are no pages in requests, the dataset would count that as a zero. This cannot be considered a null value, since this gives us insight into the performance of the program.
- Rchar and Wchar variables had null variables, which was then treated by imputing the mean value into the null values. A new user defined function was created to impute the same.



```
df.isnull().sum()
```

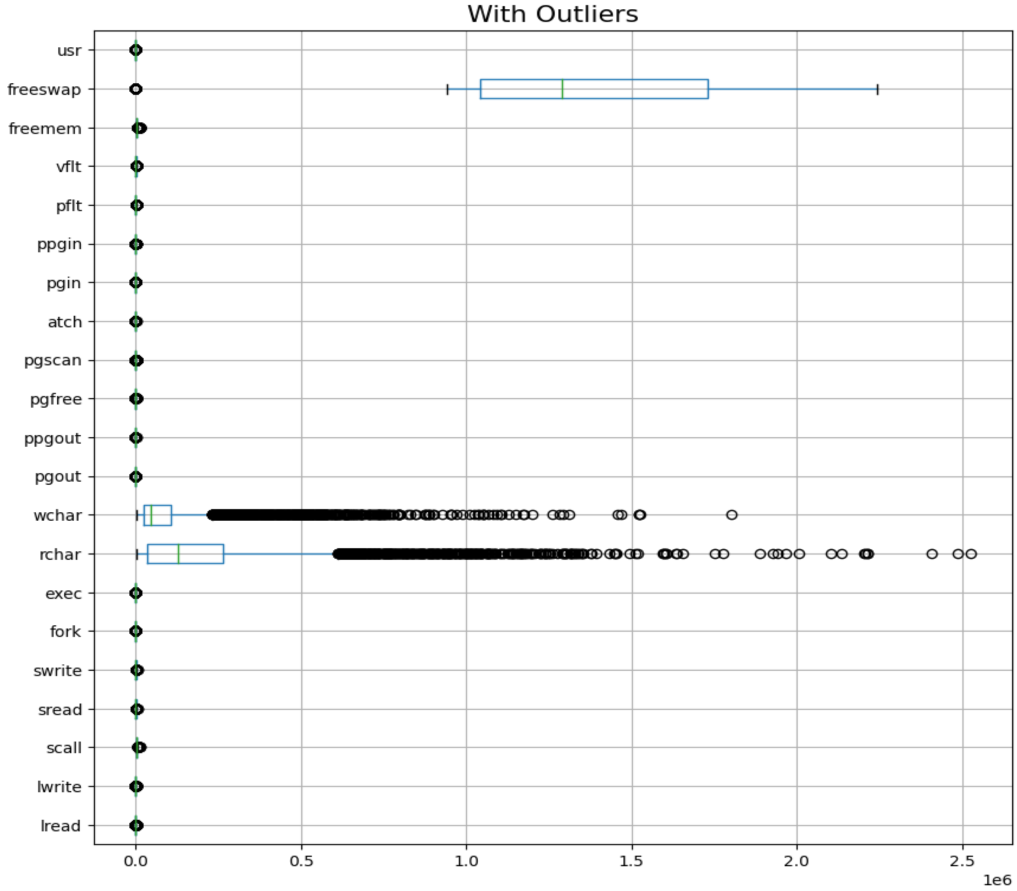
```
lread                  0
lwrite                 0
scall                  0
sread                  0
swrite                 0
fork                   0
exec                   0
rchar                104
wchar                 15
pgout                  0
ppgout                 0
pgfree               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



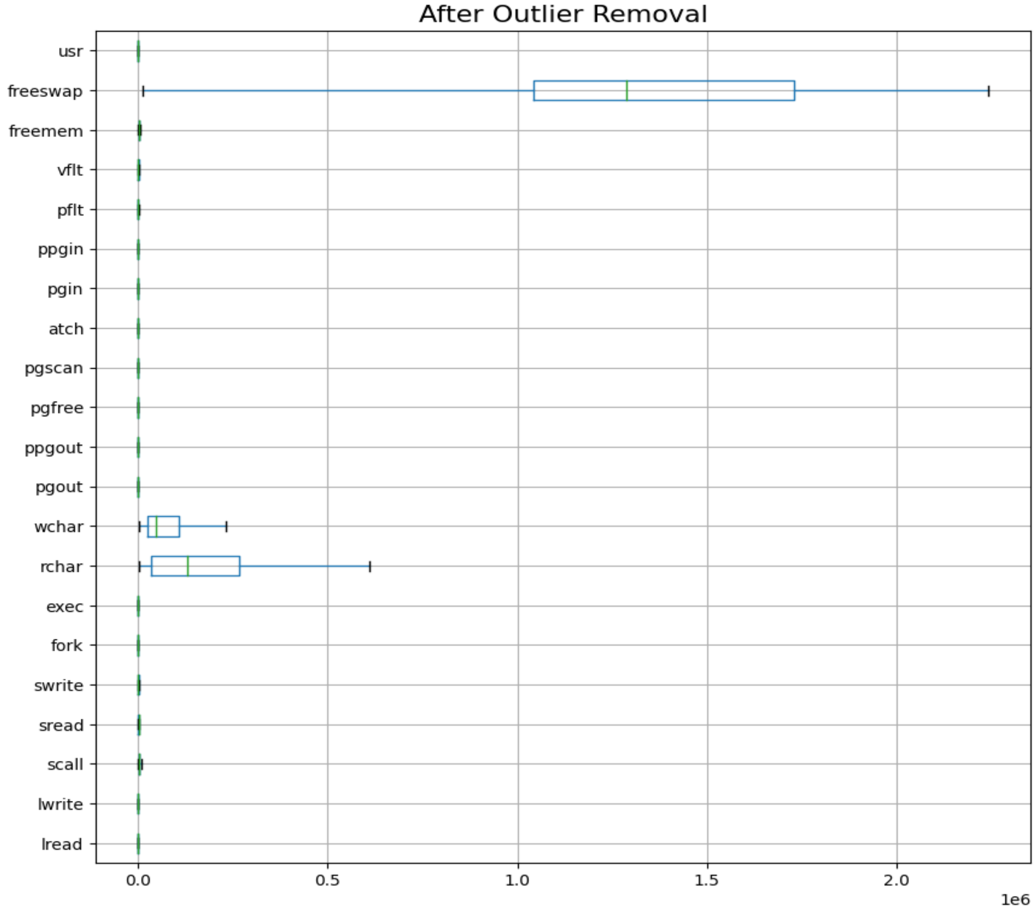
```
df.isnull().sum()
```

```
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
freemem                0
freeswap               0
usr                    0
runqsz_Not_CPU_Bound   0
dtype: int64
```

- Next step was to check outliers:



We can notice that there are certain outliers in wchar and rchar variables. These outliers can affect the model performance. Thus IQR outlier treatment using a user defined function was performed on these variables.

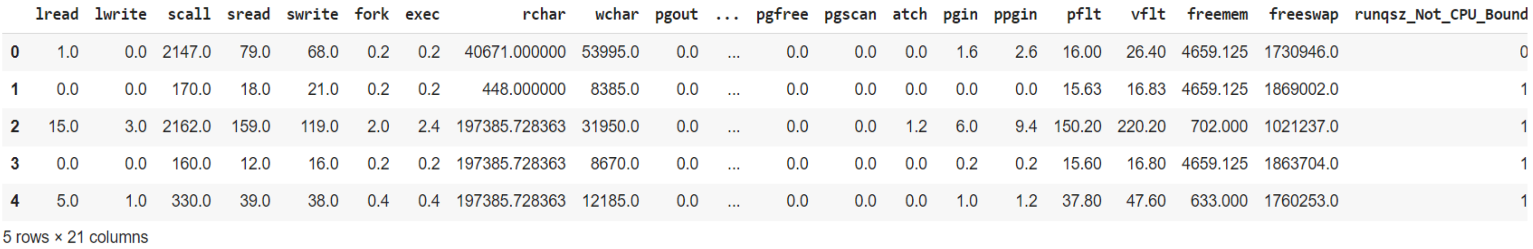


- There are no duplicated rows in the given dataset.

```
Number of duplicate rows = 0
```

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.

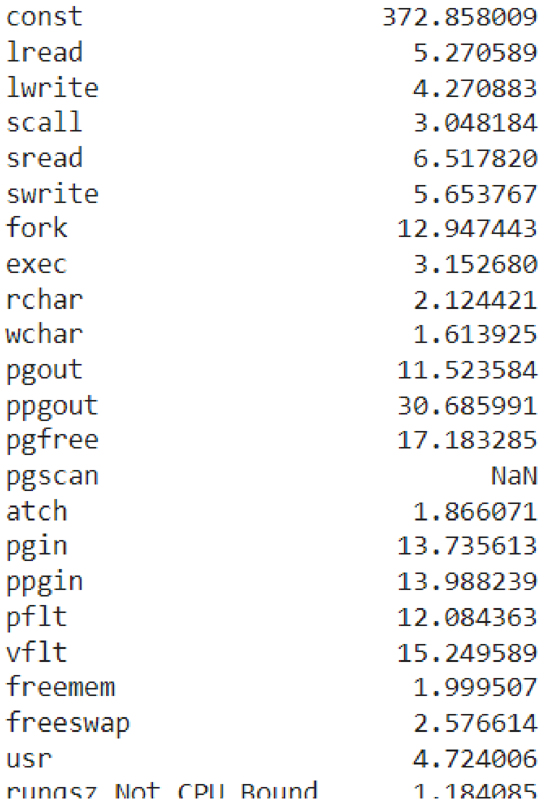
- Next, the runsqz variable was converted into an integer variable through dummy encoding for better prediction in the model.
- X and Y variables were defined and below given is the head of the X variable:



	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	runqsz_Not_CPU_Bound
0	1.0	0.0	2147.0	79.0	68.0	0.2	0.2	40671.000000	53995.0	0.0	...	0.0	0.0	0.0	1.6	2.6	16.00	26.40	4659.125	1730946.0	0
1	0.0	0.0	170.0	18.0	21.0	0.2	0.2	448.000000	8385.0	0.0	...	0.0	0.0	0.0	0.0	0.0	15.63	16.83	4659.125	1869002.0	1
2	15.0	3.0	2162.0	159.0	119.0	2.0	2.4	197385.728363	31950.0	0.0	...	0.0	0.0	1.2	6.0	9.4	150.20	220.20	702.000	1021237.0	1
3	0.0	0.0	160.0	12.0	16.0	0.2	0.2	197385.728363	8670.0	0.0	...	0.0	0.0	0.0	0.2	0.2	15.60	16.80	4659.125	1863704.0	1
4	5.0	1.0	330.0	39.0	38.0	0.4	0.4	197385.728363	12185.0	0.0	...	0.0	0.0	0.0	1.0	1.2	37.80	47.60	633.000	1760253.0	1

5 rows × 21 columns

- Using the sklearn function, the given dataset was split into test and train data with a 70:30 ratio.
- After splitting the train and test data, we shall create a regression model.
- From this model, we must try to reduce multicollinearity, i.e., where the independent variables have a high level of correlation.
- We use variance inflation factor to ascertain the correlation among the variables:



```
const                    372.858009
lread                      5.270589
lwrite                     4.270883
scall                      3.048184
sread                      6.517820
swrite                     5.653767
fork                      12.947443
exec                       3.152680
rchar                      2.124421
wchar                      1.613925
pgout                     11.523584
ppgout                    30.685991
pgfree                    17.183285
pgscan                          NaN
atch                       1.866071
pgin                      13.735613
ppgin                     13.988239
pflt                      12.084363
vflt                      15.249589
freemem                    1.999507
freeswap                   2.576614
usr                        4.724006
runqsz Not CPU Bound       1.184085
```

- Usually VIF value more than 10 is unfavourable to the regression model. From the table we can notice that some variables have more that 10 VIF values - fork, pgout, ppgout, pgfree, pgin, ppgin, pflt, vflt.
- Next step is to ascertain whether removing either of these variable would affect the predictability of the model, that is the r-square variable. Following were the result of making sure whether dropping one variable would affect the rsquare.

“Fork”

```
R-squared: 0.796 Adjusted R-squared: 0.795
```

“pgout”

```
R-squared: 0.796 Adjusted R-squared: 0.795
```

“ppgout”

```
R-squared: 0.796 Adjusted R-squared: 0.795
```

“pgfree”

```
R-squared: 0.796 Adjusted R-squared: 0.795
```

“pgin”

```
R-squared: 0.796 Adjusted R-squared: 0.795
```

“ppgin”

```
R-squared: 0.796 Adjusted R-squared: 0.795
```

“Pflt”

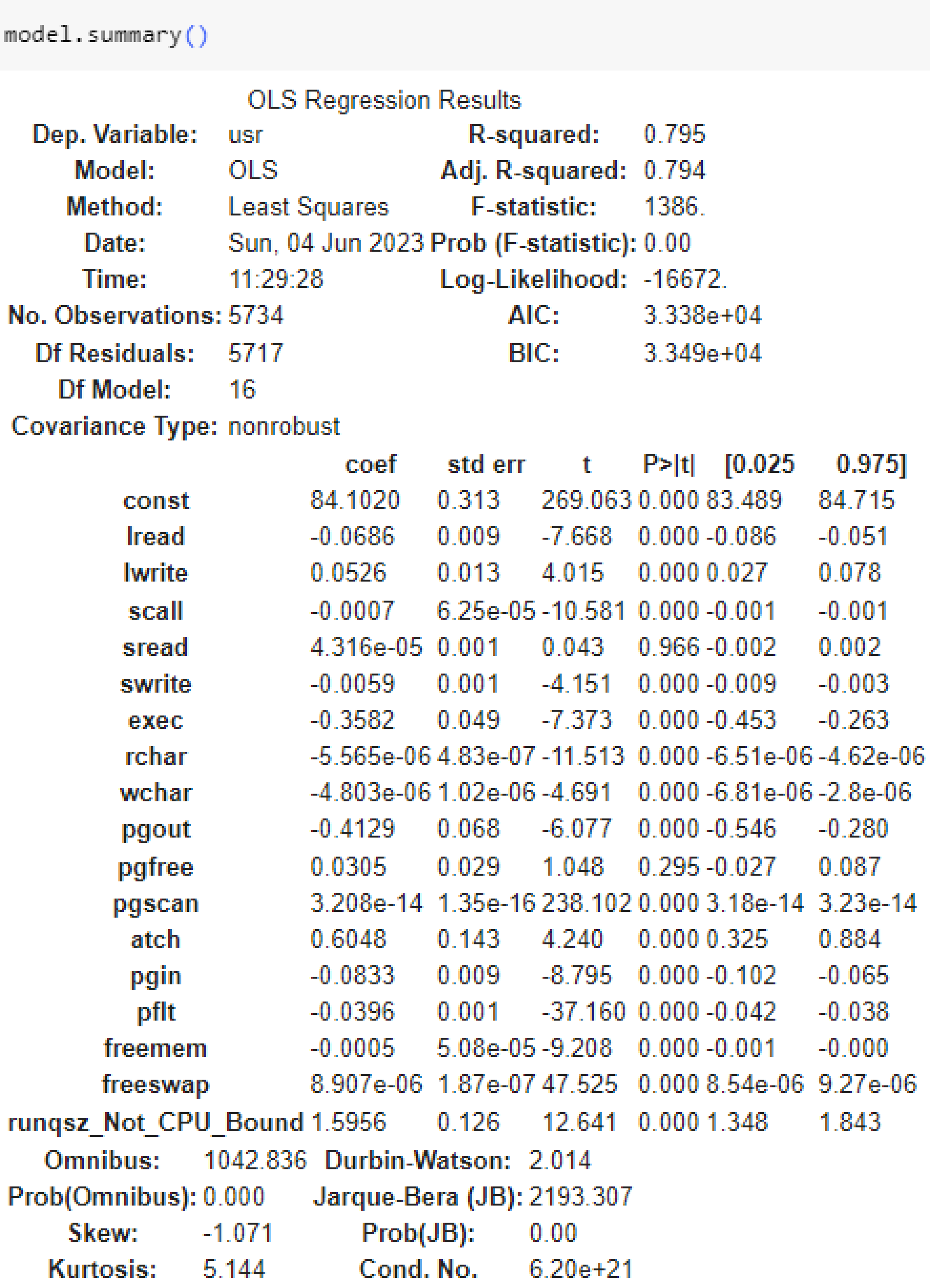
```
R-squared: 0.786 Adjusted R-squared: 0.785
```

“Vflt”

`R-squared: 0.796 Adjusted R-squared: 0.795`

As we can see, from removing these variables, the predictability of the model will not be affected in any way. Thus, I removed the variables with highest VIF value, until the multi-collinearity among the independent variables were reduced to optimum level. This process was performed four times and the OLS results for the final model is as follows.

- From the last model fitting, we received the following OLS regression results. Let us break down the same.



```
model.summary()
```

```
                         OLS Regression Results
   Dep. Variable:      usr                    R-squared:          0.795
          Model:       OLS                Adj. R-squared:         0.794
         Method:       Least Squares          F-statistic:        1386.
           Date:       Sun, 04 Jun 2023 Prob (F-statistic):       0.00
           Time:       11:29:28            Log-Likelihood:        -16672.
No. Observations:      5734                          AIC:         3.338e+04
    Df Residuals:      5717                          BIC:         3.349e+04
       Df Model:       16
Covariance Type:       nonrobust
                           coef       std err    t        P>|t|  [0.025     0.975]
        const              84.1020    0.313     269.063  0.000  83.489     84.715
        lread              -0.0686    0.009     -7.668   0.000  -0.086     -0.051
        lwrite             0.0526     0.013     4.015    0.000  0.027      0.078
        scall              -0.0007    6.25e-05  -10.581  0.000  -0.001     -0.001
        sread              4.316e-05  0.001     0.043    0.966  -0.002     0.002
        swrite             -0.0059    0.001     -4.151   0.000  -0.009     -0.003
        exec               -0.3582    0.049     -7.373   0.000  -0.453     -0.263
        rchar              -5.565e-06 4.83e-07  -11.513  0.000  -6.51e-06  -4.62e-06
        wchar              -4.803e-06 1.02e-06  -4.691   0.000  -6.81e-06  -2.8e-06
        pgout              -0.4129    0.068     -6.077   0.000  -0.546     -0.280
        pgfree             0.0305     0.029     1.048    0.295  -0.027     0.087
        pgscan             3.208e-14  1.35e-16  238.102  0.000  3.18e-14   3.23e-14
        atch               0.6048     0.143     4.240    0.000  0.325      0.884
        pgin               -0.0833    0.009     -8.795   0.000  -0.102     -0.065
        pflt               -0.0396    0.001     -37.160  0.000  -0.042     -0.038
        freemem            -0.0005    5.08e-05  -9.208   0.000  -0.001     -0.000
        freeswap           8.907e-06  1.87e-07  47.525   0.000  8.54e-06   9.27e-06
runqsz_Not_CPU_Bound       1.5956     0.126     12.641   0.000  1.348      1.843
     Omnibus:      1042.836  Durbin-Watson:     2.014
Prob(Omnibus):     0.000     Jarque-Bera (JB):  2193.307
        Skew:      -1.071         Prob(JB):     0.00
    Kurtosis:      5.144         Cond. No.      6.20e+21
```

- From the generated model, we have the r-square value of 0.795. Therefore, 79.5% of variability in the dependent variable is successfully explained by the model.
- Similarly, adjusted r-square is also calculated at 0.794. Thus, we can say the the given predictors are significant in predicting the dependent variable.

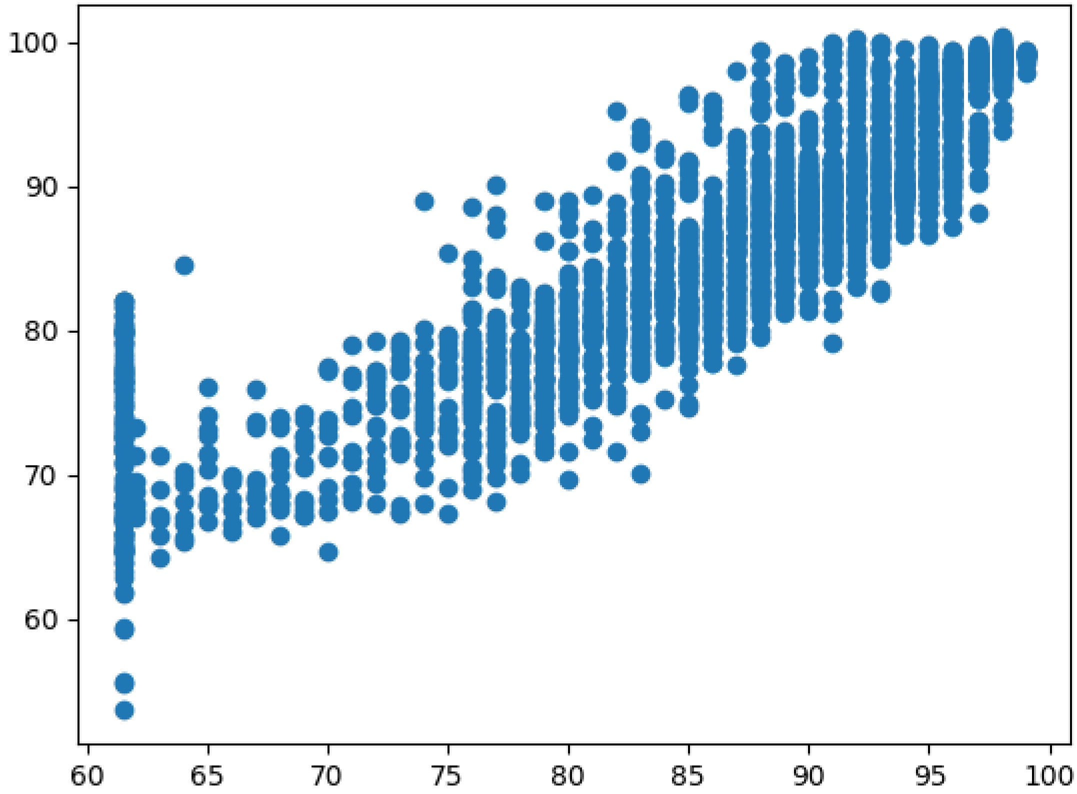
- Jarque-bera test shows us whether the given dataset is normally distributed or not. Therefore the hypothesis for this test can be written as:

Ho: Data is normally distributed
Ha: Data is not normally distributed
Since, the given p-value is 0. We fail to accept the null hypothesis at 5% level of significance and the given data is not normally distributed.

- The Kurtosis value is 5.144, i.e the distribution curve is much curvier than being flatter. The skew is negative in value, -1.071, denoting that the data is negatively skewed.
- The Omnibus probability value denotes whether the regression model satisfies all the assumptions. Therefore the hypothesis is:

Ho: All the assumptions as satisfied
Ha: All the assumptions are not satisfied
Since the p-value is 0. We fail to accept the null hypothesis at 5% level of significance and the regression model does not satisfy all the assumptions.

- The root mean squared error for the train data is `4.42` and test data is `4.65` , this means that the model is not optimised very well.
- Below given is the scatterplot of actual variables and the predicted values of the variables.



This shows that the actual and predicted values are closer to each other thus the point are closer to a formation of a line.

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

The final linear regression will be:

usr = (84.1) * intercept + (-0.07) * lread + (0.05) * lwrite + (-0.0) * scall + (0.0) * sread + (-0.01) * swrite + (-0.36) * exec + (-0.0) * rchar + (-0.0) * wchar + (-0.41) * pgout + (0.03) * pgfree + (0.0) * pgscan + (0.6) * atch + (-0.08) * pgin + (-0.04) * pflt + (-0.0) * freemem + (0.0) * freeswap + (1.6) * runqsz_Not_CPU_Bound

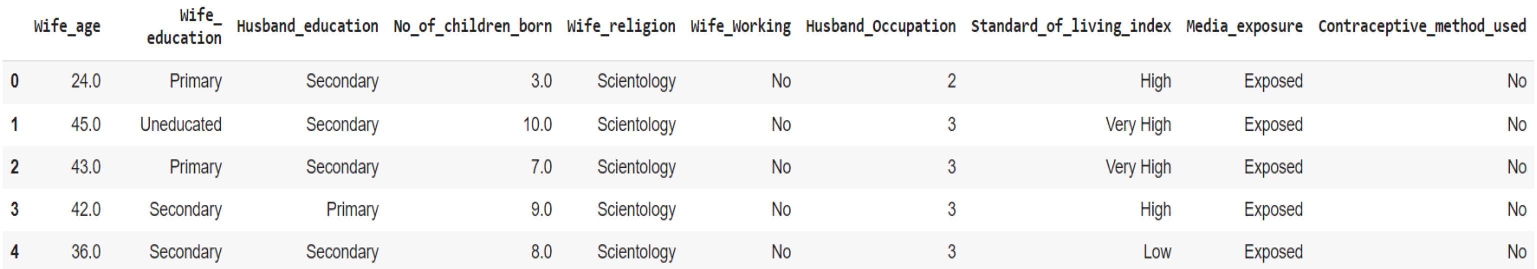
Thus we can infer that:

- As there is 1 unit increase in runsqz_Not_CPU_Bound, there is a 1.6 unit increase in usr keeping all other variables constant.. Etc.
- There are some negative coefficient values, affecting the dependent variable. For instance the is 1 unit increase in exec as there is 0.36 unit decrease in usr.
- The wchar variable has some significant negative correlation with usr. That is the change in number of characters transferred due to write cells negatively affects the portion of time cpus is in user mode.
- Process run queue size is highly positively correlated with the time cpus is in user 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.
Data Dictionary:
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

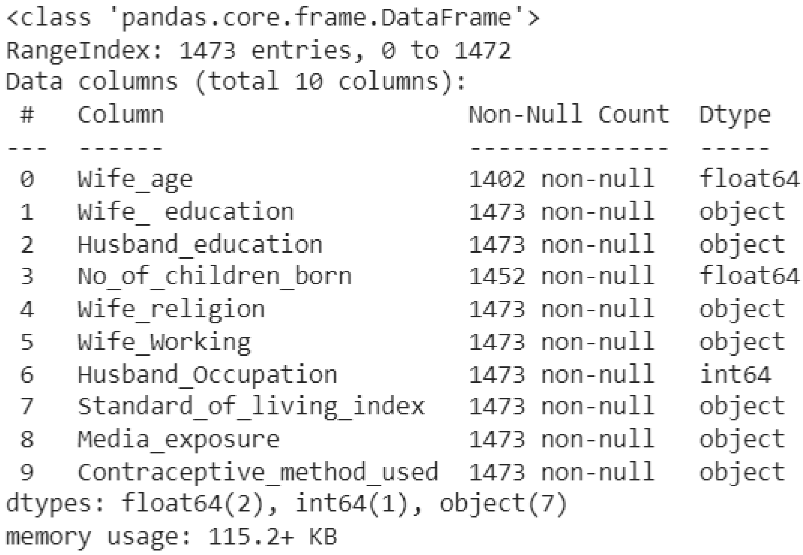
2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, check for duplicates and outliers and write an inference on it. Perform Univariate and Bivariate Analysis and Multivariate Analysis.

- There are totally 1473 entries/rows and 10 variables/columns in the given dataset.
- Below given is the head of the dataset:



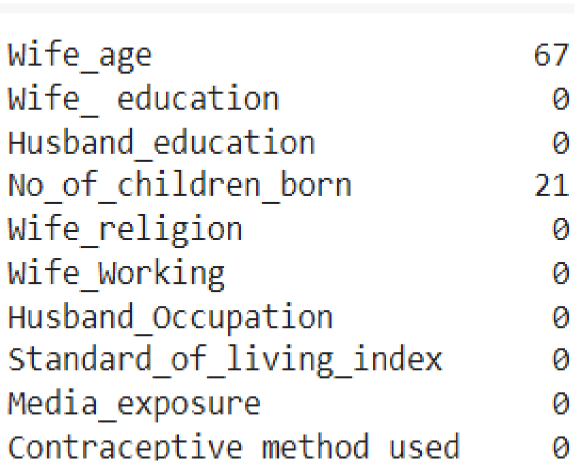
	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure	Contraceptive_method_used
0	24.0	Primary	Secondary	3.0	Scientology	No	2	High	Exposed	No
1	45.0	Uneducated	Secondary	10.0	Scientology	No	3	Very High	Exposed	No
2	43.0	Primary	Secondary	7.0	Scientology	No	3	Very High	Exposed	No
3	42.0	Secondary	Primary	9.0	Scientology	No	3	High	Exposed	No
4	36.0	Secondary	Secondary	8.0	Scientology	No	3	Low	Exposed	No

- Below given table shows the information on the variables:

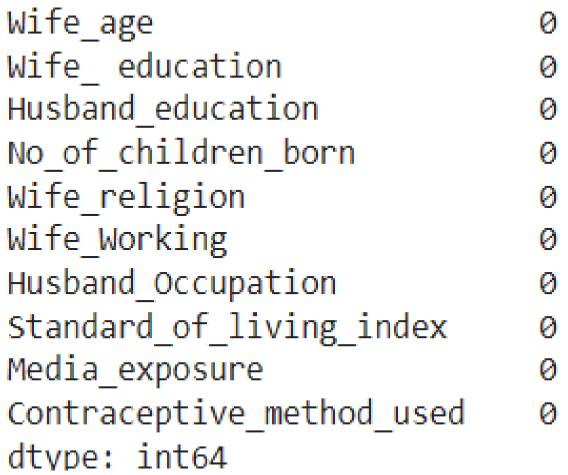


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1473 entries, 0 to 1472
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Wife_age                   1402 non-null   float64
 1   Wife_ education            1473 non-null   object 
 2   Husband_education          1473 non-null   object 
 3   No_of_children_born        1452 non-null   float64
 4   Wife_religion              1473 non-null   object 
 5   Wife_Working               1473 non-null   object 
 6   Husband_Occupation         1473 non-null   int64  
 7   Standard_of_living_index   1473 non-null   object 
 8   Media_exposure             1473 non-null   object 
 9   Contraceptive_method_used  1473 non-null   object 
dtypes: float64(2), int64(1), object(7)
memory usage: 115.2+ KB
```

- We can see that most of the variables are of float or object data type and almost all the variables are categorical. Except for the wife's age variable which is continuous.
- NULL VALUE CHECK



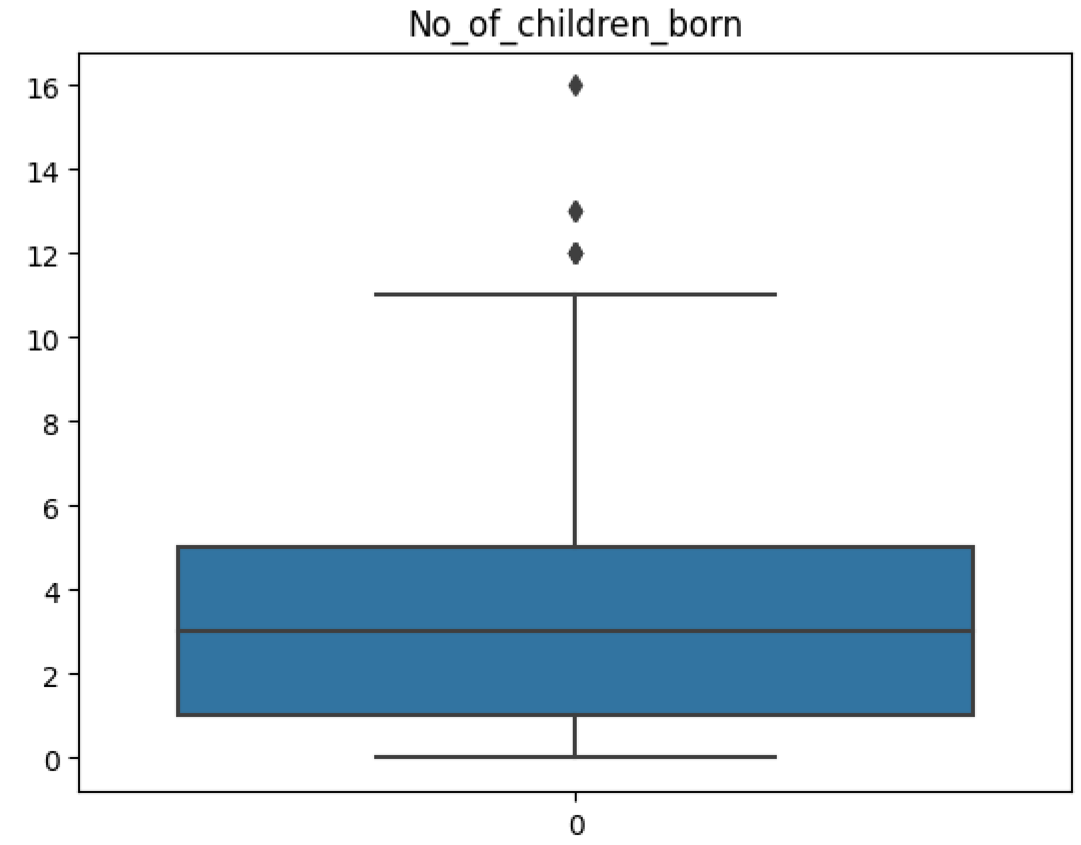
```
Wife_age                     67
Wife_ education               0
Husband_education             0
No_of_children_born          21
Wife_religion                 0
Wife_Working                  0
Husband_Occupation            0
Standard_of_living_index      0
Media_exposure                0
Contraceptive_method_used     0
```



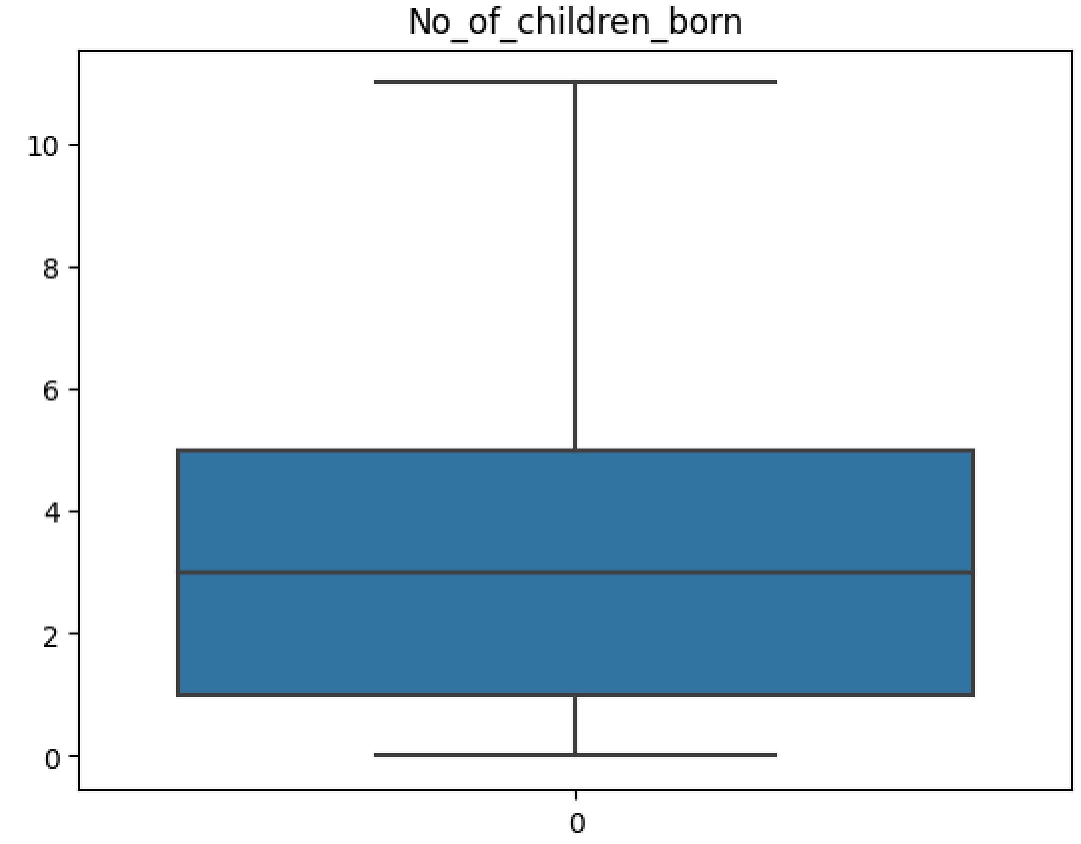
```
Wife_age                     0
Wife_ education              0
Husband_education            0
No_of_children_born          0
Wife_religion                0
Wife_Working                 0
Husband_Occupation           0
Standard_of_living_index     0
Media_exposure               0
Contraceptive_method_used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

The above given table shows us the output for the null value check. There were some null values in Wife_age and No_of_children_born variable. These null values were imputed with the mean value. Thus producing the output on the right hand side.

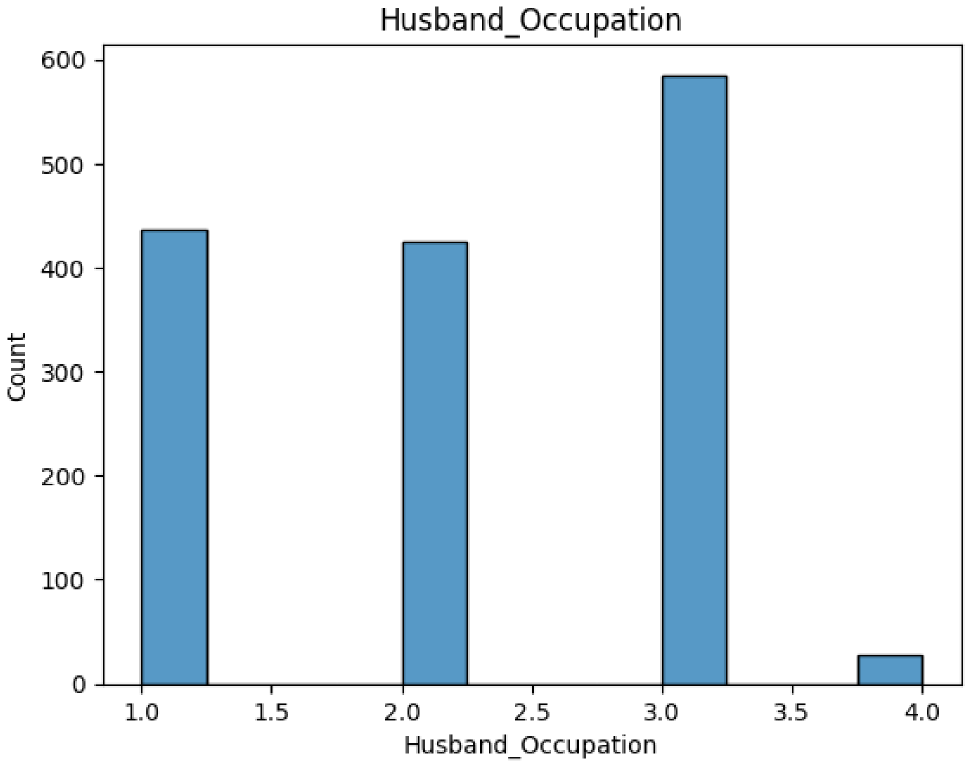
- OUTLIER TREATMENT



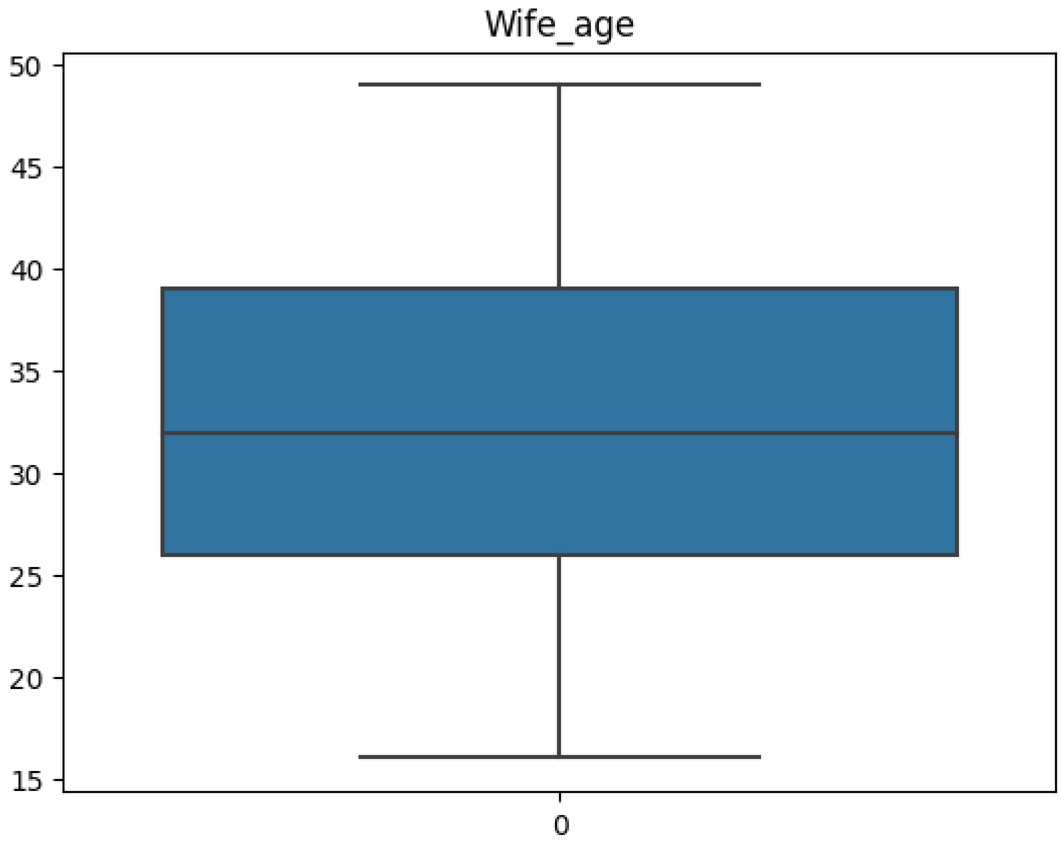
There were some outliers in the no_of children_born variable which was then treated using IQR outlier treatment. Thus producing the following boxplot.



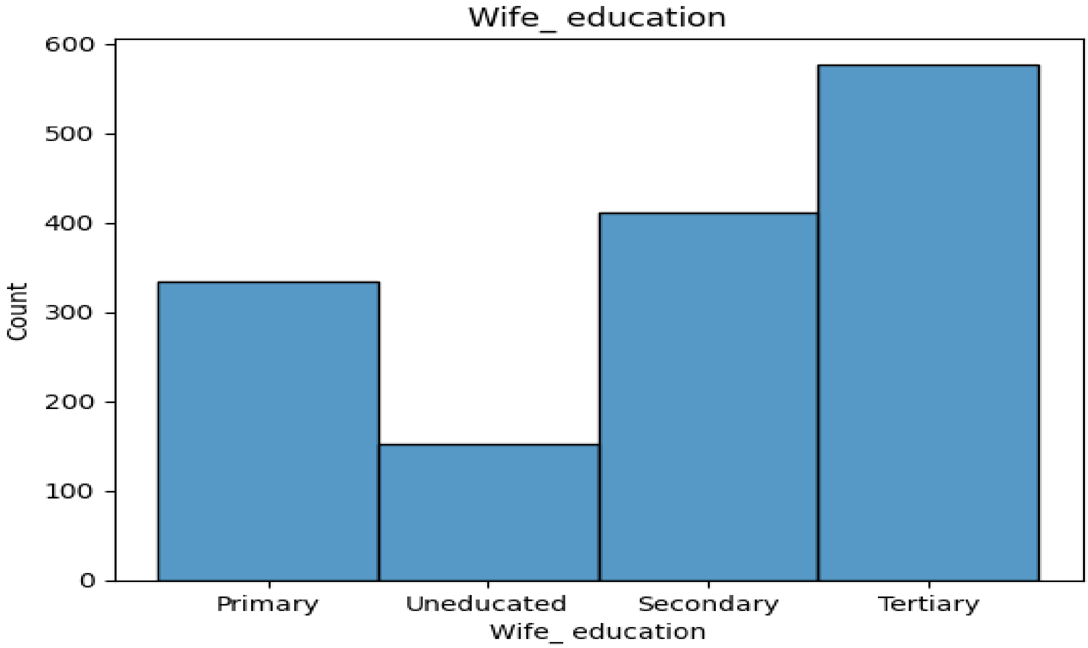
Let us move along and analyse the variables, firstly the no_of_children born variable. There are an average of 3 children born for each mother among the sample population.



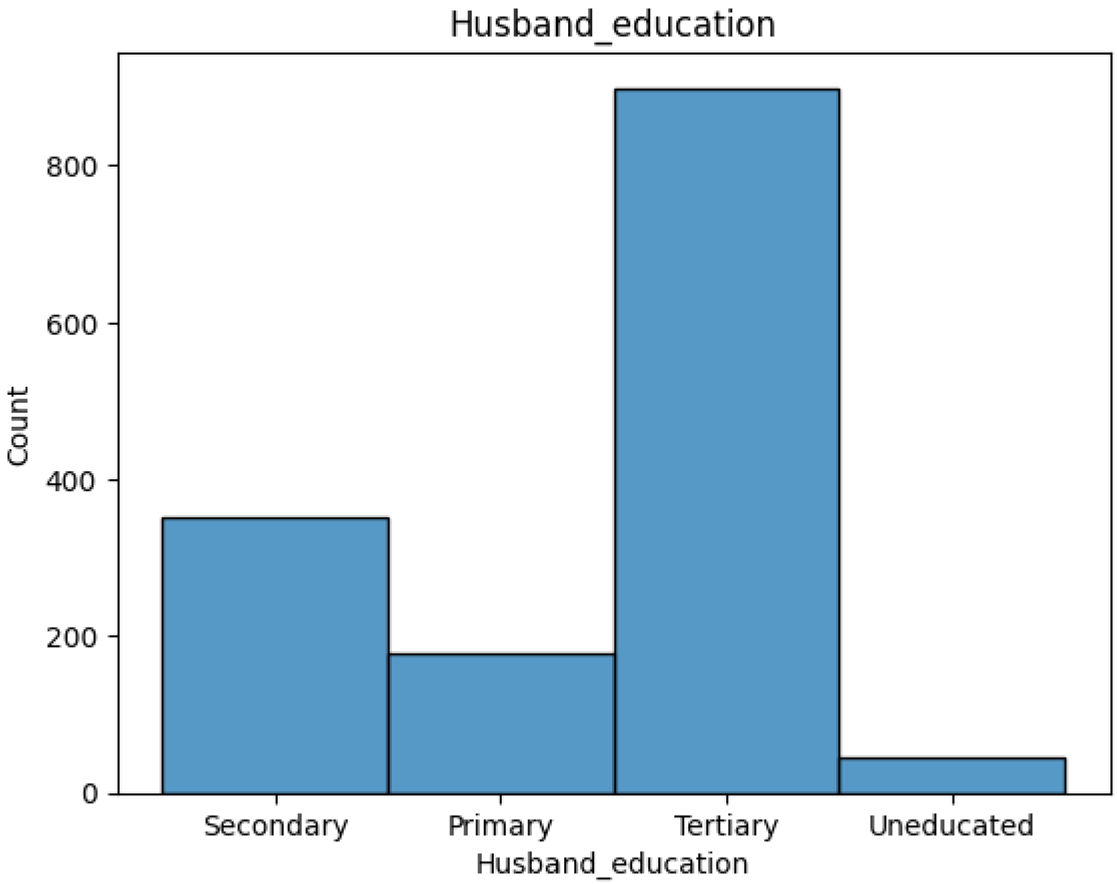
The husband occupation variable is coded in a categorical manner scaling from 1 to 4. Therefore, we can notice that on an average, most husbands are occupied with occupation type 3 and least number of husbands are employed in job type 4.



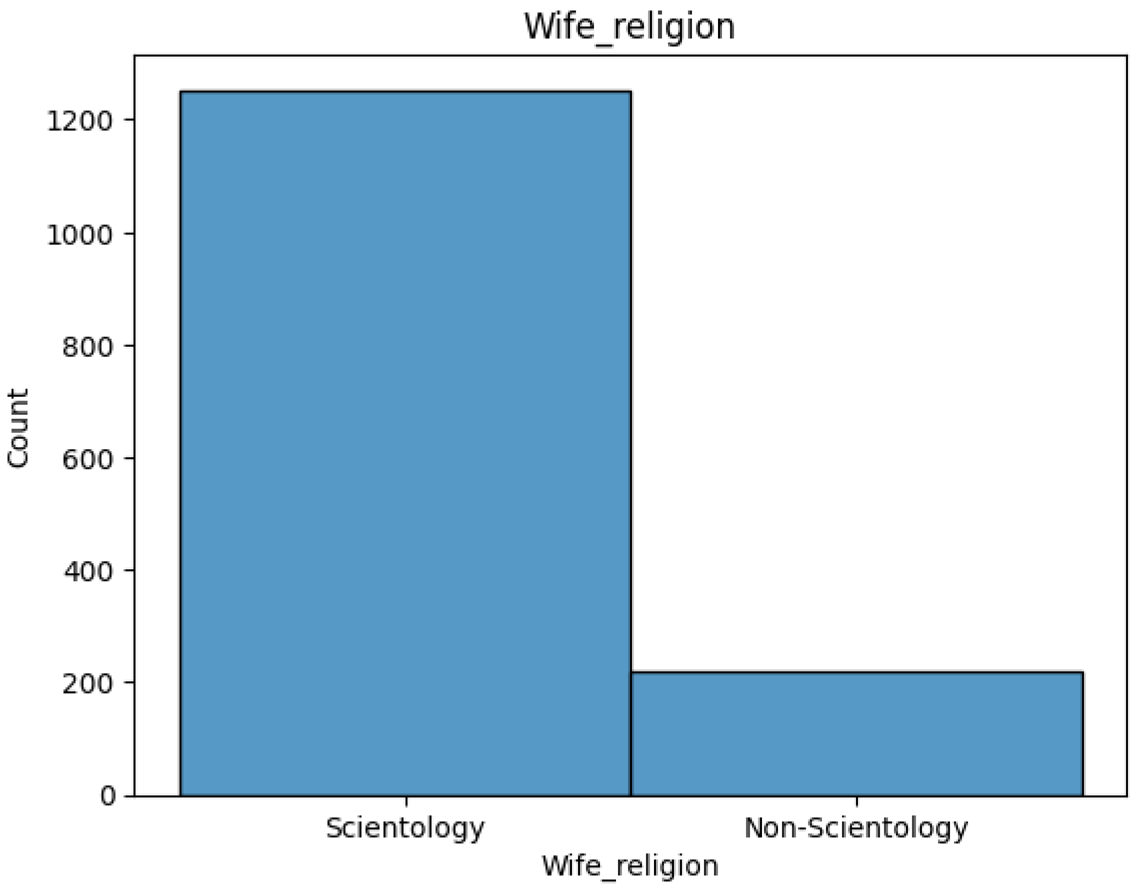
Wife_age is a continuous variable and from the boxplot we can see that most wives are ranging from 25 to 40 with the median age of 32. The boxplot is also not skewed either to the left or right side.



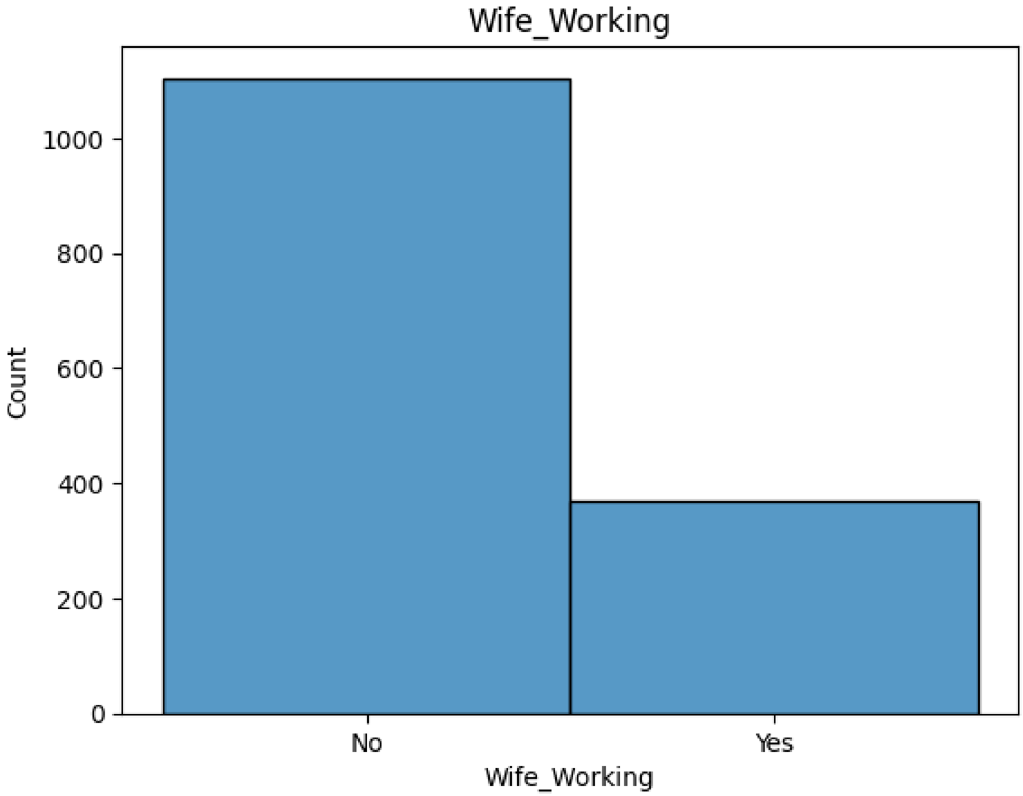
We can see that most women belong to tertiary education, i.e, they have specialisation in a field. Least amount of women are uneducated.



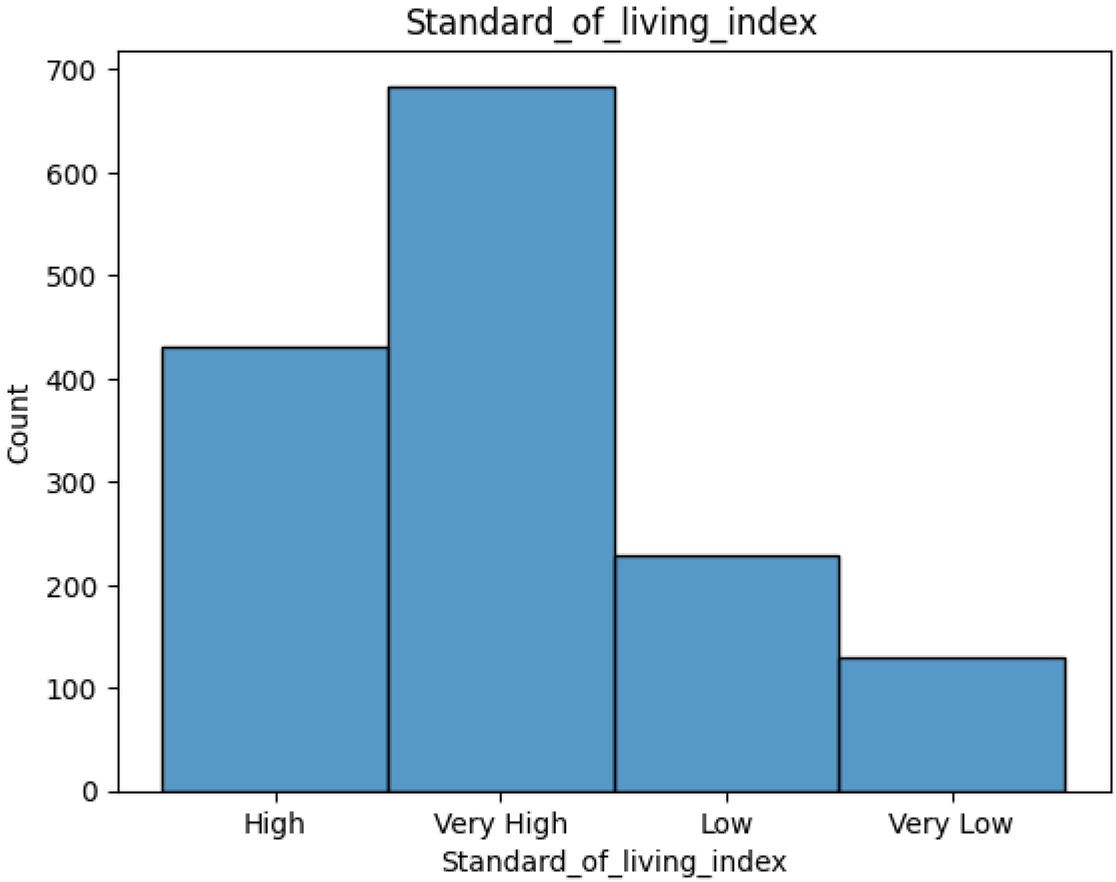
Majority of the husbands are also educated in tertiary sector and least amount of men are uneducated.



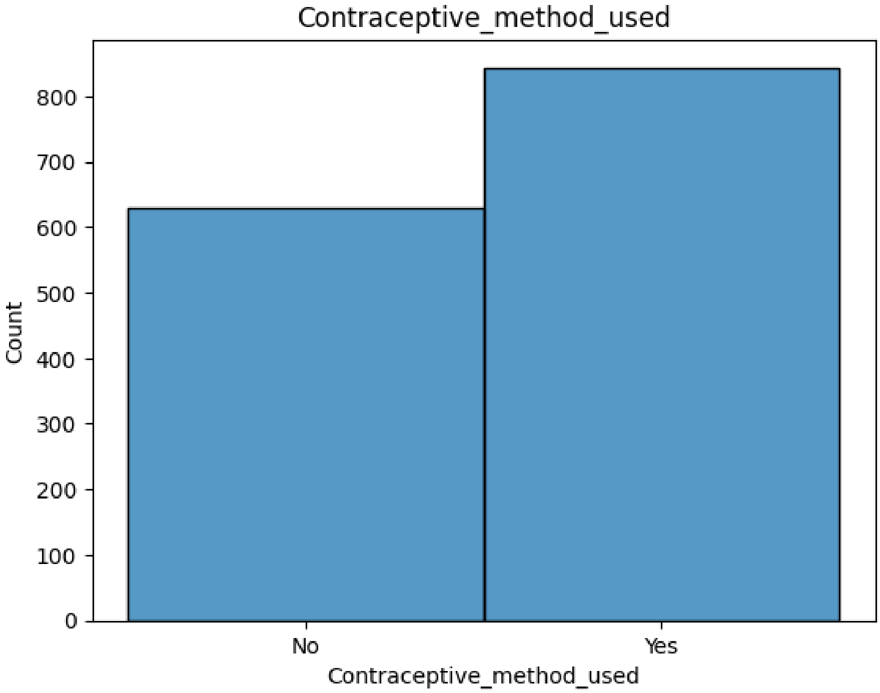
Overwhelming majority of women have scientology religious beliefs. Whereas, least number of women who have religious belief is non- scientology.



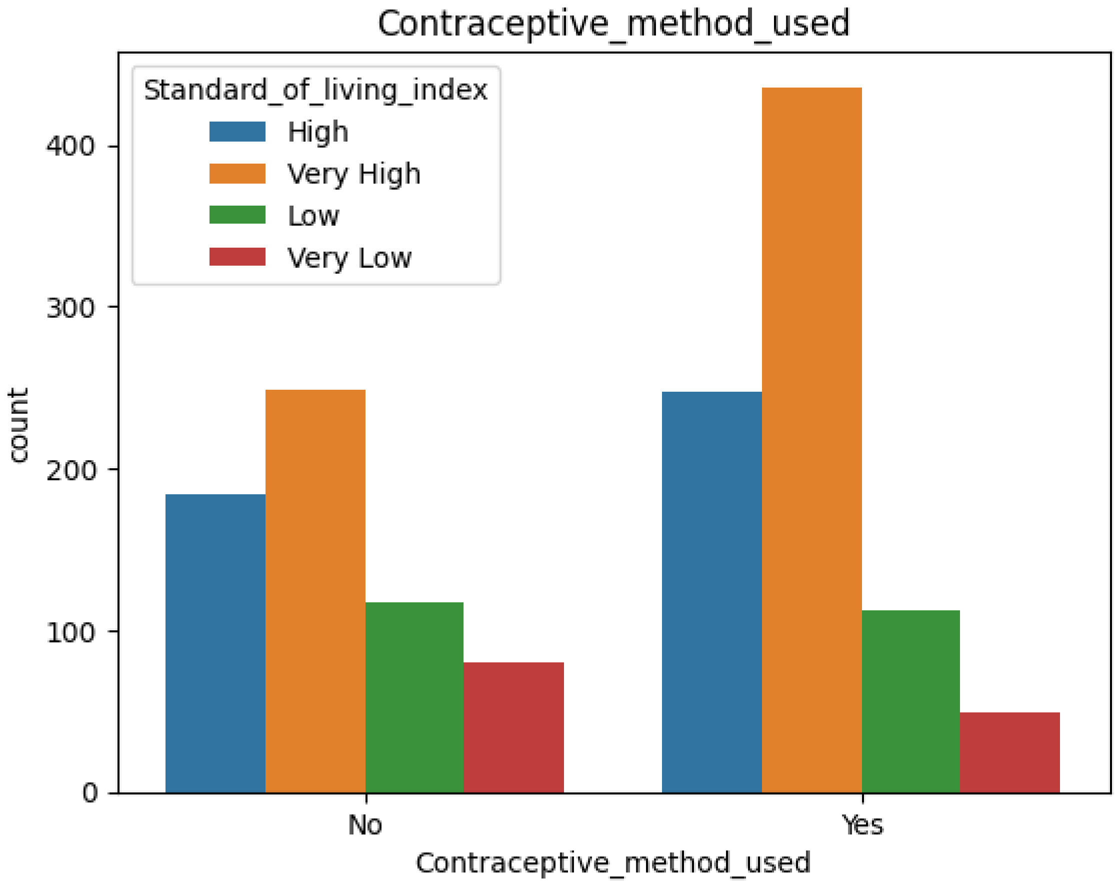
Most women from sample population are not working, less than half of the majority are working.



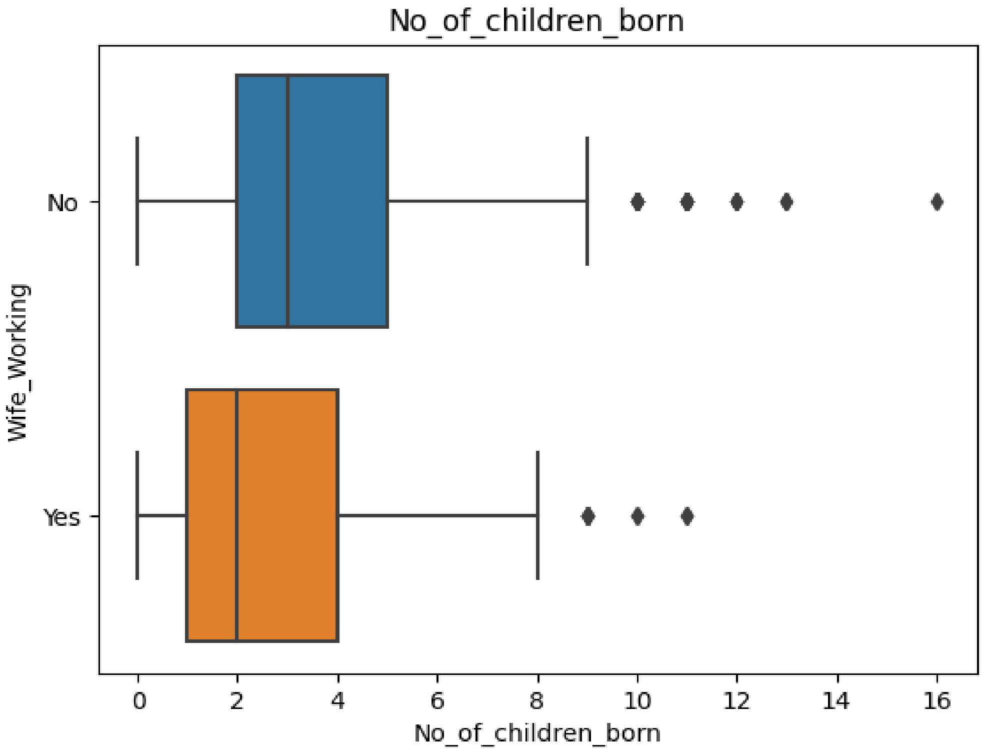
Most women live a very high standard of living, Whereas minority of women live a very low standard of living.



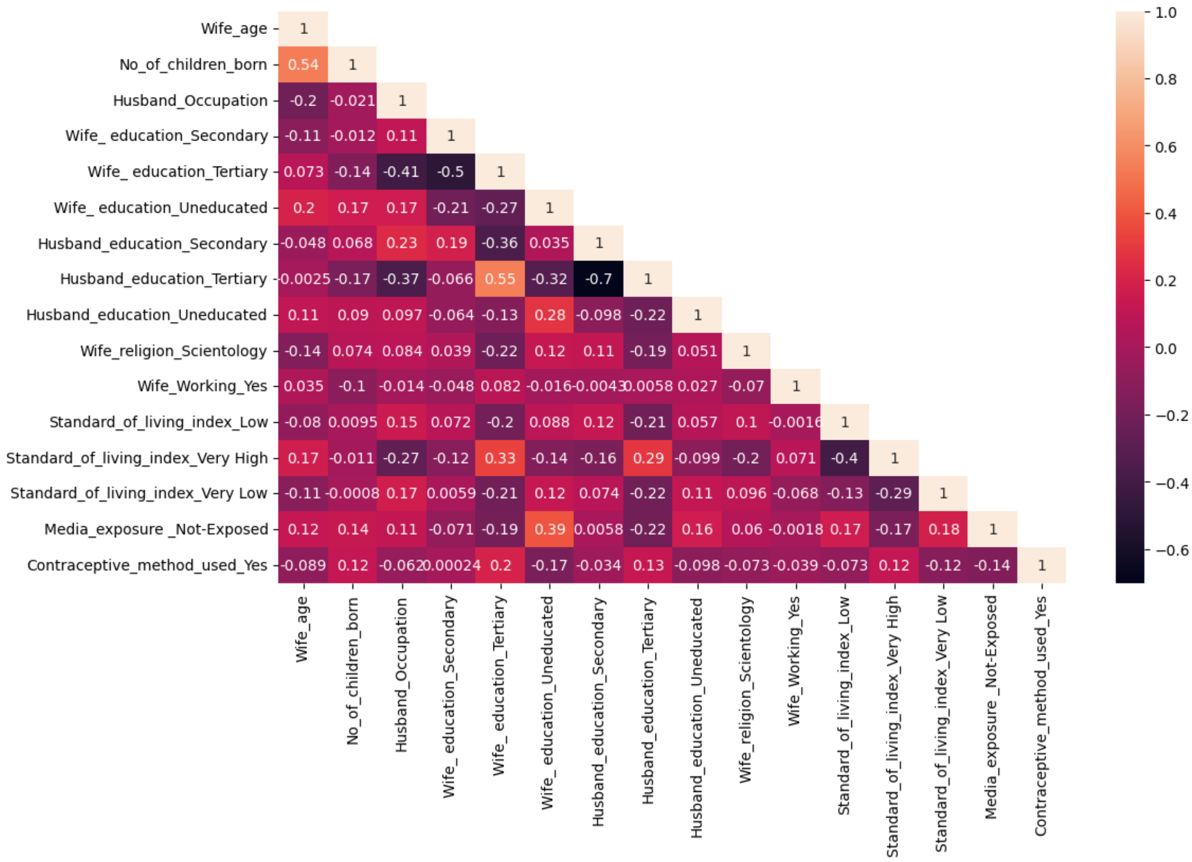
Most women from the sample population use contraceptives. Whereas significant part of the sample population do not use contraceptives.



From the above given plot we can see the distribution of married women of different living standards based on whether they use contraceptives or not. We can see that the majority of women with very high standard of living use contraceptives and fewer people with very low standard of living use contraceptives. Interestingly, the population that do not use contraceptives are distributed similarly to that of the population that use contraceptives.



From the given box plot we can see the distribution of married women based on the number of children they have and whether they work or not. We can see that both boxplots are positively skewed with few outliers. But the median number of children for working women is lesser than unemployed women.



Above given heatmap shows us the correlation among the variables. We can certainly see that almost all the variables are not correlated to each other. Except for Husband education and wife education are somewhat positively correlated. Another set of variables that are somewhat positively correlated are no_of_children_born and wife_age.

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.

- Firstly, a new dataset was created by dummy encoding the given dataset for better performance.
- The X and Y variables were coded and the dataset was split onto train and test data in 70:30 ratio.
- The split data was then used to create a logistic regression model with ‘newton-cg’ as solver, max-iteration of 10000, ‘none’ penalty, n-jobs at 2 and verbose set at true.
- A Linear discriminant analysis model was also created with the split dataset.
- A Classification and Regression tree was created on the dataset and classified further to create an interpretable CART tree.
- The results of all the applications will be given in 2.4 for inference.

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.

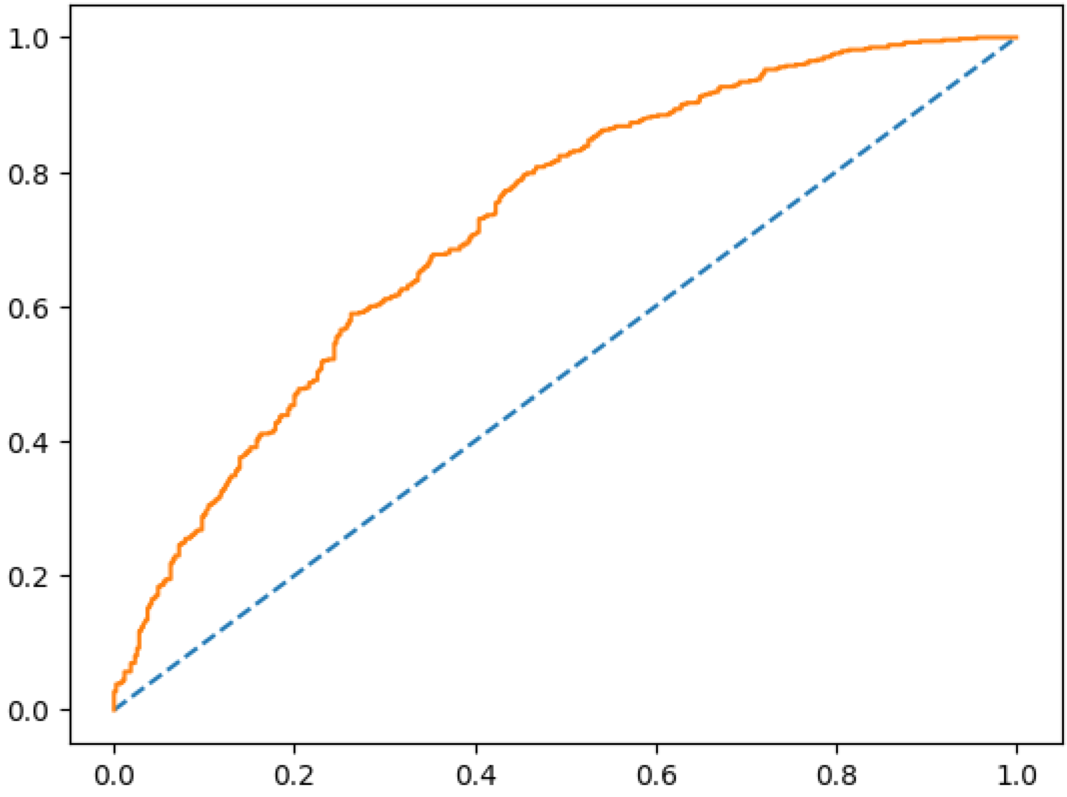
LOGISTIC REGRESSION MODEL

MODEL.SCORE

Model score gives us the accuracy score of the model in predicting the dependent variable.
The train model score is 0.68.
The test model score is 0.63.
Since, there are no huge differences between the scores. We can certainly say that the test model is not underfit, thus having the interpretability of the train model.

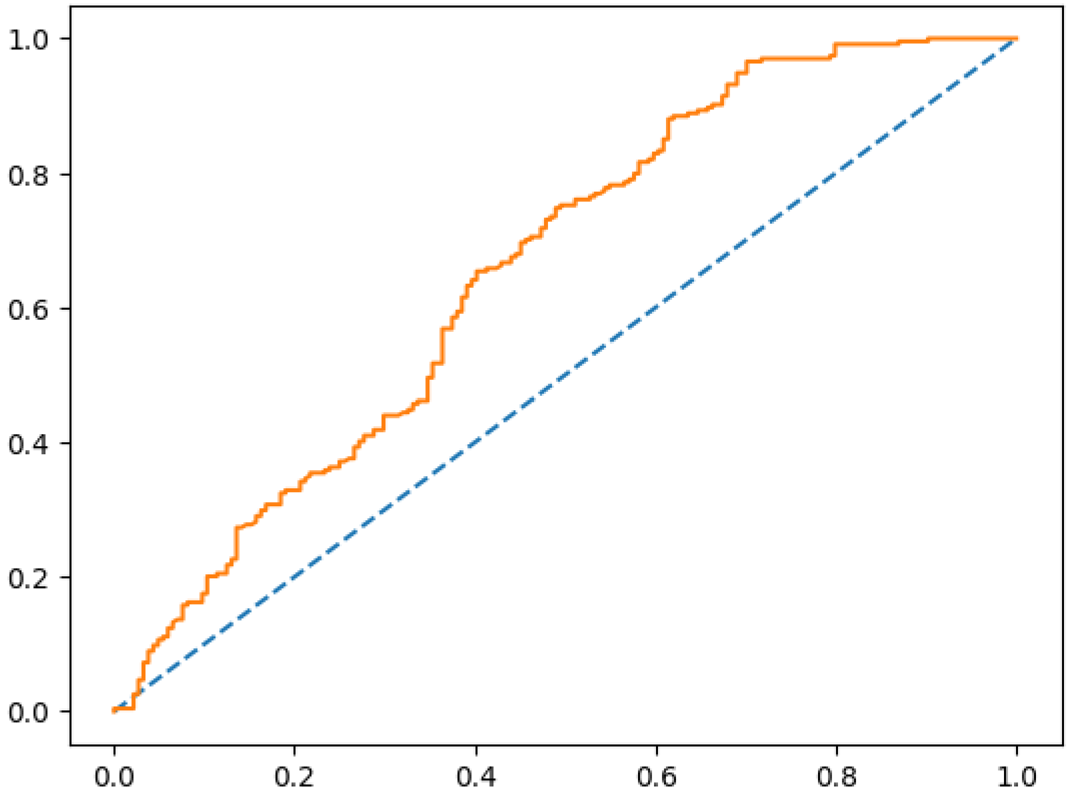
ROC CURVE & ROC AUC SCORE

Receiver operating characteristic curve shows the performance of the classification model under all classification thresholds. The ROC curve of the train model is as follows:



The larger the curve from the dotted line, the better the classification model. Here, the curve is slightly curved. The ROC AUC score for the train model is 0.724. Usually ROC AUC score closer to and higher than 0.8 is favourable.

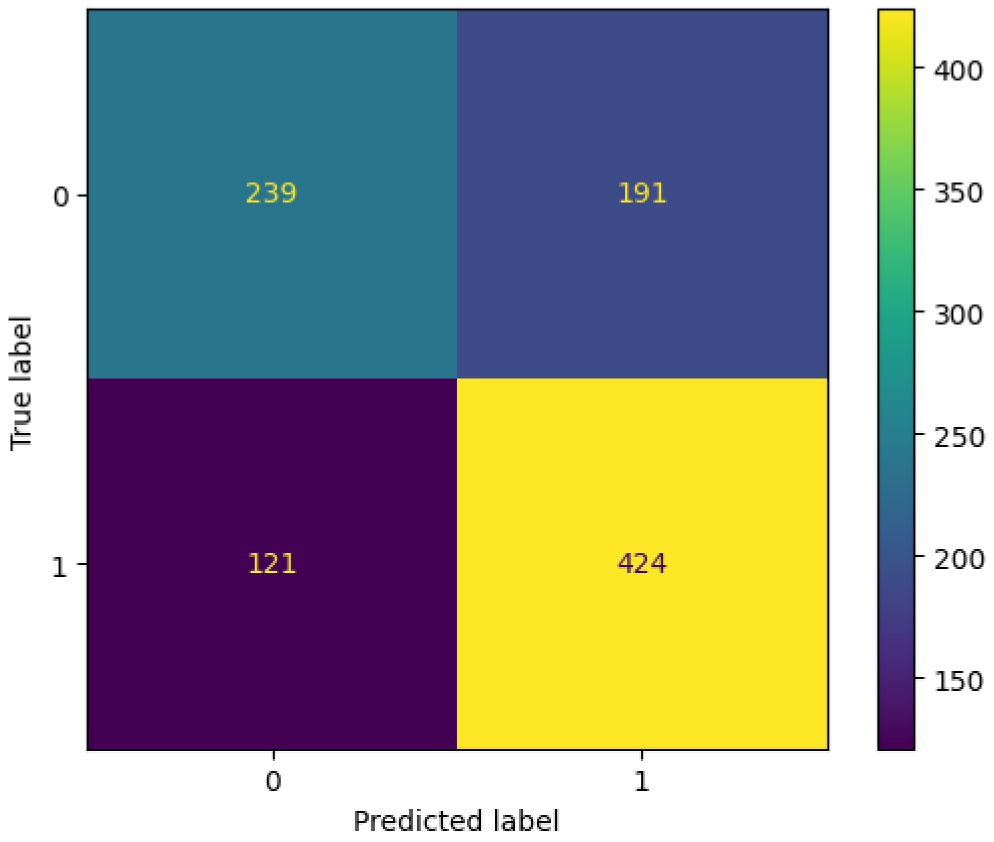
The ROC curve of the test model is as follows:

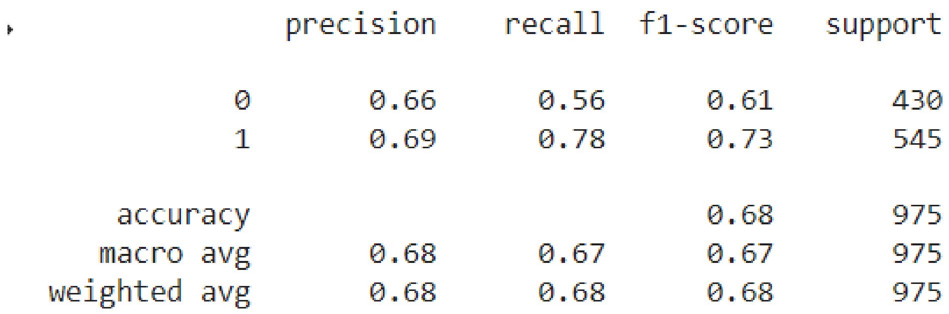


We can notice that the ROC curve for the test model is flatter and closer to the dotted line than the train data. Thus the performance of the test model is lesser than the train model. The ROC AUC score for the test model is 0.724, i.e., the same as the train model.

CONFUSION MATRIX & CLASSIFICATION REPORT

The confusion matrix is the table that defines the performance of the classification matrix. Classification report gives us the performance evaluation metrics like recall (the positive values that have been predicted correctly), precision (from all the classes we have predicted as positive, how many are actually positive), accuracy (From all the classes (positive and negative), how many of them we have predicted correctly) and F-measure (F-score helps to measure Recall and Precision at the same time). Below given is the confusion matrix and classification report of the train model.

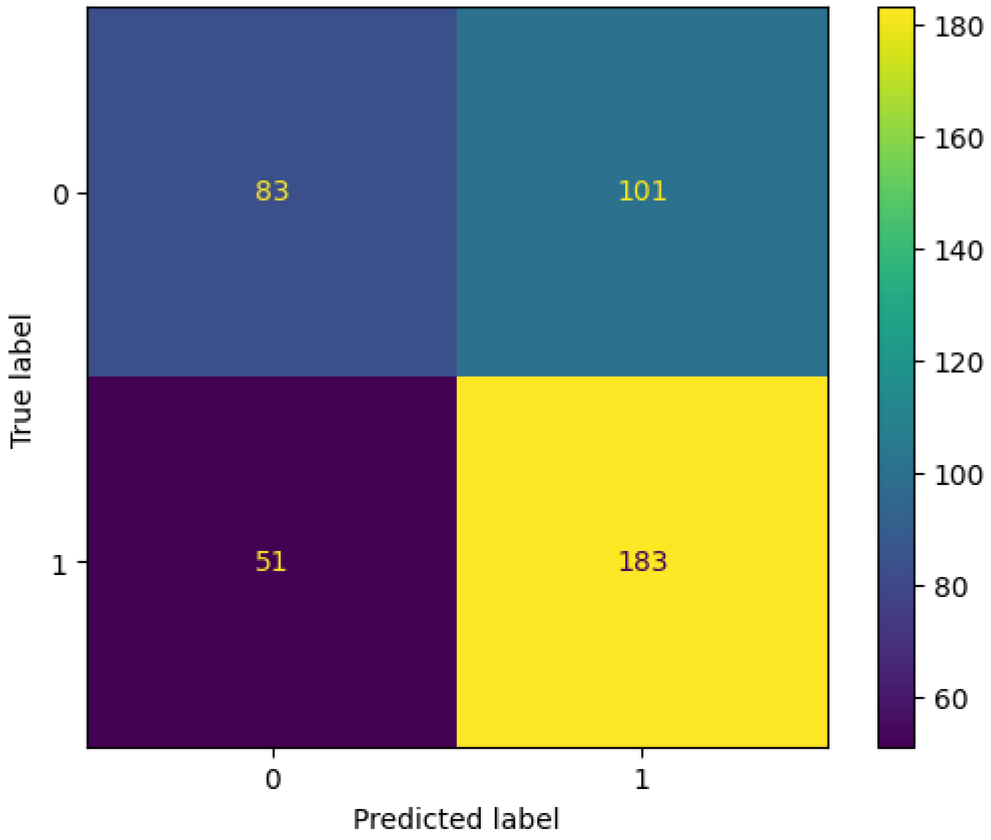


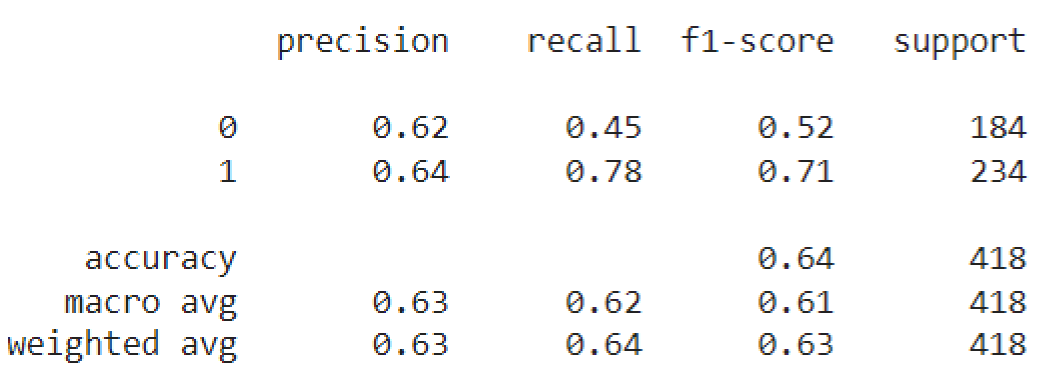


```
              precision    recall  f1-score   support

           0       0.66      0.56      0.61       430
           1       0.69      0.78      0.73       545

    accuracy                           0.68       975
   macro avg       0.68      0.67      0.67       975
weighted avg       0.68      0.68      0.68       975
```

From the confusion matrix for the train model, we can notice that the true negative is 239, that is the prediction was negative and it is true. Whereas, the true positive, the prediction was positive and it is positive, is 424.
From the classification model we can notice that recall is 0.78, and precision is 0.69. Which can be considered a decent score, since the higher the recall and precision score the better. The f1 score is 0.73.

Below given is the confusion matrix for the test model,





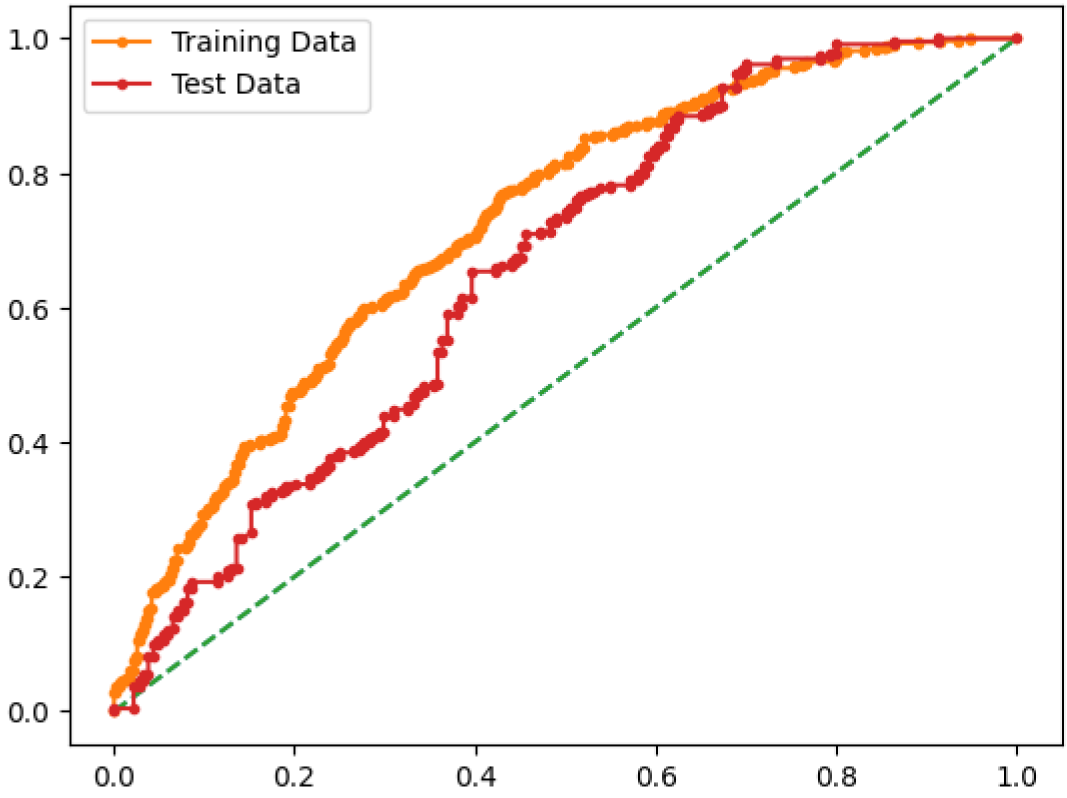
```
              precision    recall  f1-score   support

           0       0.62      0.45      0.52       184
           1       0.64      0.78      0.71       234

    accuracy                           0.64       418
   macro avg       0.63      0.62      0.61       418
weighted avg       0.63      0.64      0.63       418
```

From the confusion matrix we can notice that the true negative is 83, that is the prediction was negative and it is true. Whereas, the true positive, the prediction was positive and it is positive, is 183.
From the classification model we can notice that recall is 0.78, and precision is 0.64. Which can be considered a decent score, since the higher the recall and precision score (closer to 1) the better. The f1 score is 0.71.

LINEAR DISCRIMINANT ANALYSIS MODEL

ROC CURVE & ROC AUC SCORE

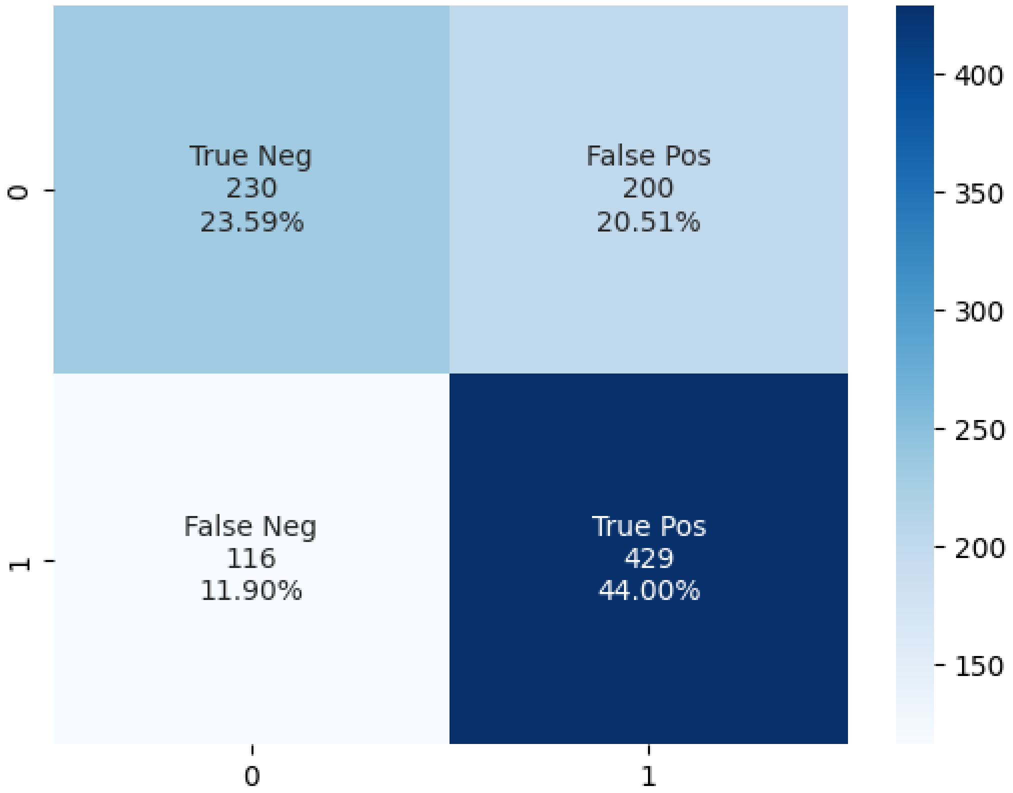
Receiver operating characteristic curve shows the performance of the classification model under all classification thresholds. The ROC curve of the train and test model is as follows:



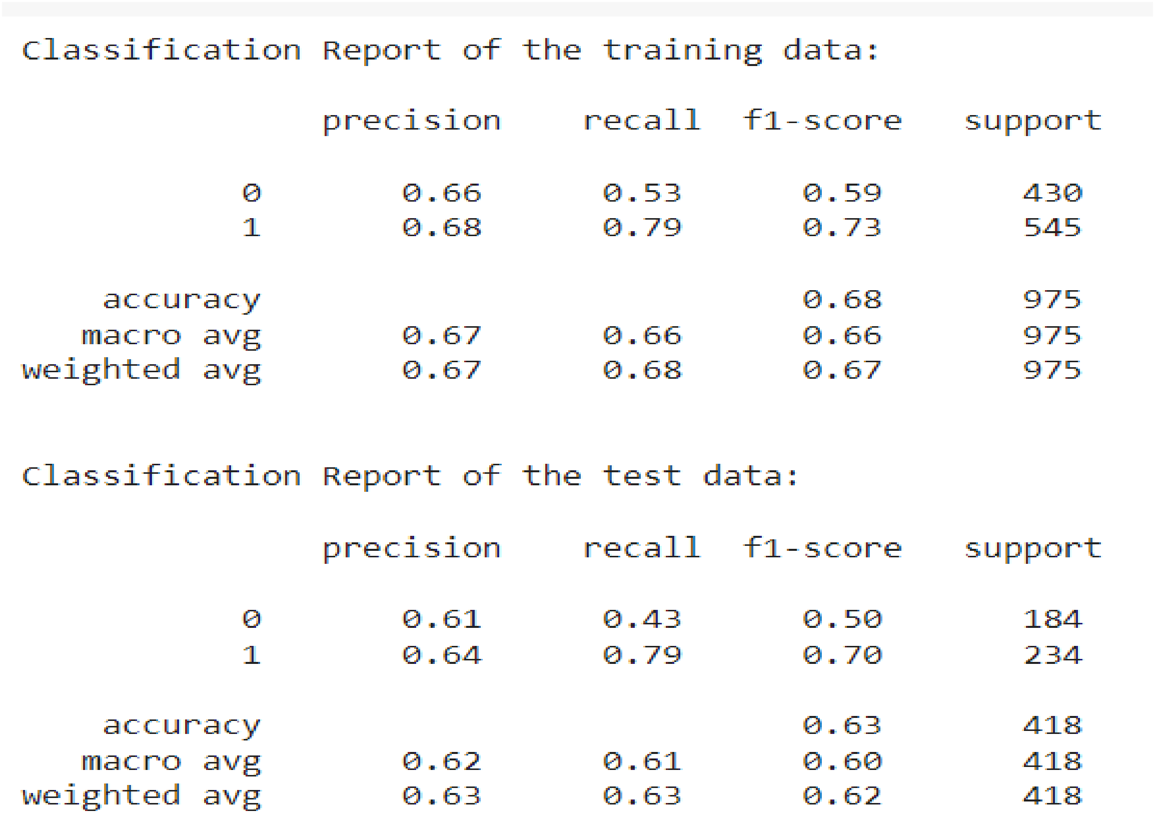
We can see that the ROC curve for the test model lies beneath the ROC curve for the train model. Thus, the prediction accuracy of the test model is not better than the train model. The ROC AUC score for the train model is 0.723 and the test model is 0.658. Thus, we can say that there is underfitting in the test model.

CONFUSION MATRIX & CLASSIFICATION REPORT

Below given is the confusion matrix and classification report of the train model.



From the confusion matrix we can see that 44% of the data is true positive and 23% of the data is true negative. Since, the model was able to predict the positives successfully, the model has good accuracy.



```
Classification Report of the training data:

              precision    recall  f1-score   support

           0       0.66      0.53      0.59       430
           1       0.68      0.79      0.73       545

    accuracy                           0.68       975
   macro avg       0.67      0.66      0.66       975
weighted avg       0.67      0.68      0.67       975


Classification Report of the test data:

              precision    recall  f1-score   support

           0       0.61      0.43      0.50       184
           1       0.64      0.79      0.70       234

    accuracy                           0.63       418
   macro avg       0.62      0.61      0.60       418
weighted avg       0.63      0.63      0.62       418
```

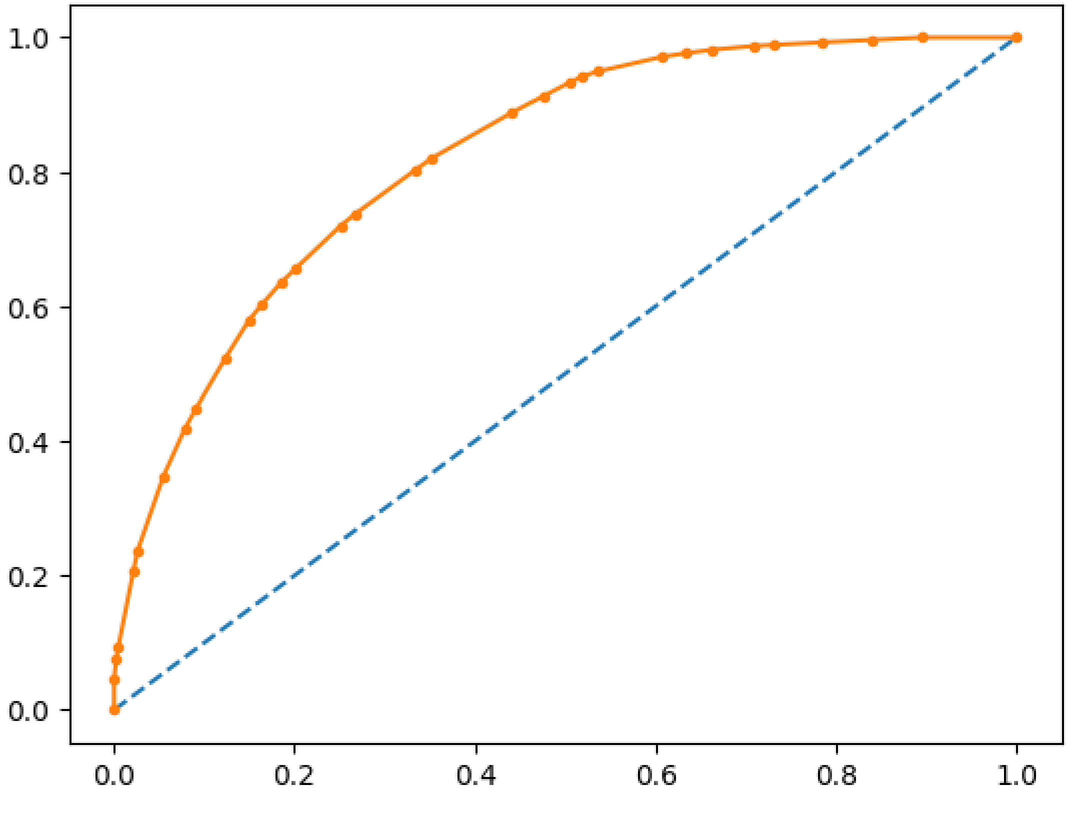
From the classification table for train data, we can see that the recall score is 0.79 and precision score is 0.68. These scores indicate that the model performs well in classifying the data. The f1 score amounts to 0.67.
Similarly, in the test model classification table, we can notice that the recall is 0.79 and precision is 0.64. Thus the test data has less precision score than the train data. The f1 score has increased from 0.67 to 0.70.
Thus we can say that the classification performance of both test and train model are well and similar.

CART

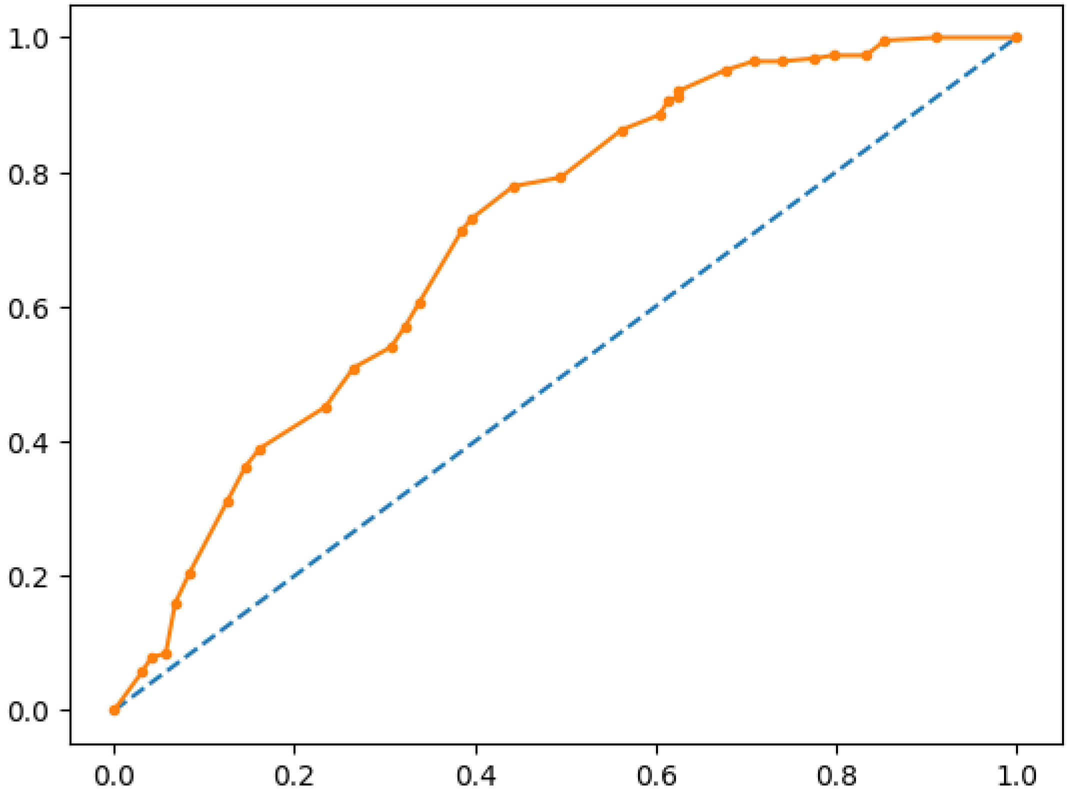
ACCURACY SCORE
The model accuracy score for the train model is 0.74. Whereas the accuracy score for the test model is 0.66. Thus, we can say that there is a case of underfitting with test model in terms of predictive accuracy.

ROC CURVE & ROC AUC SCORE

Receiver operating characteristic curve shows the performance of the classification model under all classification thresholds. The ROC curve of the train and test model is as follows:

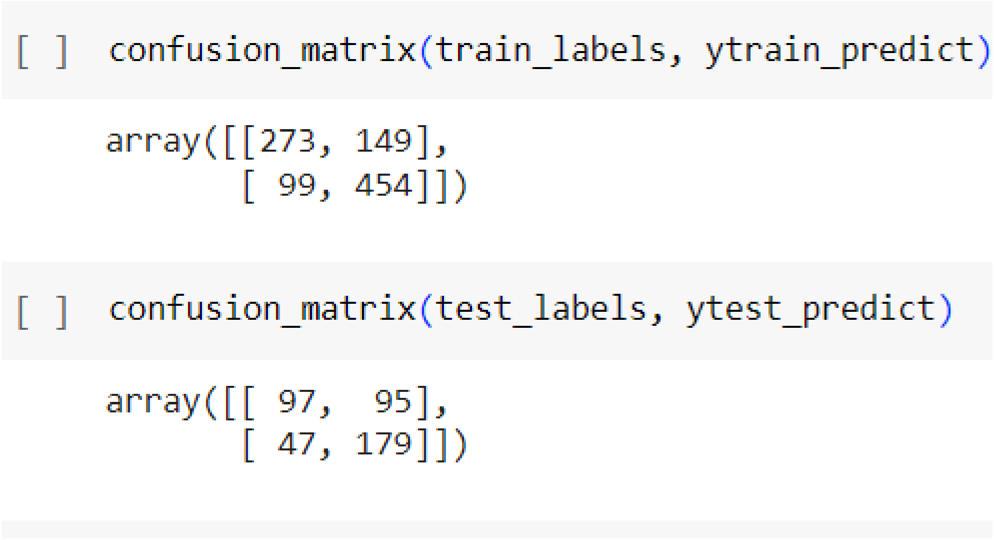


From the ROC AUC curve, we can see that the train model is performing very well, since there is a bigger area under the curve. The AUC score is 0.823, thus we can certainly say that the train model is performing very well as a classification model.



From the above given ROC AUC curve, we can see that the test model is not performing as well as the train model. THE AUC score of 0.708 further proves that the Classification performance of the test model is not as good as the train model.

CONFUSION MATRIX AND CLASSIFICATION REPORT

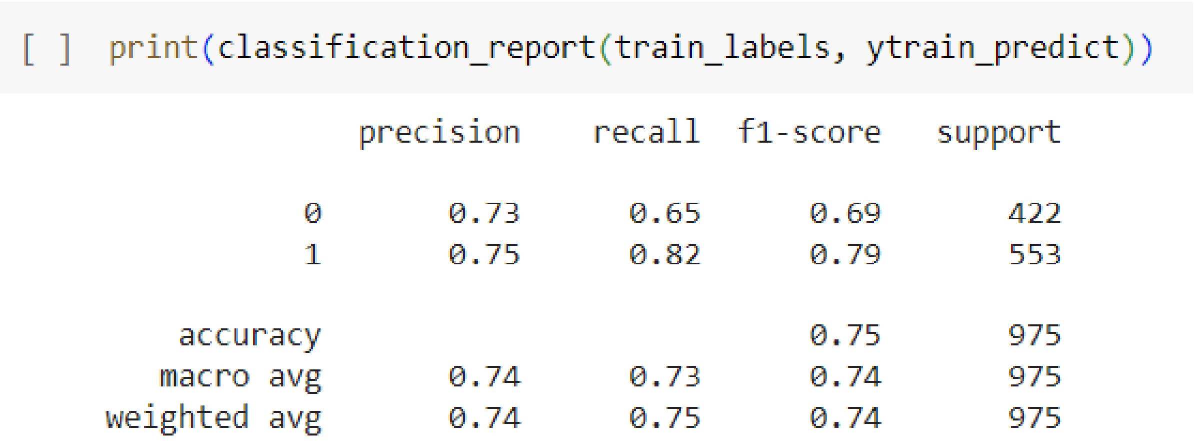


```
[ ]  confusion_matrix(train_labels, ytrain_predict)

     array([[273, 149],
            [ 99, 454]])
```

```
[ ]  confusion_matrix(test_labels, ytest_predict)

     array([[ 97,  95],
            [ 47, 179]])
```

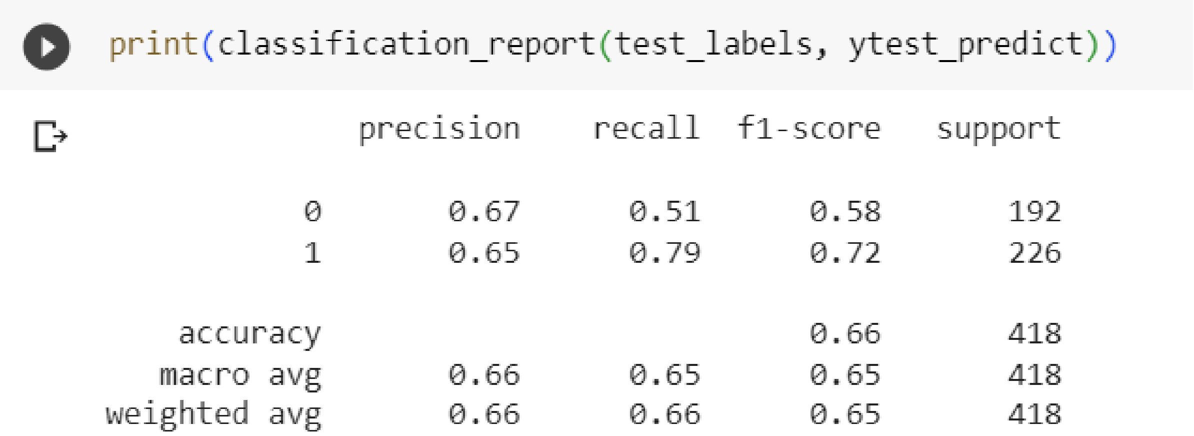
The confusion matrix of the train model tells us that there are 454 true positives and 273 true negatives. This is favourable since the model is effective in predicting the positive variables.

Similarly, the confusion matrix of the test model tells us that there are 179 true positives and 97 true negatives. This is generally favourable, however the The false positive ratio is very close to true negative. This could be troublesome, since the model may not have good predictability.



```
[ ]  print(classification_report(train_labels, ytrain_predict))
```

```
              precision    recall  f1-score   support

           0       0.73      0.65      0.69       422
           1       0.75      0.82      0.79       553

    accuracy                           0.75       975
   macro avg       0.74      0.73      0.74       975
weighted avg       0.74      0.75      0.74       975
```



```
print(classification_report(test_labels, ytest_predict))
```

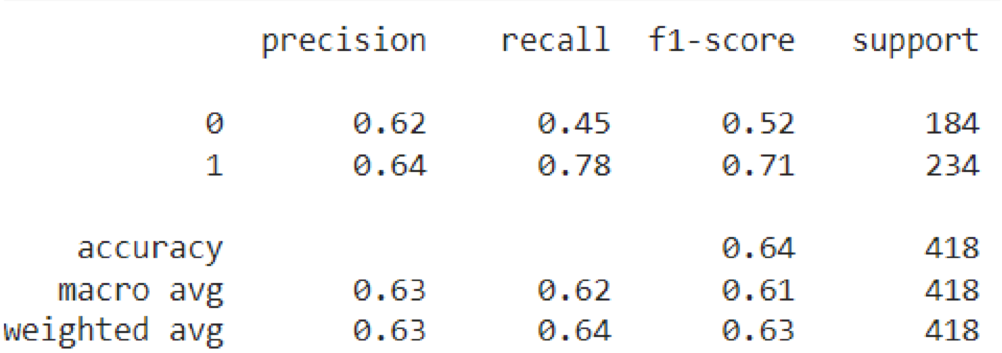
```
              precision    recall  f1-score   support

           0       0.67      0.51      0.58       192
           1       0.65      0.79      0.72       226

    accuracy                           0.66       418
   macro avg       0.66      0.65      0.65       418
weighted avg       0.66      0.66      0.65       418
```

Above given are the classification report for the train model, where the precision is 0.75 and recall is 0.82, which is generally favourable. The f1 score amounts to 0.79.

The test model has the precision of 0.65 and recall of 0.79. The f1 score is 0.65. Thus, we can see that the performance indicators of the test model is lesser than the train model. Thus, we can certainly say that the test model is not as optimised as the train model.

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

LOGISTIC REGRESSION MODEL



```
              precision    recall  f1-score   support

           0       0.62      0.45      0.52       184
           1       0.64      0.78      0.71       234

    accuracy                           0.64       418
   macro avg       0.63      0.62      0.61       418
weighted avg       0.63      0.64      0.63       418
```

From the produced test model, we can infer that:

- The model has an overall accuracy of 63%. That means the model is good at predicting 63% of the dependent variable.
- For predicting whether a married women would not use contraceptives (Label 0):
- Precision (62%) – 62% of married women predicted are actually not using contraceptives out of all married women predicted to have.
- Recall (45%) – Out of all the married women who are actually not using contraceptives, 45% of employees have been predicted correctly .
- For predicting salary >50k (Label 1):
- Precision (64%) – 64% of employees predicted are actually using contraceptives out of all married women predicted to have.
- Recall (78%) – Out of all the married who are actually using contraceptives , 78% of the married women have been predicted correctly .

LDA

Below given are the resulting coefficients of the LDA model:

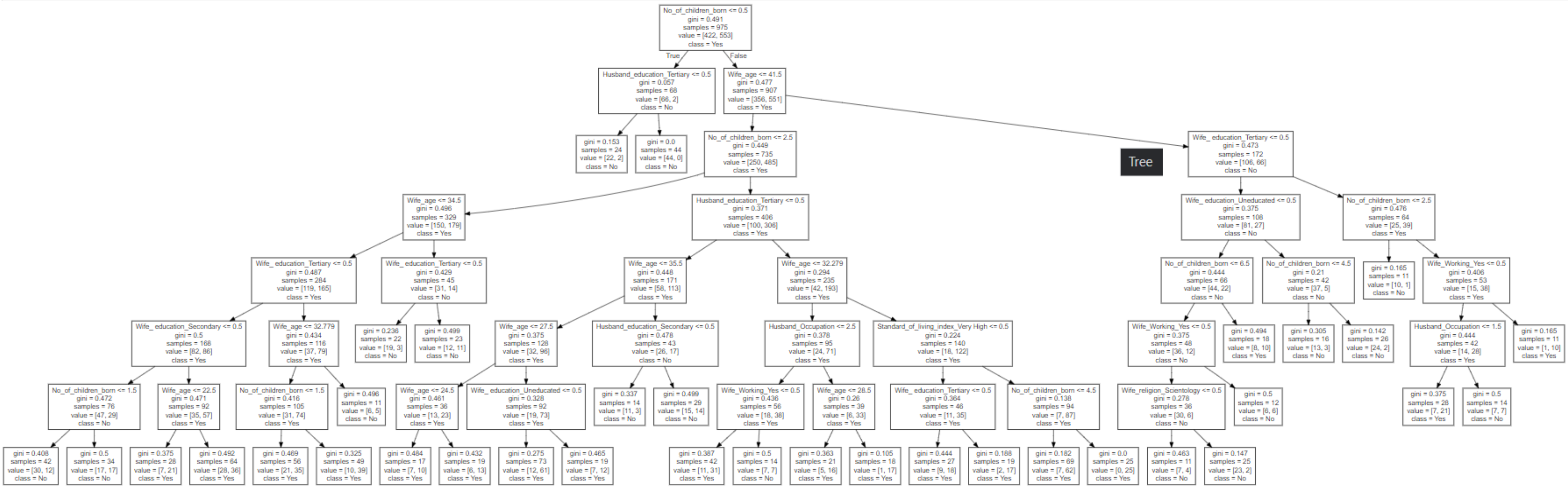
Intercept : 1.04546371
Wife_age : -0.08
No_of_children_born : 0.33
Husband_Occupation : 0.15
Wife_ education_Secondary : 0.44
Wife_ education_Tertiary : 1.17
Wife_ education_Uneducated : -0.29
Husband_education_Secondary : 0.34
Husband_education_Tertiary : 0.26
Husband_education_Uneducated : 0.
Wife_Working_Yes : -0.18
Standard_of_living_index_Low : -0.36
Standard_of_living_index_Very High : 0.34
Standard_of_living_index_Very Low : -0.52
Media_exposure _Not-Exposed : -0.41

Thus we can infer that:

- predictor 'Wife_ education_Secondary' has the largest magnitude thus this helps in classifying the best.
- predictor 'Standard_of_living_index_Very Low' has the smallest magnitude thus this helps in classifying the least.

CART

Below given CART tree is the final model.


No_of_children_born <= 0.5 gini = 0.491 samples = 975 value = [422, 553] class = Yes
True
False
Husband_education_Tertiary <= 0.5 gini = 0.057 samples = 68 value = [66, 2] class = No
Wife_age <= 41.5 gini = 0.477 samples = 907 value = [356, 551] class = Yes
gini = 0.153 samples = 24 value = [22, 2] class = No
gini = 0.0 samples = 44 value = [44, 0] class = No
No_of_children_born <= 2.5 gini = 0.449 samples = 735 value = [250, 485] class = Yes
Tree
Wife_ education_Tertiary <= 0.5 gini = 0.473 samples = 172 value = [106, 66] class = No
Wife_age <= 34.5 gini = 0.496 samples = 329 value = [150, 179] class = Yes
Husband_education_Tertiary <= 0.5 gini = 0.371 samples = 406 value = [100, 306] class = Yes
Wife_ education_Uneducated <= 0.5 gini = 0.375 samples = 108 value = [81, 27] class = No
No_of_children_born <= 2.5 gini = 0.476 samples = 64 value = [25, 39] class = Yes
Wife_ education_Tertiary <= 0.5 gini = 0.487 samples = 284 value = [119, 165] class = Yes
Wife_ education_Tertiary <= 0.5 gini = 0.429 samples = 45 value = [31, 14] class = No
Wife_age <= 35.5 gini = 0.448 samples = 171 value = [58, 113] class = Yes
Wife_age <= 32.279 gini = 0.294 samples = 235 value = [42, 193] class = Yes
No_of_children_born <= 6.5 gini = 0.444 samples = 66 value = [44, 22] class = No
No_of_children_born <= 4.5 gini = 0.21 samples = 42 value = [37, 5] class = No
gini = 0.165 samples = 11 value = [10, 1] class = No
Wife_Working_Yes <= 0.5 gini = 0.406 samples = 53 value = [15, 38] class = Yes
Wife_ education_Secondary <= 0.5 gini = 0.5 samples = 168 value = [82, 86] class = Yes
Wife_age <= 32.779 gini = 0.434 samples = 116 value = [37, 79] class = Yes
gini = 0.236 samples = 22 value = [19, 3] class = No
gini = 0.499 samples = 23 value = [12, 11] class = No
Wife_age <= 27.5 gini = 0.375 samples = 128 value = [32, 96] class = Yes
Husband_education_Secondary <= 0.5 gini = 0.478 samples = 43 value = [26, 17] class = No
Husband_Occupation <= 2.5 gini = 0.378 samples = 95 value = [24, 71] class = Yes
Standard_of_living_index_Very High <= 0.5 gini = 0.224 samples = 140 value = [18, 122] class = Yes
Wife_Working_Yes <= 0.5 gini = 0.375 samples = 48 value = [36, 12] class = No
gini = 0.494 samples = 18 value = [8, 10] class = Yes
gini = 0.305 samples = 16 value = [13, 3] class = No
gini = 0.142 samples = 26 value = [24, 2] class = No
Husband_Occupation <= 1.5 gini = 0.444 samples = 42 value = [14, 28] class = Yes
gini = 0.165 samples = 11 value = [1, 10] class = Yes
No_of_children_born <= 1.5 gini = 0.472 samples = 76 value = [47, 29] class = No
Wife_age <= 22.5 gini = 0.471 samples = 92 value = [35, 57] class = Yes
No_of_children_born <= 1.5 gini = 0.416 samples = 105 value = [31, 74] class = Yes
gini = 0.496 samples = 11 value = [6, 5] class = No
Wife_age <= 24.5 gini = 0.461 samples = 36 value = [13, 23] class = Yes
Wife_ education_Uneducated <= 0.5 gini = 0.328 samples = 92 value = [19, 73] class = Yes
gini = 0.337 samples = 14 value = [11, 3] class = No
gini = 0.499 samples = 29 value = [15, 14] class = No
Wife_Working_Yes <= 0.5 gini = 0.436 samples = 56 value = [18, 38] class = Yes
Wife_age <= 28.5 gini = 0.26 samples = 39 value = [6, 33] class = Yes
Wife_ education_Tertiary <= 0.5 gini = 0.364 samples = 46 value = [11, 35] class = Yes
No_of_children_born <= 4.5 gini = 0.138 samples = 94 value = [7, 87] class = Yes
Wife_religion_Scientology <= 0.5 gini = 0.278 samples = 36 value = [30, 6] class = No
gini = 0.5 samples = 12 value = [6, 6] class = No
gini = 0.375 samples = 28 value = [7, 21] class = Yes
gini = 0.5 samples = 14 value = [7, 7] class = No
gini = 0.408 samples = 42 value = [30, 12] class = No
gini = 0.5 samples = 34 value = [17, 17] class = No
gini = 0.375 samples = 28 value = [7, 21] class = Yes
gini = 0.492 samples = 64 value = [28, 36] class = Yes
gini = 0.469 samples = 56 value = [21, 35] class = Yes
gini = 0.325 samples = 49 value = [10, 39] class = Yes
gini = 0.484 samples = 17 value = [7, 10] class = Yes
gini = 0.432 samples = 19 value = [6, 13] class = Yes
gini = 0.275 samples = 73 value = [12, 61] class = Yes
gini = 0.465 samples = 19 value = [7, 12] class = Yes
gini = 0.387 samples = 42 value = [11, 31] class = Yes
gini = 0.5 samples = 14 value = [7, 7] class = No
gini = 0.363 samples = 21 value = [5, 16] class = Yes
gini = 0.105 samples = 18 value = [1, 17] class = Yes
gini = 0.444 samples = 27 value = [9, 18] class = Yes
gini = 0.188 samples = 19 value = [2, 17] class = Yes
gini = 0.182 samples = 69 value = [7, 62] class = Yes
gini = 0.0 samples = 25 value = [0, 25] class = Yes
gini = 0.463 samples = 11 value = [7, 4] class = No
gini = 0.147 samples = 25 value = [23, 2] class = No

- The final regularised model had a minimum sample split of thirty.
- Accuracy on the Training Data: 74%
  Accuracy on the Test Data: 66%

The accuracy of the test data is lesser than the training data. This might be because of lesser sample split and underfitting. However, the test model has high recall value, thus there is some significant probability of the model to predict whether a married woman is using contraceptives or not.

- AUC on the Training Data: 82.3%
  AUC on the Test: 70.8%

The AUC of the test is less than training data due to underfitting. Thus it has a lesser possibility of predicting whether a married woman uses contraceptives or not.

- Wife_age and no_of_children_born are the most important variables in determining whether a married woman is using contraceptives or not.