PREDICTIVE MODELLING BUSSINESS REPORT

THANUSRI A

14-01-2024

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

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

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.

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

The top 5 rows of the dataset:

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap
0	1	0	2147	79	68	0.2	0.2	40671.0	53995.0	0.0	...	0.0	0.0	1.6	2.6	16.00	26.40	CPU_Bound	4670	1730946
1	0	0	170	18	21	0.2	0.2	448.0	8385.0	0.0	...	0.0	0.0	0.0	0.0	15.63	16.83	Not_CPU_Bound	7278	1869002
2	15	3	2162	159	119	2.0	2.4	NaN	31950.0	0.0	...	0.0	1.2	6.0	9.4	150.20	220.20	Not_CPU_Bound	702	1021237
3	0	0	160	12	16	0.2	0.2	NaN	8670.0	0.0	...	0.0	0.0	0.2	0.2	15.60	16.80	Not_CPU_Bound	7248	1863704
4	5	1	330	39	38	0.4	0.4	NaN	12185.0	0.0	...	0.0	0.0	1.0	1.2	37.80	47.60	Not_CPU_Bound	633	1760253

- Note that there are many zeroes in few of the features

The last 5 rows of the dataset:

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freesv
8187	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	986
8188	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	1055
8189	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	969
8190	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	1022
8191	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	1756

The number of rows and columns in the dataset:

```
no.of rows:  8192
no.of columns: 22
```

- The shape of the data is (8192,22)

Dataset summary:

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

Basic info about the dataset:

```
<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 of 8192 rows and 22 columns in the dataset. Out of 22, 13 are float 8 are integer type and 1 object type variable.

Dataset null value check:

```
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
```

- There are missing values present in ‘rchar’ , ‘wchar’.

Let us treat them using median value

```
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
```

There are no duplicate rows present:

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

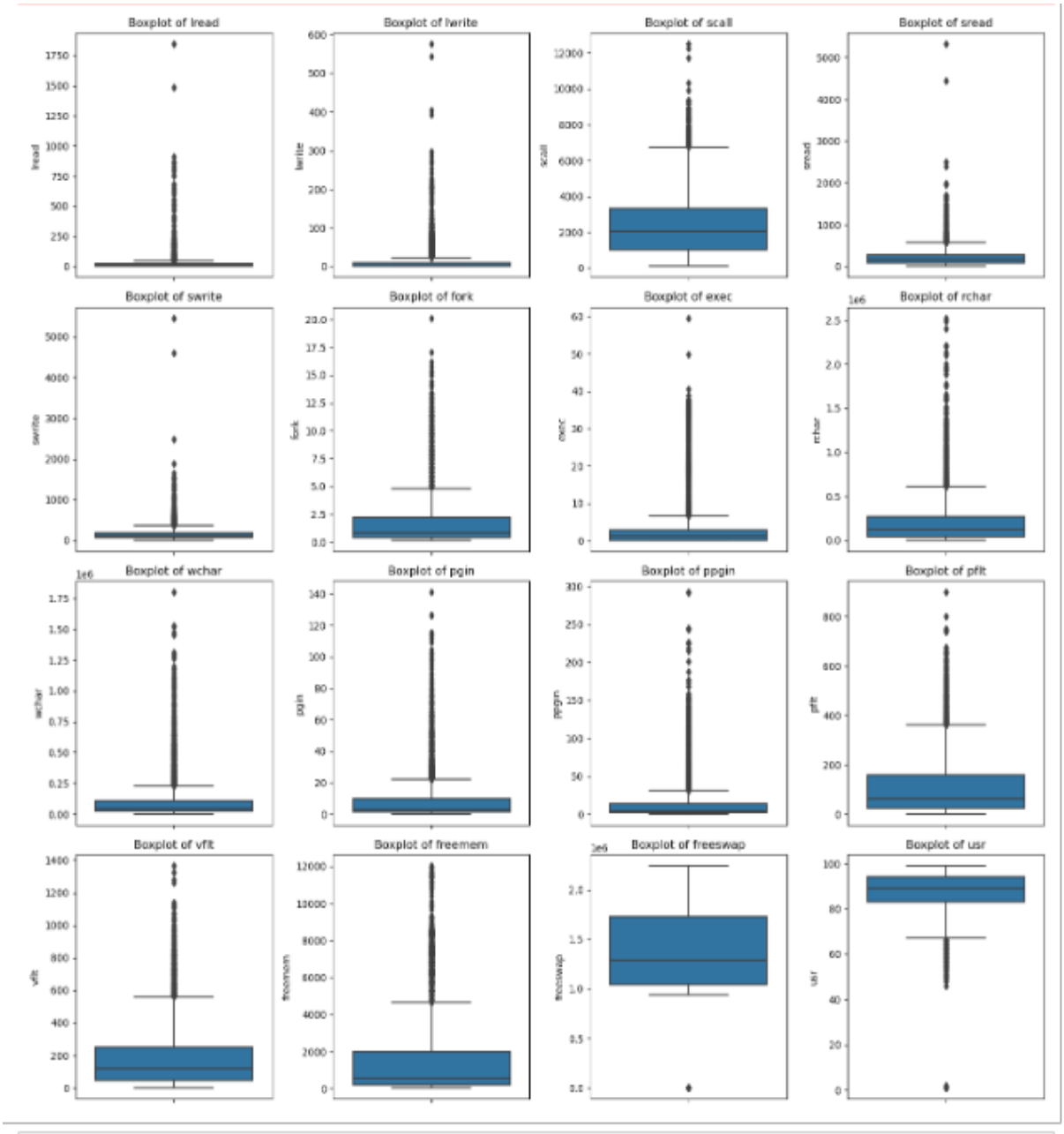
- There are many 0 values present in few of the variables
- Let us check the number of zeroes in each feature and remove accordingly if more than 50 percent of the values are 0's
- The following features have more than 50 percent of 0's in the variables 'pgout','ppgout','pgfree','pgscan','atch'
- So we drop these 5 columns from the dataset.
- For rest of the features, let us treat all the zeroes with median values

The dataset after removing the required features and computing median values for the rest of the features looks like:

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
0	1	1	2147	79	68	0.2	0.20	40671.0	53995.0	1.60	2.60	16.00	26.40	CPU_Bound	4670	1730946	95
1	7	1	170	18	21	0.2	0.20	448.0	8385.0	2.80	3.80	15.63	16.83	Not_CPU_Bound	7278	1869002	97
2	15	3	2162	159	119	2.0	2.40	125473.5	31950.0	6.00	9.40	150.20	220.20	Not_CPU_Bound	702	1021237	87
3	7	1	160	12	16	0.2	0.20	125473.5	8670.0	0.20	0.20	15.60	16.80	Not_CPU_Bound	7248	1863704	98
4	5	1	330	39	38	0.4	0.40	125473.5	12185.0	1.00	1.20	37.80	47.60	Not_CPU_Bound	633	1760253	90
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8187	16	12	3009	360	244	1.6	5.81	405250.0	85282.0	35.87	47.90	139.28	270.74	CPU_Bound	387	986647	80
8188	4	1	1596	170	146	2.4	1.80	89489.0	41764.0	3.80	4.40	122.40	212.60	Not_CPU_Bound	263	1055742	90
8189	16	5	3116	289	190	0.6	0.60	325948.0	52640.0	28.40	45.20	60.20	219.80	Not_CPU_Bound	400	969106	87
8190	32	45	5180	254	179	1.2	1.20	62571.0	29505.0	23.05	24.25	93.19	202.81	CPU_Bound	141	1022458	83
8191	2	1	985	55	46	1.6	4.80	111111.0	22256.0	3.40	6.20	91.80	110.00	CPU_Bound	659	1756514	94

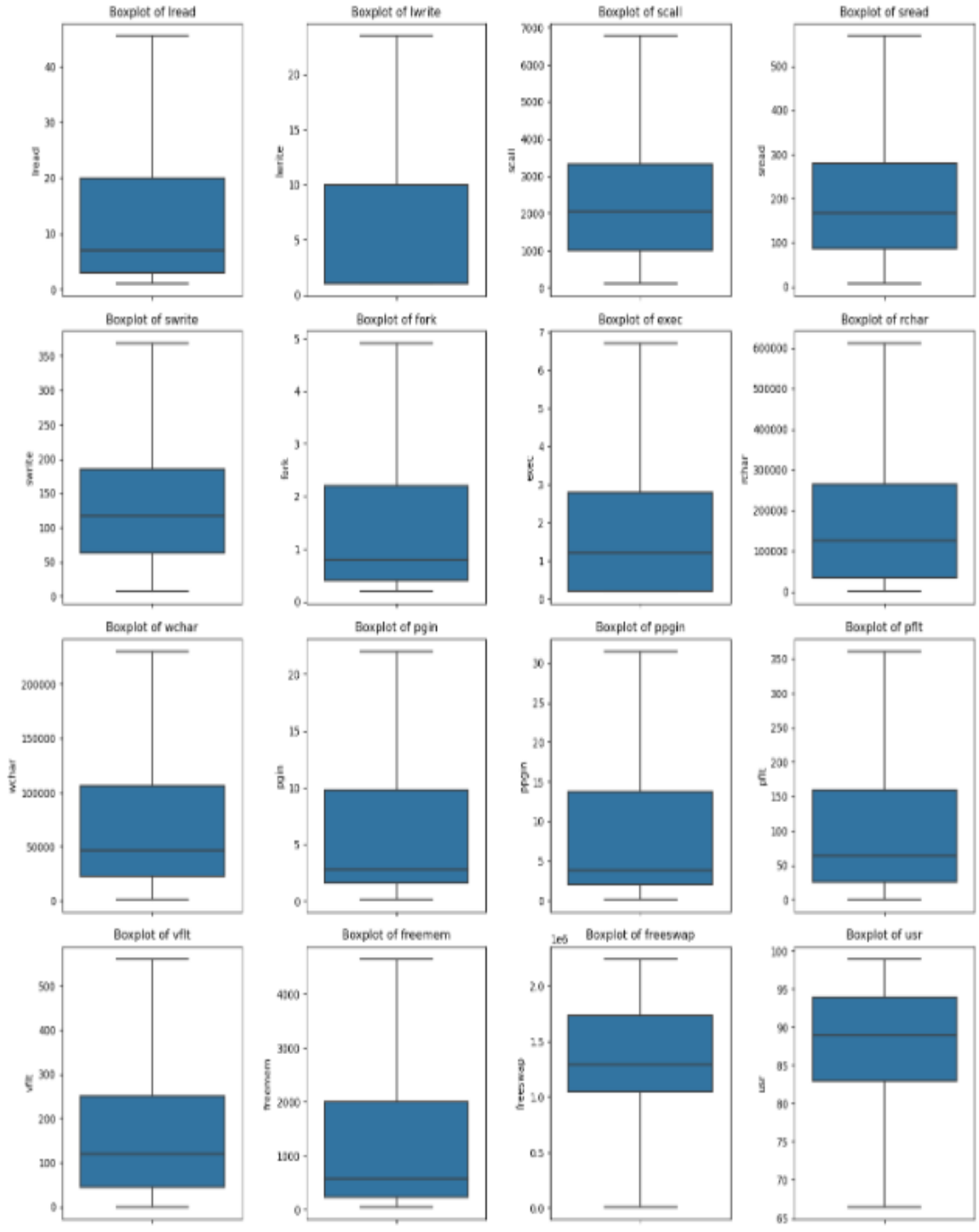
8192 rows × 17 columns

- Now we create a dataframe that contains only the integer and float type variables and try printing the boxplots for these features.



- There are outliers present in the data, these needs to be treated
- There are many methods in which outliers can be treated
- We choose IQR method to treat them
- So, we treat them using IQR method. In this method, any observation that is less than Q1- 1.5 IQR or more than Q3 + 1.5 IQR is considered an outlier.

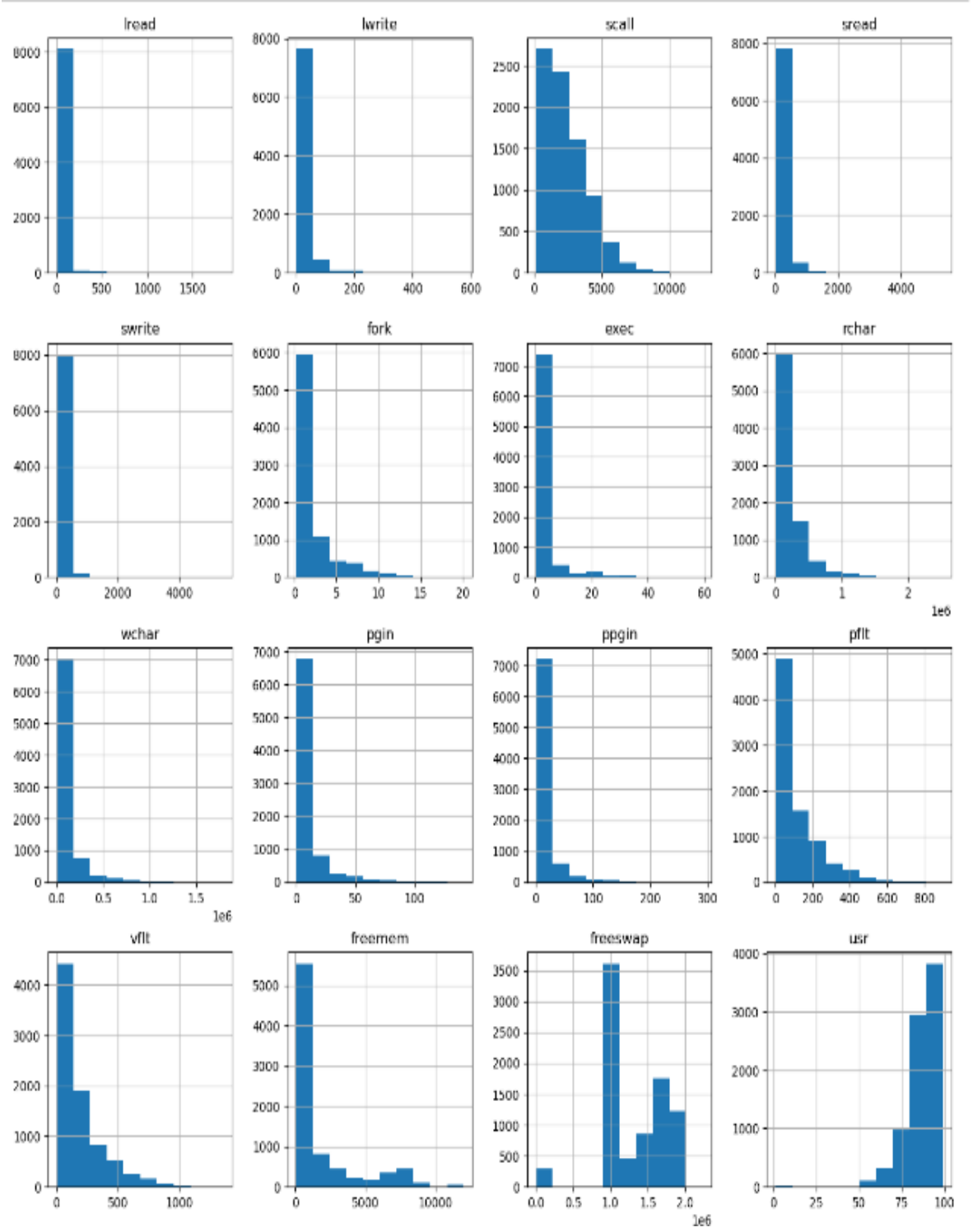
After outlier treatment :



- Outliers have been successfully treated from the dataset now.

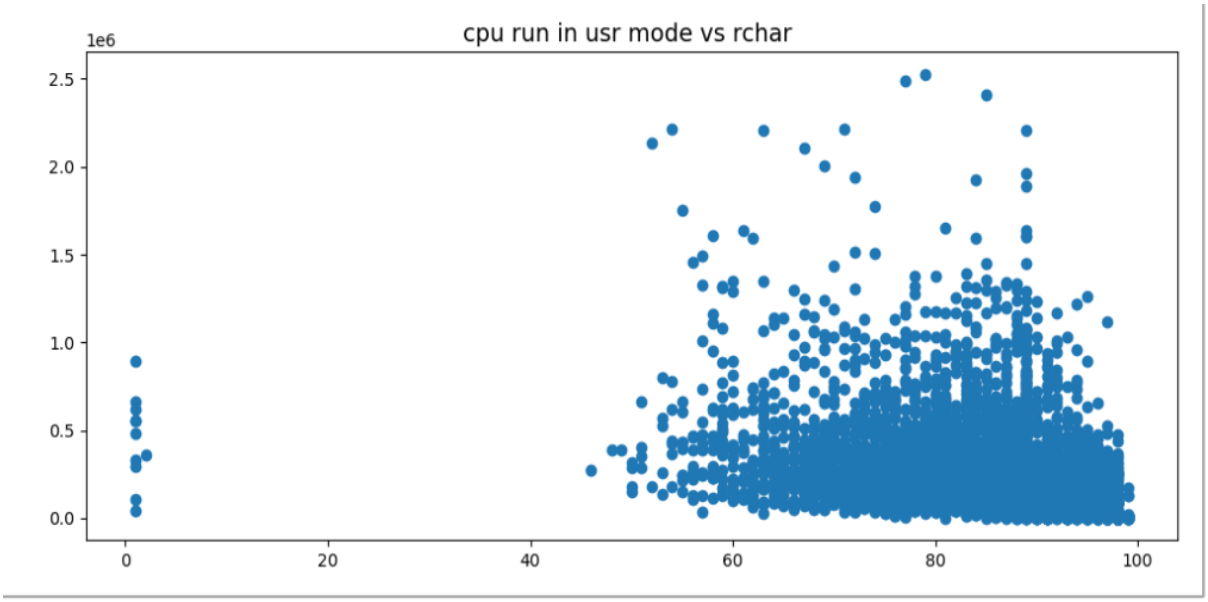
UNIVARIATE ANALYSIS:

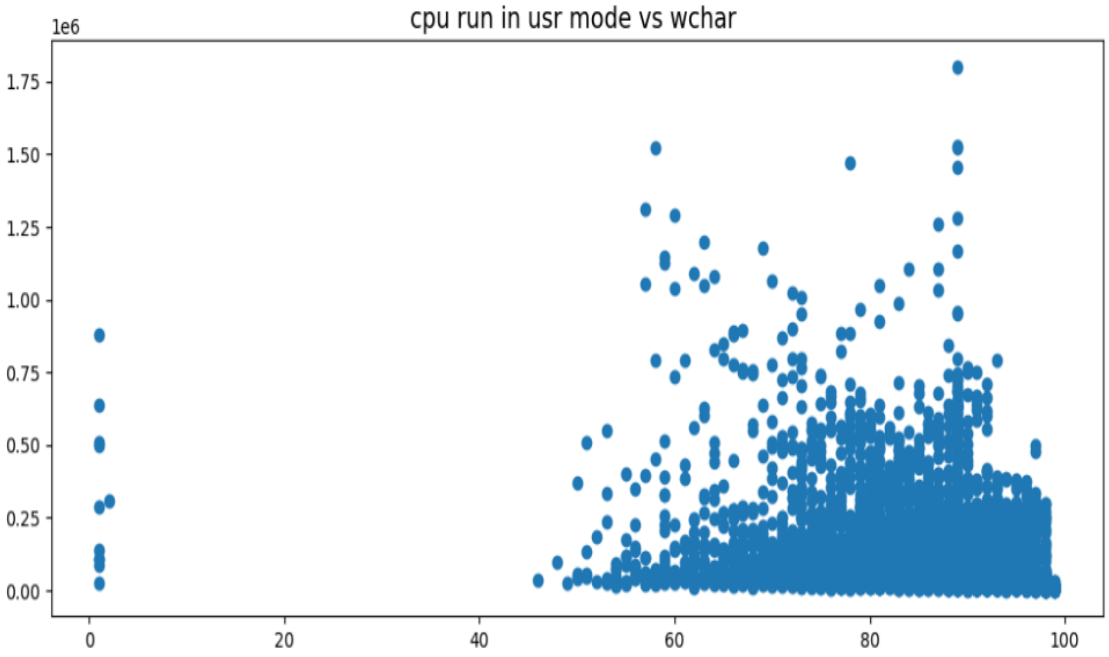
we plot the histograms for the different feature present in the dataset.



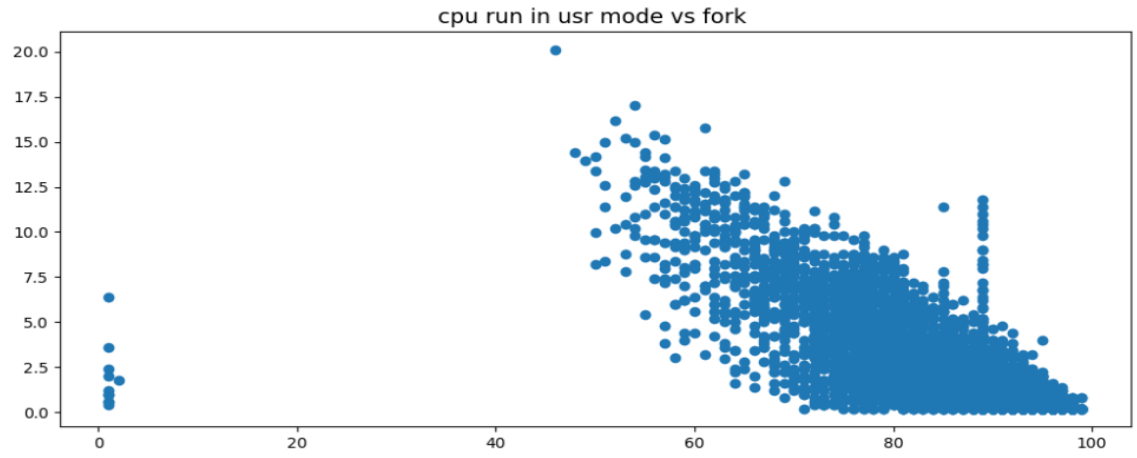
Bivariate analysis:

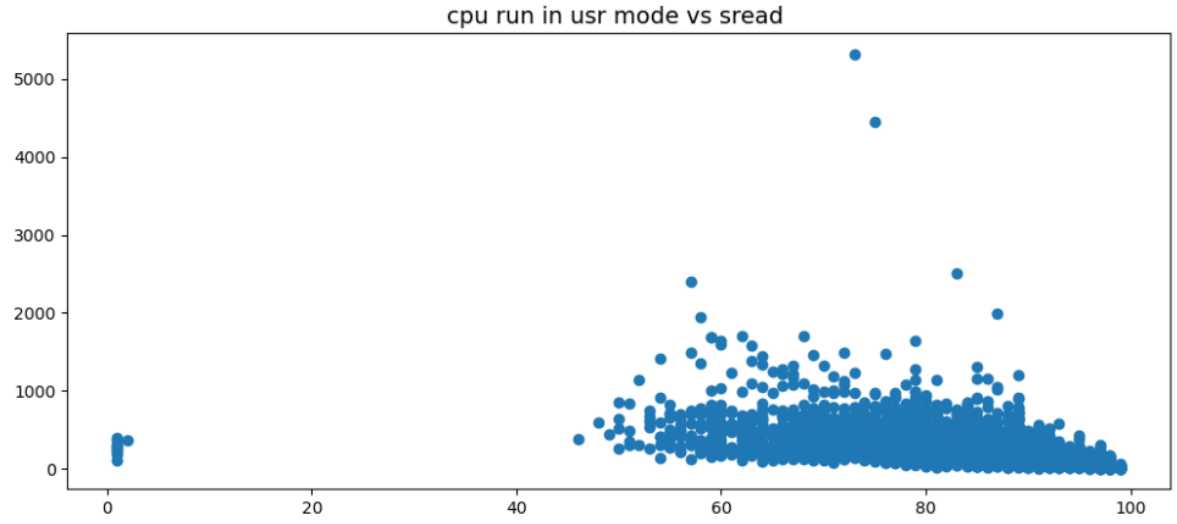
The scatteplots between the dependent variable and different independent variable is as follows :

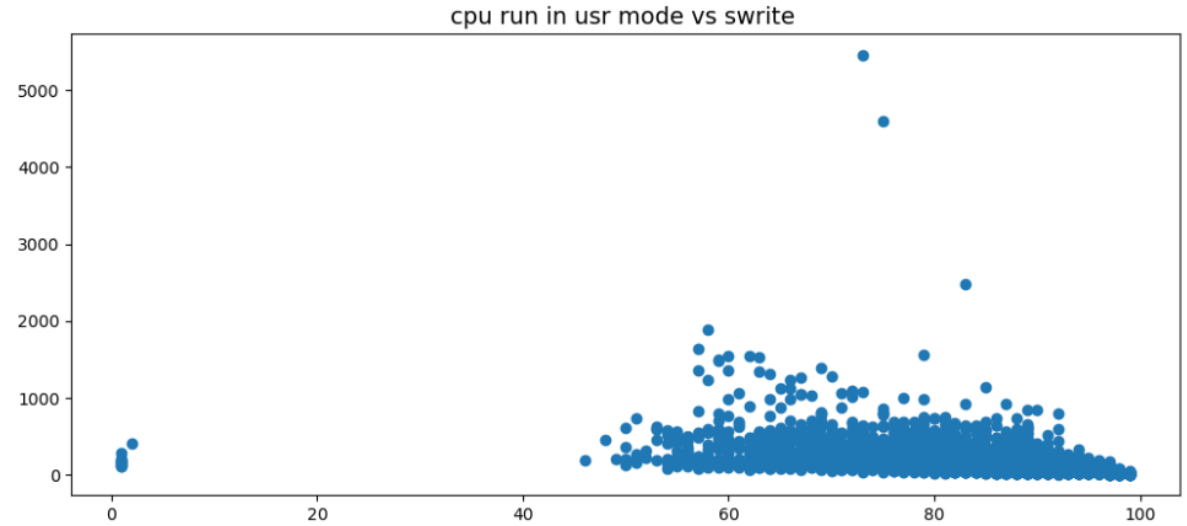




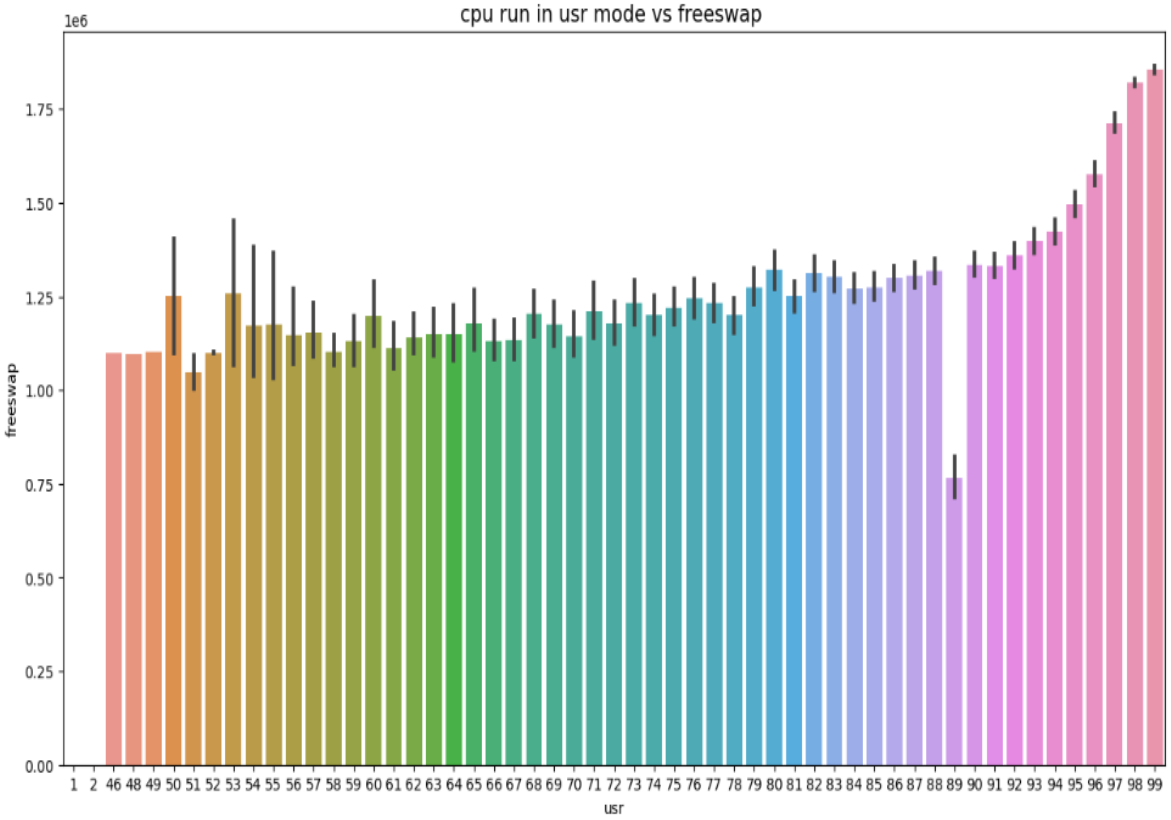
.


cpu run in usr mode vs fork
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
20
40
60
80
100

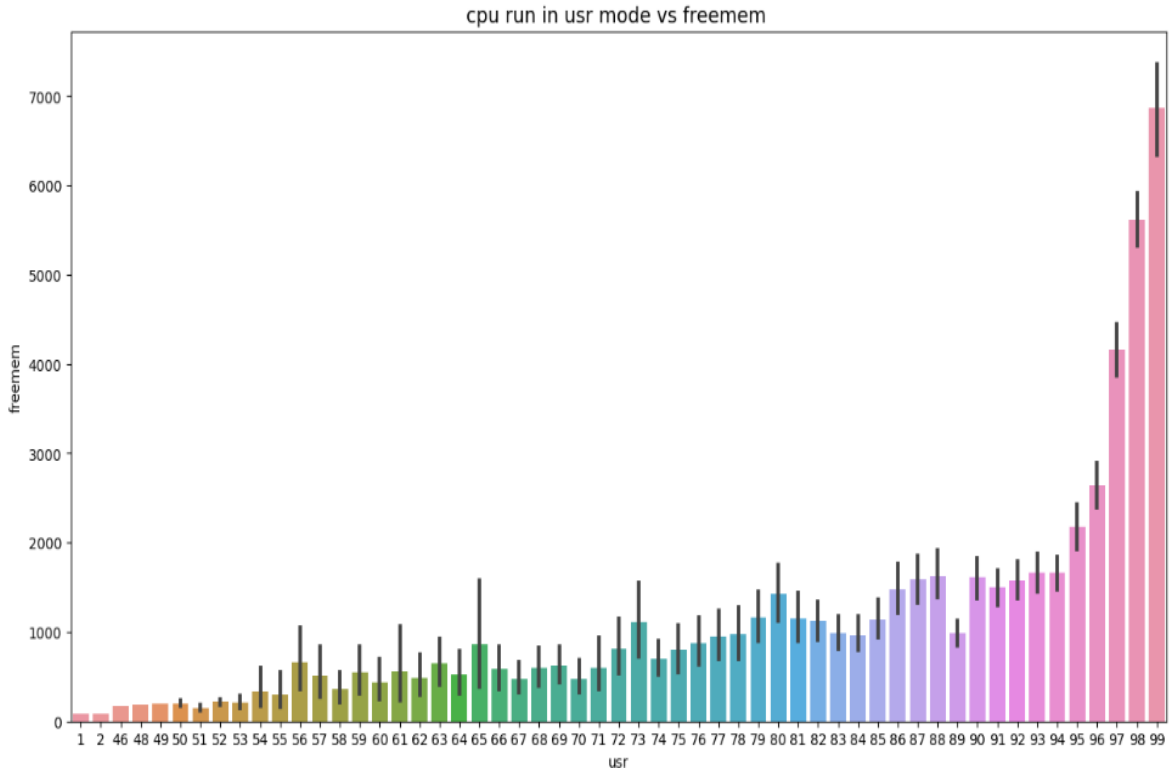

cpu run in usr mode vs sread
5000
4000
3000
2000
1000
0
0
20
40
60
80
100


cpu run in usr mode vs swrite
5000
4000
3000
2000
1000
0
0
20
40
60
80
100

A bat plot between cpu run in usr mode vs freeswap

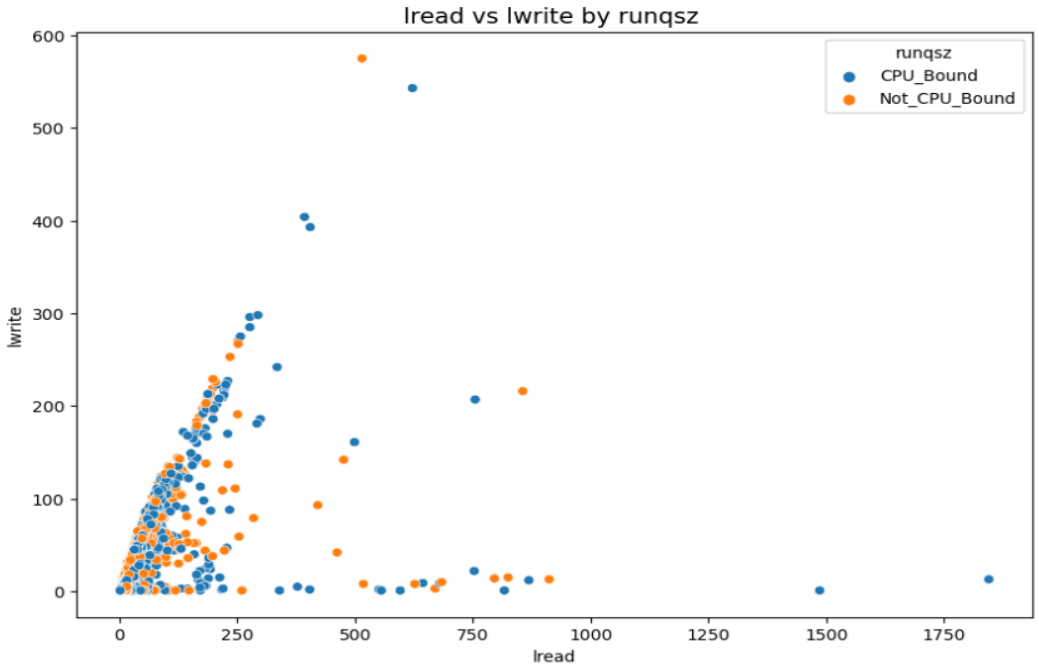


A bat plot between cpu run in usr mode vs freemem

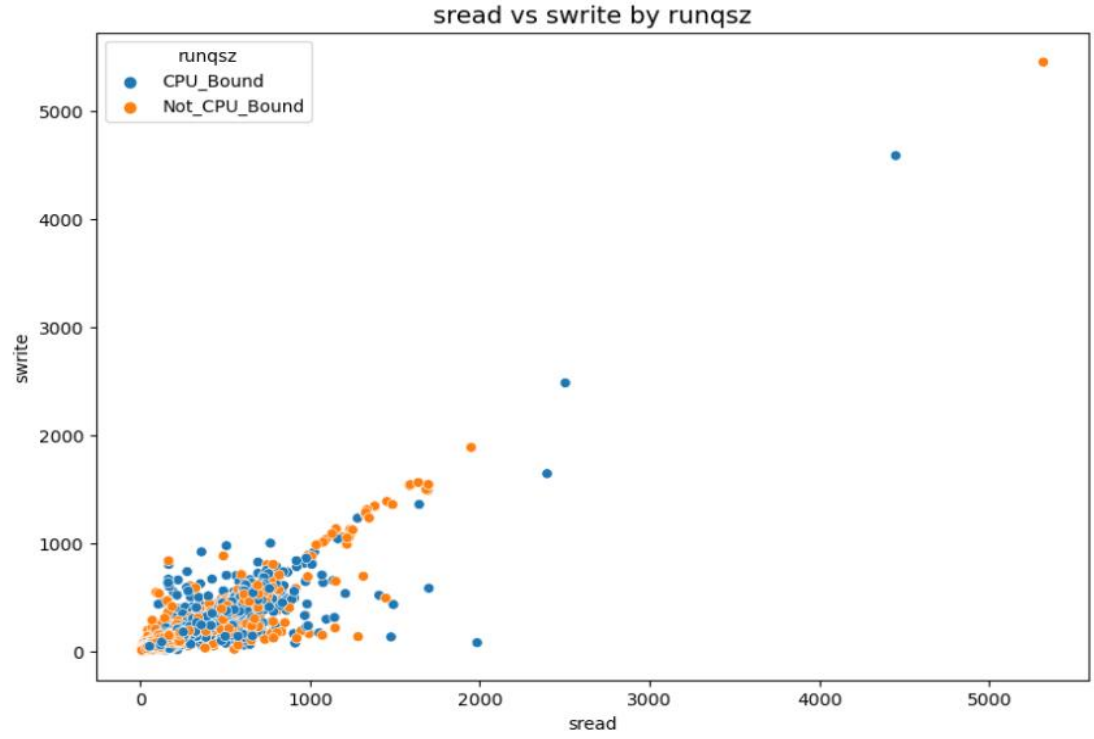


Multivariate analysis:

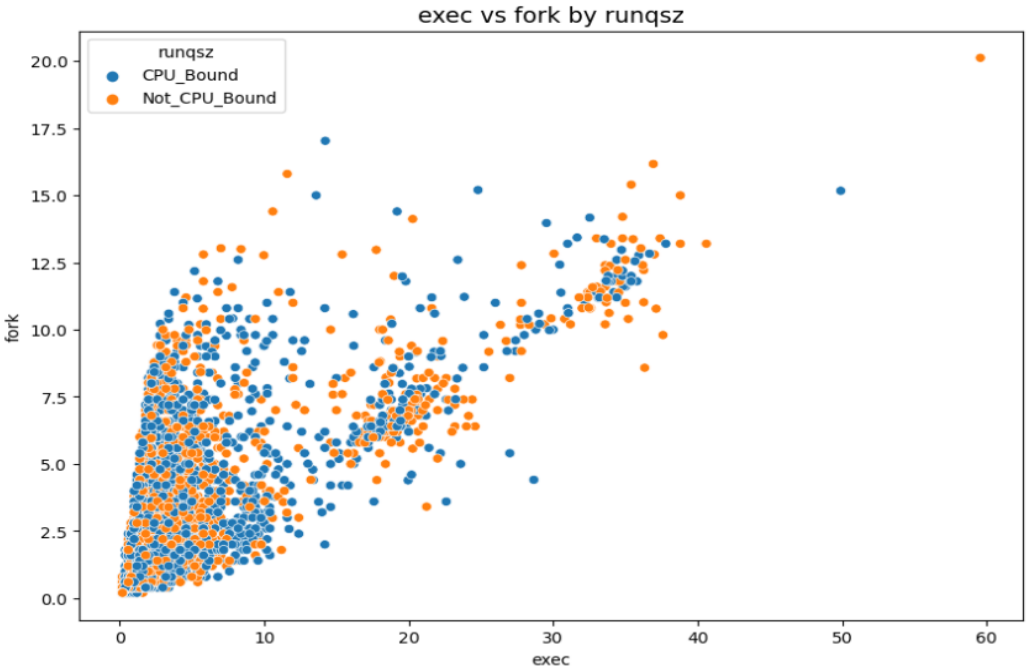
Scatterplot ‘lread’ and ‘lwrite’ seperated by ‘runqsz’



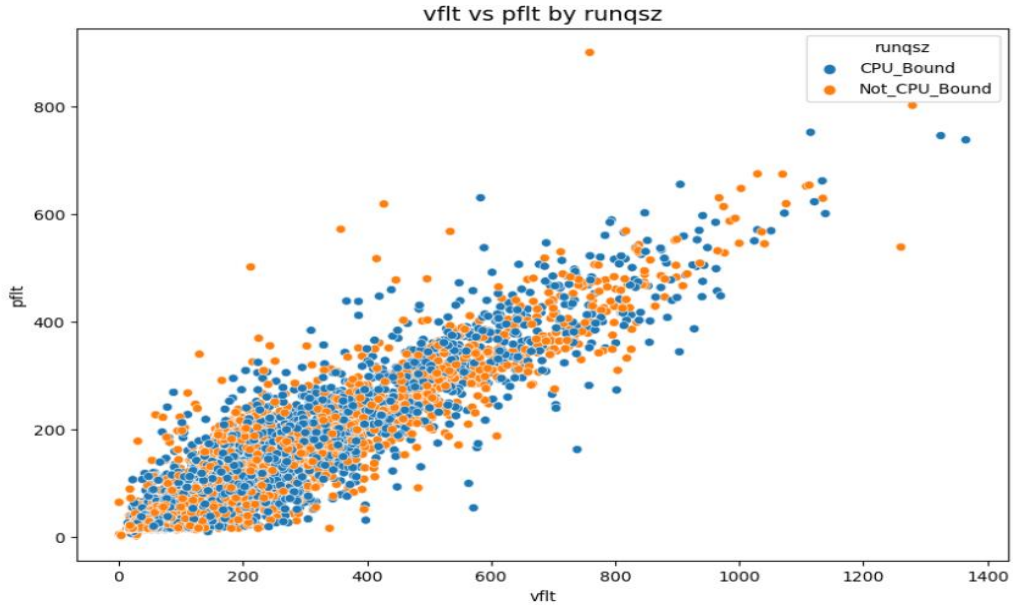
Scatterplot ‘sread’ and ‘swrite’ seperated by ‘runqsz’



Scatterplot between ‘exec’ and ‘fork’ seperated by ‘runqsz’



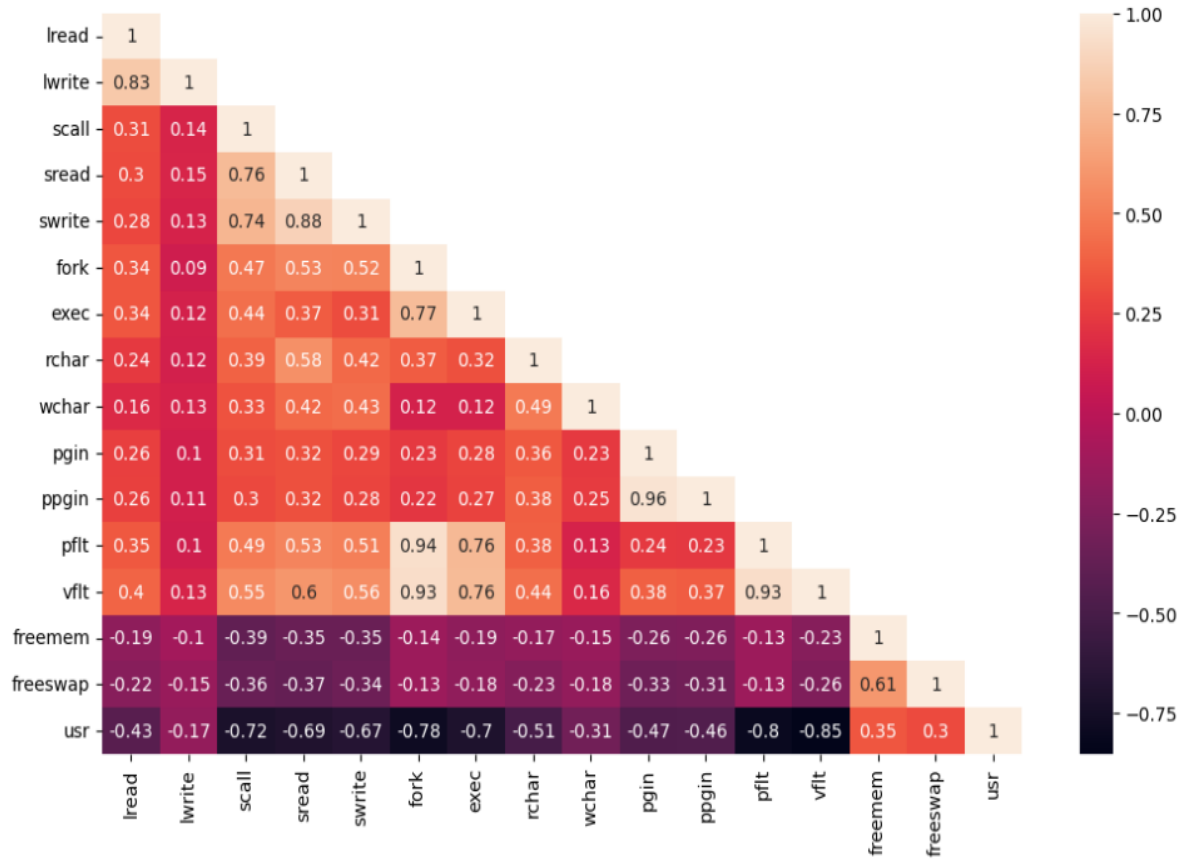
Scatterplot between ‘vflt’ and ‘pflt’ seperated by ‘runqsz’



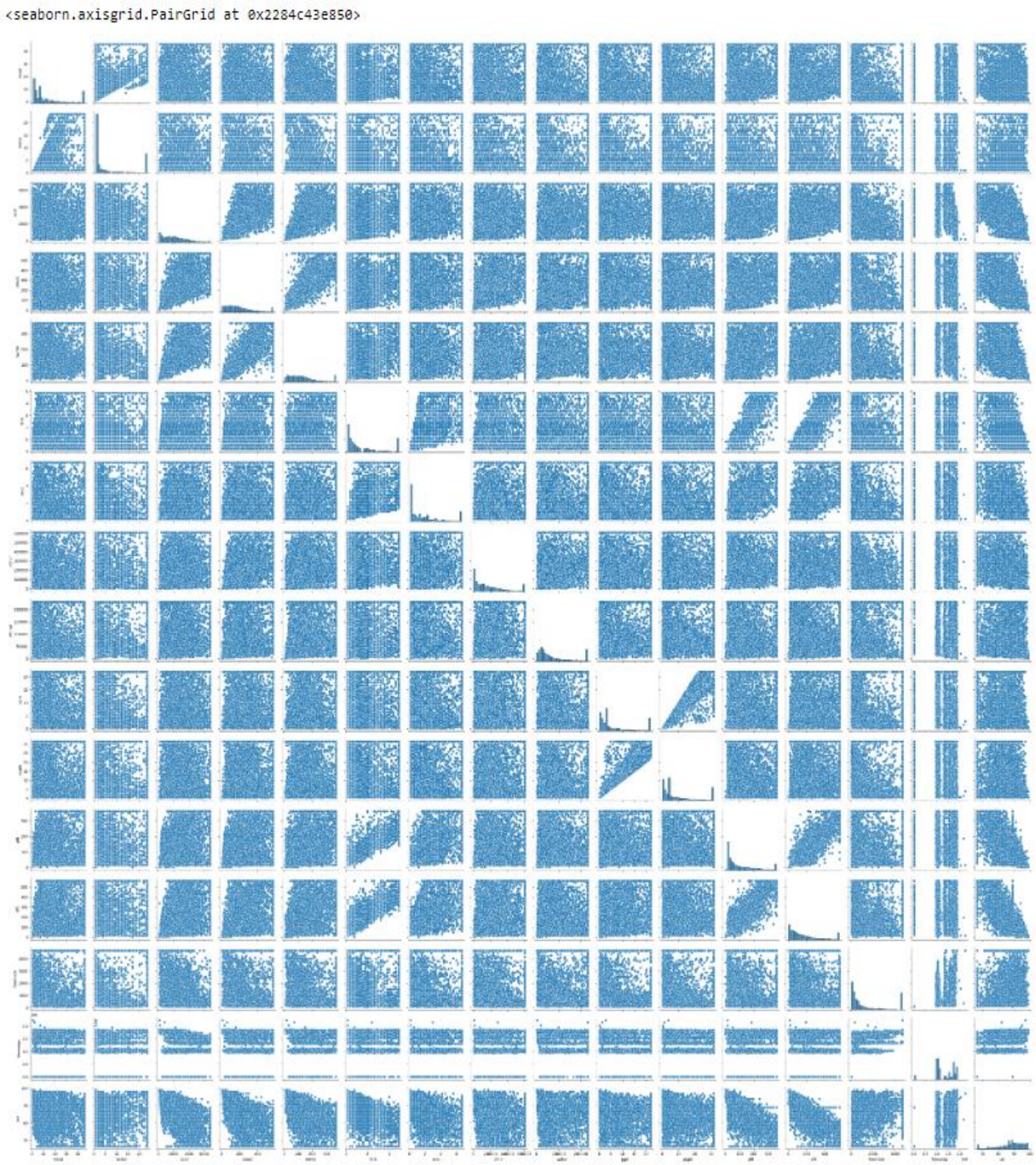
Checking for correlation:

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgin	ppgin	pflt	vflt	freemem
lread	1.000000	0.827548	0.311692	0.303606	0.284089	0.344897	0.343601	0.238675	0.161430	0.255872	0.264319	0.354210	0.399510	-0.189052
lwrite	0.827548	1.000000	0.137430	0.145805	0.129290	0.089957	0.118182	0.115714	0.129947	0.103827	0.109948	0.099747	0.132512	-0.101668
scall	0.311692	0.137430	1.000000	0.763001	0.742206	0.473931	0.439110	0.386875	0.331933	0.306777	0.298207	0.485143	0.548081	-0.388969
sread	0.303606	0.145805	0.763001	1.000000	0.876652	0.527536	0.368862	0.576127	0.415059	0.323644	0.322648	0.529167	0.597892	-0.349887
swrite	0.284089	0.129290	0.742206	0.876652	1.000000	0.518853	0.312562	0.419759	0.430126	0.285817	0.281703	0.505362	0.563163	-0.350318
fork	0.344897	0.089957	0.473931	0.527536	0.518853	1.000000	0.773772	0.370491	0.122580	0.234980	0.223515	0.938801	0.932208	-0.135541
exec	0.343601	0.118182	0.439110	0.368862	0.312562	0.773772	1.000000	0.324253	0.123445	0.280719	0.269849	0.757563	0.762709	-0.191383
rchar	0.238675	0.115714	0.386875	0.576127	0.419759	0.370491	0.324253	1.000000	0.486317	0.355068	0.377337	0.381335	0.438870	-0.165552
wchar	0.161430	0.129947	0.331933	0.415059	0.430126	0.122580	0.123445	0.486317	1.000000	0.233760	0.245300	0.125595	0.157922	-0.146838
pgin	0.255872	0.103827	0.306777	0.323644	0.285817	0.234980	0.280719	0.355068	0.233760	1.000000	0.958522	0.236922	0.380663	-0.263493
ppgin	0.264319	0.109948	0.298207	0.322648	0.281703	0.223515	0.269849	0.377337	0.245300	0.958522	1.000000	0.229187	0.368406	-0.257606
pflt	0.354210	0.099747	0.485143	0.529167	0.505362	0.938801	0.757563	0.381335	0.125595	0.236922	0.229187	1.000000	0.931791	-0.132542
vflt	0.399510	0.132512	0.548081	0.597892	0.563163	0.932208	0.762709	0.438870	0.157922	0.380663	0.368406	0.931791	1.000000	-0.228413
freemem	-0.189052	-0.101668	-0.388969	-0.349887	-0.350318	-0.135541	-0.191383	-0.165552	-0.146838	-0.263493	-0.257606	-0.132542	-0.228413	1.000000
freeswap	-0.220320	-0.147760	-0.357864	-0.369897	-0.336869	-0.132593	-0.183675	-0.230327	-0.175309	-0.326718	-0.311739	-0.132484	-0.258084	0.607000
usr	-0.427215	-0.169155	-0.719677	-0.694638	-0.666116	-0.783740	-0.701395	-0.510129	-0.314052	-0.465194	-0.459377	-0.802507	-0.852012	0.346472

Correlation between variables":



Pairplot is as follows:



- Pair plot shows the relationship between the variables in the form of scatterplot and the distribution of the variable in the form of histogram .
- As the given data set contains number of columns the pair plot is looking a little messy.
- In some plots, we will be able to see possitive correlation, some having negative correlation and some having no correlation
- Now we convert the categorical variable ‘runqsz’ into numerical by encoding using dummy variable:

The first 5 rows of the dataset now :

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgin	ppgin	pflt	vflt	freemem	freeswap	usr	runqsz_Not_CPU_Bound
0	1	1	2147	79	68	0.2	0.2	40671.0	53995.0	1.6	2.6	16.00	26.40	4670	1730946	95	0
1	7	1	170	18	21	0.2	0.2	448.0	8385.0	2.8	3.8	15.63	16.83	7278	1869002	97	1
2	15	3	2162	159	119	2.0	2.4	125473.5	31950.0	6.0	9.4	150.20	220.20	702	1021237	87	1
3	7	1	160	12	16	0.2	0.2	125473.5	8670.0	0.2	0.2	15.60	16.80	7248	1863704	98	1
4	5	1	330	39	38	0.4	0.4	125473.5	12185.0	1.0	1.2	37.80	47.60	633	1760253	90	1

- Now we make a copy of the dataset present at this moment so that it is feasible for various things.
- Now we separate the dataset given into X independent variables and Y dependent variable.

SPLITTING THE DATASET INTO TRAINING AND TESTING DATASET

- Let us create the x and y variable data with respect to ‘usr’ column as the target variable. Now x having every data except the target variable and y having only the targetvariable .
- We split the independent variables X into two parts, one for training X_train and one for testing X_test
- we split the dependent variable Y into two parts, one for training Y_train and one for the testing Y_test
- Using stats model api as SM to intercept the X variable.
- Using sklearn to split the data into x_train and y_train

X data frame :

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgin	ppgin	pflt	vflt	freemem	freeswap	runqsz_Not_CPU_Bound
0	1	1	2147	79	68	0.2	0.2	40671.0	53995.0	1.6	2.6	16.00	26.40	4670	1730946	0
1	7	1	170	18	21	0.2	0.2	448.0	8385.0	2.8	3.8	15.63	16.83	7278	1869002	1
2	15	3	2162	159	119	2.0	2.4	125473.5	31950.0	6.0	9.4	150.20	220.20	702	1021237	1
3	7	1	160	12	16	0.2	0.2	125473.5	8670.0	0.2	0.2	15.60	16.80	7248	1863704	1
4	5	1	330	39	38	0.4	0.4	125473.5	12185.0	1.0	1.2	37.80	47.60	633	1760253	1

Y data frame :

	usr
0	95
1	97
2	87
3	98
4	90

The coefficients are:

```
The coefficient for lread is -0.012146380034548925
The coefficient for lwrite is -0.0016011667237528858
The coefficient for scall is -0.001240024487339167
The coefficient for sread is -0.00010576209172118666
The coefficient for swrite is -0.006442206641492385
The coefficient for fork is 0.3304251318479866
The coefficient for exec is -0.38950042722690004
The coefficient for rchar is -1.713818012907695e-06
The coefficient for wchar is -3.8637149520586104e-06
The coefficient for pgin is -0.04197267747231756
The coefficient for ppgin is -0.027936580799646765
The coefficient for pflt is -0.016836475444208562
The coefficient for vflt is -0.015064857428819092
The coefficient for freemem is 0.00017662414361003866
The coefficient for freeswap is -8.642658039727524e-08
The coefficient for runqsz_Not_CPU_Bound is -0.04136754471609674
```

- The intercept of the model is

```
The intercept for our model is 97.50199997677092
```

- R square on training data:

```
0.7664568891348148
```

- R square on testing data:

```
0.8616707861990465
```

- RMSE on training data:

```
4.57438741900247
```

- RMSE on testing data:

```
3.3068938651440583
```

LINEAR REGRESSION USING STATS MODELS :

As the Train and the test data split up we can process with creating the linearmodel. Now for creating the OLS model, we can use the .ols from stats model api package.And Fit the data with x_train and y_train.

The model summary is :

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    usr   R-squared:                       0.766
Model:                            OLS   Adj. R-squared:                  0.766
Method:                 Least Squares   F-statistic:                     1173.
Date:                Sat, 30 Dec 2023   Prob (F-statistic):               0.00
Time:                        08:04:25   Log-Likelihood:                -16855.
No. Observations:                5734   AIC:                         3.374e+04
Df Residuals:                    5717   BIC:                         3.386e+04
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const                     97.5020      0.299    325.667      0.000      96.915      98.089
lread                     -0.0121      0.001     -9.415      0.000      -0.015      -0.010
lwrite                    -0.0016      0.002     -0.661      0.509      -0.006       0.003
scall                     -0.0012   6.02e-05    -20.603      0.000      -0.001      -0.001
sread                     -0.0001      0.001     -0.135      0.893      -0.002       0.001
swrite                    -0.0064      0.001     -6.520      0.000      -0.008      -0.005
fork                       0.3304      0.104      3.165      0.002       0.126       0.535
exec                      -0.3895      0.022    -18.111      0.000      -0.432      -0.347
rchar                -1.714e-06   3.67e-07     -4.675      0.000   -2.43e-06   -9.95e-07
wchar                -3.864e-06    5.7e-07     -6.778      0.000   -4.98e-06   -2.75e-06
pgin                      -0.0420      0.012     -3.554      0.000      -0.065      -0.019
ppgin                     -0.0279      0.007     -3.891      0.000      -0.042      -0.014
pflt                      -0.0168      0.002     -9.624      0.000      -0.020      -0.013
vflt                      -0.0151      0.001    -11.346      0.000      -0.018      -0.012
freemem                    0.0002   3.16e-05      5.594      0.000       0.000       0.000
freeswap             -8.643e-08   1.85e-07     -0.468      0.640   -4.49e-07    2.76e-07
runqsz_Not_CPU_Bound      -0.0414      0.128     -0.323      0.746      -0.292       0.209
==============================================================================
Omnibus:                     9069.058   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          8083159.894
Skew:                          -9.974   Prob(JB):                         0.00
Kurtosis:                     185.851   Cond. No.                     7.06e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.06e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

- The R-square value tells that the model can explain 76.6 % of the variance in the training set

- Adjusted R-square also nearly to the R-square,76.6%.

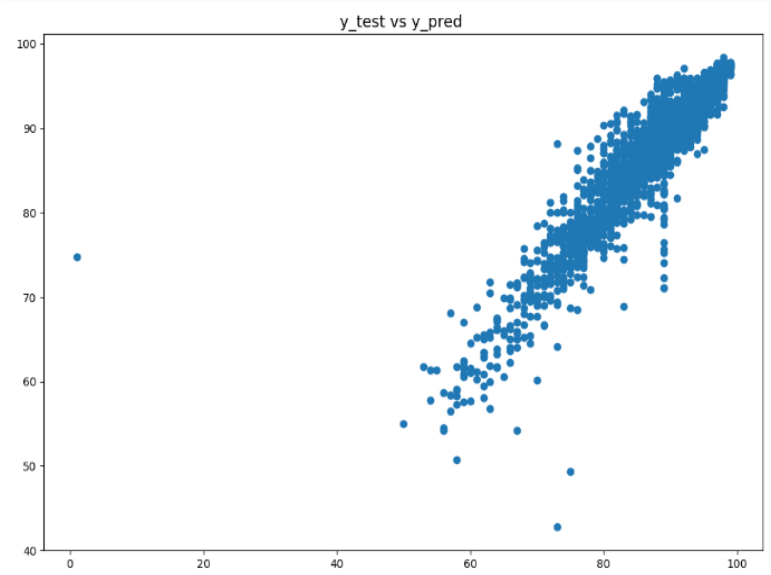
RMSE on train data:

```
4.57438741900247
```

RMSE on test data:

```
3.3068938651439566
```

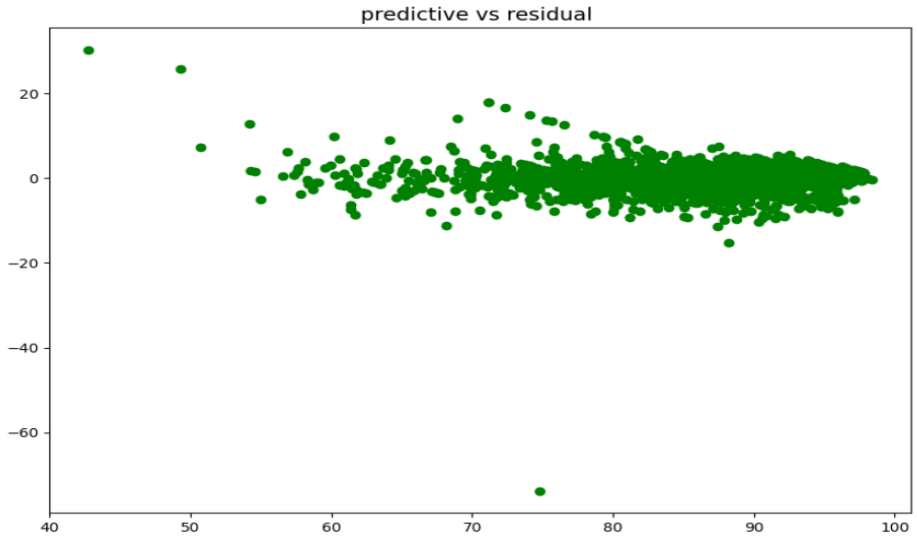
Scatterplot between the actual y value and predicted y value:



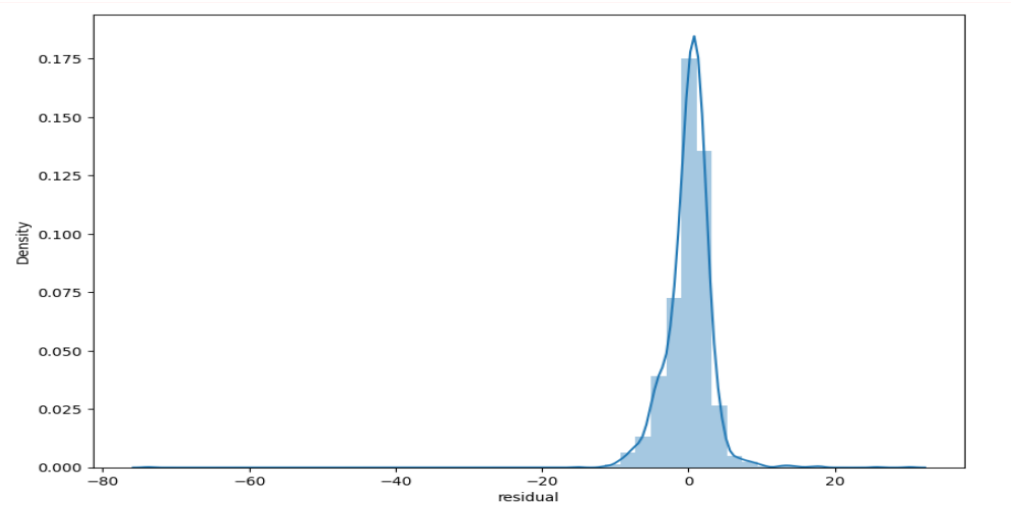
The following table shows the comparision between the actual and predicted values and their difference i.e residual

	usr	predictive	residual
5455	96	94.048316	1.951684
152	76	76.457852	-0.457852
6394	95	93.606355	1.393645
7509	83	86.702820	-3.702820
6343	92	93.263191	-1.263191

Graph between predicted and residual values



The residual density graph:



The final linear model equation of the data is :

```
(97.502) * const + (-0.0121) * lread + (-0.0016) * lwrite + (-0.0012) * scall + (-0.0001) * sread + (-0.0064) * swrite + (0.330
4) * fork + (-0.3895) * exec + (-0.0) * rchar + (-0.0) * wchar + (-0.042) * pgin + (-0.0279) * ppgin + (-0.0168) * pflt + (-0.0
151) * vflt + (0.0002) * freemem + (-0.0) * freeswap + (-0.0414) * runqsz_Not_CPU_Bound +
```

From the above linear equation it can be predcited that ,

- There are many negative coefficients present in the linear equation.
- Except ‘fork’, ‘freemem’ all coefficients are decrease when implies.
- When ‘fork - Number of system fork calls per second’ is increased by a unit then the ‘usr’ value increases by 33 % and also ‘Number of system exec calls per second’ is increased by a unit then the ‘usr’ gets decresed by 38.9 %

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

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

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

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.

Top five rows of the dataset:

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

Last five rows of the dataset:

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_expos
1468	33.0	Tertiary	Tertiary	NaN	Scientology	Yes	2	Very High	Expos
1469	33.0	Tertiary	Tertiary	NaN	Scientology	No	1	Very High	Expos
1470	39.0	Secondary	Secondary	NaN	Scientology	Yes	1	Very High	Expos
1471	33.0	Secondary	Secondary	NaN	Scientology	Yes	2	Low	Expos
1472	17.0	Secondary	Secondary	1.0	Scientology	No	2	Very High	Expos

Shape of the data:

```
no.of rows:  1473 no.of columns: 10
```

Dataset 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

Basic info about the dataset:

```
<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
```

- There are 2 features with float datatype , 1 feature with integer datatype , 7 features with object datatype.

Null value check:

```
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
```

- There are null values present in the ‘Wife_age’ feature and ‘No_of_children_born’ feature.
- Lets treat the null values by imputing null calue with median value of that particular feature.

```
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
```

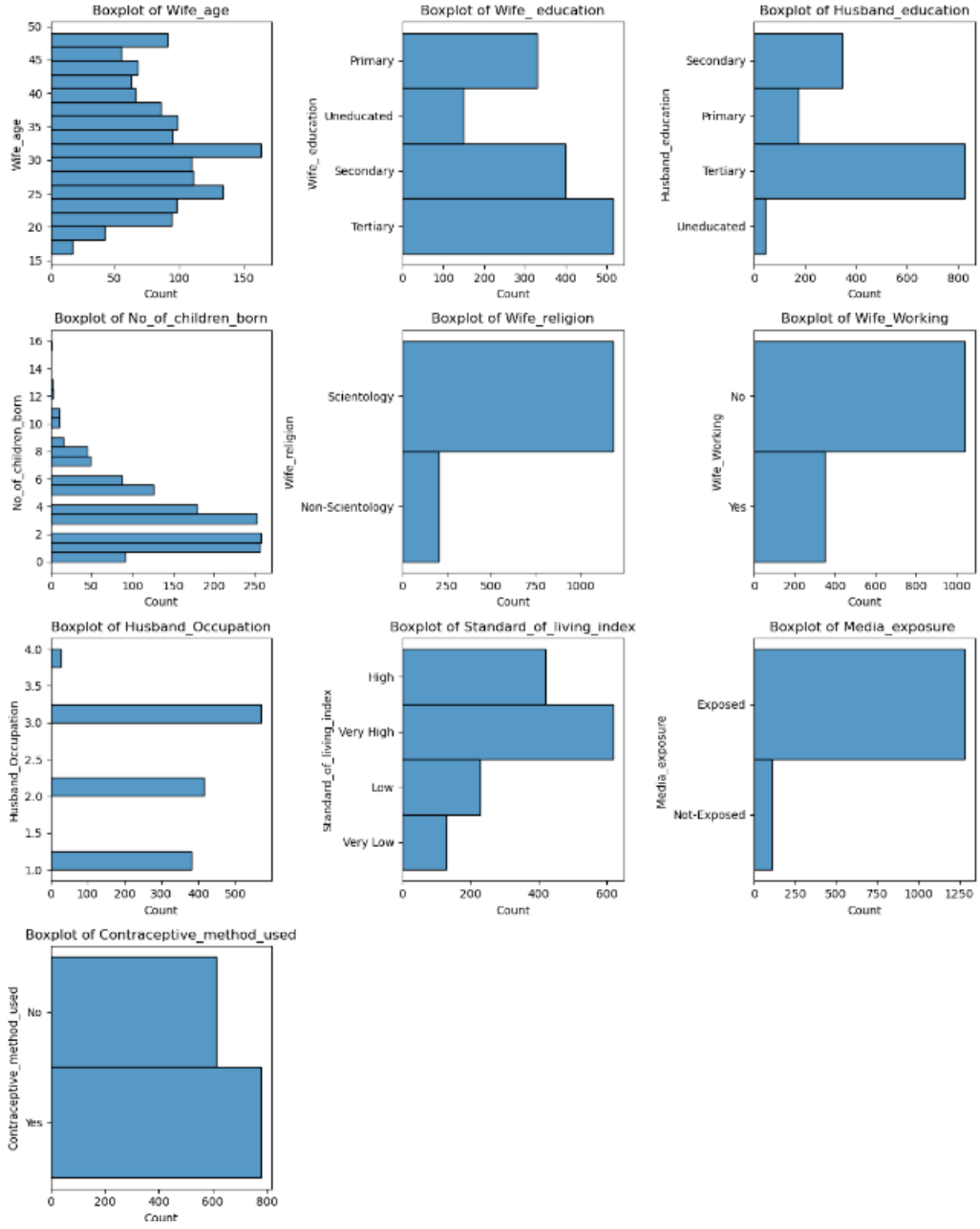
- There are 80 duplicate rows present in the dataset.

```
Number of duplicate rows = 80
```

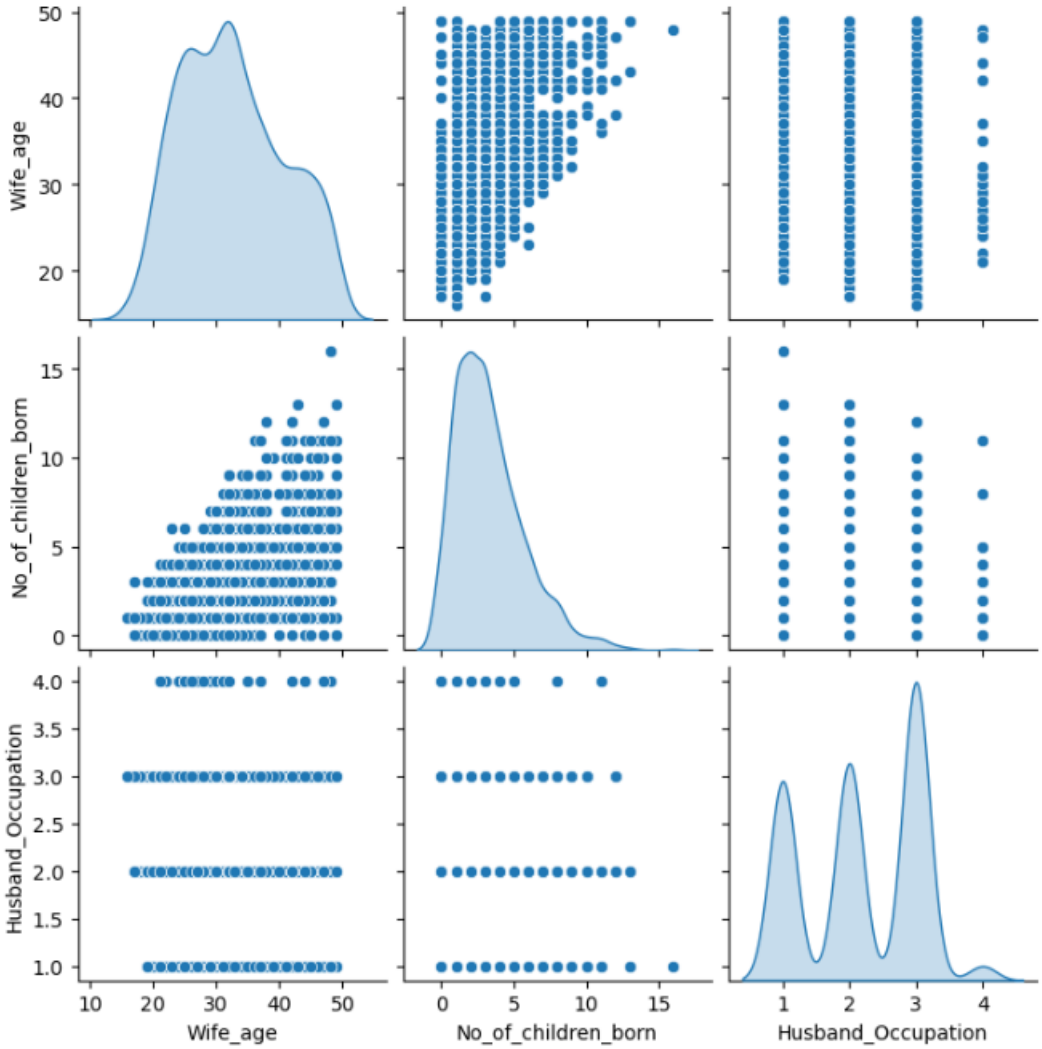
- Lets drop all the duplicate rows present in the dataset.
- Now the shape of the data is

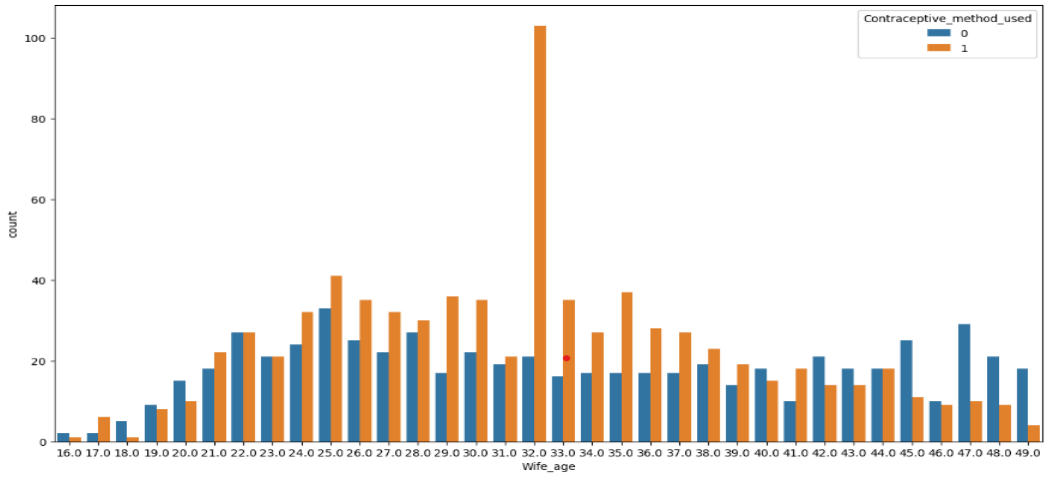
```
no.of rows:  1393 no.of columns: 10
```

UNIVARIATE ANALYSIS:



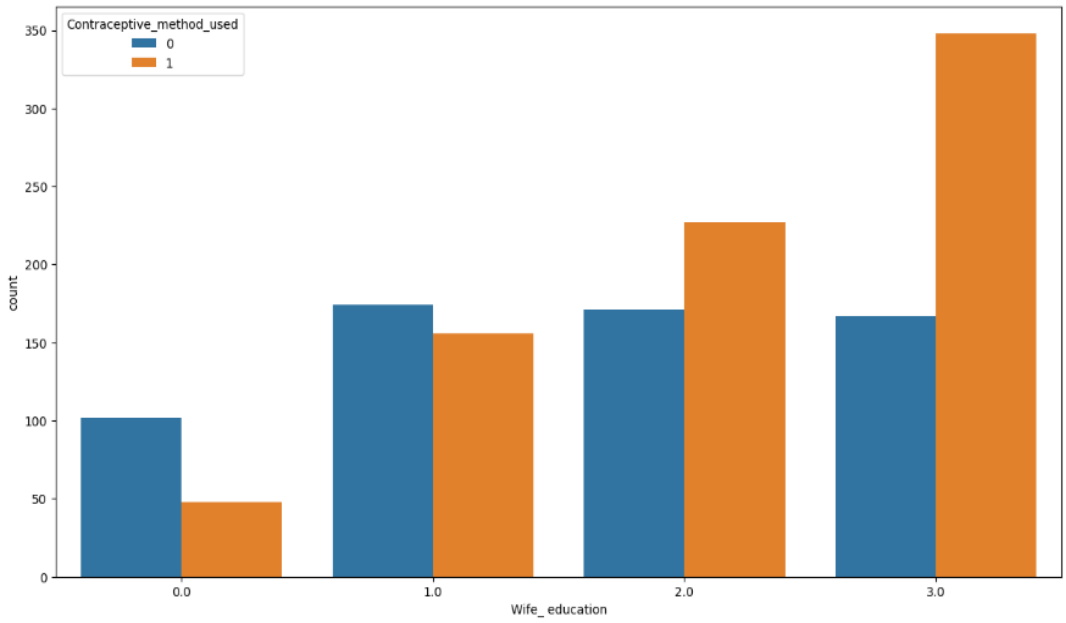
BIVARIATE ANALYSIS:



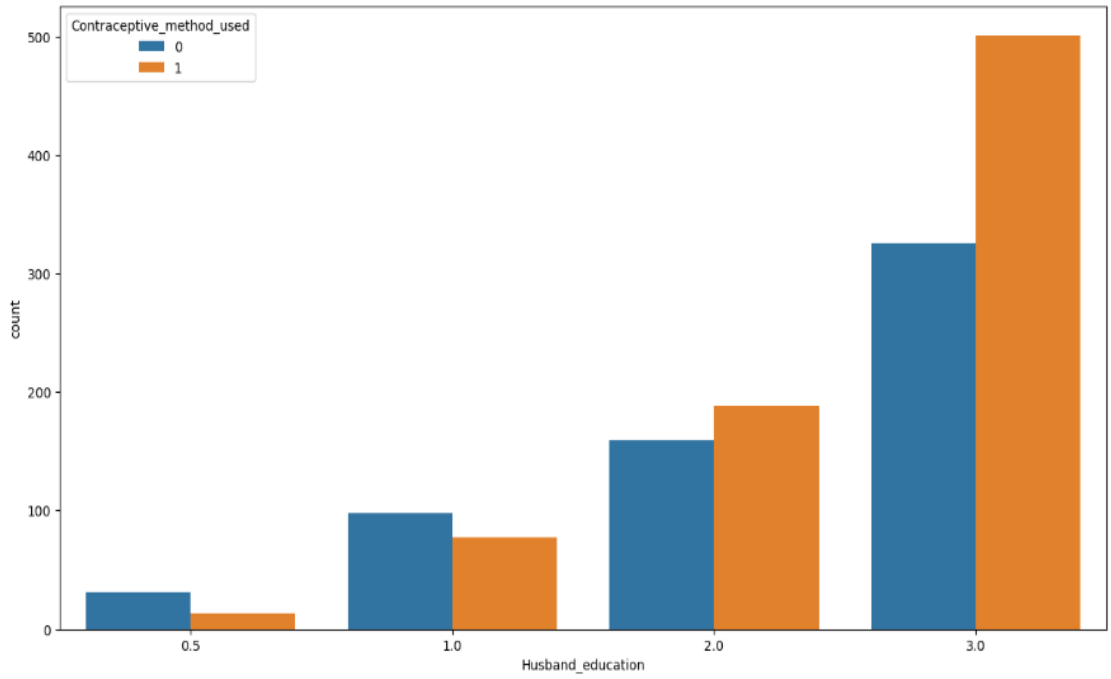


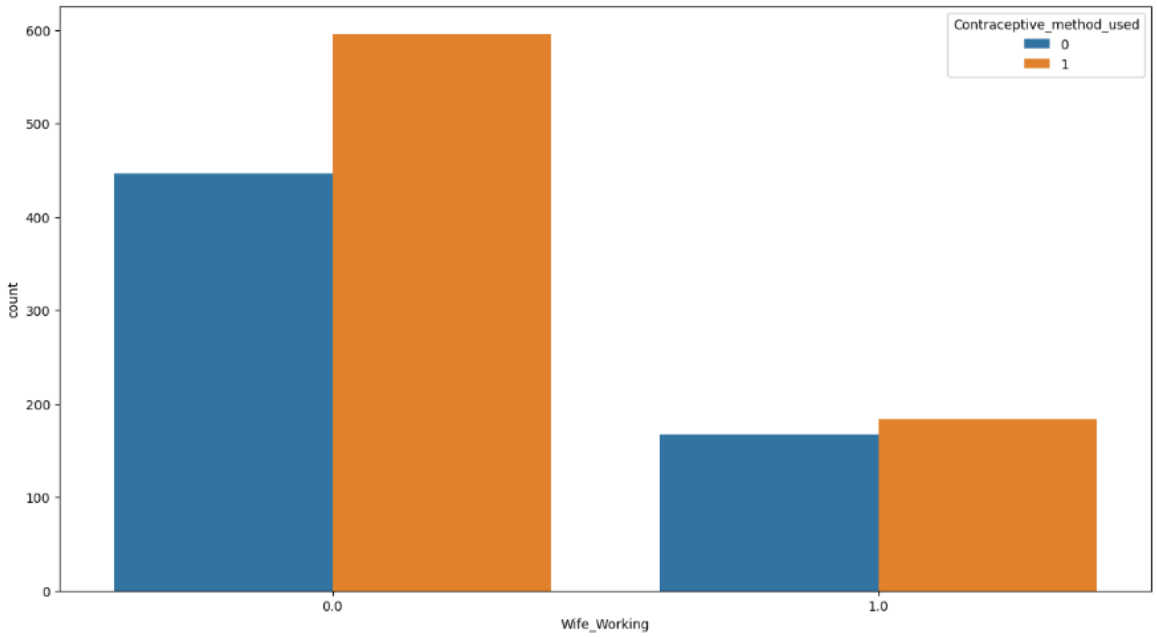
The above plot shows the relation between different age group of women and contraceptive method used.

From the below plot we may note that, tertiary educated women use the most contraceptive methods

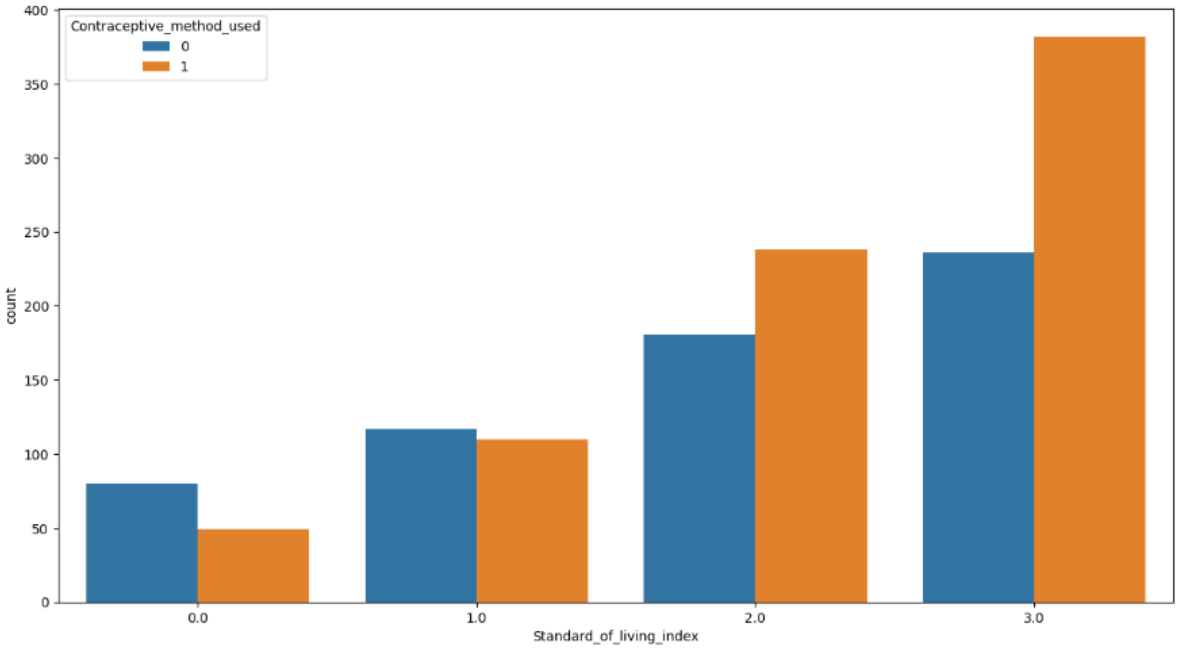


The below plot shows that wives with highest husband's education have used the most contraceptive methods.



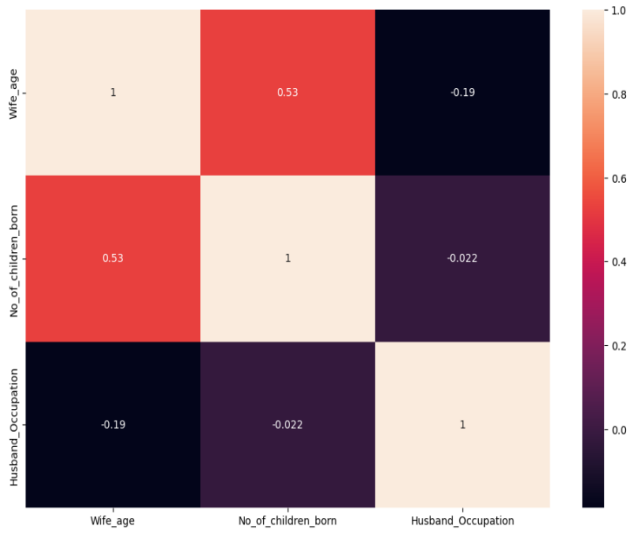


The above plot shows that the non working women use the most contraceptive methods.



From the above plot, Women with very high standard living index use the most contraceptive measures.

Correlation between variables:



We have noticed that there are only three features in integer form which can be plotted,

Now we convert all the objects to categorical codes

Encoded the categorical variables Wife_ education, Husband_education, Wife_religion,Standard_of_living_index, Media_exposure and Contraceptive_method_used in the ascending orderfrom worst to best since LDA does not take string variables as parameters into model building.

Below isthe 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 and non-Scientology = 2.

Wife_Working: Yes = 1 and No = 2.

Standard_of_living _index: Very Low = 1, Low = 2, High = 3, Very High = 4.

Media_exposure: Exposed = 1 and Not-Exposed = 2.

Contraceptive_method_used: Yes = 1 and No = 0

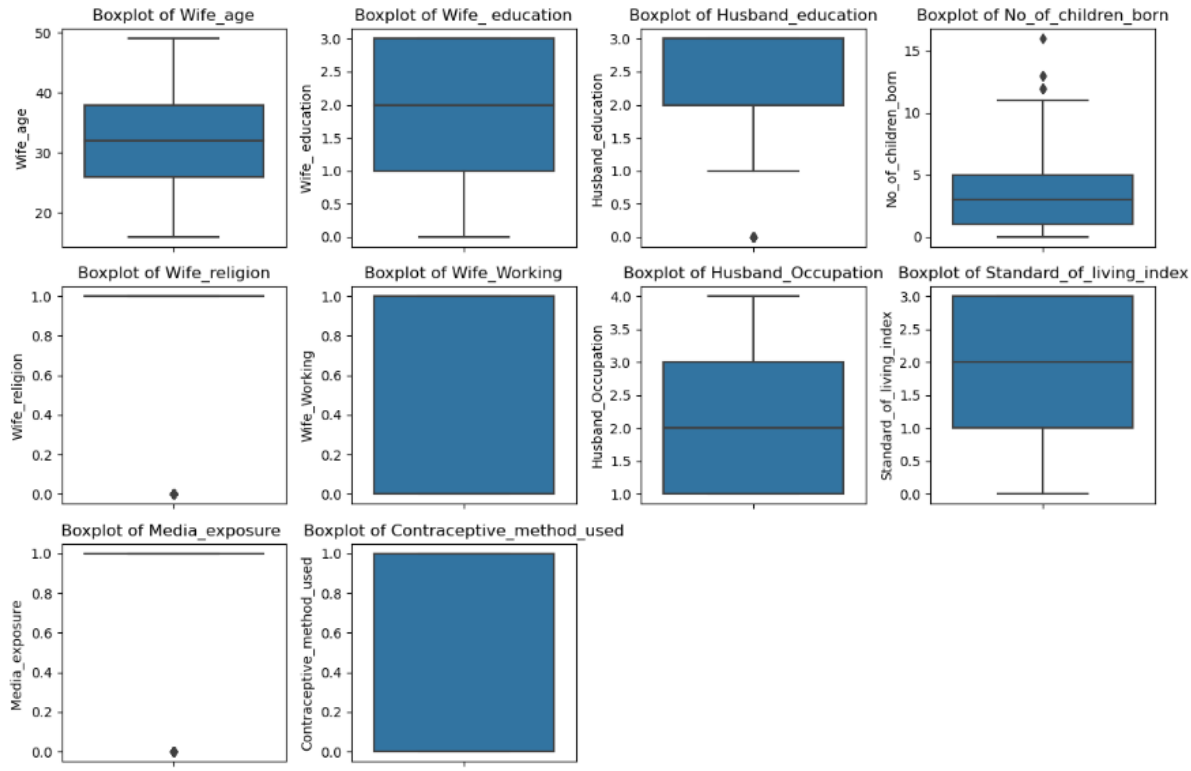
The first rows of the dataset:

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

Info :

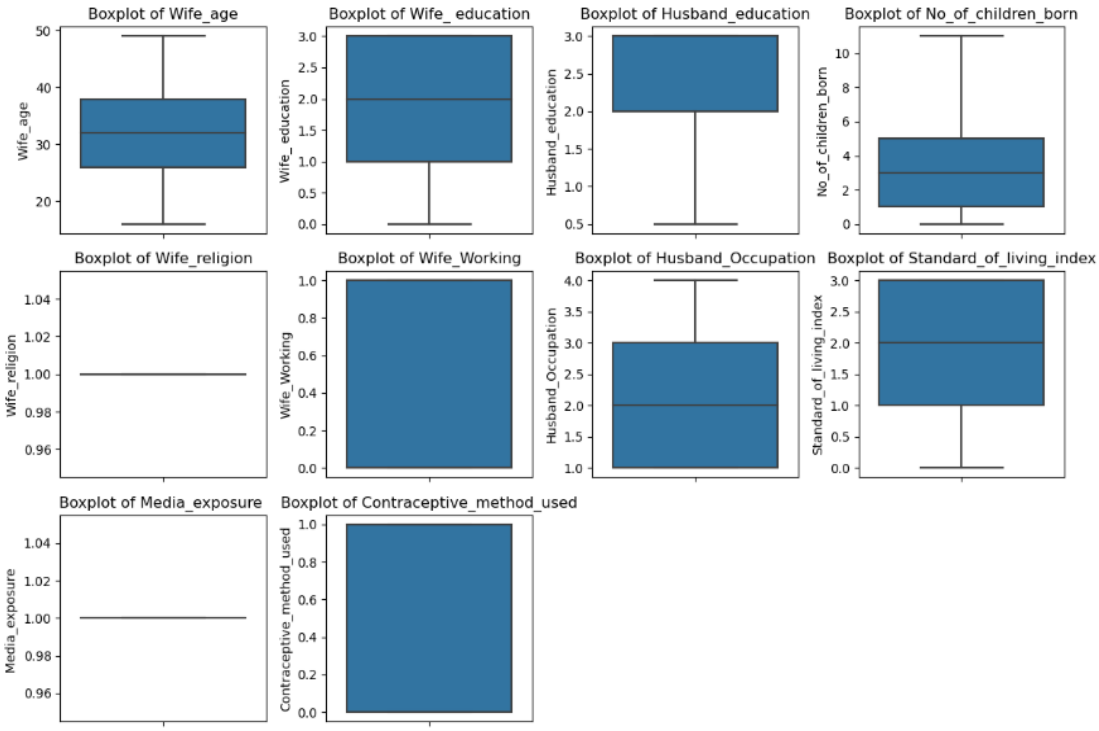
```
<class 'pandas.core.frame.DataFrame'>
Index: 1393 entries, 0 to 1472
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Wife_age                  1393 non-null   float64
 1   Wife_ education           1393 non-null   object
 2   Husband_education         1393 non-null   object
 3   No_of_children_born       1393 non-null   float64
 4   Wife_religion             1393 non-null   object
 5   Wife_Working              1393 non-null   object
 6   Husband_Occupation        1393 non-null   int64
 7   Standard_of_living_index  1393 non-null   object
 8   Media_exposure            1393 non-null   object
 9   Contraceptive_method_used 1393 non-null   object
dtypes: float64(2), int64(1), object(7)
memory usage: 119.7+ KB
```

Outliers are as follows:

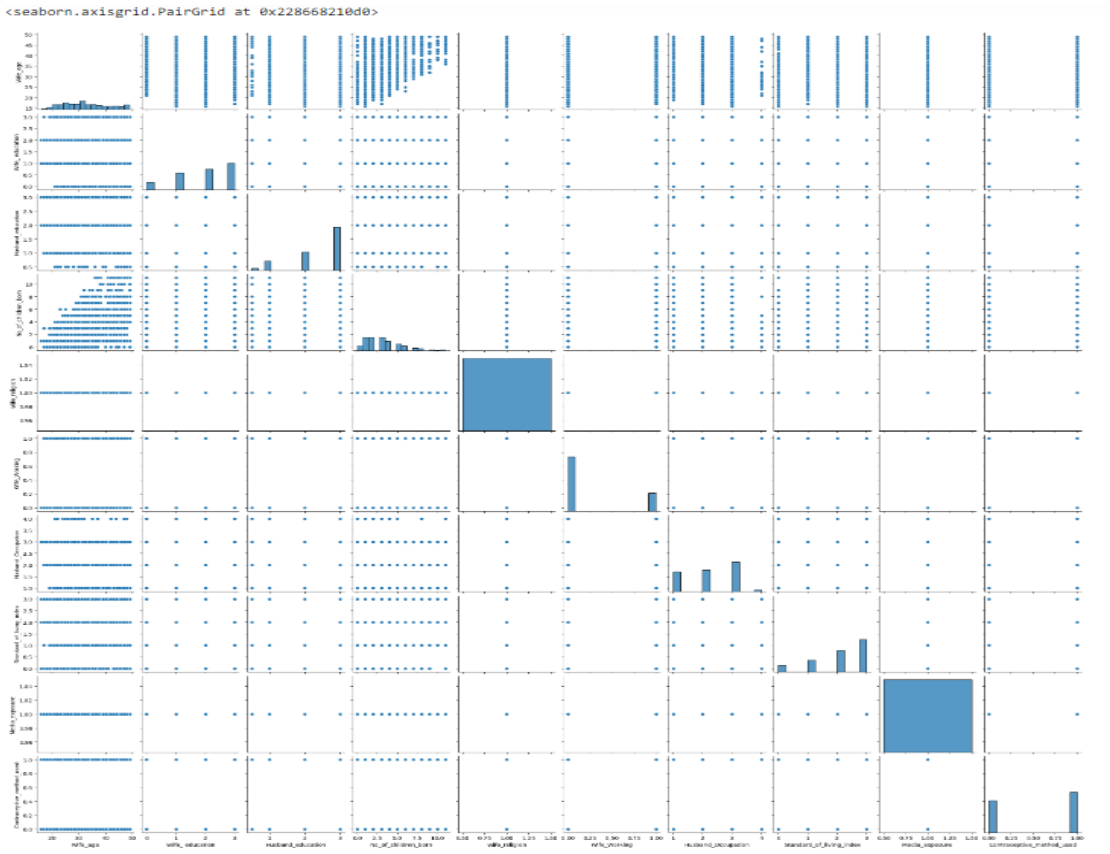


There are outliers present in the data

We treat them using IQR method



Pairplot:



Correlation between variables :

LOGISTIC REGRESSION

Train and Test Split:Let us create the x and y variable data with respect to ‘‘Contraceptive_method_used’’column as the target variable. Now x having every data except the target variable and yhaving only the target variable .

- Before we proceed the process , we need to import the required libraries or checking it. In this encoding for ‘‘Contraceptive_method_used’’ 1 as yes and 0 as No.
- We use Label Encoder form sklearn library to encode the data if we havent encoded the data previously.
- The encoding is for creating the dummy variables .
- Now the Train set and the test set has been spitted up by using the sklearnmodel.Using logistic regression model method to fit the data and creating a logistic model.
- The proportion of 1s and 0s i.e.(Customers using Contraceptive_method_used Yes /No) as follows,

```
Contraceptive_method_used
1.0    0.558974
0.0    0.441026
Name: proportion, dtype: float64
```

- Now we need to fit the Logistic regression model by using newton cg as solver, 1000 as maximum iteration, then we get the predicted data frame model as,

	0	1
0	0.269587	0.730413
1	0.626732	0.373268
2	0.331280	0.668720
3	0.366515	0.633485
4	0.305626	0.694374

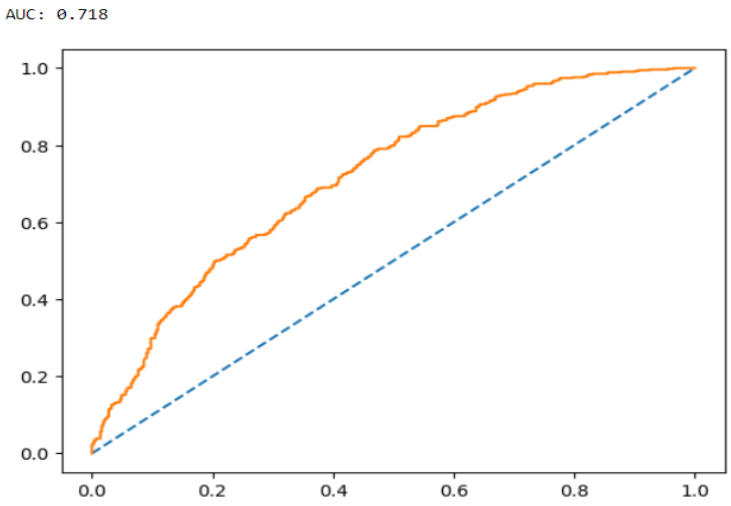
In the above data frame, we will be able to see that 1 is having the highest accuracy 69.43 %

The model accuracy is 67.3 %

AUC and ROC curve

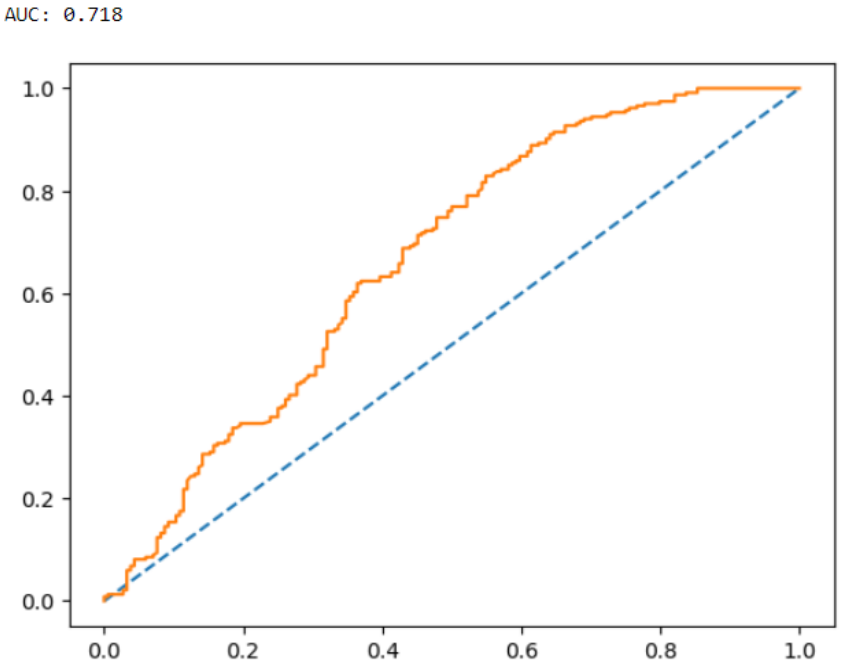
Now we can plot plot the AUC and ROC curve of the model and get the separatecurve and auc score of Train dataset and test dataset.

AUC curve for the train data:



In this curve, if the plot occurs below the do□ed lines, then it accept as worst modelever, Eventhough the curve is not perfect but the curve is OK ,the AUC(Area underthe curve) of the train data model is 71.8%.

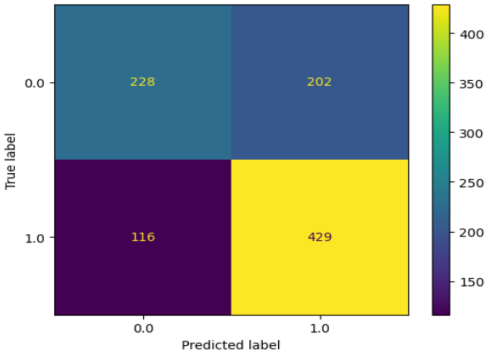
AUC curve for the test data:



- This curve is similar to the train data curve AUC but slightly vary in the initial locations.
- the curve is ok as plotted above the dotted lines.
- The Area under curve is same as the train data as 71.8%.
- For the comparision the train data AUC with the test data AUC, mostly both curve issimilar with some variation only as the AUC of both is same as 71.8.10%. Lets move to theconfusion matrix,

Confusion matrix for train data:

```
array([[228, 202],
       [116, 429]], dtype=int64)
```



This plot shows the realtionship between the true label and predicted label as 0's and 1's

Classification report is as follows

	precision	recall	f1-score	support
0.0	0.66	0.53	0.59	430
1.0	0.68	0.79	0.73	545
accuracy			0.67	975
macro avg	0.67	0.66	0.66	975
weighted avg	0.67	0.67	0.67	975

For Contraceptive_method_used (Label 0):

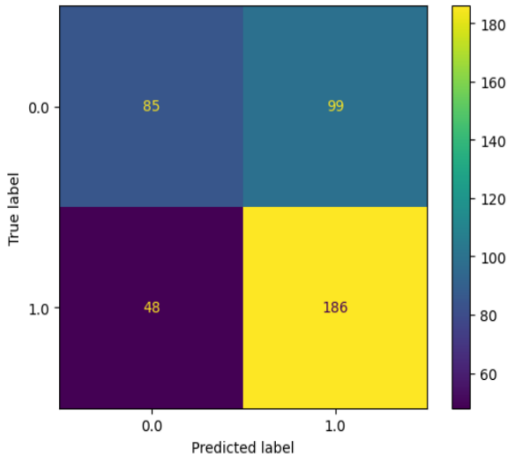
- Precision (66%) – 66% of married women predicted are actually not using Contraceptive method out of all married women predicted to not using Contraceptive method.
- Recall (53%) – Out of all the married women not using Contraceptive method , 53%of married women have been predicted correctly .

For Contraceptive_method_used (Label 1):

- Precision (68%) – 68% of married women predicted are actually using Contraceptive method out of all married women predicted to be using Contraceptive method .
- Recall (79%) – Out of all the married women actually using contraceptive method 79% of married women have been predicted correctly .
- And the Accuracy is 67% which is more than 50%, so the model is Good.

Confusion matrix for test data:

```
array([[ 85,  99],
       [ 48, 186]], dtype=int64)
```



This plot shows the relationship between the true labels and predicted labels as 0's and 1's

And the classification report is as follows

	precision	recall	f1-score	support
0.0	0.64	0.46	0.54	184
1.0	0.65	0.79	0.72	234
accuracy			0.65	418
macro avg	0.65	0.63	0.63	418
weighted avg	0.65	0.65	0.64	418

For Contraceptive_method_used (Label 0):

- Precision (64%) – 64% of married women predicted are actually not using Contraceptive method out of all married women predicted to not using Contraceptive method.
- Recall (46%) – Out of all the married women not using Contraceptive method , 46%of married women have been predicted correctly .

For Contraceptive_method_used (Label 1):

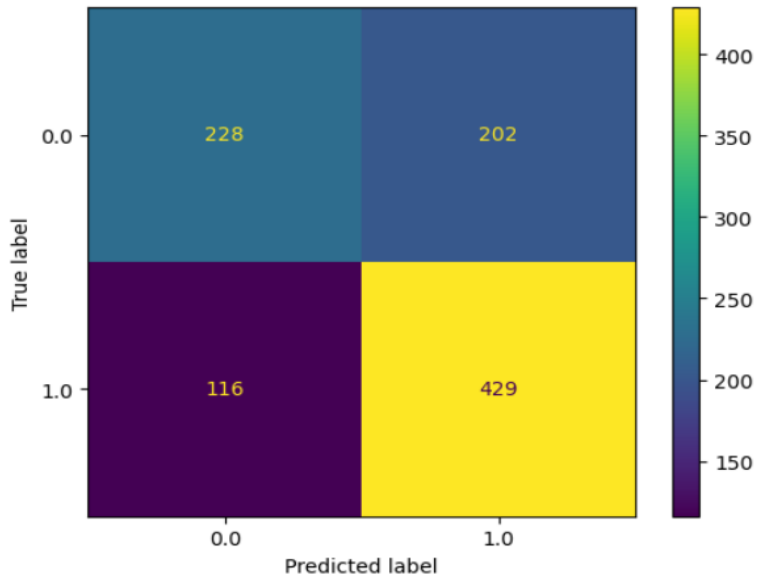
- Precision (65%) – 65% of married women predicted are actually using Contraceptive method out of all married women predicted to be using Contraceptive method .
- Recall (79%) – Out of all the married women actually using contraceptive method ,79% of married women have been predicted correctly .
- And the Accuracy is 65% which is more than 50%, so the model is also Good asTraining Data.

Grid search :

By using the grid search CV from sklearn model to get predict the best model. Theprocess is same as the above and we get,

For train data

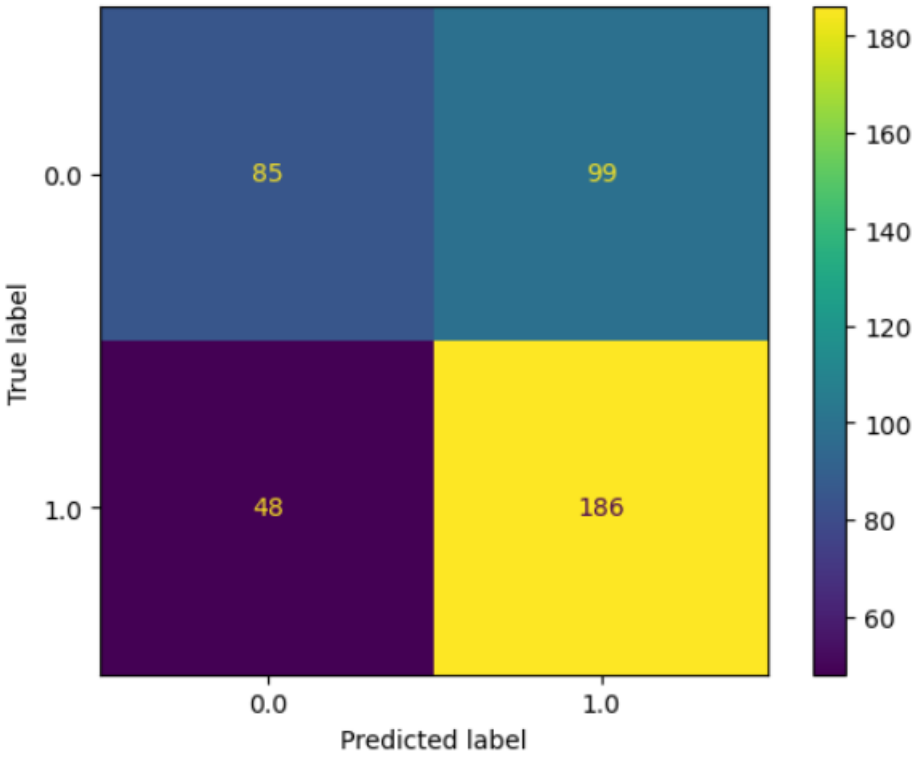
	precision	recall	f1-score	support
0.0	0.66	0.53	0.59	430
1.0	0.68	0.79	0.73	545
accuracy			0.67	975
macro avg	0.67	0.66	0.66	975
weighted avg	0.67	0.67	0.67	975



As the above method, here also we get similar values and accuracy is still 67 %

For test data

	precision	recall	f1-score	support
0.0	0.64	0.46	0.54	184
1.0	0.65	0.79	0.72	234
accuracy			0.65	418
macro avg	0.65	0.63	0.63	418
weighted avg	0.65	0.65	0.64	418



As the above method, here also we get similar values and accuracy is still 65 %

- Overall accuracy of the model – 67 % of total predictions are correct
- Accuracy, AUC, Precision and Recall for test data is almost inline with training data.This proves no overfitting or underfitting has happened, and overall the model is agood model for classification

LINEAR DISCRIMINANT ANALYSIS

Train and Test Split:

The procedure is same as the above Logistics regression for splitting the Train and test data.

Need to import the LDA(Linear Discriminant analysis) from the sklearn library and the results is as follows,

	precision	recall	f1-score	support
0	0.66	0.53	0.59	430
1	0.68	0.79	0.73	545
accuracy			0.67	975
macro avg	0.67	0.66	0.66	975
weighted avg	0.67	0.67	0.67	975

	precision	recall	f1-score	support
0.0	0.64	0.46	0.54	184
1.0	0.65	0.79	0.72	234
accuracy			0.65	418
macro avg	0.65	0.63	0.63	418
weighted avg	0.65	0.65	0.64	418

There is some slight difference with the Training and the test data reports, but its ok as the Accuracy of train data is as 67% and the accuracy for the test data is as 65%

CART

In CART we can use the dataset with outliers as its not sensitive with outliers.

Train and Test Split:

The Same procedure as the above Logis□c regression and the LDA, Train and testdata need to be splitted, and before that the necessary libraries need to be imported.

In cart , the decision tree is the most important,

Decision tree:

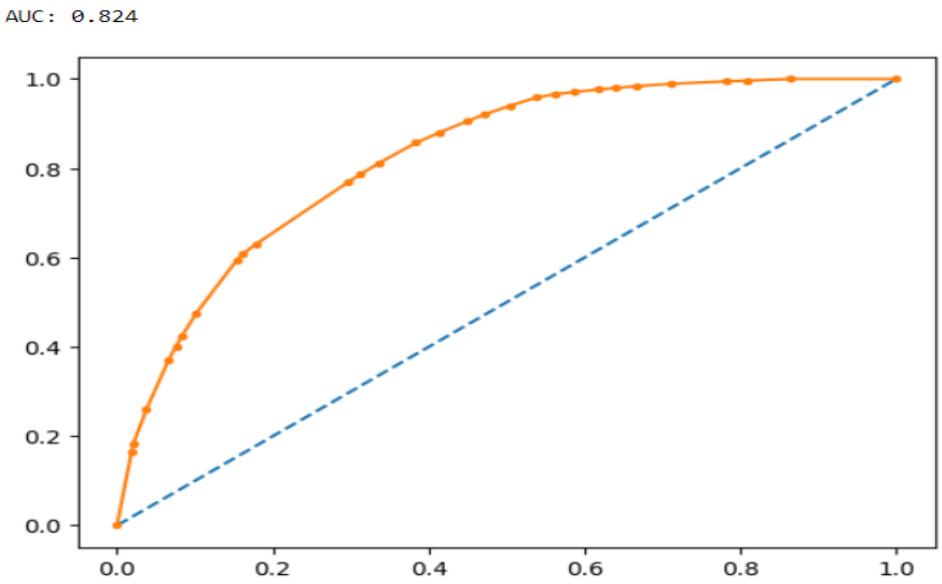
- Fit the train and test data into decision tree. We need to create in new worddocument and saved in Project folder.
- Now we can copy and paste the code in http://webgraphviz.com/. For checking thedecision tree we can delete the existing codes and paste it there.
- The tree will be little messy as the data contains vast information or classifications,sowe will reduce the max.leaf , max.depth of the tree and the min. sample size.
- Here “GINI” ,a decision tree classifier plays the important role. And creating a newword document with reduced branches as 30, leaf is 10 and depth is 7 and saved thedocument in project folder.
- Now decision tree is looking better than before

Now Let us check the feature Importance, where Feature importance refers totechniques that assign a score to input features based on how useful they are at predic□nga target variable.

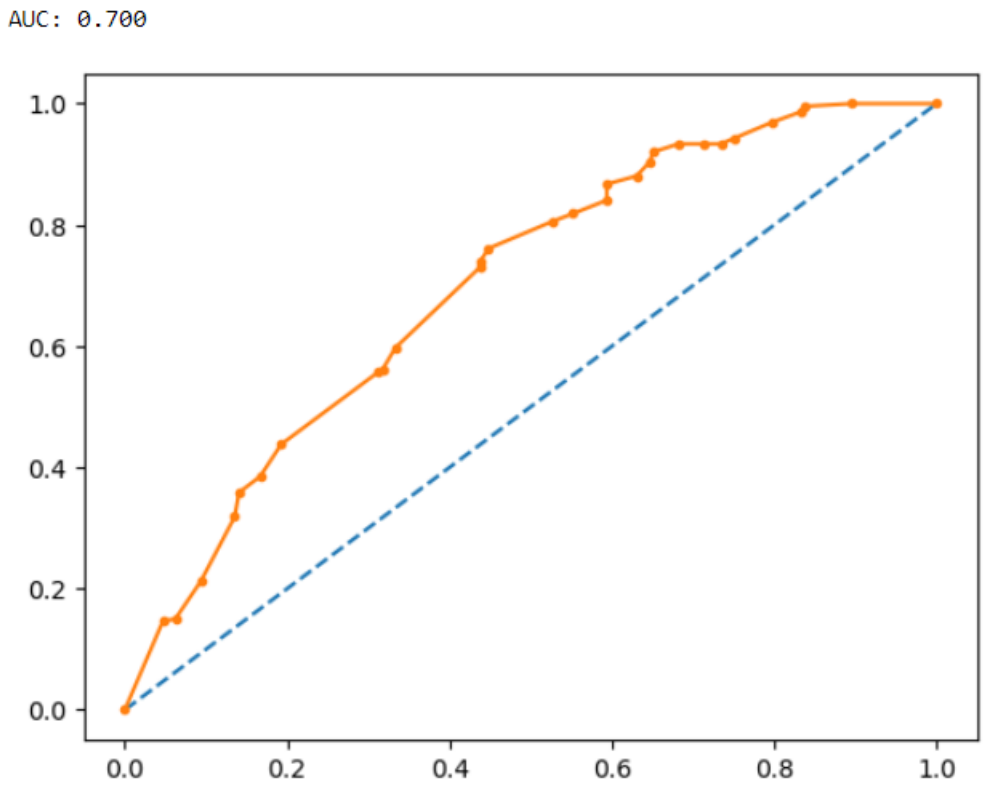
	Imp
Wife_age	0.316134
Wife_ education	0.112728
Husband_education	0.055614
No_of_children_born	0.256892
Wife_religion	0.000000
Wife_Working	0.055761
Husband_Occupation	0.117332
Standard_of_living_index	0.085540
Media_exposure	0.000000

As we see ,depend upon the ‘wife_age’ having more importance, we can slightly predict that the contraceptive method can be used depend upon the age factors of women.

AUC plot:



As we see the AUC curve bending high , the model will be good and its AUCvalue for train data is 82.4%



Here the plot is not quite smooth , but over the area its keeping up the bend formation and its AUC value for test data is 70.0%

Confusion matrix for train data:

```
array([[260, 162],
       [ 79, 474]], dtype=int64)
```

	precision	recall	f1-score	support
0.0	0.77	0.62	0.68	422
1.0	0.75	0.86	0.80	553
accuracy			0.75	975
macro avg	0.76	0.74	0.74	975
weighted avg	0.75	0.75	0.75	975

By checking up the confusion matrix of the train data, we can get the value of True Positive as 260 and the True Negative as 474.

For Contraceptive_method_used (Label 0):

- Precision (77%) – 77% of married women predicted are actually not using Contraceptive method out of all married women predicted to not using Contraceptive method.
- Recall (62%) – Out of all the married women not using Contraceptive method , 62%of married women have been predicted correctly .

For Contraceptive_method_used (Label 1):

- Precision (75%) – 75% of married women predicted are actually using Contraceptive method out of all married women predicted to be using Contraceptive method .
- Recall (86%) – Out of all the married women actually using contraceptive method ,86% of married women have been predicted correctly .
- And the Accuracy is 75% which is more than 50%, so the model is also Good as Training Data.

Confusion matrix for test data:

```
array([[ 91, 101],
       [ 44, 182]], dtype=int64)
```

	precision	recall	f1-score	support
0.0	0.67	0.47	0.56	192
1.0	0.64	0.81	0.72	226
accuracy			0.65	418
macro avg	0.66	0.64	0.64	418
weighted avg	0.66	0.65	0.64	418

By checking up the confusion matrix of the train data, we can get the value of True Positive as 91 and the True Negative as 182.

For Contraceptive_method_used (Label 0):

- Precision (67%) – 67% of married women predicted are actually not using Contraceptive method out of all married women predicted to not using Contraceptive method.
- Recall (47%) – Out of all the married women not using Contraceptive method , 47%of married women have been predicted correctly .

For Contraceptive_method_used (Label 1):

- Precision (64%) – 64% of married women predicted are actually using Contraceptive method out of all married women predicted to be using Contraceptive method .
- Recall (81%) – Out of all the married women actually using contraceptive method ,81% of married women have been predicted correctly .
- And the Accuracy is 65% which is more than 50%, so the model is also Good as Training Data.

CONCLUSION

- From these above models , in Every models the Encoded label ‘1’(conceptive method used) predicted as high and the Accuracy and the F1 score of the models also favourfor the label ’1’.
- But we can’t conclude that the contraceptive method used or not , but we canpredict that the married women used the Contraceptive method as prediction and the final prediction also showing the same things only.