PREDICTIVE MODELLING PROJECT

BUSINESS REPORT

Problem : 1.LINEAR REGRESSION

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.

Data head:

ead	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	us
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	9
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	9
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	8
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	9
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	9

Data tail:

read	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap
16	12	3009	360	244	1.6	5.81	405250.0	85282.0	8.02	...	55.11	0.6	35.87	47.90	139.28	270.74	CPU_Bound	387	986647
4	0	1596	170	146	2.4	1.80	89489.0	41764.0	3.80	...	0.20	0.8	3.80	4.40	122.40	212.60	Not_CPU_Bound	263	1055742
16	5	3116	289	190	0.6	0.60	325948.0	52640.0	0.40	...	0.00	0.4	28.40	45.20	60.20	219.80	Not_CPU_Bound	400	969106
32	45	5180	254	179	1.2	1.20	62571.0	29505.0	1.40	...	18.04	0.4	23.05	24.25	93.19	202.81	CPU_Bound	141	1022458
2	0	985	55	46	1.6	4.80	111111.0	22256.0	0.00	...	0.00	0.2	3.40	6.20	91.80	110.00	CPU_Bound	659	1756514

× 22 columns

Data info:

```
<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
```

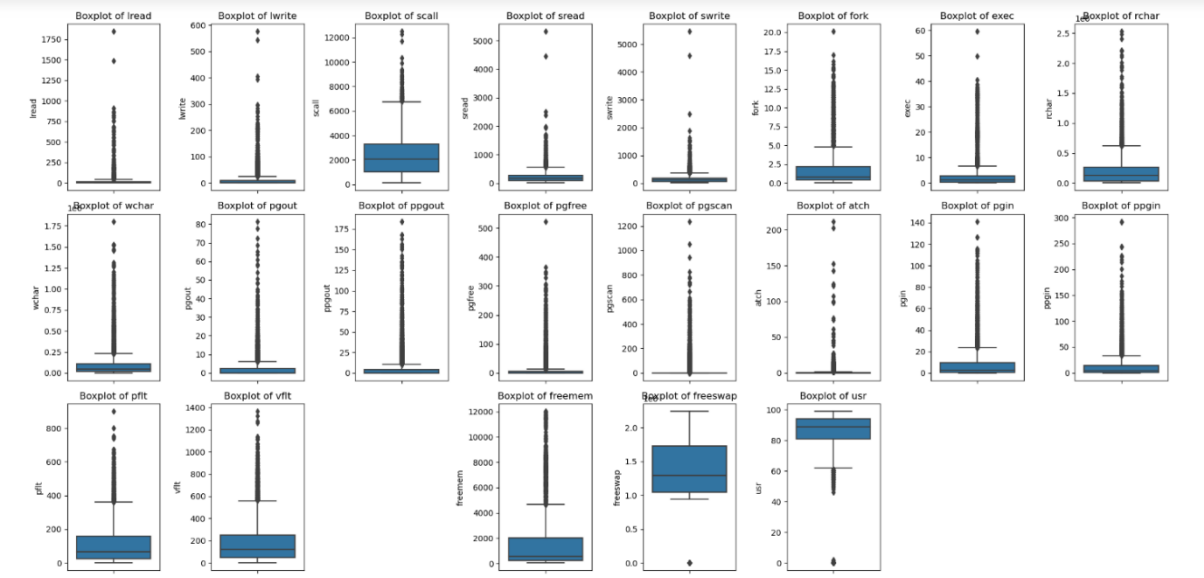
- There are a total 8192 rows and 22 columns in the dataset. Out of 22, 13 are flot 8 are integer and 1 is object type variable.
- We have 8192 rows and 22 columns in our dataset.

Data Summary:

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

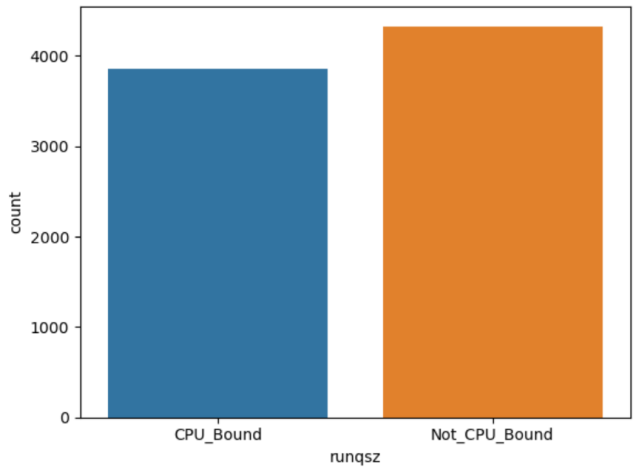
Univariate Analysis:

(Boxplot)



- From this boxplot we can see that we have outliers in our columns we need to treat them before further analysis.

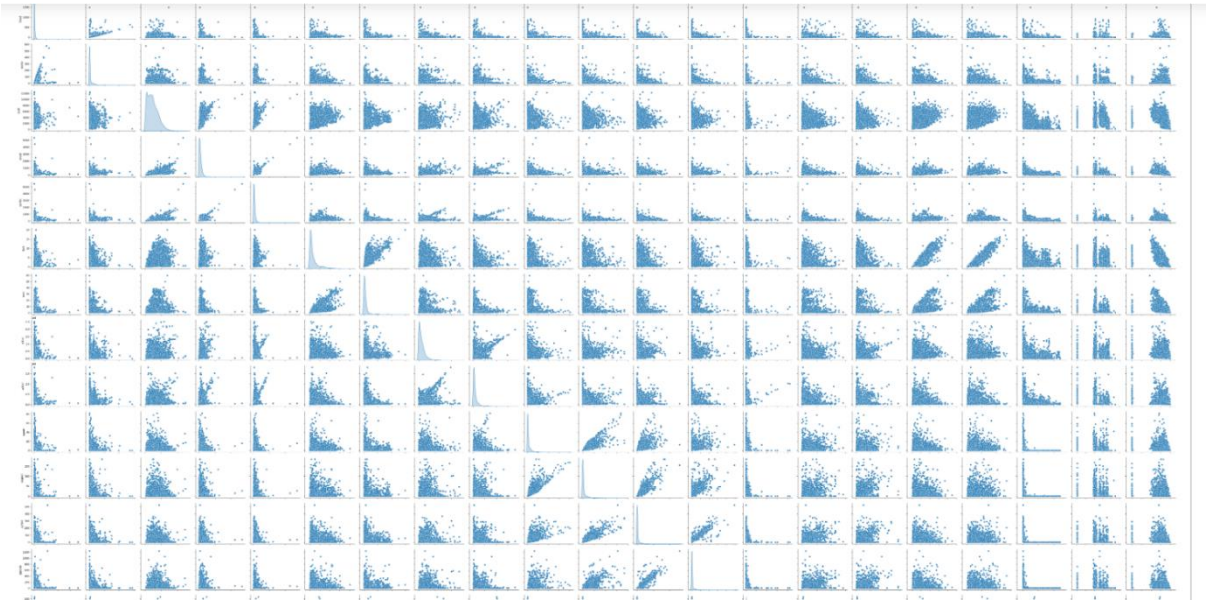
Univariate Analysis for Categorical Data :



- In runqsz variable, 4331 datapints are as Not_CPU_Bound and 3861 datapoints are as CPU_Bound.

Bivariate Analysis:

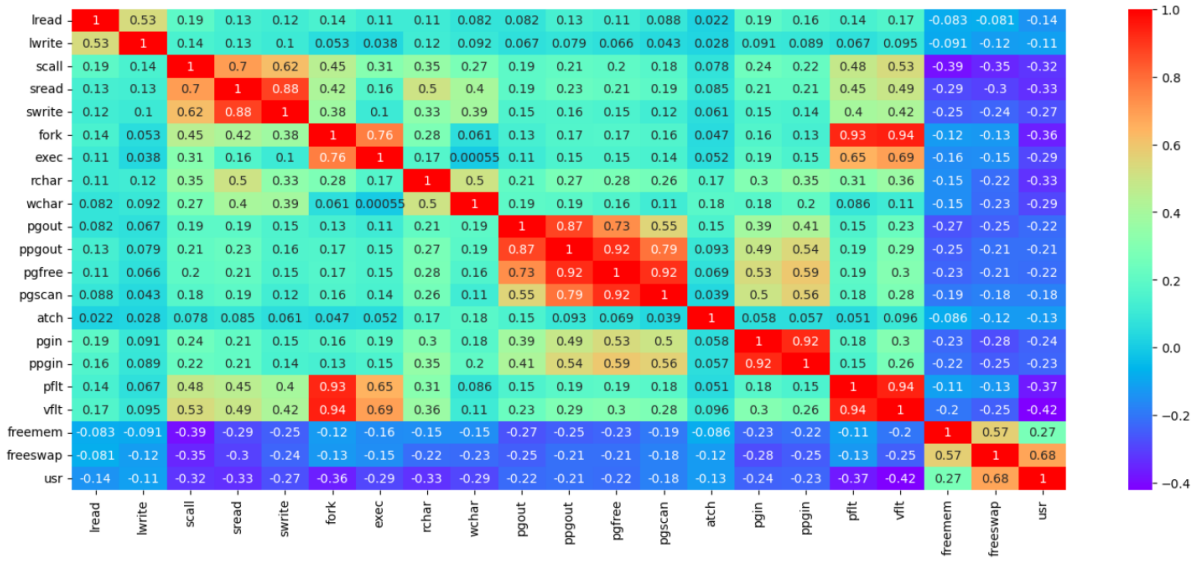
Pairplot (we can see here relation between the variables)



- In this plot we can see some variable are likely linearly related.

Bivariate Analysis

Heatmap (we can see the correlation between the variables, means how strong they are correlated)



- Here we can see some variables has strong correlations. The higher value indicating that.

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

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

- In the data info we have already seen that there are some null values are present in out data.

As they are contentious variables, mean value can be imputed.

```
lread         675
lwrite       2684
scall           0
sread           0
swrite          0
fork           21
exec           21
rchar           0
wchar           0
pgout        4878
ppgout       4878
pgfree       4869
pgscan       6448
atch         4575
pgin         1220
ppgin        1220
pflt            3
vflt            0
runqsz          0
freemem         0
freeswap        0
usr           283
dtype: int64
```

- We have some zero values in our dataset.
- We can keep them in our dataset for further analysis as we have some features in our dataset which can be 0 if the system stay Idle.

Unique values in categorical variable:

```
array(['CPU_Bound', 'Not_CPU_Bound'], dtype=object)
```

- Converting categorical into dummy variable.

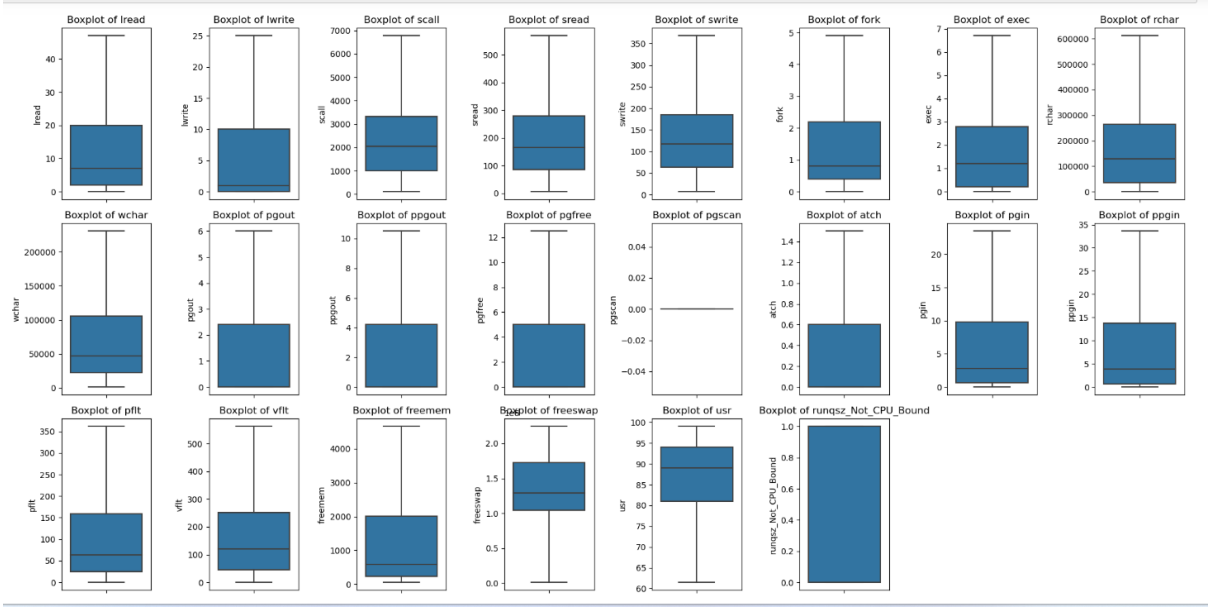
Sample data after data encoding

scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	usr	runqsz_Not_CPU_Bound
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	4670	1730946	95	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	7278	1869002	97	1
2162	159	119	2.0	2.4	NaN	31950.0	0.0	...	0.0	1.2	6.0	9.4	150.20	220.20	702	1021237	87	1
160	12	16	0.2	0.2	NaN	8670.0	0.0	...	0.0	0.0	0.2	0.2	15.60	16.80	7248	1863704	98	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	633	1760253	90	1

umns

- From univariate Analysis we can clearly see that we have outliers in our dataset so we need to treat this outliers.
- IQR is used for treating outliers.

Outliers check after treatment



- In this plot we can see outliers has been treated.

There are no duplicate rows in our dataset.

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

0 rows × 22 columns

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.

We have already encoded the object data to Categorical variable.

After applying the linear regression now let’s check the VIF values which will tells us the multicollinearity in the variables.

```
VIF values:

const                  27.597251
lread                   5.163698
lwrite                  4.260489
scall                   2.963615
sread                   6.735500
swrite                  5.799549
fork                   12.595784
exec                    3.104367
rchar                   2.117378
wchar                   1.612945
pgout                  11.490281
ppgout                 30.813527
pgfree                 17.461990
pgscan                       NaN
atch                    1.836885
pgin                   13.531714
ppgin                  13.828682
pflt                   11.224761
vflt                   14.386823
freemem                 1.956149
freeswap                1.788994
runqsz_Not_CPU_Bound    1.146916
dtype: float64
```

Here pgscan we can see the NaN value showing. in describe data we have already seen that, this variable is containing only higher value and after treating outliers its become zero. That why its showing NaN in VIF values.

If VIF is 1, that means there is no multicollinearity. If VIF is more then 1 we need to check after dropping the variable which is more then 1, is it effecting the r^2 value. If R^2 come down then we cannot drop if R^2 remain same we can drop this.

For p value, if p value is high we should drop the variable and again check the R^2 is it same or come down.

From the above I can say, they are moderate correlation.

Linear Regression Using Scikit-learn:

The intercepts and coefficients for each of independent attributes:

Independent attributes are :

```
lread
lwrite
scall
sread
swrite
fork
exec
rchar
wchar
pgout
ppgout
pgfree
pgscan
atch
pgin
ppgin
pflt
vflt
freemem
freeswap
runqsz_Not_CPU_Bound
```

And there intercept and coefficients are:

```
Intercept: [83.05839481]
Coefficients: [[-4.88128702e-02  3.79393055e-02 -6.55344637e-04  1.16554146e-03
  -6.25997074e-03 -6.11743317e-02 -3.00464139e-01 -4.94214110e-06
  -5.38414435e-06 -4.64270118e-01  4.41758974e-02  2.35973908e-02
  -3.33066907e-16  7.73146181e-01  2.67911125e-02 -7.46791796e-02
  -3.27448535e-02 -4.90936288e-03 -4.52008454e-04  9.29393792e-06
   1.88652746e+00]]
```

For training Data:

```
Training Set Evaluation:
Mean Squared Error (MSE): 20.188694872384822
R-squared (R2) Score: 0.7855998601202907
RMSE: 4.493183155891247
```

For Testing Data:

```
Test Set Evaluation:
Mean Squared Error (MSE): 20.048051022018672
R-squared (R2) Score: 0.7933360631038957
RMSE: 4.477504999664285
```

Linear Regression Model Accuracy of Training Data: 0.786, so 78% of the variation in the usr is explained by the predictors in the model for train set.

Linear Regression Model Accuracy of Test Data: 0.793, so 79% of the variation in the usr is explained by the predictors in the model for test set.

Linear Regression Using Stats Models(OLS):

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    usr   R-squared:                       0.786
Model:                            OLS   Adj. R-squared:                  0.785
Method:                 Least Squares   F-statistic:                     1047.
Date:                Sat, 15 Jul 2023   Prob (F-statistic):               0.00
Time:                        16:21:52   Log-Likelihood:                -16752.
No. Observations:                5734   AIC:                         3.355e+04
Df Residuals:                    5713   BIC:                         3.369e+04
Df Model:                          20
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         83.0584      0.312    265.968      0.000      82.446      83.671
lread         -0.0488      0.009     -5.481      0.000      -0.066      -0.031
lwrite         0.0379      0.013      2.888      0.004       0.012       0.064
scall         -0.0007   6.41e-05    -10.226      0.000      -0.001      -0.001
sread          0.0012      0.001      1.107      0.268      -0.001       0.003
swrite        -0.0063      0.001     -4.286      0.000      -0.009      -0.003
fork          -0.0612      0.133     -0.461      0.645      -0.321       0.199
exec          -0.3005      0.052     -5.781      0.000      -0.402      -0.199
rchar      -4.942e-06   4.93e-07    -10.030      0.000   -5.91e-06   -3.98e-06
wchar      -5.384e-06   1.06e-06     -5.095      0.000   -7.46e-06   -3.31e-06
pgout         -0.4643      0.091     -5.107      0.000      -0.642      -0.286
ppgout         0.0442      0.081      0.544      0.586      -0.115       0.203
pgfree         0.0236      0.050      0.476      0.634      -0.074       0.121
pgscan      -3.57e-14   1.77e-16   -202.042      0.000    -3.6e-14   -3.54e-14
atch           0.7731      0.142      5.428      0.000       0.494       1.052
[illegible]
pgin           0.0268      0.028      0.941      0.347      -0.029       0.083
ppgin         -0.0747      0.020     -3.776      0.000      -0.113      -0.036
pflt          -0.0327      0.002    -16.656      0.000      -0.037      -0.029
vflt          -0.0049      0.001     -3.525      0.000      -0.008      -0.002
freemem       -0.0005   5.18e-05     -8.730      0.000      -0.001      -0.000
freeswap   9.294e-06   1.89e-07     49.244      0.000    8.92e-06    9.66e-06
runqsz_Not_CPU_Bound     1.8865      0.127     14.798      0.000       1.637       2.136
==============================================================================
Omnibus:                      978.352   Durbin-Watson:                   2.011
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1917.427
Skew:                          -1.040   Prob(JB):                         0.00
Kurtosis:                       4.924   Cond. No.                     1.75e+22
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 3.7e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Mean Squared Error of Train and Test Data:

```
MSE for train data: 20.188694872384815
MSE for test data: 20.04805102202073
```

Root Mean Squared Error & R^2 of train and test data:

```
RMSE for train data: 4.493183155891246
RMSE for test data: 4.477504999664515
R^2 for train data: 0.7855998601202908
R^2 for test data: 0.7933360631038745
```

The linear Regression is:

```
y = 83.05839 - 0.04881 * lread + 0.03794 * lwrite - 0.00066 * scall + 0.00117 * sread - 0.00626 * swrite - 0.06117 * fork -
0.30046 * exec - 0.00000 * rchar - 0.00001 * wchar - 0.46427 * pgout + 0.04418 * ppgout + 0.02360 * pgfree - 0.00000 * pgsca
n + 0.77315 * atch + 0.02679 * pgin - 0.07468 * ppgin - 0.03274 * pflt - 0.00491 * vflt - 0.00045 * freemem + 0.00001 * free
swap + 1.88653 * runqsz_Not_CPU_Bound
```

[Here y= usr]

PROBLEM: 2 LOGISTIC REGRESSION, LDA, CART

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

Data head:

ıd_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure	Contraceptive_method_used
Secondary	3.0	Scientology	No	2	High	Exposed	No
Secondary	10.0	Scientology	No	3	Very High	Exposed	No
Secondary	7.0	Scientology	No	3	Very High	Exposed	No
Primary	9.0	Scientology	No	3	High	Exposed	No
Secondary	8.0	Scientology	No	3	Low	Exposed	No

Data tail:

Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure
33.0	Tertiary	Tertiary	NaN	Scientology	Yes	2	Very High	Exposed
33.0	Tertiary	Tertiary	NaN	Scientology	No	1	Very High	Exposed
39.0	Secondary	Secondary	NaN	Scientology	Yes	1	Very High	Exposed
33.0	Secondary	Secondary	NaN	Scientology	Yes	2	Low	Exposed
17.0	Secondary	Secondary	1.0	Scientology	No	2	Very High	Exposed

Data info:

```
<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
```

The Dataset has 7 objects, 2 float type & 1 integer type variable.

Contraceptive_methode_used is the dependent variable.

We have 1473 rows and 10 columns in our dataset.

Data 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

Check Null Values:

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

Here we can see that the columns Wife_age & No_of_children_born has 71 & 21 null values. As they are continuous variable mean value can be imputed.

After Treating the Null Values:

```
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 Data has 80 duplicated rows:

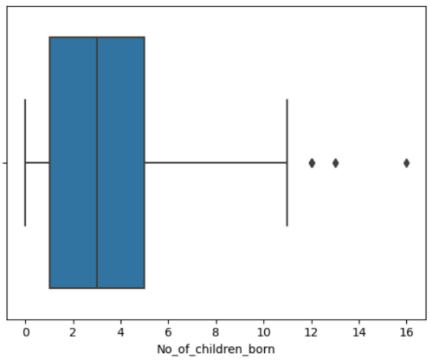
```
Number of duplicated rows =80
```

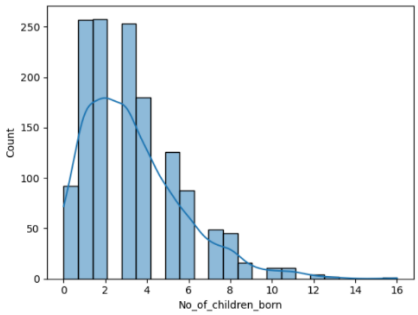
Let's drop them:

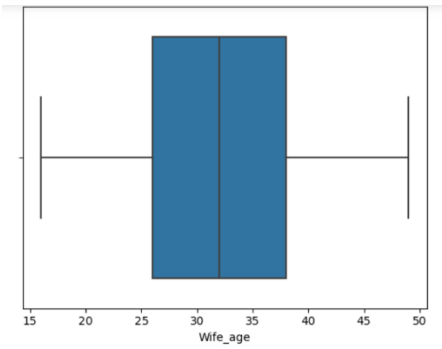
```
Number of duplicated rows =0
```

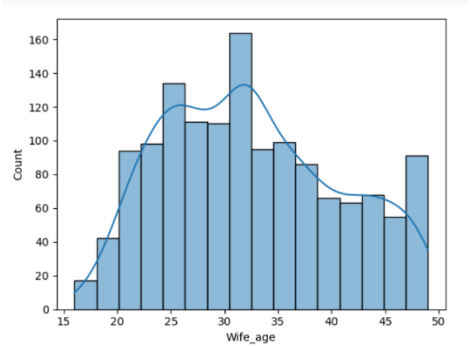
Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposı

Univariate Analysis:

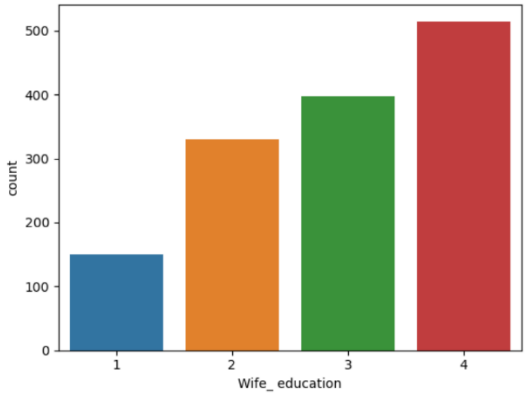


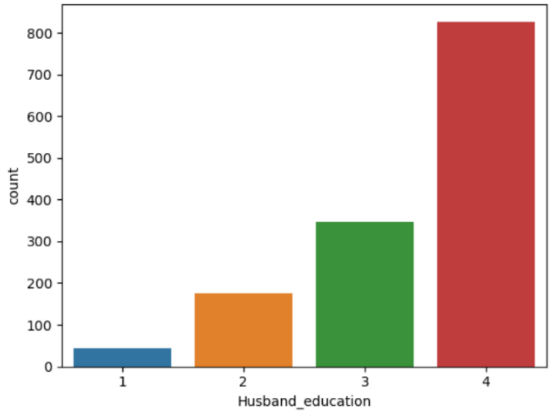


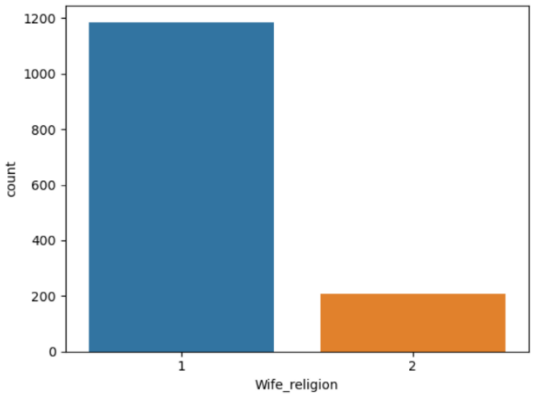


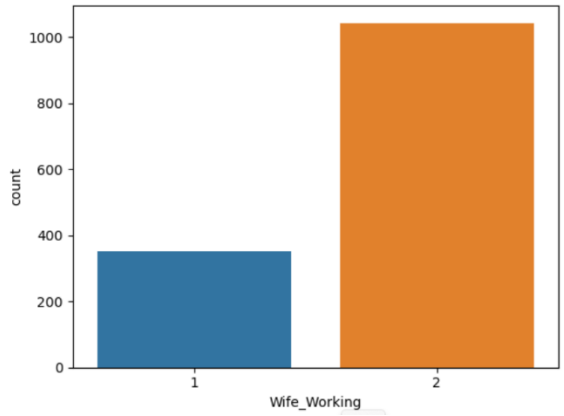


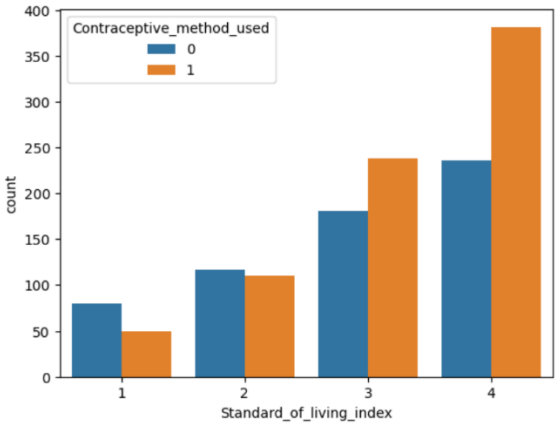
For categorical data:

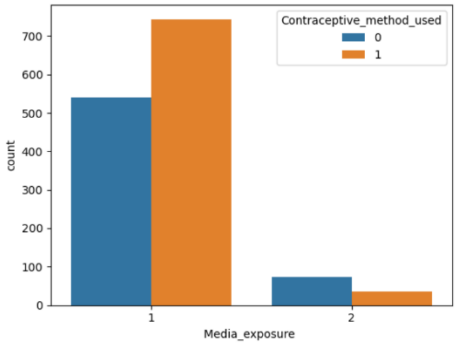




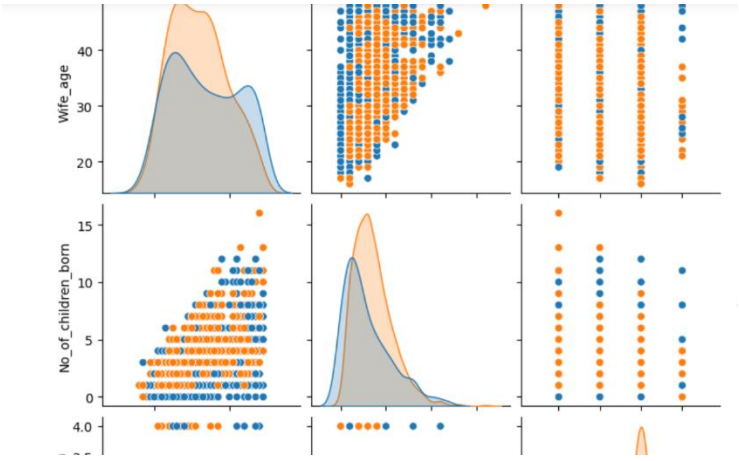


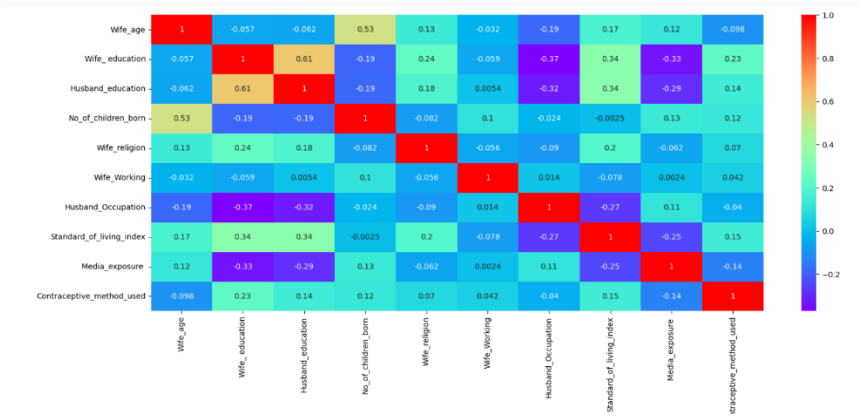






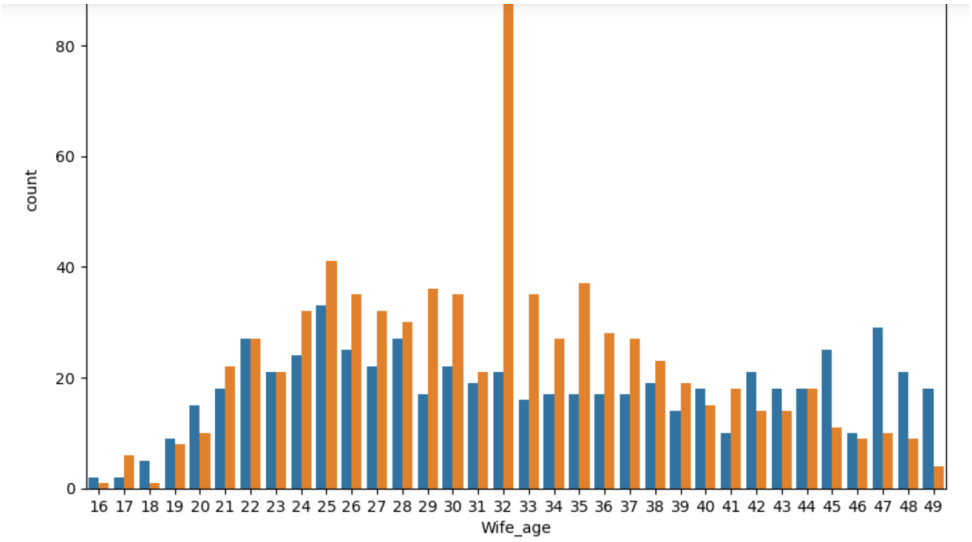
Bivariate Analysis:

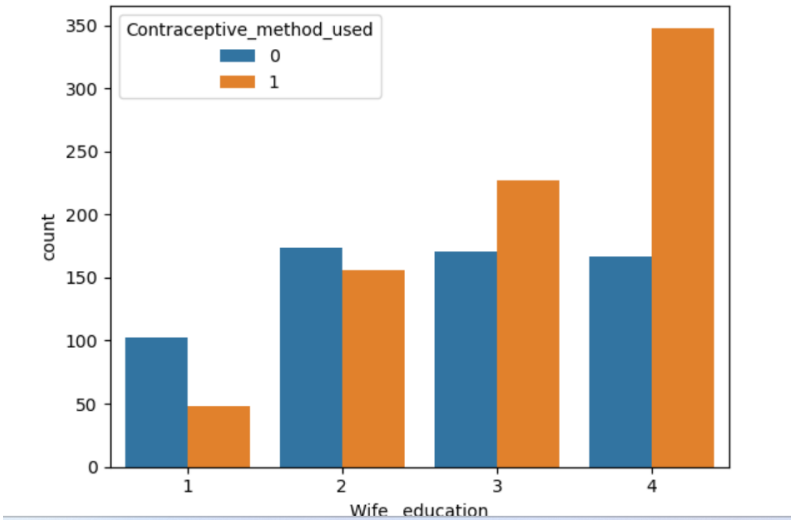


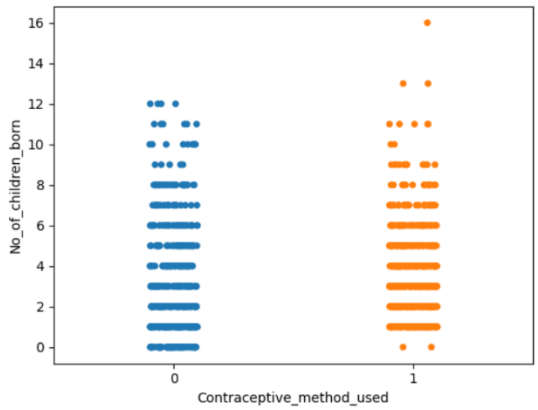


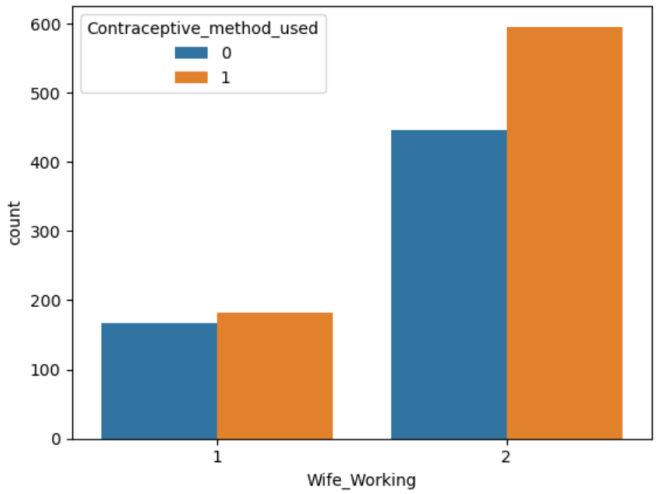
No_of_children_born and Wife_age is slightly correlated.

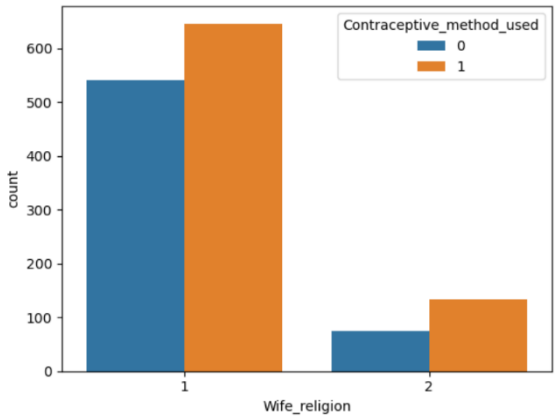
Multivariate Analysis:

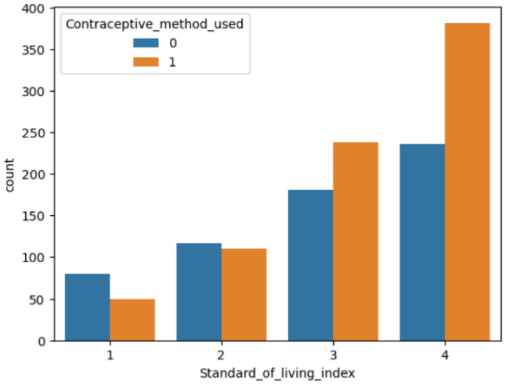


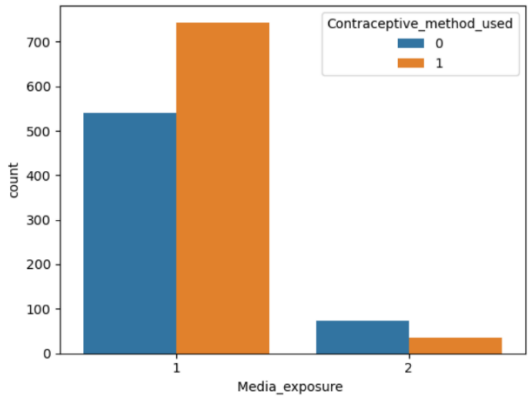



No_of_children_born
Contraceptive_method_used


Contraceptive_method_used
0
1
count
Wife_Working


Contraceptive_method_used
0
1
count
Wife_religion


Contraceptive_method_used
0
1
count
Standard_of_living_index



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.

Encoded the categorical variables Wife_education,Husband_education, Wife_religion, Standard_of_living_index, Media_exposure and Contraceptive_method_used in ascending order from worst of best since LDA does not take string variables as parameters into model building. Below is encoding for ordinal values:

Wife_education: Uneducated = 1, Primary =2, Secondary = 3, Tertiary = 4.

Husband_education: Uneducated = 1, Primary =2, Secondary = 3, Tertiary = 4.

Wife_religion: Scientology = 1, non-Scientology = 2.

Wife_Working: Yes=1, No=2

Standard_of_living_index: Very Low=1, Low=2, High=3, Very high=4

Media_exposure: Exposed = 1, Not-Exposed=2

Contraceptive_method_used: Yes = 1, No= 0. (Dependent)

Wife_ ıcation	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure	Contraceptive
2	3	3	1	2	2	3	1	
1	3	10	1	2	3	4	1	
2	3	7	1	2	3	4	1	
3	2	9	1	2	3	3	1	
3	3	8	1	2	3	2	1	

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1393 entries, 0 to 1472
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Wife_age                  1393 non-null   int32
 1   Wife_ education           1393 non-null   int64
 2   Husband_education         1393 non-null   int64
 3   No_of_children_born       1393 non-null   int32
 4   Wife_religion             1393 non-null   int64
 5   Wife_Working              1393 non-null   int64
 6   Husband_Occupation        1393 non-null   int64
 7   Standard_of_living_index  1393 non-null   int64
 8   Media_exposure            1393 non-null   int64
 9   Contraceptive_method_used 1393 non-null   int64
dtypes: int32(2), int64(8)
memory usage: 141.1 KB
```

There is all int datatype now.

Bivariate Analysis:

Logistic Regression Model:

```
LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='newton-cg',
                   verbose=True)
```

Logistic Regression Model Prediction on Train set:

```
[1 1 1 0 1 0 1 1 0 1 0 1 0 1 1 1 0 1 0 1 1 0 1 1 1 1 1 0 0 1 0 0 1 0 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0
 1 0 1 0 0 0 1 0 1 0 1 1 0 0 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 0
 0 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 0 1 1 0 0 1
 1 1 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1 1
 0 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 0 1 1
 1 0 0 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 1 0 1
 0 1 0 0 0 0 0 0 1 0 0 1 1 1 0 1 1 0 1 1 1 1 0 0 1 1 0 1 0 1 0 1 1 0 0 1 0
 1 0 0 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1 0 1 0 0 1 1 0 1 1 1 1 1 0 1 1 1 0 1 0 1 1
 1 1 1 0 0 1 1 0 1 1 0 0 0 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 0
 1 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 0 0 1 0 0 1 0 1 1 1 1 1 1 1 0 0 1 0
 1 1 0 1 0 0 0 0 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1
 1 1 0 1 0 0 0 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1 0
 1 1 1 0 0 0 0 0 1 0 1 0 1 1 1 1 1 0 0 1 1 1 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1
 1 1 0 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0 1 1 0 1 1 0 1
 1 0 1 0 1 1 1 0 1 0 0 0 1 0 0 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 1 1 1
 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 0 0 0 1 1 1 1 0 1 0 1 1 1 0 0 0 0 1 1
 0 0 1 1 0 1 1 0 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1
 1 0 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
 1 1 1 0 1 0 1 1 0 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1
 1 1 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 1 0 1 0 0 0 1 1 1 1 0 1 1 0 1 1 0 1
 1 0 1 1 0 1 0 1 0 1 1 0 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
 0 1 1 0 1 0 1 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0 0 1 1 1 1 0 0 1 0 1 0 1 0 0 1
 1 1 0 1 1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 0 1 1
 1 1 1 1 1 1 1 1 0 1 0 1 1]
```

Logistic Regression Model Prediction on Test set:

```
[1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 1 0 1 0 0 1 1
 1 1 1 0 0 0 1 1 1 1 0 1 0 1 1 1 1 0 0 0 0 1 0 1 1 0 0 1 1 1 0 1 1 1 0 1 1
 1 0 1 0 1 1 1 1 1 1 1 0 1 0 0 1 1 0 1 0 0 1 0 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 1
 1 0 1 1 1 0 1 1 0 1 0 0 0 1 0 0 1 0 0 1 0 1 0 0 1 1 0 0 0 0 1 1 1 1 0 1 1
 0 1 1 0 1 1 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 1 1 0 0 1 0 1 1 0 1 0 0 1 1 1 0 1
 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 0 1 1 0 1
 1 0 1 1 0 1 1 1 0 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0
 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1
 1 1 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0 1 1 0 1 1 1 0 0 0 0 1 1 1 1 0
 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
 1 1 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1
 0 1 0 0 1 1 0 0 1 1 0]
```

Probabilities on Train set:

	0	1
0	0.380982	0.619018
1	0.351590	0.648410
2	0.298429	0.701571
3	0.570795	0.429205
4	0.495381	0.504619

Probabilities on Test set:

	0	1
0	0.316618	0.683382
1	0.319686	0.680314
2	0.079646	0.920354
3	0.201060	0.798940
4	0.396030	0.603970

Logistic Regression Model Accuracy on Train Set: `0.6574358974358975`
Logistic Regression model Accuracy on Test Set: `0.6602870813397129`

Applying GridSearchCV for Logistic Regression:

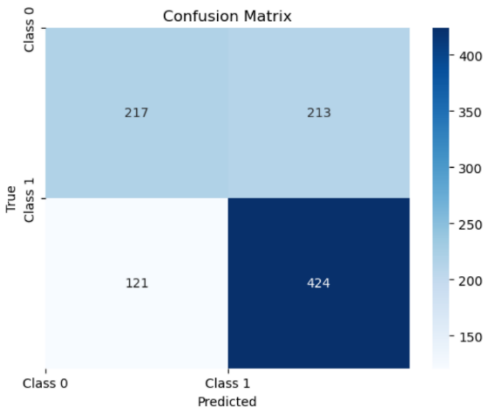
```
GridSearchCV(cv=3, estimator=LogisticRegression(max_iter=10000, n_jobs=2),
             n_jobs=-1,
             param_grid={'penalty': ['12', 'none'], 'solver': ['sag', 'lbfgs'],
                         'tol': [0.0001, 1e-05]},
             scoring='f1')
```

```
{'penalty': 'none', 'solver': 'sag', 'tol': 0.0001}
```

```
LogisticRegression(max_iter=10000, n_jobs=2, penalty='none', solver='sag')
```

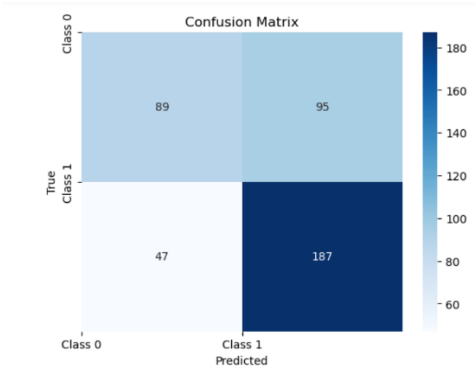
Classification Report and Confusion Matrix(train set):

	precision	recall	f1-score	support
0	0.64	0.50	0.57	430
1	0.67	0.78	0.72	545
accuracy			0.66	975
macro avg	0.65	0.64	0.64	975
weighted avg	0.66	0.66	0.65	975

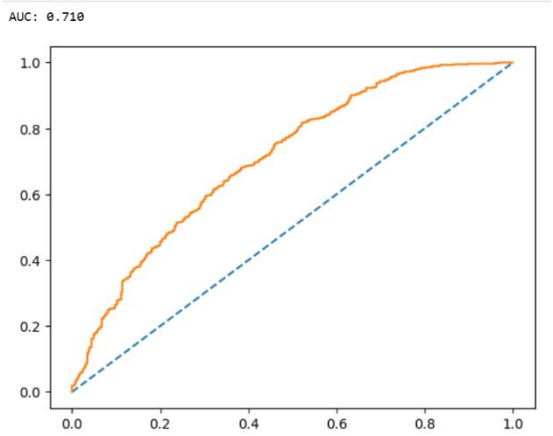


Classification Report and Confusion Matrix(test set):

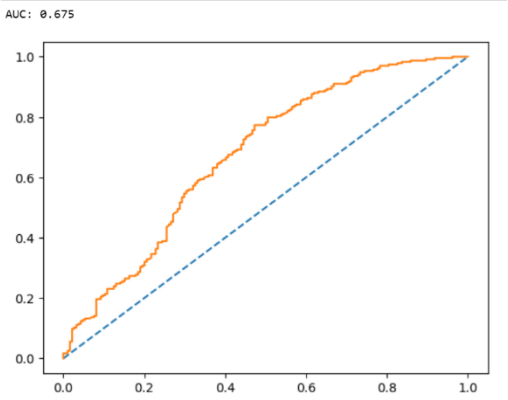
	precision	recall	f1-score	support
0	0.65	0.48	0.56	184
1	0.66	0.80	0.72	234
accuracy			0.66	418
macro avg	0.66	0.64	0.64	418
weighted avg	0.66	0.66	0.65	418



AUC and ROC for Train set:



AUC and ROC for Test set:



Logistic Regression Accuracy on Train set: 66.5%

Logistic Regression Accuracy on Test set: 66%

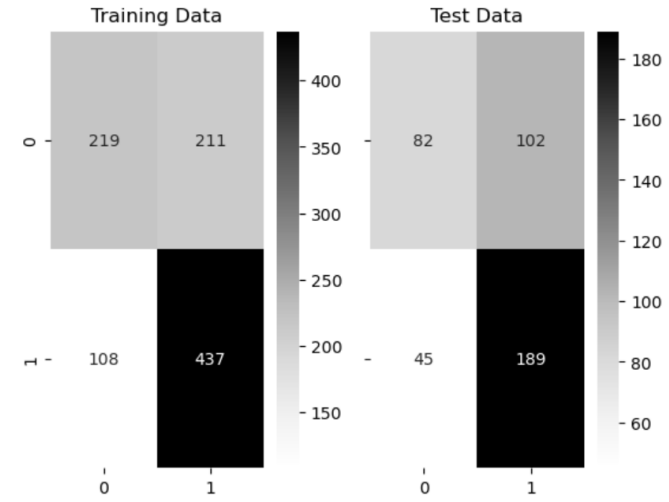
LDA Model:

Intercept value LDA Model: 0.29091934
Co-efficients for LDA Model:

```
array([[-0.59179189,  0.52315047,  0.02985055,  0.73958093,  0.16005024,
         0.07410451,  0.11734836,  0.31440571, -0.0920293 ]])
```

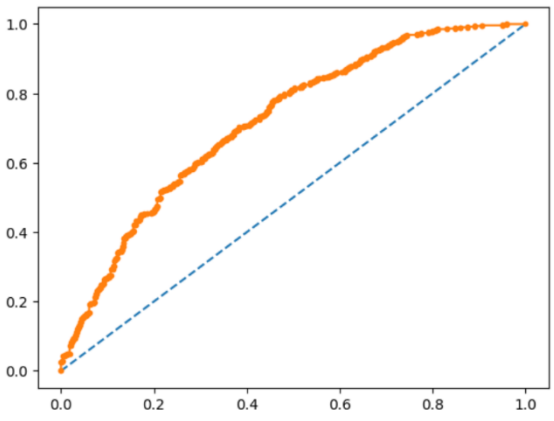
Classification Report & Confusion Matrix:

```
classification report of training data:

              precision    recall  f1-score   support

           0       0.67      0.51      0.58       430
           1       0.67      0.80      0.73       545

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


classification report of test data:

              precision    recall  f1-score   support

           0       0.65      0.45      0.53       184
           1       0.65      0.81      0.72       234

    accuracy                           0.65       418
   macro avg       0.65      0.63      0.62       418
weighted avg       0.65      0.65      0.64       418
```



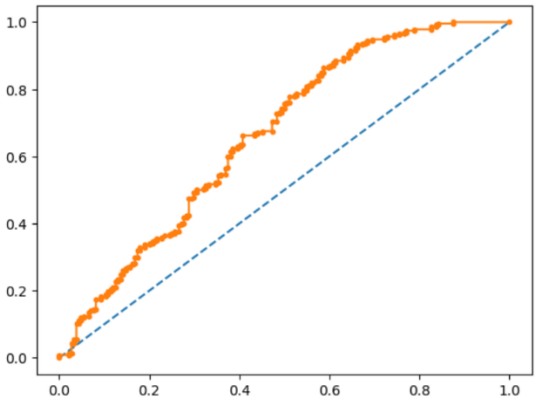
AUC & ROC for train set:

```
AUC for the Traning Data: 0.719
Out[133]: [<matplotlib.lines.Line2D at 0x179cb7315b0>]
```



For Test set:

```
AUC for the Traning Data: 0.664
34]: [<matplotlib.lines.Line2D at 0x179cee4e9d0>]
```



LDA Model Accuracy for Train set: 67%

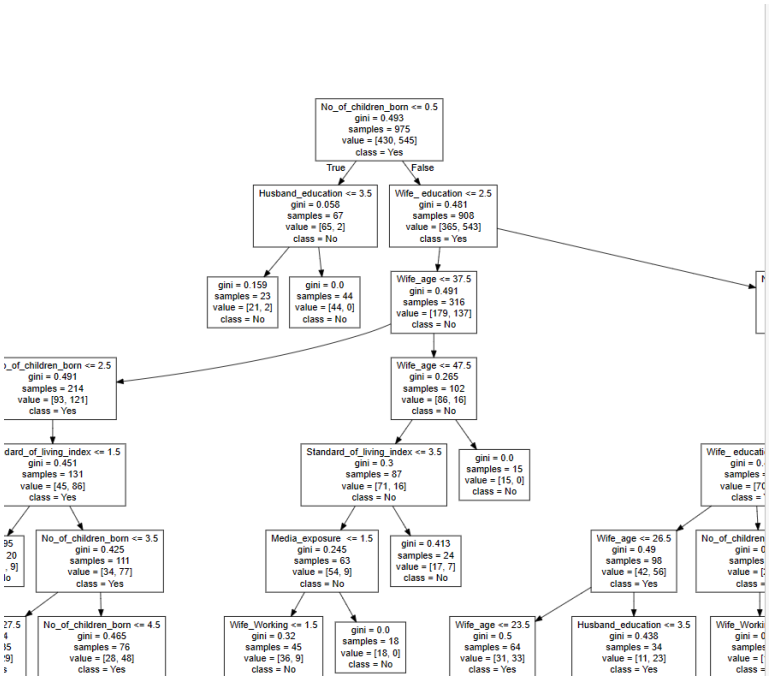
LDA Model Accuracy For Test set: 65%

CART MODEL

```
DecisionTreeClassifier()
```

Decision tree using Graphviz:

	Imp
Wife_age	0.391566
No_of_children_born	0.202849
Wife_ education	0.086672
Standard_of_living_index	0.086631
Husband_education	0.063957
Husband_Occupation	0.062836
Wife_Working	0.053269
Wife_religion	0.028906
Media_exposure	0.023314



CART Model prediction on train set:

```
[1 1 1 1 0 1 1 1 1 1 0 1 0 1 0 0 0 1 1 1 1 0 1 1 1 1 0 0 0 1 0 0 0 0 1 1 0
 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 0
 1 0 0 0 0 0 1 0 0 0 1 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0
 0 0 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1
 1 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1
 1 0 1 1 1 0 0 1 0 1 0 1 0 0 1 1 1 0 1 0 0 1 0 1 1 0 1 1 1 1 1 1 1 1 1 0 1
 0 0 1 0 1 1 1 1 1 1 1 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1
 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 1 0 1 1 1 0 0 0 0 1 0 1
 1 1 0 0 1 0 1 0 1 1 0 1 1 1 0 1 1 0 1 0 1 1 0 0 1 1 0 1 1 1 0 0 0 0 0 1 0
 1 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 0 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1
 1 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 1 1 1 1 1 0 0 0 1 1 1 1 0 1 0 1 0 0
 1 1 1 0 0 1 0 0 1 0 1 0 0 1 0 1 1 1 0 0 0 1 1 0 1 1 0 1 1 1 1 1 1 0 1 0 0 0
 1 0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 0 0 0 1 1 1
 1 1 1 1 1 0 0 0 1 1 1 0 1 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 0 0 1 1 1
 1 0 1 1 1 0 0 0 1 0 1 0 1 1 1 1 0 0 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 0 1 1 0 1
 1 1 0 1 1 1 0 1 1 0 1 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 0 0
 1 1 1 0 0 1 1 0 1 1 0 0 1 0 0 1 1 1 1 1 0 0 1 1 0 1 1 0 0 1 1 1 0 1 0 1
 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1
 0 1 0 1 0 1 1 0 0 1 0 0 0 1 1 1 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 1 0 1 0 1 1 1 1
 1 0 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 0 1 0 0 1 1
 1 1 0 0 1 0 1 1 1 0 0 1 1 1 0 1 1 1 0 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1
 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 1 1 0 1 1 0 1 1 0 1
 1 0 1 0 0 1 0 1 0 1 1 0 1 1 1 1 1 0 0 1 0 1 0 0 1 1 1 1 1 1 1 0 1 1 1 0 1 0
 1 1 1 0 1 0 1 0 1 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 0 0
 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 0 1 1 0 1 1 0 0 0 1 0 0 0 1 1
 1 1 1 1 1 1 0 0 0 1 1 1 1]
```

CART Model prediction on Test set:

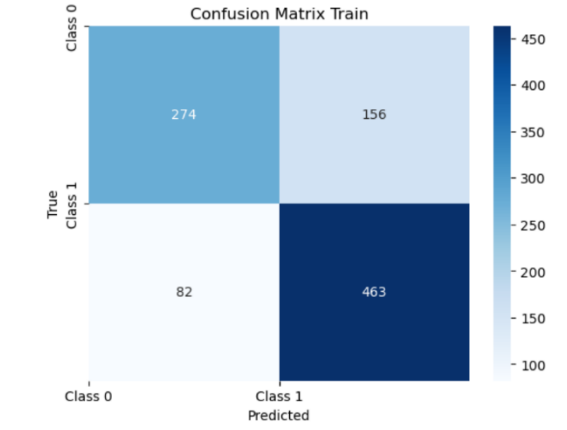
```
[1 1 1 1 0 0 1 1 1 1 0 1 1 0 1 0 1 1 1 0 1 1 1 0 1 0 0 0 0 1 0 0 1 0 0 0 1
 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 0 0 0 0 1 1 0 1 0 0 1 1 1 0 1 1 1 1 1 1
 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1
 1 0 1 1 1 0 1 1 0 1 0 0 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 0 0 1 1 0 1 1 1 0
 0 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 1 0 0 1 1 1 1 1
 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1
 1 0 0 1 0 1 1 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1 1 0
 0 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1
 1 0 1 1 0 1 0 0 1 0 1 1 1 1 0 1 1 1 1 0 0 0 1 1 1 0 1 1 0 0 0 1 1 1 1 1 0 0
 1 1 1 1 0 1 0 1 0 1 1 0 0 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0
 1 0 1 1 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 0 1 0 1 1 0 1 1 1 1 1 1 0 1 1 0 1
 1 0 1 0 1 1 0 0 1 1 1 1]
```

CART Model Test prediction:

	0	1
0	0.142857	0.857143
1	0.321429	0.678571
2	0.133333	0.866667
3	0.104167	0.895833
4	0.900000	0.100000

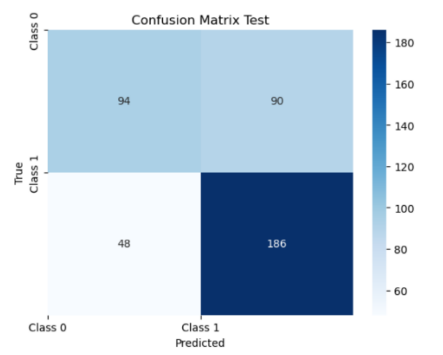
CART Model Classification Report and Confusion Matrix on train data:

	precision	recall	f1-score	support
0	0.77	0.64	0.70	430
1	0.75	0.85	0.80	545
accuracy			0.76	975
macro avg	0.76	0.74	0.75	975
eighted avg	0.76	0.76	0.75	975



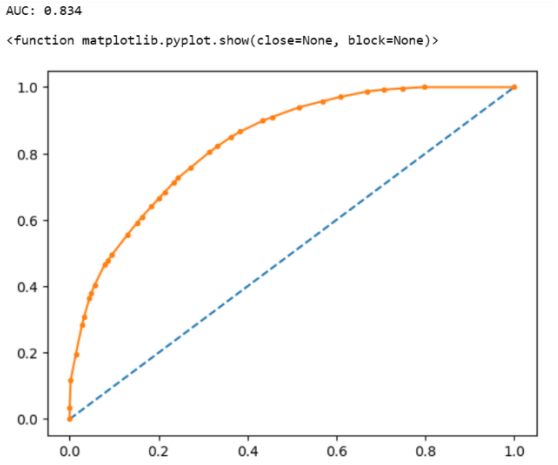
CART Model Classification Report and Confusion Matrix on test data:

	precision	recall	f1-score	support
0	0.66	0.51	0.58	184
1	0.67	0.79	0.73	234
accuracy			0.67	418
macro avg	0.67	0.65	0.65	418
eighted avg	0.67	0.67	0.66	418

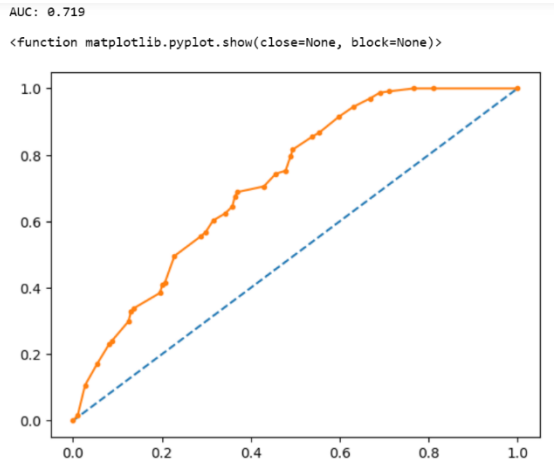


AUC & ROC for Train and Test data:

Train:



Test:



CART Model Accuracy on Train data:76%

CART Model Accuracy on Test data:67%

Comparing both these model, we find both result are same, but Logistic Regression works better when there is category target variable.

#The EDA analysis clearly indicates that women with a tertiary education and very high standard of living used contraceptive methods Women ranging from 21 to 38 generally use contraceptive methods

more

#The usage of contraceptive methods need not depend on their demographic or socioeconomic backgrounds since

the use of contraceptive methods were almost the same for both working and non-working women

#The use of contraceptive method was high for both Scientology and Non-scientology women