PREDICTIVE MODELLING CODED PROJECT

BY

SRINIVASAN T

Table of Content:

List of Tables:

List of Figures:

Scoring guide (Rubric) - PM Project Rubric

Criteria	Points
Problem 1 - Define the problem and perform exploratory Data Analysis - Problem definition - Check shape, Data types, statistical summary - Univariate analysis - Multivariate analysis - Use appropriate visualizations to identify the patterns and insights - Key meaningful observations on individual variables and the relationship between variables	8
Problem 1 - Data Pre-processing Prepare the data for modelling: - Missing Value Treatment (if needed) - Outlier Detection (treat, if needed) - Feature Engineering - Encode the data - Train-test split	5
Problem 1- Model Building - Linear regression - Apply linear Regression using Sklearn - Using Statsmodels Perform checks for significant variables using the appropriate method - Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare.	8
Problem 1 - Business Insights & Recommendations - Comment on the Linear Regression equation from the final model and impact of relevant variables (atleast 2) as per the equation - Conclude with the key takeaways (actionable insights and recommendations) for the business	5
Problem 2 - Define the problem and perform exploratory Data Analysis - Problem definition - Check shape, Data types, statistical summary - Univariate analysis - Multivariate analysis - Use appropriate visualizations to identify the patterns and insights - Key meaningful observations on individual variables and the relationship between variables	8
Problem 2 - Data Pre-processing Prepare the data for modelling: - Missing value Treatment (if needed) - Outlier Detection(treat, if needed) - Feature Engineering (if needed) - Encode the data - Train-test split	3
Problem 2 - Model Building and Compare the Performance of the Models - Build a Logistic Regression model - Build a Linear Discriminant Analysis model - Build a CART model - Prune the CART model by finding the best hyperparameters using GridSearch - Check the performance of the models across train and test set using different metrics - Compare the performance of all the models built and choose the best one with proper rationale	11
Problem 2 - Business Insights & Recommendations - Comment on the importance of features based on the best model - Conclude with the key takeaways (actionable insights and recommendations) for the business	6
Business Report Quality - Adhere to the business report checklist	6

Problem 1

Data Description

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 transferred 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 Define the problem and perform exploratory Data Analysis

1.1.1 Problem definition

The comp-activ database comprises activity measures of computer systems. Data was gathered from a Sun Sparcstation 20/712 with 128 Mbytes of memory, operating in a multi-user university department. Users engaged in diverse tasks, such as internet access, file editing, and CPU-intensive programs.

Being an aspiring data scientist, you aim to establish a linear equation for predicting 'usr' (the percentage of time CPUs operate in user mode). Your goal is to analyse various system attributes to understand their influence on the system's 'usr' mode.

Loading the Dataset:

The dataset can be loaded using the head function to check whether the data is loaded properly or not. The dataset is properly loaded.

lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	u
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	
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	
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	
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	
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	

Table 1.1: Loading the Dataset

1.1.2 Checking the Shape:

The shape of the dataset can be determined

```
The No. of Rows    =  8192
The No. of Columns =  22
```

Figure 1.1: Shape of the Dataset

1.1.3 Data Types:

The datatypes can be identified using the info function

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8192 entries, 0 to 8191
Data columns (total 22 columns):
 #   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(1)
memory usage: 1.4+ MB
```

Figure 1.2: Data Types

There are 21 numerical and 1 categorical variable
There are 2 variables rchar & wchar has null values

1.1.4 Statistical Summary

The statistical summary can be derived from the describe function.

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

Table 1.2: Statistical Summary Numerical Variables

Number of characters transferred per second by system read calls is higher at all percentile than system write calls
Most of the variables having the min value as 0 and values starts at 75%
scall, sread, swrite, rchar, wchar, pflt, vflt & freemen having distribution across all the percentile

	count	unique	top	freq
runqsz	8192	2	Not_CPU_Bound	4331

Table 1.3: Statistical Summary Object Variable

The object variable runqsz has 2 types with Not_CPU_Bound has the highest frequency of 4331

1.1.5 Univariate analysis:

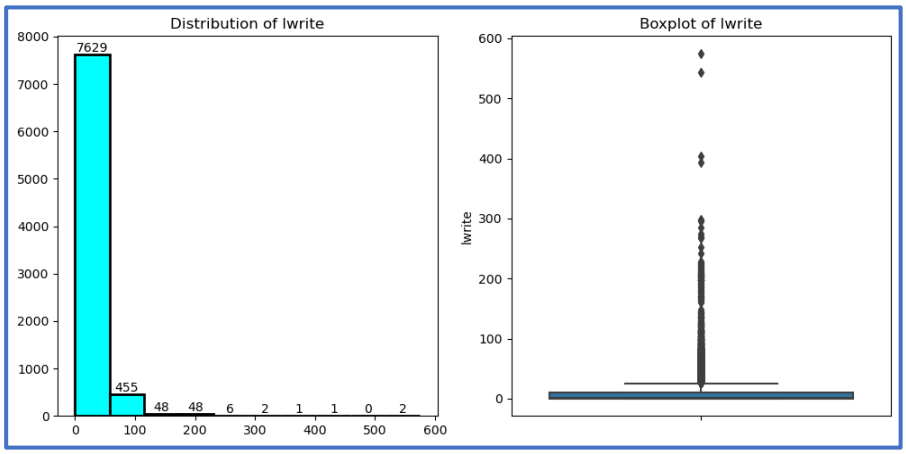

Figure 1.3: Distribution of lwrite

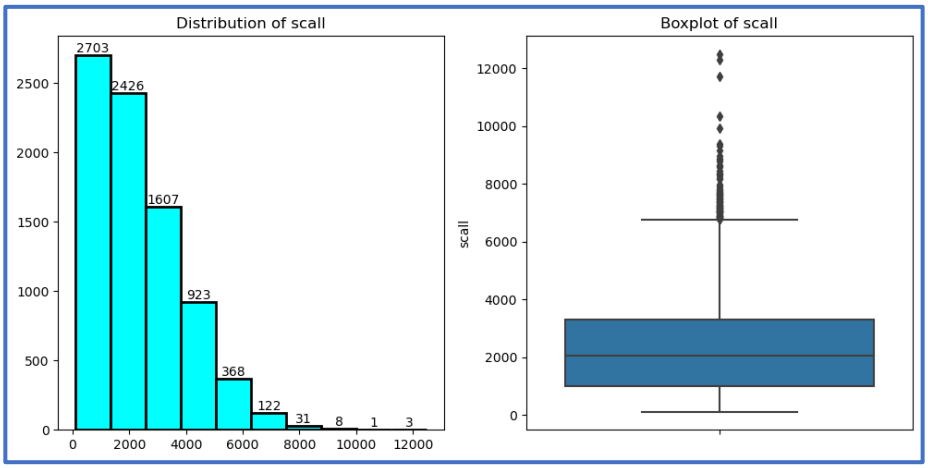

Figure 1.4: Distribution of scall

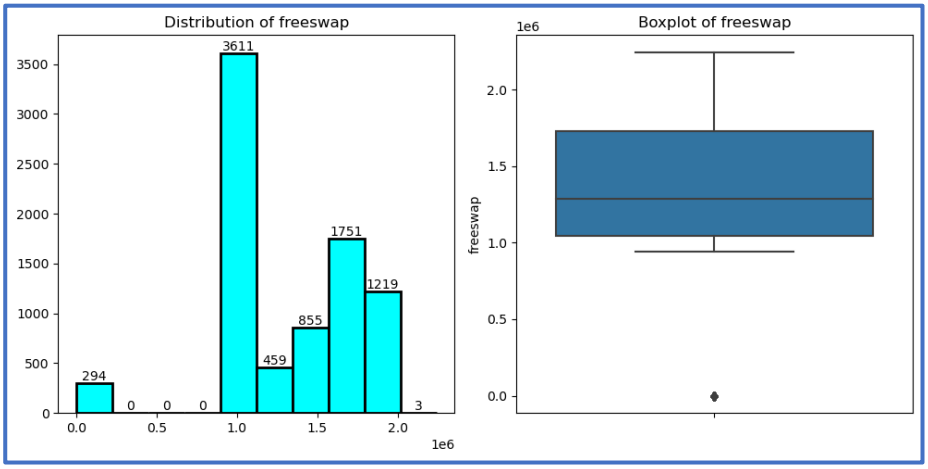

Figure 1.5: Distribution of freeswap

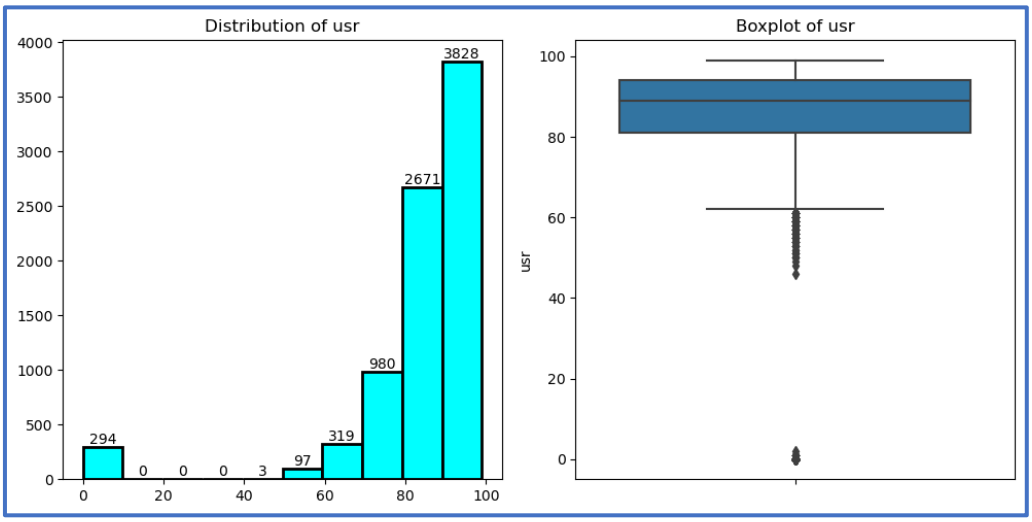

Figure 1.6: Distribution of usr

All the variables has outliers
Dependent variable usr & independent variable freeswap are left skewed and all the other variables are right skewed
Also, these two variables have outliers below their lower limit

1.1.6 Multivariate analysis

1.1.6.1 Numerical vs Numerical

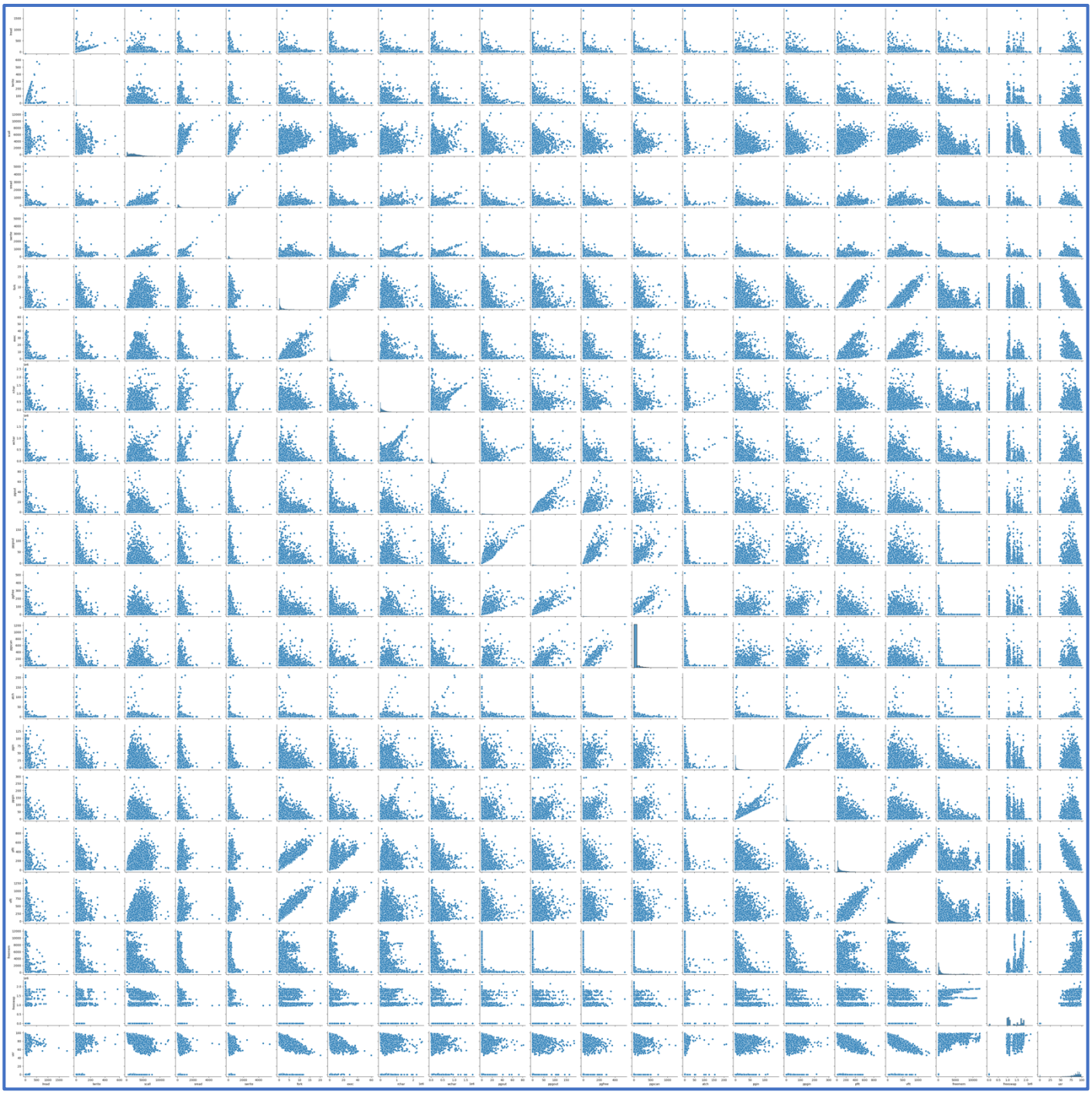

Figure 1.7: Numerical vs Numerical Variable Analysis

Variable fork is positively correlated with pflt, vflt & exec variables
Variable pgin is positively correlated with ppgin
Variable vflt is positively correlated with pflt
Variable freeswap does not seems to have a correlation with any of the variables
Variable usr is negatively correlated with pflt & vflt

1.1.6.2 Numerical vs Categorical

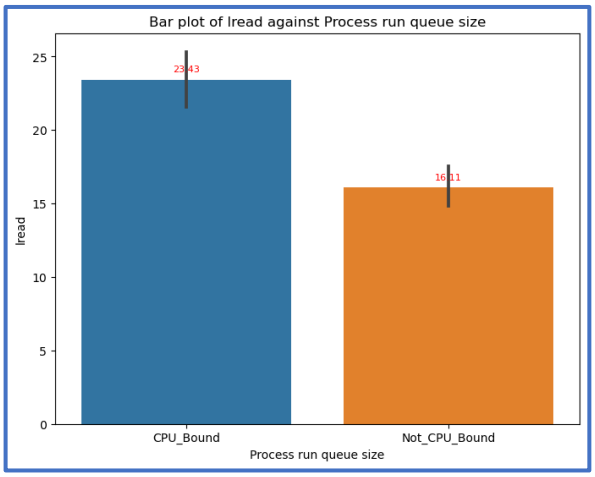

Figure 1.8: Count of lread

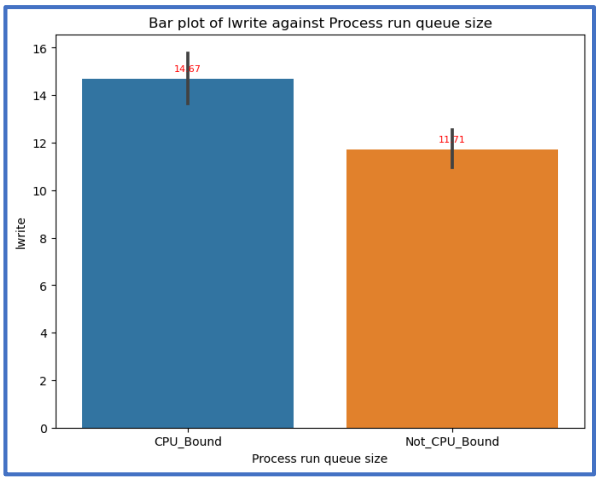

Figure 1.9: Count of lwirte

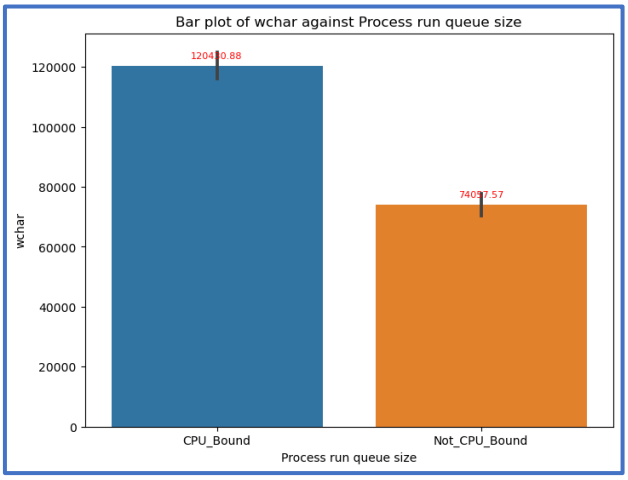

Figure 1.10: Count of wchar

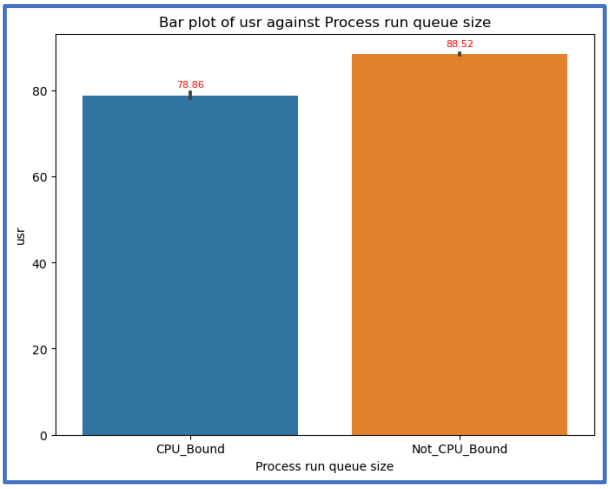

Figure 1.11: Count of usr

Except the variables usr, freeswap, freemen all other variables are having higher value for CPU_Bound variable

1.1.7 Patterns and insights - Key meaningful observations

- Number of characters transferred per second by system read calls is higher at all percentile than system write calls
- Most of the variables having the min value as 0 and values starts at 75%
- scall, sread, swrite, rchar, wchar, pflt, vflt & freemen having distribution across all the percentile
- The object variable runqsz has 2 types with Not_CPU_Bound has the highest frequency of 4331
- All the variables has outliers
- Dependent variable usr & independent variable freeswap are left skewed and all the other variables are right skewed
- Also, these two variables have outliers below their lower limit
- Variable fork is positively correlated with pflt, vflt & exec variables
- Variable pgin is positively correlated with ppgin
- Variable vflt is positively correlated with pflt
- Variable freeswap does not seems to have a correlation with any of the variables
- Variable usr is negatively correlated with pflt & vflt
- Except the variables usr, freeswap, freemen all other variables are having higher value to be a CPU_Bound variable

1.2 Data Pre-processing

1.2.1 Missing Values

The variables rchar and wchar has missing values.

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

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

Figure 1.12: Missing value treatment before and after

The missing values are treated using the fillna function with their median.

1.2.2 Outlier Treatment:

Before:

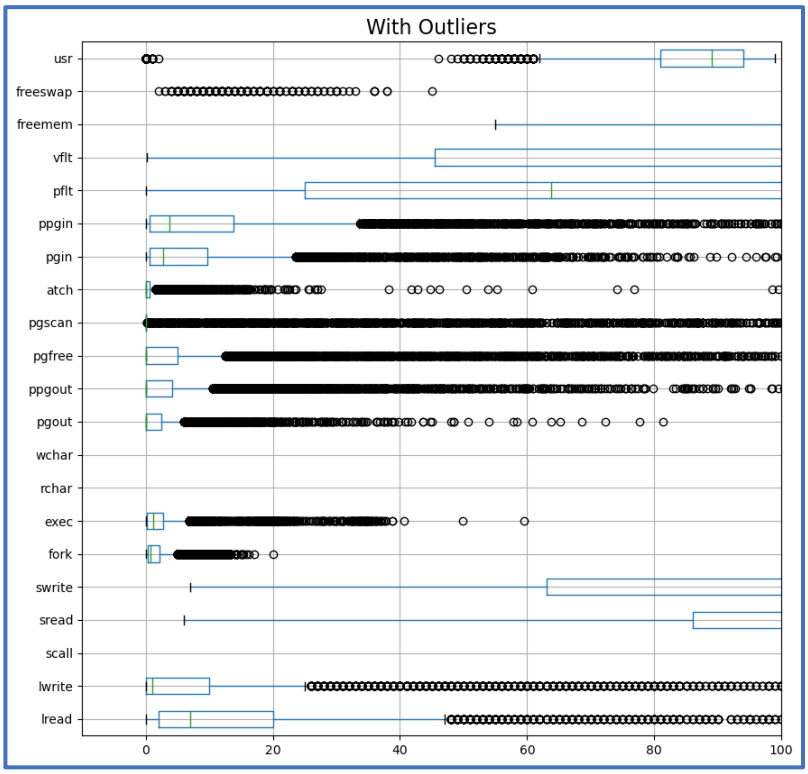

Figure 1.13: Outlier treatment before

After:

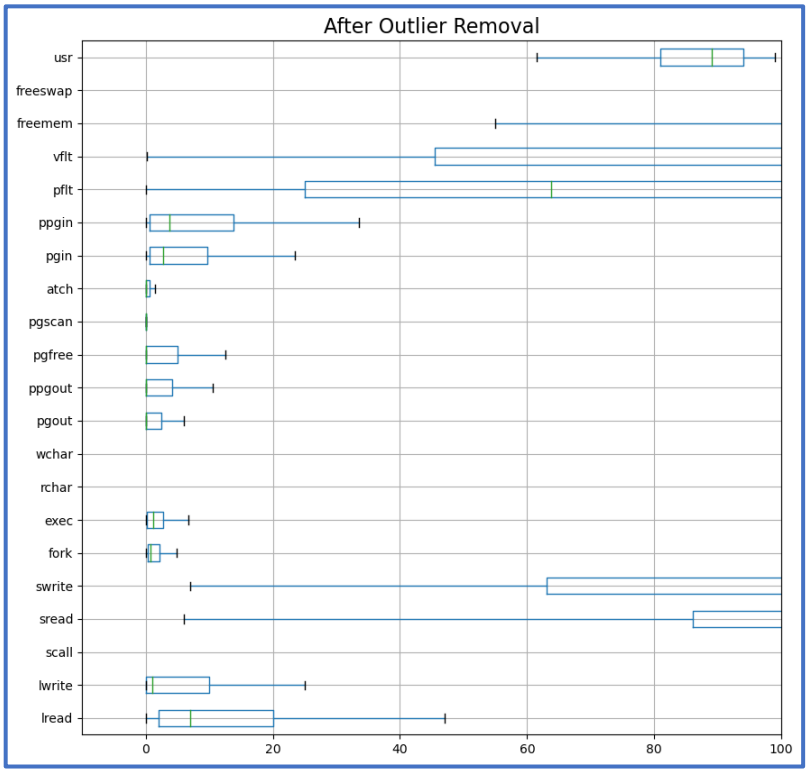

Figure 1.14: Outlier treatment after

The outliers are capped to their Upper limit and Lower limit respectively

1.2.3 Feature Engineering

1.2.4 Encode the Data

scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	usr	runqsz_Not_CPU_Bound
2147.0	79.0	68.0	0.2	0.2	40671.0	53995.0	0.0	...	0.0	0.0	1.6	2.6	16.00	26.40	4659.125	1730946.0	95.0	0
170.0	18.0	21.0	0.2	0.2	448.0	8385.0	0.0	...	0.0	0.0	0.0	0.0	15.63	16.83	4659.125	1869002.0	97.0	1
2162.0	159.0	119.0	2.0	2.4	125473.5	31950.0	0.0	...	0.0	1.2	6.0	9.4	150.20	220.20	702.000	1021237.0	87.0	1
160.0	12.0	16.0	0.2	0.2	125473.5	8670.0	0.0	...	0.0	0.0	0.2	0.2	15.60	16.80	4659.125	1863704.0	98.0	1
330.0	39.0	38.0	0.4	0.4	125473.5	12185.0	0.0	...	0.0	0.0	1.0	1.2	37.80	47.60	633.000	1760253.0	90.0	1

Table 1.4: Encoded dataset

Changing the object variables into binary using the get_dummies function. The runqsz is changed to 0 & 1 output.

1.2.5 Train Test Split

X & Y datasets are created separately by classifying the dependent and independent variables.
Then train data and test data set is created using the function train_test_split function with the size as 0.30(70:30)
Checking the shape of X_train and X_test

The shape of X_train dataset is (5734, 21)

The shape of X_test dataset is (2458, 21)

1.3 Model Building - Linear regression using SKlearn Method

1.3.1 Building the model

Building the model using the LinearRegression function and fit the model using train dataset of X & Y

1.3.2 Finding intercept

The intercept can be done using the function intercept_

The intercept for our model is 84.12174079532724

1.3.3 Finding coefficients

The coefficients can be done using the function coef_

```
array([[-6.34815062e-02,  4.81612871e-02, -6.63828011e-04,
         3.08252103e-04, -5.42182230e-03,  2.93127272e-02,
        -3.21166484e-01, -5.16684176e-06, -5.40287524e-06,
        -3.68819064e-01, -7.65976821e-02,  8.44841447e-02,
        -1.11022302e-16,  6.27574157e-01,  1.99879077e-02,
        -6.73338398e-02, -3.36028294e-02, -5.46366880e-03,
        -4.58467188e-04,  8.83184026e-06,  1.61529785e+00]])
```

1.3.4 Creating a data frame for the coefficients

Creating a Data frame for coefficients using the DataFrame function in Pandas

	Variables	Coefficient Estimate
0	lread	-0.063
1	lwrite	0.048
2	scall	-0.001
3	sread	0.000
4	swrite	-0.005
5	fork	0.029
6	exec	-0.321
7	rchar	-0.000
8	wchar	-0.000
9	pgout	-0.369
10	ppgout	-0.077
11	pgfree	0.084
12	pgscan	-0.000
13	atch	0.628
14	pgin	0.020
15	ppgin	-0.067
16	pflt	-0.034
17	vflt	-0.005
18	freemem	-0.000
19	freeswap	0.000
20	runqsz_Not_CPU_Bound	1.615

Figure 1.15: Coefficients

Runqsz_Not_CPU_Bound having the higher coefficients value of 1.615 whereas scall, sread, swrite, rchar, wchar, pgscan, vflt, freemen & freeswap having negligible coefficient values

1.3.5 Visualization of the coefficients

Visualizing the coefficients using a bar plot

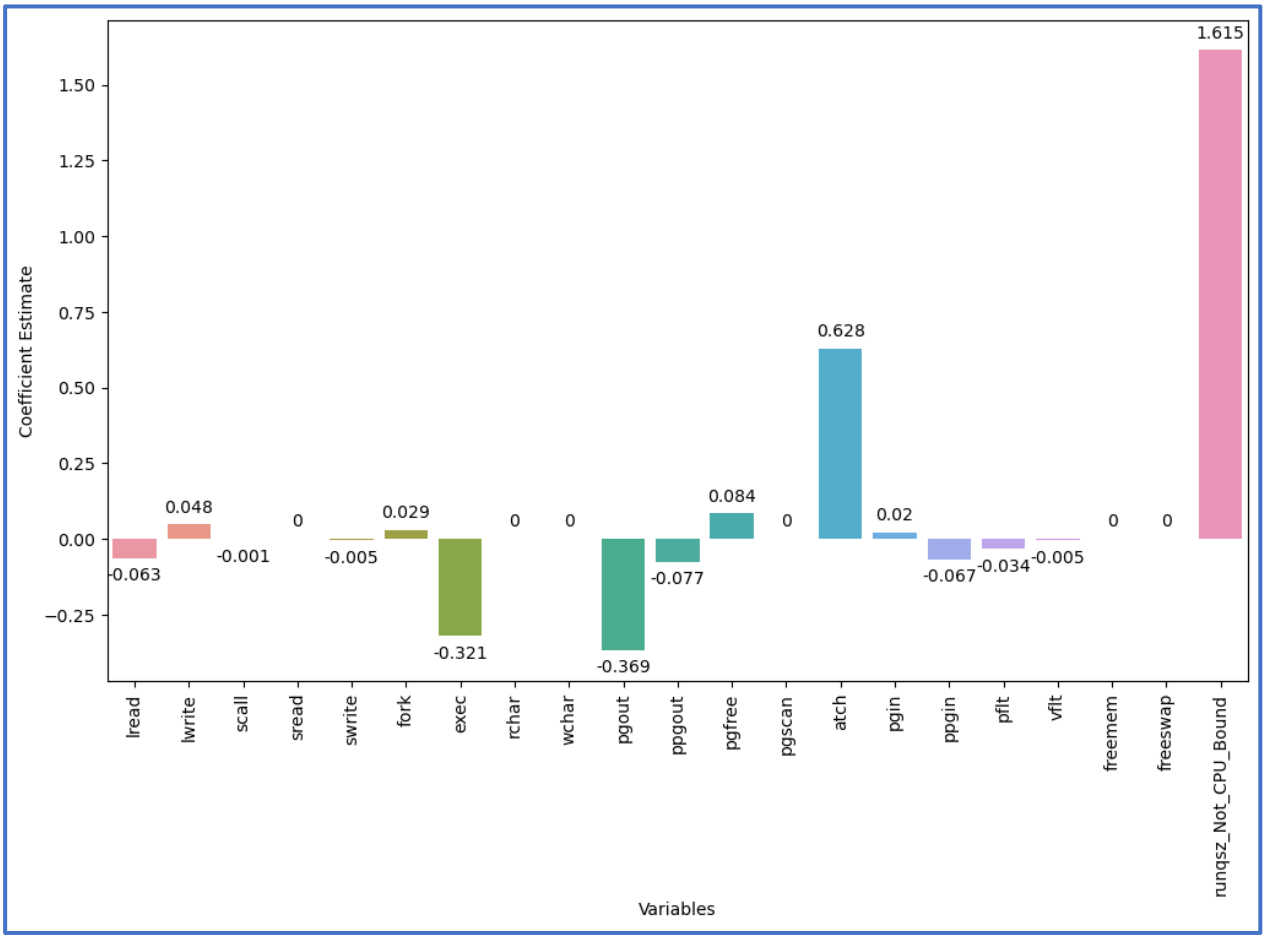

Figure 1.16: Coefficients Visualization

1.3.6 Calculating R-squared value on train & test data

The scores are calculated using score function

The R-Square score of the train dataset is 0.796108610127457

The R-Square score of the test dataset is 0.7677318597936152

The R-Square score for both the train and test dataset is almost the same which indicates there is no overfitting of the model

1.3.7 Calculating train and test predictions

The predictions are calculated using predict function

Train Predictions

```
array([[91.5078012 ],
       [91.77883105],
       [74.85526321],
       ...,
       [84.49418847],
       [84.15725271],
       [92.95606428]])
```

Test Predictions

```
array([[96.91549254],
       [90.34324284],
       [77.86534314],
       ...,
       [97.58162341],
       [90.9160509 ],
       [79.45653718]])
```

1.3.8 Calculating RMSE score for the train & test data

Using mean squared error function the RMSE values are calculated

The RMSE value of train data is 4.419536092979902

The RMSE value of test data is 4.65229570419262

1.4 Model Building - Linear regression using Statsmodel

1.4.1 Building the model using statsmodel

Building the model using the OLS function and fitting the model

1.4.2 Checking the OLS Summary

Printing the OLS summary

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.796
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1115.
Date:	Thu, 11 Jan 2024	Prob (F-statistic):	0.00
Time:	21:04:56	Log-Likelihood:	-16657.
No. Observations:	5734	AIC:	3.336e+04
Df Residuals:	5713	BIC:	3.350e+04
Df Model:	20		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.1217	0.316	266.106	0.000	83.502	84.741
lread	-0.0635	0.009	-7.071	0.000	-0.081	-0.046
lwrite	0.0482	0.013	3.671	0.000	0.022	0.074
scall	-0.0007	6.28e-05	-10.566	0.000	-0.001	-0.001
sread	0.0003	0.001	0.305	0.760	-0.002	0.002
swrite	-0.0054	0.001	-3.777	0.000	-0.008	-0.003
fork	0.0293	0.132	0.222	0.824	-0.229	0.288
exec	-0.3212	0.052	-6.220	0.000	-0.422	-0.220
rchar	-5.167e-06	4.88e-07	-10.598	0.000	-6.12e-06	-4.21e-06
wchar	-5.403e-06	1.03e-06	-5.232	0.000	-7.43e-06	-3.38e-06
pgout	-0.3688	0.090	-4.098	0.000	-0.545	-0.192
ppgout	-0.0766	0.079	-0.973	0.330	-0.231	0.078
pgfree	0.0845	0.048	1.769	0.077	-0.009	0.178
pgscan	-3.035e-14	1.45e-16	-209.209	0.000	-3.06e-14	-3.01e-14
atch	0.6276	0.143	4.394	0.000	0.348	0.908
pgin	0.0200	0.028	0.703	0.482	-0.036	0.076
ppgin	-0.0673	0.020	-3.415	0.001	-0.106	-0.029
pflt	-0.0336	0.002	-16.957	0.000	-0.037	-0.030
vflt	-0.0055	0.001	-3.830	0.000	-0.008	-0.003
freemem	-0.0005	5.07e-05	-9.038	0.000	-0.001	-0.000
freeswap	8.832e-06	1.9e-07	46.472	0.000	8.46e-06	9.2e-06
runqsz_Not_CPU_Bound	1.6153	0.126	12.819	0.000	1.368	1.862

Omnibus:	1103.645	Durbin-Watson:	2.016
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2372.553
Skew:	-1.119	Prob(JB):	0.00
Kurtosis:	5.219	Cond. No.	4.18e+22

Figure 1.17: OLS Summary

1.4.3 Checking the VIF of the Predictors

```
VIF values:

const                    29.229332
lread                     5.350560
lwrite                    4.328397
scall                     2.960609
sread                     6.420172
swrite                    5.597135
fork                     13.035359
exec                      3.241417
rchar                     2.133616
wchar                     1.584381
pgout                    11.360363
ppgout                   29.404223
pgfree                   16.496748
pgscan                         NaN
atch                      1.875901
pgin                     13.809339
ppgin                    13.951855
pflt                     12.001460
vflt                     15.971049
freemem                   1.961304
freeswap                  1.841239
runqsz_Not_CPU_Bound      1.156815
dtype: float64
```

Figure 1.18: VIF Values

Variable 'ppgout' having the highest VIF value. Dropping that variable, building the model again and comparing the Adjusted R-Squared Value from the initial Adjusted R-Squared Value.

The Adjusted R-squared value difference is 0.0

If the difference is very less or negligible value then the variable can be dropped and model is built again.
The VIF values is calculated again

```
VIF values:

const                    29.021961
lread                     5.350387
lwrite                    4.328325
scall                     2.960379
sread                     6.420135
swrite                    5.597025
fork                     13.027305
exec                      3.239231
rchar                     2.133614
wchar                     1.580894
pgout                     6.453978
pgfree                    6.172847
pgscan                         NaN
atch                      1.875553
pgin                     13.784007
ppgin                    13.898848
pflt                     12.001460
vflt                     15.966865
freemem                   1.959267
freeswap                  1.838167
runqsz_Not_CPU_Bound      1.156421
dtype: float64
```

Figure 1.19: VIF Values after dropping variable 'ppgout'

Process:

Dropping the higher VIF value variable
Checking the Adjusted R-squared value difference from the previous OLS summary
Dropping the variable if the difference is lesser or negligible
Building the model again by dropping that variable
Checking the VIF values again

Repeating the above process until all the VIF values of the variables comes less that 5 and stopping the process if there is a significant difference in the Adjusted R-squared value from the previous OLS summary.

Adjusted R-squared value difference of the independent variables with higher VIF values

The Adjusted R-squared value difference of variable ppgout is 0.0
The Adjusted R-squared value difference of variable vflt is 0.0
The Adjusted R-squared value difference of variable ppgin is 0.0
The Adjusted R-squared value difference of variable fork is 0.001
The Adjusted R-squared value difference of variable pgout is 0.001
The Adjusted R-squared value difference of variable sread is 0.0
The Adjusted R-squared value difference of variable lread is 0.002
The Adjusted R-squared value difference of variable pgscan is 0.0
The Adjusted R-squared value difference of variable pflt is 0.058

We stop the process of dropping the high VIF value variables once there is a significant drop in the Adjusted R-squared Value.

```
VIF values:

const                  28.315818
lwrite                  1.052259
scall                   2.648774
swrite                  3.012409
exec                    2.819098
rchar                   1.671676
wchar                   1.529035
pgfree                  1.917372
atch                    1.732230
pgin                    1.483892
pflt                    3.253088
freemem                 1.950475
freeswap                1.762866
runqsz_Not_CPU_Bound    1.140440
dtype: float64
```

Figure 1.20: VIF Values after dropping all the variable with lesser Adj. R-Squared Difference

1.4.4 Assumptions

	Actual Values	Fitted Values	Residuals
0	91.0	89.300965	1.699035
1	94.0	91.690164	2.309836
2	61.5	75.802891	-14.302891
3	83.0	80.511458	2.488542
4	94.0	97.522942	-3.522942

Table 1.5: Assumptions

1.4.4.1 Linearity & Independence

The plot between fitted and residual value identified by residplot function

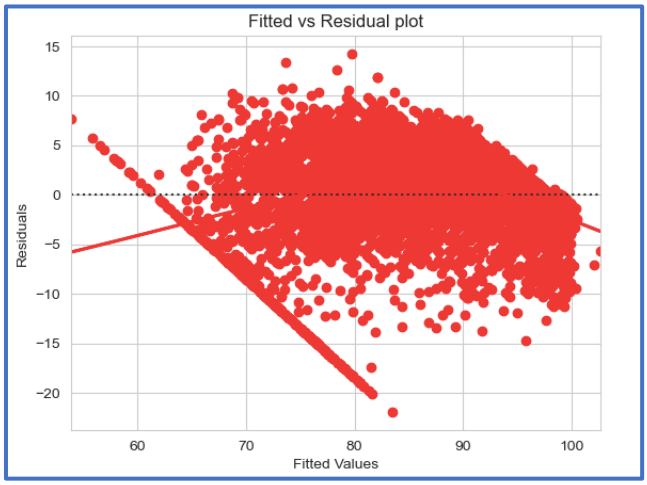

Figure 1.21: Fitted vs Residual plot

No patterns identified.

1.4.4.2 Normality

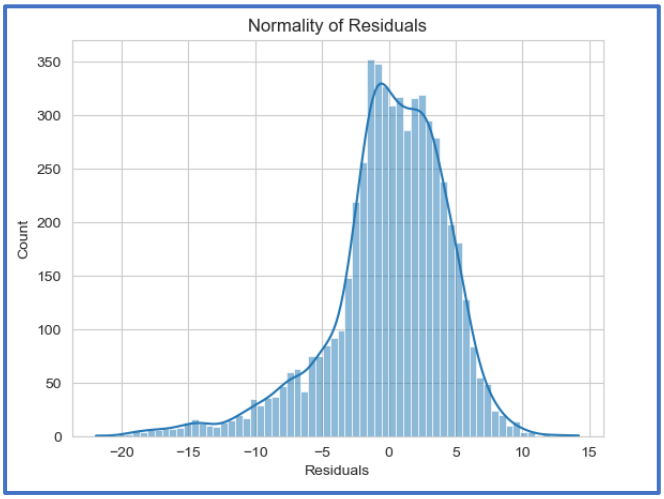

Figure 1.22: Normality of Residual

ShapiroResult(statistic=0.9430180191993713, pvalue=2.023474982485036e-42)

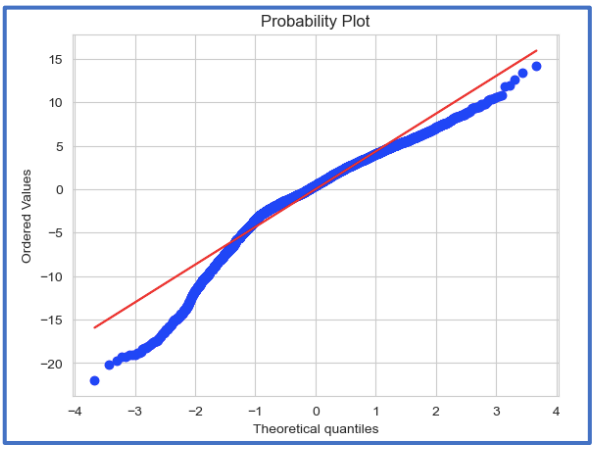

Figure 1.23: Probability plot

1.4.4.3 Homoscedasticity

[('F statistic', 1.1139220397084624), ('p-value', 0.0019860830795081015)]

1.4.5 Predictions

sread	swrite	fork	exec	rchar	wchar	pgout	...	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	usr	runqsz_Not_CPU_Bound	New
79.0	68.0	0.2	0.2	40671.0	53995.000	0.00	...	0.0	1.6000	2.60	16.00	26.40	4659.125	1730946.0	95.0	0	94.201706
18.0	21.0	0.2	0.2	448.0	8385.000	0.00	...	0.0	0.0000	0.00	15.63	16.83	4659.125	1869002.0	97.0	1	99.343028
159.0	119.0	2.0	2.4	125473.5	31950.000	0.00	...	1.2	6.0000	9.40	150.20	220.20	702.000	1021237.0	87.0	1	84.040288
12.0	16.0	0.2	0.2	125473.5	8670.000	0.00	...	0.0	0.2000	0.20	15.60	16.80	4659.125	1863704.0	98.0	1	98.627627
39.0	38.0	0.4	0.4	125473.5	12185.000	0.00	...	0.0	1.0000	1.20	37.80	47.60	633.000	1760253.0	90.0	1	98.125236
65.0	61.0	0.4	0.4	125473.5	58703.000	0.00	...	0.0	0.0000	0.00	28.40	34.40	4659.125	1877461.0	96.0	1	96.935026
168.0	190.0	0.2	0.2	125473.5	189975.000	6.00	...	1.5	0.6000	0.60	27.40	28.60	312.000	1013458.0	89.0	1	86.228569
291.0	211.0	0.6	0.4	125473.5	230625.875	2.60	...	0.0	1.0000	1.00	35.40	71.00	87.000	10989.5	61.5	0	76.145491
42.0	33.0	0.2	0.2	125473.5	10116.000	0.00	...	0.0	0.4000	0.80	15.63	18.44	1374.000	1749756.0	98.0	1	98.875789
13.0	24.0	0.2	0.2	125473.5	6777.000	0.00	...	0.0	0.0000	0.00	15.60	16.80	4659.125	1859912.0	98.0	1	98.556045
191.0	152.0	0.8	0.8	125473.5	170579.000	0.00	...	0.0	1.2000	1.60	65.00	65.60	1143.000	1535661.0	90.0	0	90.346763
144.0	103.0	2.4	0.8	125473.5	10148.000	0.20	...	0.2	1.0000	1.00	121.80	166.80	298.000	1709362.0	92.0	0	90.804766
245.0	135.0	4.2	6.7	125473.5	42433.000	0.00	...	0.2	20.8000	23.60	295.00	408.80	2630.000	1524755.0	75.0	0	74.846668
45.0	196.0	0.2	0.2	125473.5	223361.000	1.80	...	0.0	0.4000	0.40	15.60	19.60	165.000	1749568.0	96.0	1	96.989420
74.0	44.0	1.2	1.8	125473.5	8905.000	0.80	...	0.0	4.2000	7.40	108.60	142.40	133.000	1703250.0	94.0	0	90.377110
177.0	151.0	4.9	5.2	125473.5	6300.000	2.40	...	1.5	11.4000	11.80	202.00	418.40	233.000	1447301.0	85.0	0	80.836912
47.0	65.0	0.4	0.4	125473.5	13620.000	0.00	...	0.0	5.9900	5.99	28.34	39.32	701.000	1037450.0	93.0	1	90.213096
125.0	79.0	0.6	2.4	125473.5	25056.000	3.79	...	0.8	1.4000	1.80	51.30	70.66	201.000	1001517.0	94.0	1	88.607737
259.0	163.0	4.9	6.7	125473.5	37938.000	6.00	...	1.5	23.5125	33.60	361.50	561.40	159.000	1088988.0	61.5	1	69.610025
14.0	33.0	0.2	0.2	125473.5	14377.000	0.00	...	0.0	0.0000	0.00	15.40	16.80	4659.125	1858256.0	98.0	1	97.663869

Figure 1.24: Prediction variables

1.4.6 Linear Equation

usr = 83.97250861363338 + -0.034501248661689306 * (lwrite) + -0.0006924438391003225 * (scall) + -0.005816907313513595 * (swrite) + -0.38516662696822623 * (exec) + -5.473325684927575e-06 * (rchar) + -4.8610924794833105e-06 * (wchar) + -0.11841334885500272 * (pgfree) + 0.36247530227938113 * (atch) + -0.09637945268316846 * (pgin) + -0.04163081380782967 * (pflt) + -0.0004481966053920618 * (freemem) + 9.002038349886397e-06 * (freeswap) + 1.6447018685480495 * (runqsz_Not_CPU_Bound)

1.4.7 RMSE on train & test data

The RSME value of the train data is 4.467611664735454
The RSME value of the test data is 4.7053764916318706

1.4.8 MAE on train & test data

The MAE value of the train data is 3.320911925730544
The MAE value of the test data is 3.41900906922404

1.5 Business Insights & Recommendations

- The final linear equation consist of 12 out of 21 independent variables
- The 12 variables has an considerable impact on the dependent variable for predication
- The variable atch has the higher positive coefficient value with respect to the dependent variable usr.
- one-unit increase in the atch variable is associated with an increase of 0.36250 units in the dependent variable usr.
- All the other variables having a negative impact with the dependent variable usr.
- The R-Squared value is 0.796 and Adjusted R-Squared value is 0.795 which indicates that the model has good accuracy
- Assumptions: Linearity & Independence the residual plot does not show any pattern
- Assumptions: The data looks normal in the plot but fails in the shaprio test that the p value is greater than 0.05
- Assumptions: Homoscedasticity p value is less that 0.05 which indicates that the null hypothesis is rejected
- The RSME values for both the train and test are almost similar. The value is greater than 4 which indicates that the model has higher error and less precise predictions
- The positive coefficient for atch suggests that increasing the number of attached processes positively influences CPU utilization.
- Optimizing the attachment of processes to enhance system performance can be recommended
- The negative coefficient for exec indicates that longer execution times are associated with lower CPU utilization.
- Consider optimizing processes with longer execution times to improve overall CPU efficiency.
- Variables such as freemem, freeswap, and runqsz_Not_CPU_Bound have impacts on CPU utilization.
- System resource allocation can be evaluated and consider tuning parameters related to free memory, swap space, and non-CPU bound run queue sizes for optimal performance.

Problem 2:

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

2.1 Define the problem and perform exploratory Data Analysis

2.1.1 Problem definition

In your role as a statistician at the Republic of Indonesia Ministry of Health, you have been entrusted with a dataset containing information from a Contraceptive Prevalence Survey. This dataset encompasses data from 1473 married females who were either not pregnant or were uncertain of their pregnancy status during the survey.

Your task involves predicting whether these women opt for a contraceptive method of choice. This prediction will be based on a comprehensive analysis of their demographic and socio-economic attributes.

Loading the dataset:

Loading the dataset using the read function and checking whether the data is loaded properly or not using head and tail function

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

Table 2.1: Loading the dataset

The dataset is loaded properly

2.1.2 Check shape

There no. of rows = 1473
There no. of columns = 10

2.1.3 Data types

The datatypes can be identified using the info function

```
<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
```

Figure 2.1: Datatypes

There are 2 variables Wife_age and No_of_children_born have null values

2.1.4 Statistical summary

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

Table 2.2: Statistical Summary of numerical variables

All the numerical variables have a normal distribution as the mean and median are almost the same numbers.

	count	unique	top	freq
Wife_ education	1473	4	Tertiary	577
Husband_education	1473	4	Tertiary	899
Wife_religion	1473	2	Scientology	1253
Wife_Working	1473	2	No	1104
Standard_of_living_index	1473	4	Very High	684
Media_exposure	1473	2	Exposed	1364
Contraceptive_method_used	1473	2	Yes	844

Table 2.3: Statistical Summary of categorical variables

2.1.5 Univariate analysis

2.1.5.1 Numerical

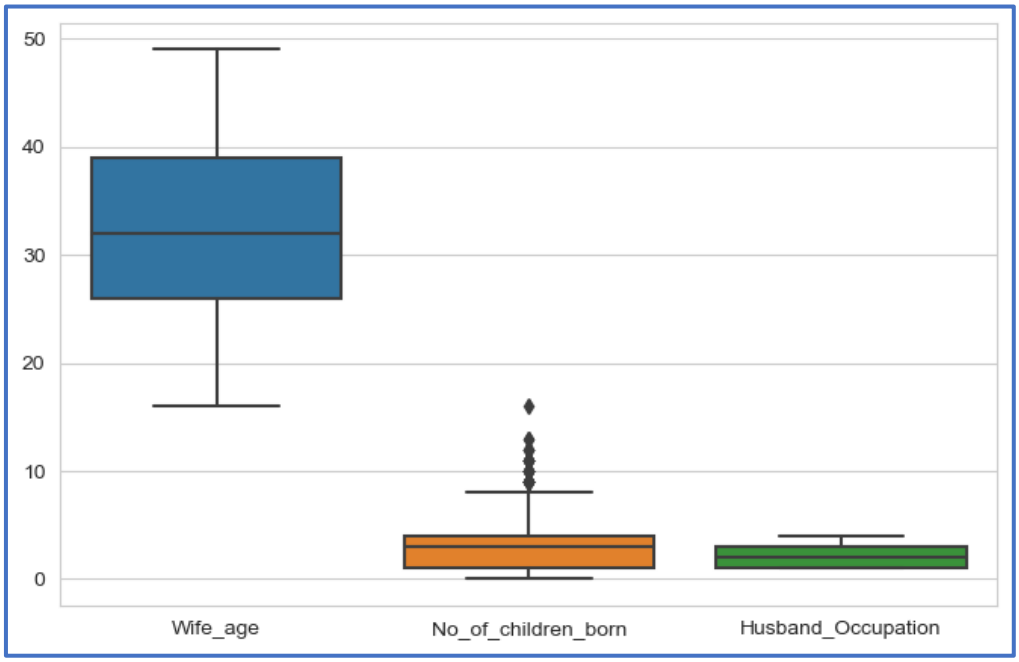

Figure 2.2: Univariate Analysis - Numerical

The variable wife age has no outliers and has the higher distribution.
The No. of children variable has outliers.
The variable Husband Occupation has the lowest distribution and mean without any outliers

2.1.5.2 Categorical

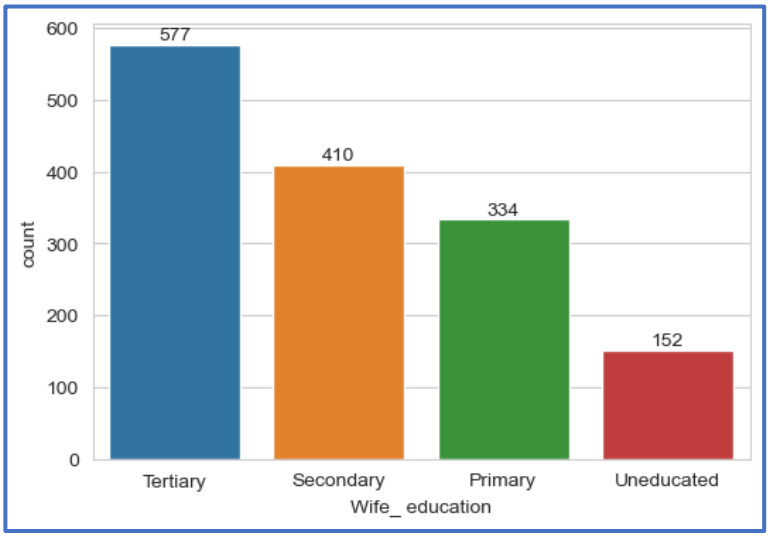

Figure 2.3: Univariate Analysis – Wife education

Wife Education: around 40% have tertiary education with 10% uneducated

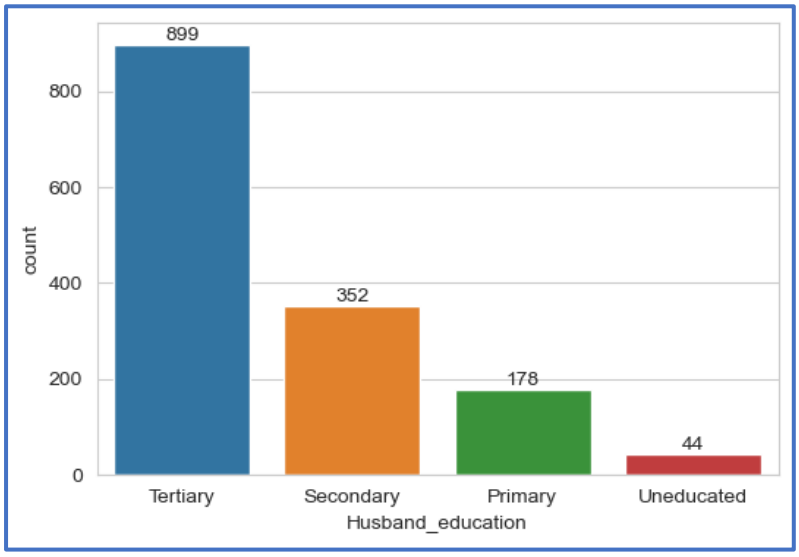

Figure 2.4: Univariate Analysis – Husband education

Husband Education: around 61% have tertiary education with 3% uneducated

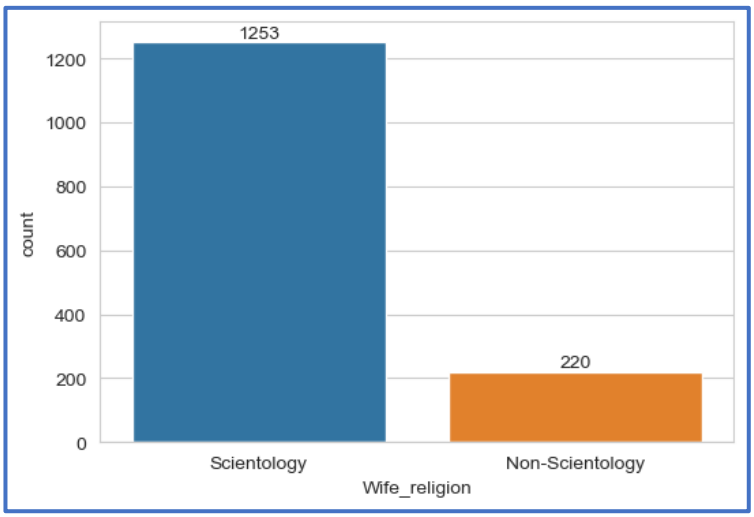

Figure 2.5: Univariate Analysis – Wife religion

1253 wife's are belong to the scientology religion compared to Non-scientology

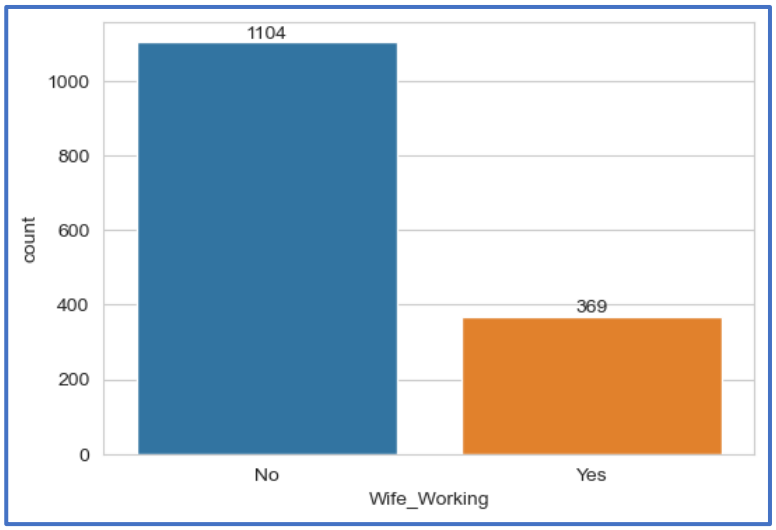

Figure 2.6: Univariate Analysis – Wife Working

Overall 78% of the wife are not working

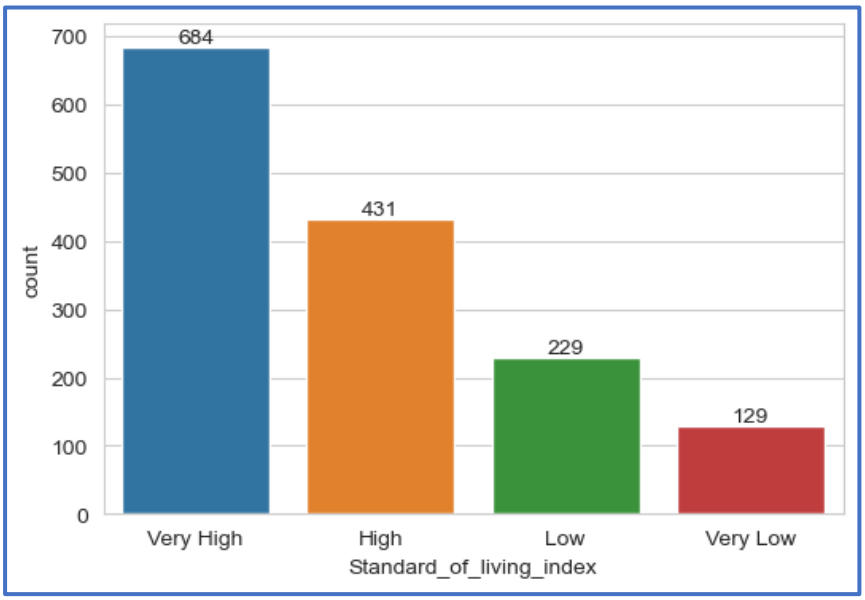

Figure 2.7: Univariate Analysis – Standard of living index

Around 75% of the total families are having a very high and high standard of living index

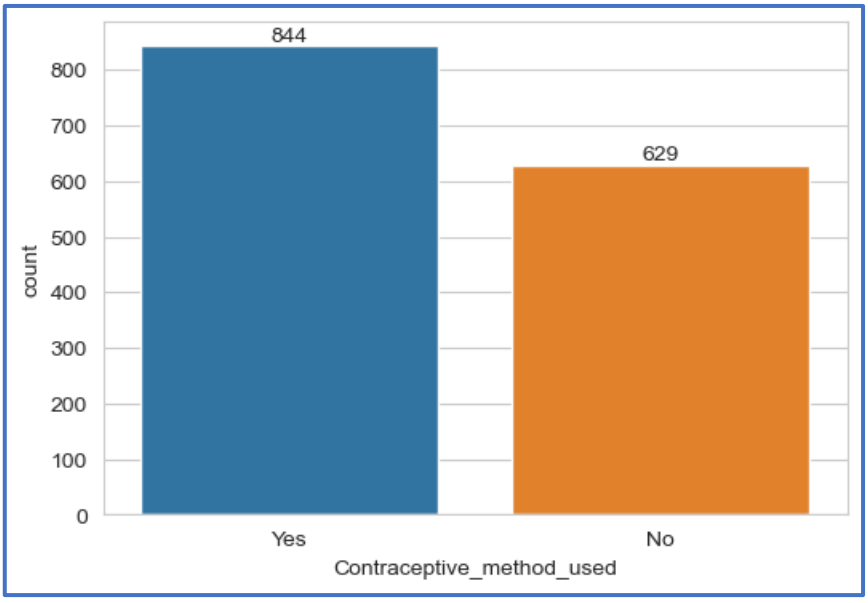

Figure 2.8: Univariate Analysis – Contraceptive method used

57% people use contraceptive method

2.1.6 Multivariate analysis

2.1.6.1 Numerical vs Numerical

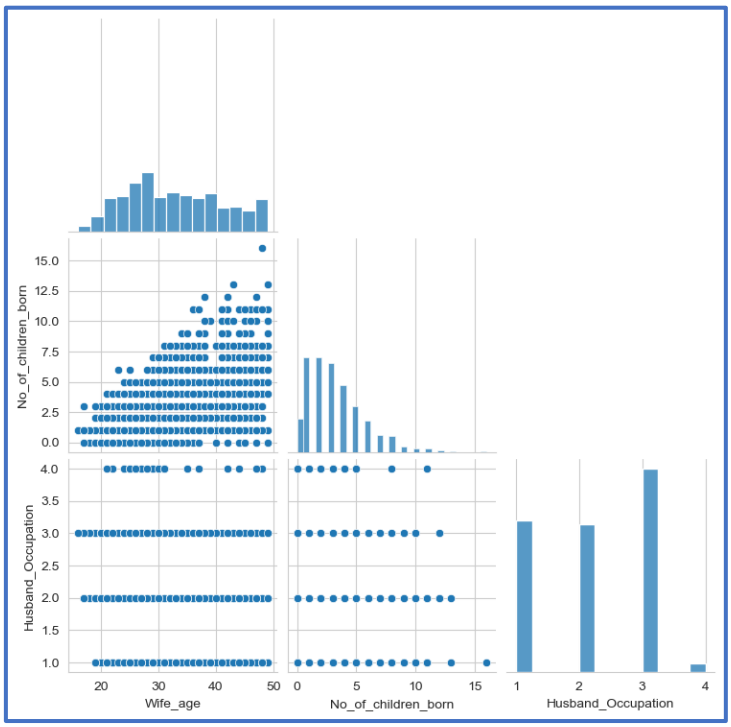

Figure 2.9: Multivariate Analysis – Numerical vs Numerical

No significant patterns or correlation recognized between these variables

2.1.6.2 Categorical vs Categorical

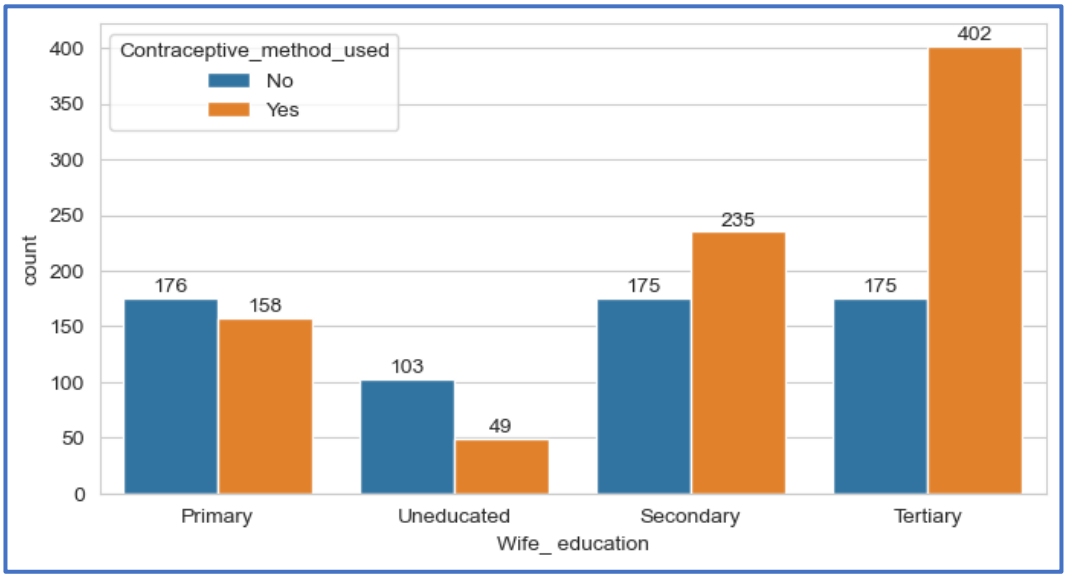

Figure 2.10: Multivariate Analysis – Contraceptive Method vs Wife Education

People who used contraceptive method having the higher number in tertiary education

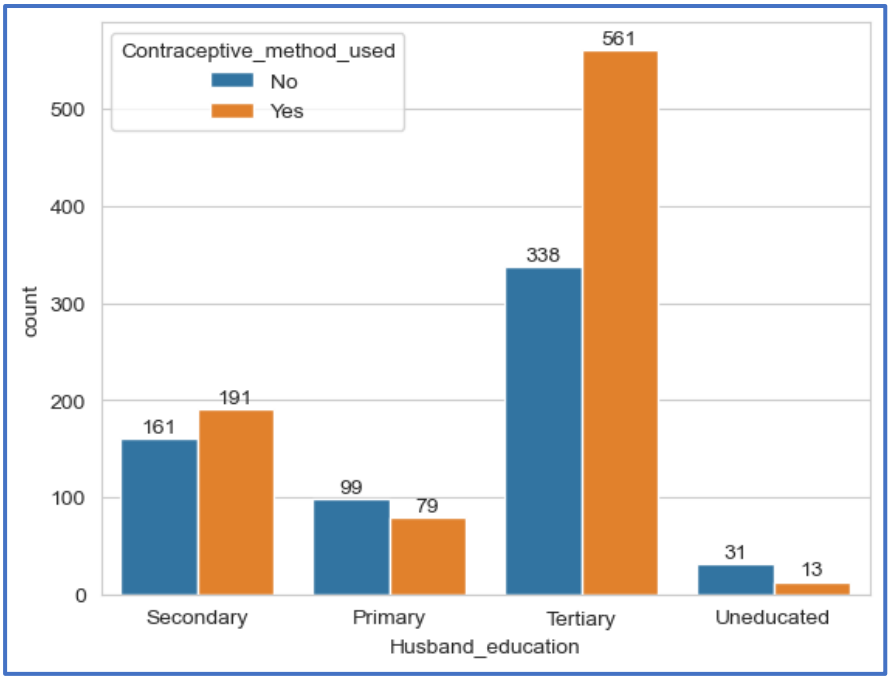

Figure 2.11: Multivariate Analysis – Contraceptive Method vs Husband Education

Similar to wife education, husband having tertiary education have used contraceptive method more in number

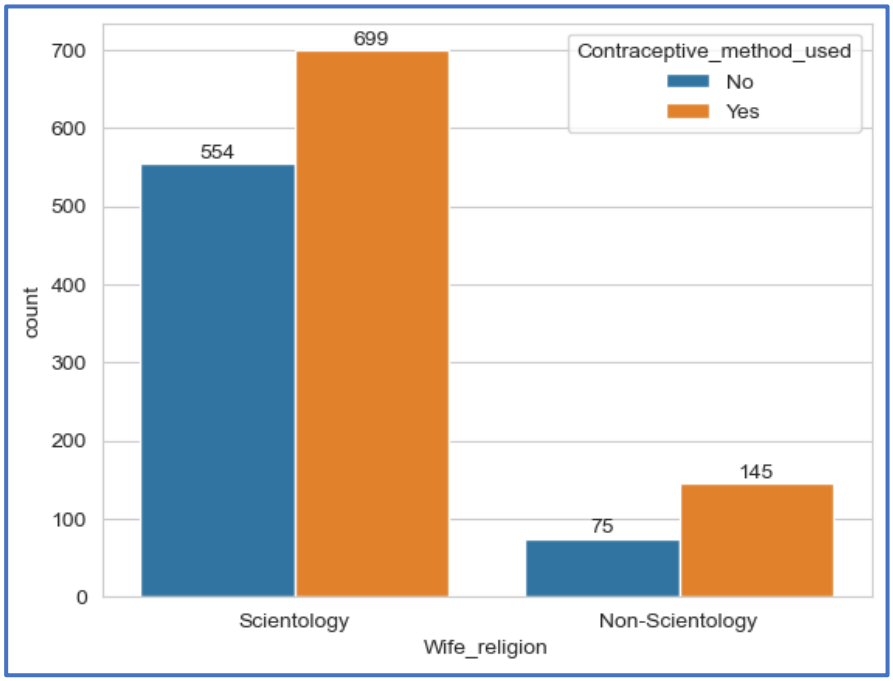

Figure 2.12: Multivariate Analysis – Contraceptive Method vs Wife Religion

Wife with scientology have the highest contribution for both contraceptive method used and not used

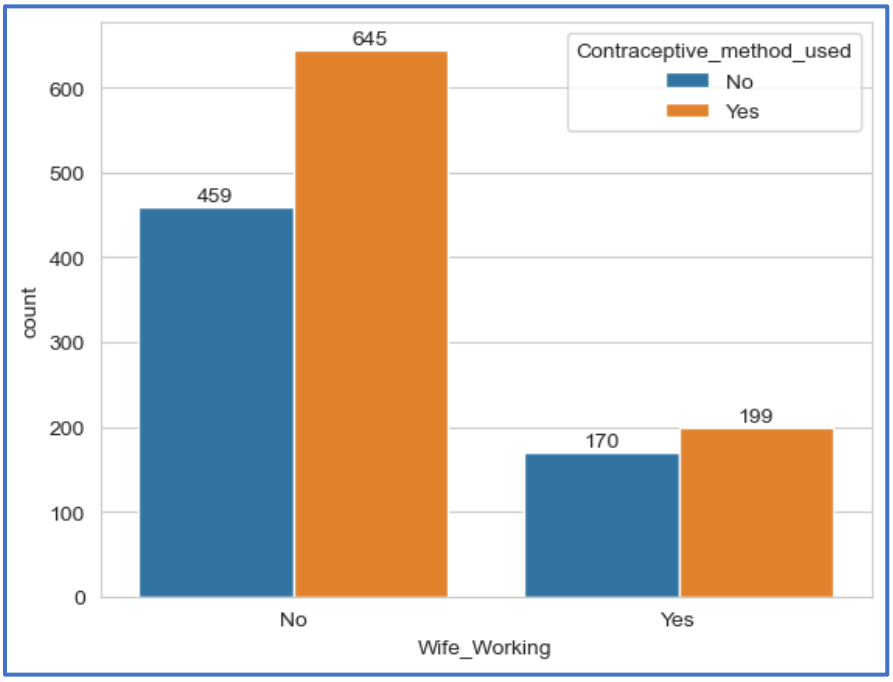

Figure 2.13: Multivariate Analysis – Contraceptive Method vs Wife Working

Wife who are not working used the contraceptive methos the most

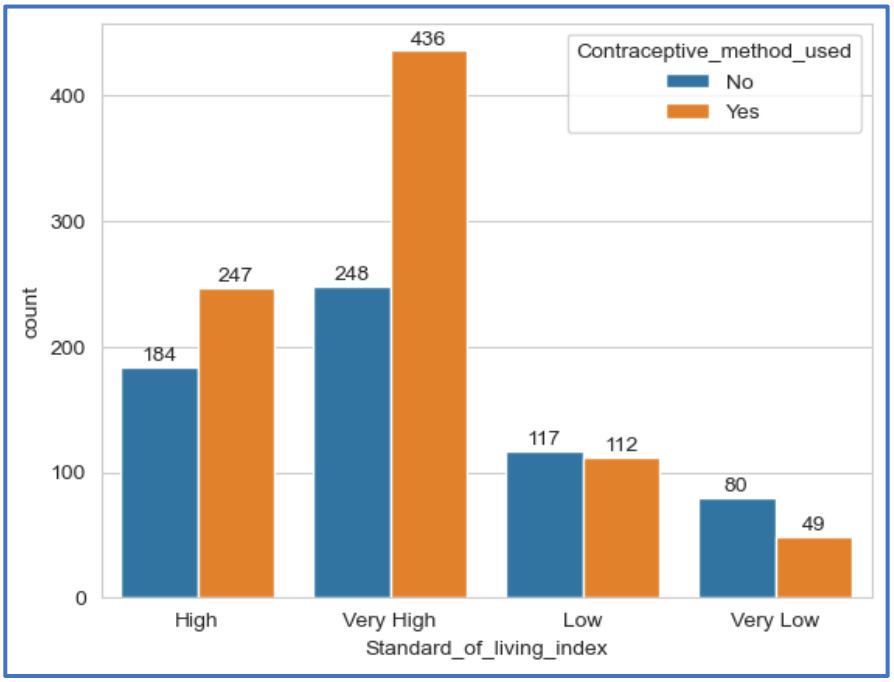

Figure 2.14: Multivariate Analysis – Contraceptive Method vs Standard of Living Index

Family with very high standard of living index used the contraceptive method the most

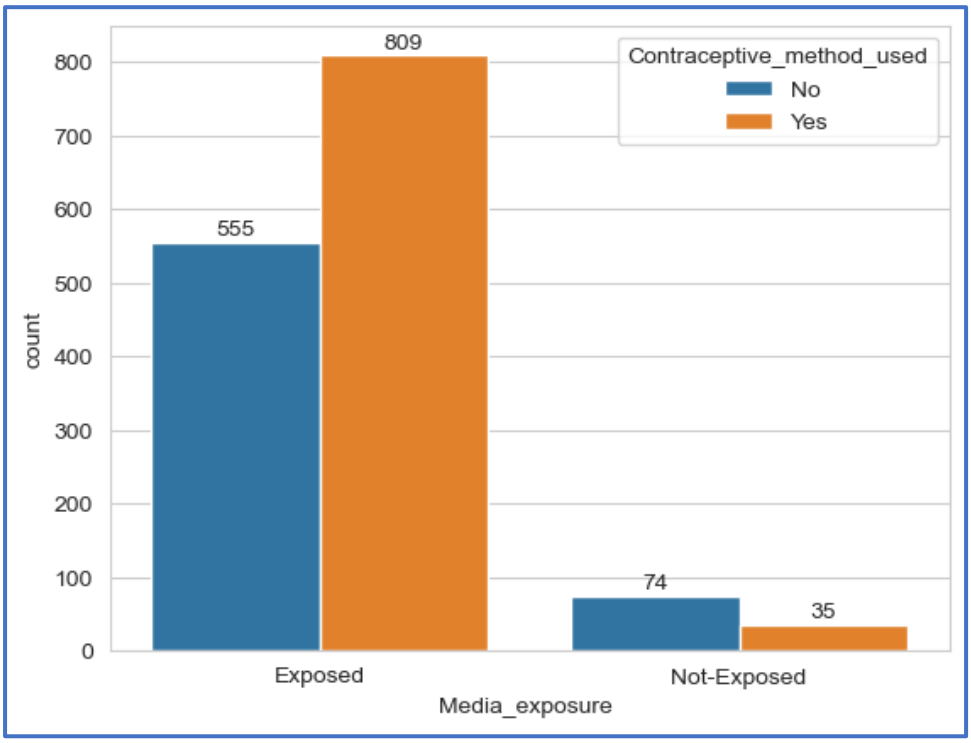

Figure 2.15: Multivariate Analysis – Contraceptive Method vs Media Exposure

55% of the wife used the contraceptive method who exposed to media

2.1.6.2 Numerical vs Categorical

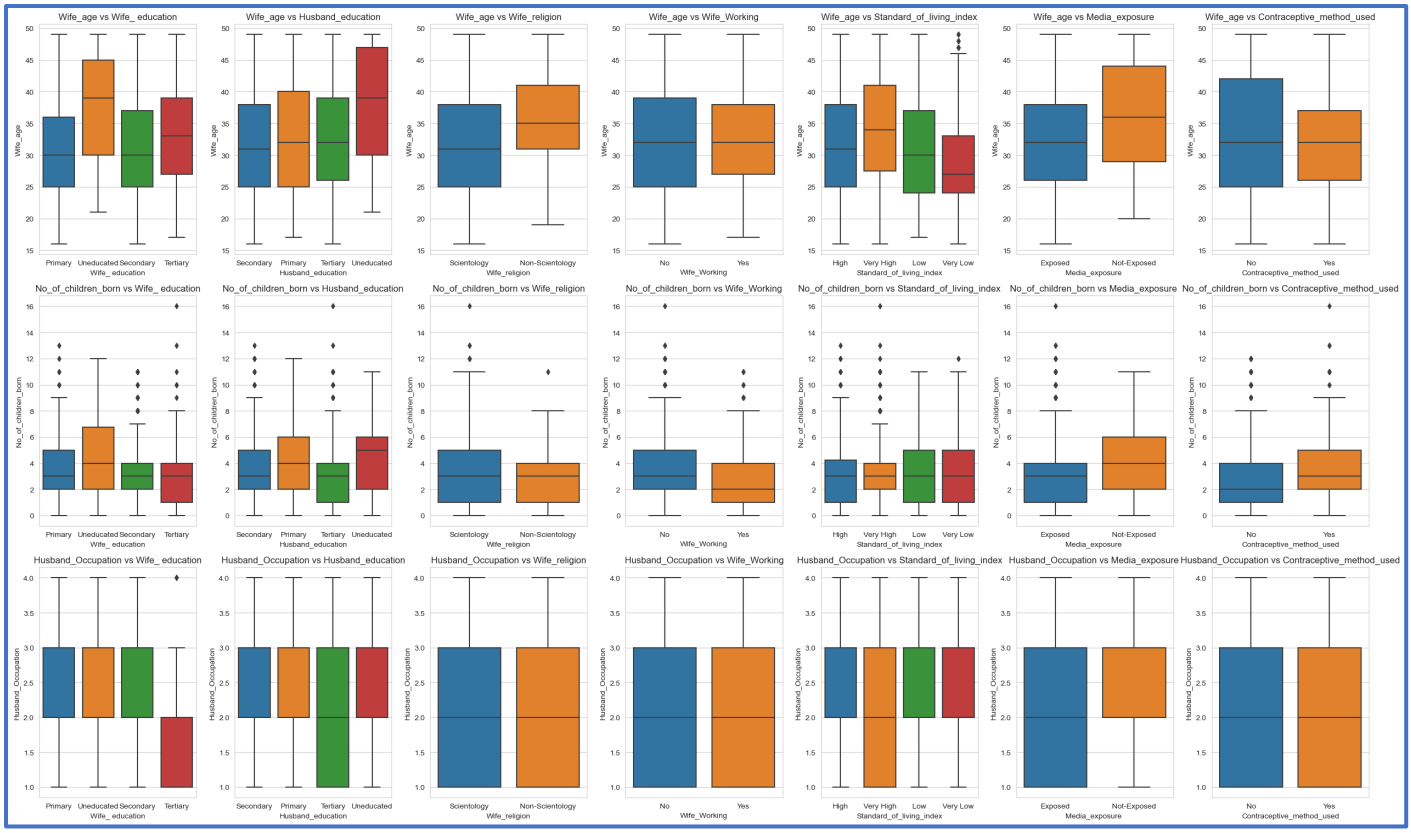

Figure 2.16: Multivariate Analysis – Categorical vs Numerical

2.1.7 Patterns and insights - Key meaningful observations

- The variable wife age has no outliers and has the higher distribution.
- The No. of children variable has outliers.
- The variable Husband Occupation has the lowest distribution and mean without any outliers
- Wife Education: around 40% have tertiary education with 10% uneducated
- Husband Education: around 61% have tertiary education with 3% uneducated
- 1253 wife's are belong to the scientology religion compared to Non-scientology
- Overall 78% of the wife are not working
- Around 75% of the total families are having a very high and high standard of living index
- 57% people use contraceptive method
- No significant patterns or correlation recognized between these variables
- People who used contraceptive method having the higher number in tertiary education
- Similar to wife education, husband having tertiary education have used contraceptive method more in number
- Wife with scientology have the highest contribution for both contraceptive method used and not used
- Wife who are not working used the contraceptive methos the most
- Family with very high standard of living index used the contraceptive method the most
- 55% of the wife used the contraceptive method who exposed to media

2.2 Data Pre-processing

2.2.1 Missing value Treatment (if needed)

The null values are identified using the isnull function. There are null values found in two variables

```
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
```

Figure 2.17: Null Values Identification

The null values are filled using the variables mean by fillna function

```
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
```

Figure 2.18: After missing value treatment

2.2.2 Outlier Detection(treat, if needed)

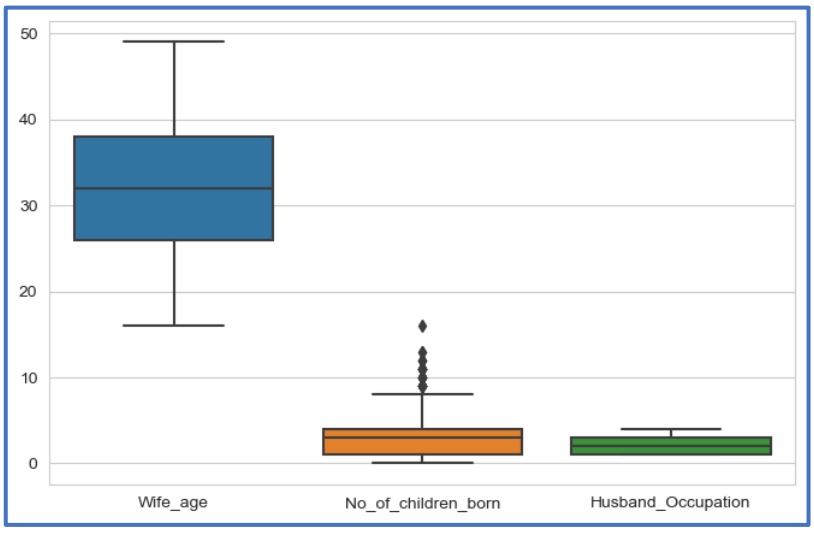

Figure 2.19: Outliers Detection

2.2.3 Feature Engineering (if needed)

2.2.4 Encode the data

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure
0	24.0	1	2	3.0	1	0	2	2	1
1	45.0	0	2	10.0	1	0	3	3	1
2	43.0	1	2	7.0	1	0	3	3	1
3	42.0	2	1	9.0	1	0	3	2	1
4	36.0	2	2	8.0	1	0	3	1	1

Table 2.4: Encoded Dataset

The data is encoded such as

Wife Education:

Uneducated - 0
Primary - 1
Secondary - 2
Tertiary - 3

Husband Education:

Uneducated - 0
Primary - 1
Secondary - 2
Tertiary - 3

Wife Religion:

Non-Scientology - 0
Scientology - 1

2.2.5 Train-test split

Contraceptive_method_used	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_livi
0	629	629	629	629	629	629	629	
1	844	844	844	844	844	844	844	

Table 2.5: Train Test Split

The dependent variable classification/split is 42% & 58% which indicates that the data is not highly skewed towards a single output. Accuracy can be used to measure

2.3 Model Building and Compare the Performance of the Models

2.3.1 Build a Logistic Regression model

Building the model using the LinearRegression function and fitting the same using train data

```
▾ LogisticRegression
LogisticRegression()
```

Predicating the model using the predict function

```
array(['1', '0', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '0',
       '1', '1', '0', '1', '1', '1', '1', '1', '0', '0', '1', '1', '0',
       '0', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '1', '1',
       '1', '0', '1', '1', '1', '0', '1', '1', '1', '0', '1', '1', '1',
       '1', '1', '1', '0', '1', '1', '1', '0', '1', '0', '0', '1', '1',
       '0', '0', '0', '0', '1', '0', '0', '1', '1', '1', '1', '1', '1',
       '1', '1', '1', '0', '1', '1', '0', '1', '0', '1', '1', '1', '1',
       '0', '0', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '0',
       '1', '1', '1', '1', '0', '1', '0', '1', '1', '0', '1', '1', '1',
       '1', '1', '1', '1', '0', '0', '0', '1', '0', '1', '1', '1', '1',
       '1', '1', '0', '1', '1', '1', '0', '0', '1', '0', '1', '0', '0',
       '1', '0', '1', '1', '1', '1', '1', '0', '0', '1', '1', '1', '1',
       '1', '0', '0', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1',
       '1', '0', '1', '0', '1', '1', '0', '1', '1', '0', '1', '0', '0',
       '0', '0', '1', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1',
       '0', '1', '1', '0', '0', '1', '0', '1', '1', '1', '1', '0', '1',
       '1', '1', '1', '1', '1', '1', '0', '1', '1', '0', '1', '0', '0',
       '1', '1', '1', '0', '0', '1', '1', '0', '1', '1', '1', '1', '1',
       '0', '1', '1', '1', '1', '1', '1', '1', '1', '1', '0', '0', '1',
       '1', '1', '1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '0',
       '1', '0', '1', '0', '0', '1', '1', '1', '0', '0', '1', '1', '0',
       '1', '1', '0', '0', '1', '1', '1', '1', '1', '1', '1', '1', '1',
       '0', '1', '0', '1', '1', '1', '1', '1', '0', '1', '0', '0', '1',
       '1', '0', '1', '1', '1', '0', '1', '1', '1', '1', '1', '1', '1',
       '1', '1', '1', '1', '1', '0', '0', '0', '1', '1', '1', '0', '1',
       '1', '1', '0', '1', '0', '0', '1', '1', '1', '0', '1', '0', '1',
       '1', '1', '1', '1', '1', '1', '0', '0', '0', '1', '1', '1', '1',
       '1', '1', '1', '0', '1', '0', '1', '1', '1', '1', '1', '1', '1',
       '0', '0', '0', '1', '0', '0', '1', '1', '1', '1', '0', '1', '1',
       '1', '0', '0', '1', '1', '0', '1', '0', '1', '0', '0', '1', '0',
       '1', '0', '0', '1', '1', '1', '0', '1', '1', '0', '1', '0', '1',
       '1', '0', '0', '0', '0', '1', '0', '1', '1', '1', '0', '1', '0',
       '1', '1', '1', '1', '0', '0', '1', '1', '1', '0', '1', '1', '1',
       '1', '1', '1', '1', '1', '0', '1', '1', '1', '1', '0', '0', '1'],
      dtype=object)
```

Figure 2.20: Predicted values

2.3.1.1 Model Score

The model score of the train data is 0.67

The model score of the test data is 0.69

2.3.1.2 Confusion Matrix

[[95 98]
 [41 208]]

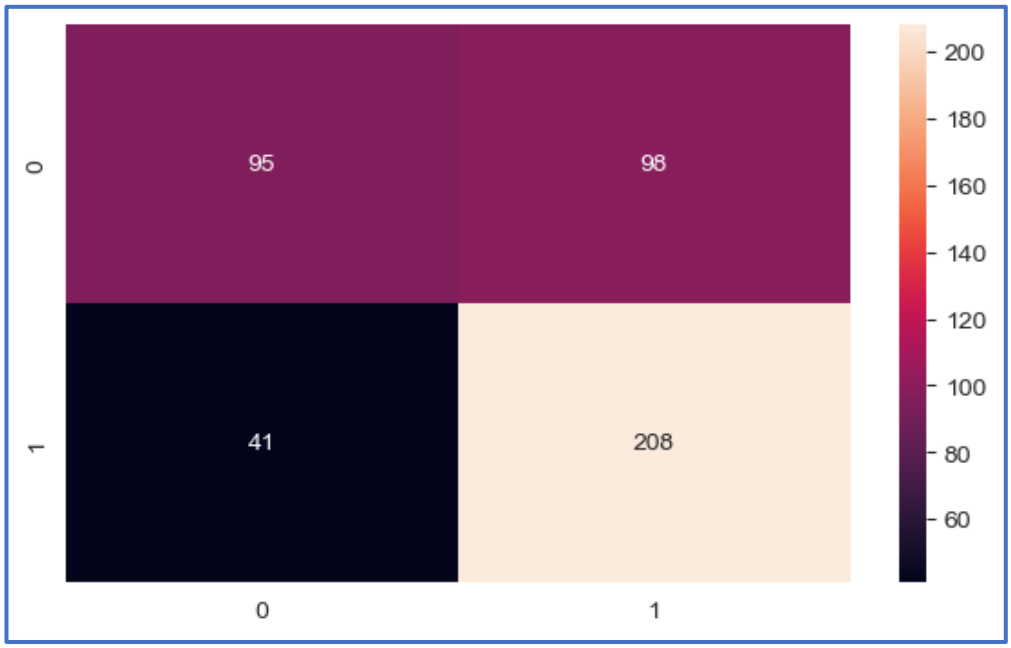

Figure 2.21: Confusion Matrix of LR

True Negative – 95: Actual 0's are identified as 0's
False Positive(Type I Error) – 98: Actual 0's are identified as 1's
True Positive – 208: Actual 1's are identified as 0's
False Negative(Type II Error) – 41: Actual 1's are identified as 1's

2.3.1.3 Classification Report

	precision	recall	f1-score	support
0	0.70	0.49	0.58	193
1	0.68	0.84	0.75	249
accuracy			0.69	442
macro avg	0.69	0.66	0.66	442
weighted avg	0.69	0.69	0.67	442

Figure 2.22: Classification Table of LR

The model predicts with 70% accuracy for 0's as 0's
The model predicts with 68% accuracy for 1's as 1's

The model predicts 84% correctly that the women opt for the Contraceptive method of choice

F1 score – Hormonic mean of Precision & Recall. 75% of class 1 indicates that the balanced measure for class 1.

2.3.2 Build a Linear Discriminant Analysis model

2.3.2.1 Model Building

Model building using the LinearDiscriminantAnalysis function and fitting the model with the dataset

2.3.2.2 Prediction

Prediction is done using the predict function on the independent variables

2.3.2.3 Checking the correlation

	Wife_age	No_of_children_born	Husband_Occupation
Wife_age	1.000000	0.527019	-0.199495
No_of_children_born	0.527019	1.000000	-0.020857
Husband_Occupation	-0.199495	-0.020857	1.000000

Table 2.6: Correlation

2.3.2.4 Confusion matrix

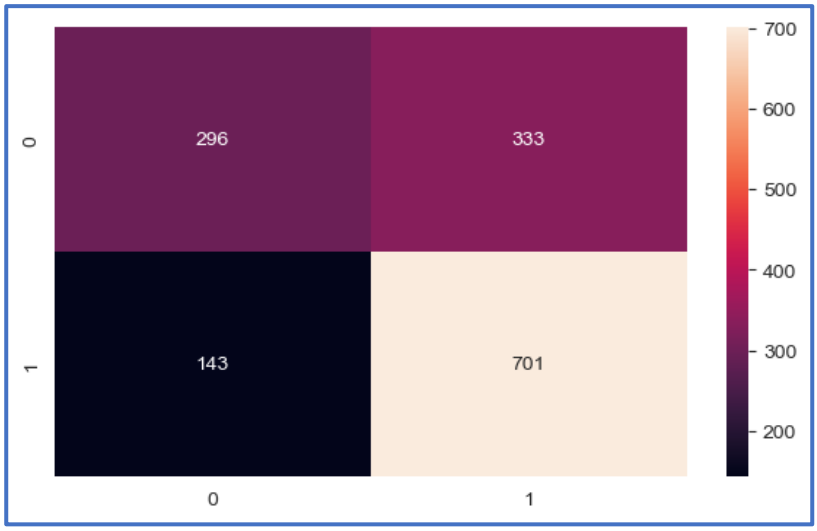

Figure 2.23: Confusion Matrix of LDA

True Negative – 296: Actual 0's are identified as 0's
False Positive(Type I Error) – 333: Actual 0's are identified as 1's
True Positive – 701: Actual 1's are identified as 0's
False Negative(Type II Error) – 143: Actual 1's are identified as 1's

2.3.2.5 Classification Report

	precision	recall	f1-score	support
0	0.67	0.47	0.55	629
1	0.68	0.83	0.75	844
accuracy			0.68	1473
macro avg	0.68	0.65	0.65	1473
weighted avg	0.68	0.68	0.66	1473

Figure 2.24: Classification Table of LDA

The model predicts with 67% accuracy for 0's as 0's
The model predicts with 68% accuracy for 1's as 1's

The model predicts 83% correctly that the women opt for the Contraceptive method of choice

F1 score – Hormonic mean of Precision & Recall. 75% of class 1 indicates that the balanced measure for class 1.

2.3.3 Build a CART model

2.3.3.1 Changing variables to categorical values

```
<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
```

Figure 2.25: Dataset before changing to categorical object type

Decision tree in Python can take only numerical / categorical columns. It cannot take string / objects types.
The following code loops through each column and checks if the column type is object then converts those columns into categorical with each distinct value becoming a category or code.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1473 entries, 0 to 1472
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Wife_age                    1473 non-null   float64
 1   Wife_ education             1473 non-null   int8
 2   Husband_education           1473 non-null   int8
 3   No_of_children_born         1473 non-null   float64
 4   Wife_religion               1473 non-null   int8
 5   Wife_Working                1473 non-null   int8
 6   Husband_Occupation          1473 non-null   int64
 7   Standard_of_living_index    1473 non-null   int8
 8   Media_exposure              1473 non-null   int8
 9   Contraceptive_method_used   1473 non-null   int8
dtypes: float64(2), int64(1), int8(7)
memory usage: 44.7 KB
```

Figure 2.26: Dataset after changing to categorical object type

2.3.3.2 Train Test Split

Train test split done using the function train_test_split

2.3.3.3 Model Building

Model is build using DecisionTreeClassifier with criterion as gini and with train variables

2.3.3.4 Importing Tree

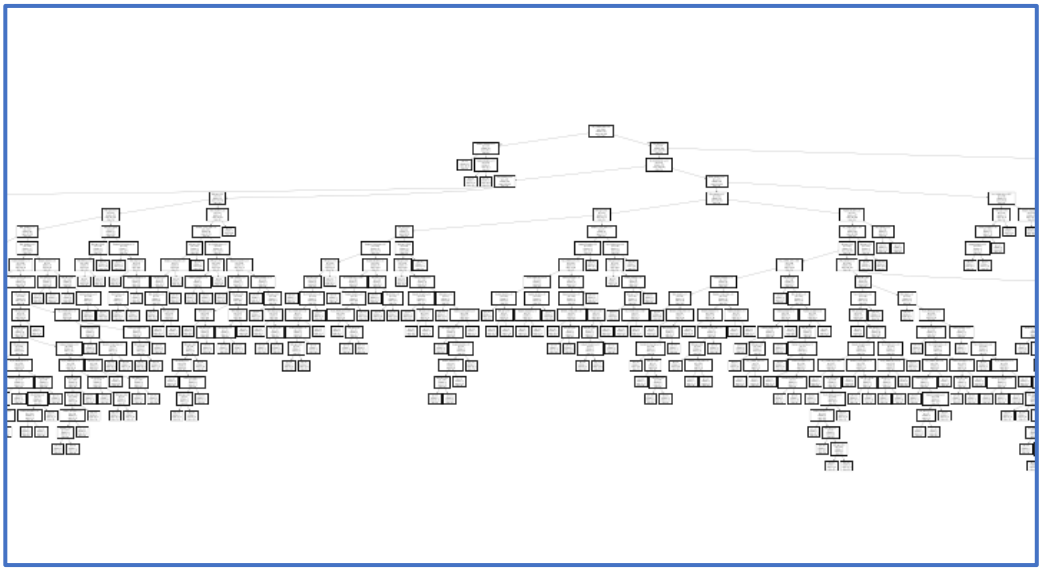

Figure 2.27: Importing Tree

2.3.3.5 Prune the CART model by finding the best hyperparameters using Grid Search

```
▾                    DecisionTreeClassifier
DecisionTreeClassifier(max_depth=7, min_samples_leaf=10, min_samples_split=30)
```

Figure 2.28: Building the Model after Pruning

2.3.3.5.1 Importing Tree

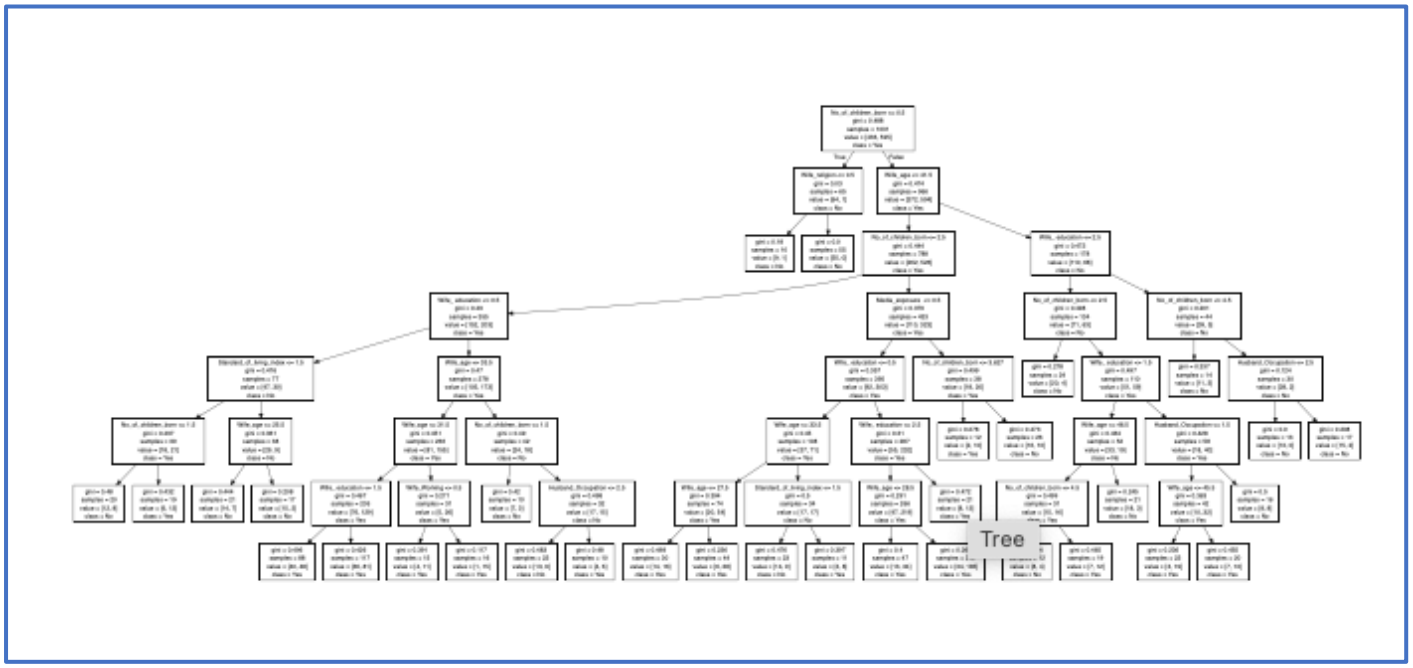

Figure 2.29: importing Tree after Pruning

2.3.3.6 Prediction

array([1, 1, 1, ..., 1, 0, 1], dtype=int8)

array([1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0,
0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1,
1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1,
0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1,
0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,
0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1,
1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0,
1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1,
0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0,
1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0,
0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1,
0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1,
1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1,
0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1,
1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1,
1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
0, 1], dtype=int8)

Check the performance of the models across train and test set using different metrics

2.3.3.7 AUC and ROC for the train data

AUC: 0.805

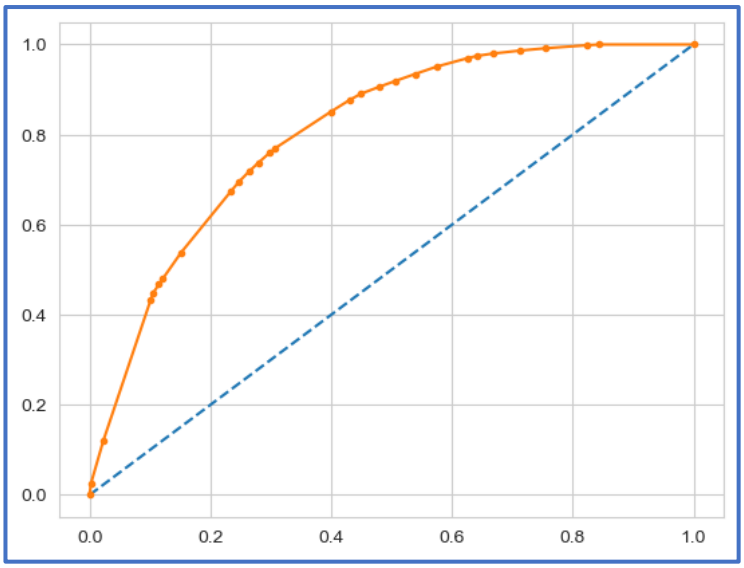

Figure 2.30: AUC of train data

The model is 80% perfect in terms of performance

2.3.3.8 AUC and ROC for the test data

AUC: 0.755

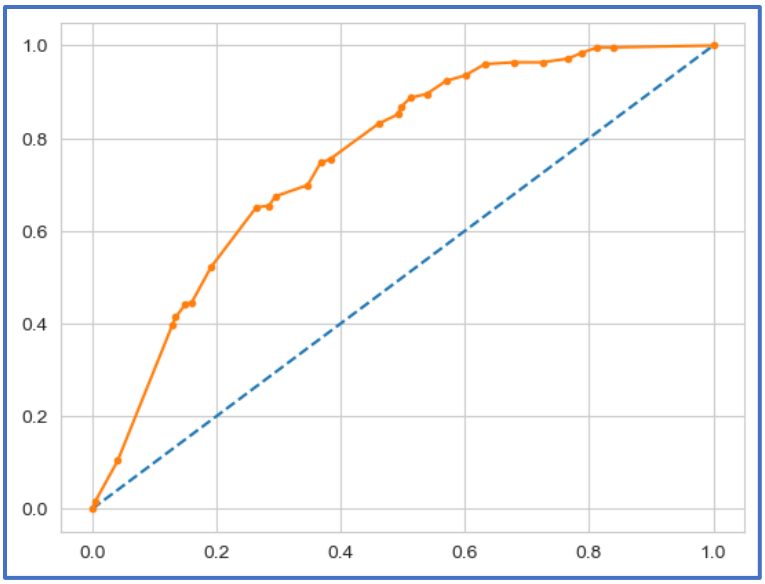

Figure 2.31: AUC of test data

The model is 75% perfect in terms of performance

2.3.3.9 Classification Report for the train data

	precision	recall	f1-score	support
0	0.77	0.57	0.66	436
1	0.74	0.88	0.80	595
accuracy			0.75	1031
macro avg	0.75	0.72	0.73	1031
weighted avg	0.75	0.75	0.74	1031

Figure 2.32: Classification of train data

The model predicts 74% correctly that the women opt for the Contraceptive method of choice

F1 score – Hormonic mean of Precision & Recall. 80% of class 1 indicates that the balanced measure for class 1.

2.3.3.10 Classification Report for the test data

	precision	recall	f1-score	support
0	0.73	0.51	0.60	193
1	0.69	0.85	0.76	249
accuracy			0.70	442
macro avg	0.71	0.68	0.68	442
weighted avg	0.71	0.70	0.69	442

Figure 2.33: Classification of test data

The model predicts 69% correctly that the women opt for the Contraceptive method of choice

F1 score – Hormonic mean of Precision & Recall. 76% of class 1 indicates that the balanced measure for class 1.

2.3.3.11 Confusion Matrix for the train data

array([[248, 188],
[73, 522]])

True Negative – 248: Actual 0's are identified as 0's
False Positive(Type I Error) – 188: Actual 0's are identified as 1's
True Positive – 522: Actual 1's are identified as 0's
False Negative(Type II Error) – 73: Actual 1's are identified as 1's

2.3.3.12 Confusion Matrix for the test data

array([[98, 95],
[37, 212]])

True Negative – 98: Actual 0's are identified as 0's
False Positive(Type I Error) – 95: Actual 0's are identified as 1's
True Positive – 212: Actual 1's are identified as 0's
False Negative(Type II Error) – 37: Actual 1's are identified as 1's

2.3.3.13 Model Score for the train data

The train data model score is 0.747

2.3.3.14 Model Score for the test data

The test data model score is 0.701

2.3.3.15 Compare the performance of all the models built and choose the best one with proper rationale

Class 1 (Contraceptive method used)			
Metric	LR	LDA	CART
Precision	↓ 0.68	↓ 0.68	↑ 0.74
Recall	↓ 0.84	↓ 0.83	↑ 0.88
F1	↓ 0.75	↓ 0.75	↑ 0.8
Accuracy	↓ 0.69	↓ 0.68	↑ 0.75
AUC Score	-	-	↑ 0.8
Model Score	↓ 0.67	↓ 0.68	↑ 0.75

Comparing the metrics of all the 3 classification models the CART (Decision Tree) seems to have higher value of the model. So we can decide to proceed with the CART model.

CART Model:

- The model is 80% perfect in terms of performance
- The model predicts 74% correctly that the women opt for the Contraceptive method of choice
- Type II error is lesser in value compared to other models

Recommendations:

Women who are working, having very high standard of living index, who's education & Husband's Education is Tertiary, with higher media exposure and religion belongs to scientology are highly tend to use a contraceptive method of choice.