Predictive Modelling - Project

Chetan R Deshpande

Contents **Page No.**

Problem 1: Linear Regression

The comp-activ databases is a collection of a computer systems activity measures . The data was collected from a Sun Sparcstation 20/712 with 128 Mbytes of memory running in a multi-user university department. Users would typically be doing a large variety of tasks ranging from accessing the internet, editing files or running very cpu-bound programs.

As you are a budding data scientist you thought to find out a linear equation to build a model to predict 'usr'(Portion of time (%) that cpus run in user mode) and to find out how each attribute affects the system to be in 'usr' mode using a list of system attributes.

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

- *Using head(), we can take a quick glimpse of the data and roughly analyze the information that we can grasp.*

	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	173094
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	186900
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	102123
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	186370
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	176025

From the above picture, we can see that there are NaN values present in the rchar column and runqsz column has to be encoded.

- (8192, 22) *is the shape of the entire dataset.*

- *The following data types are present in the dataset.*

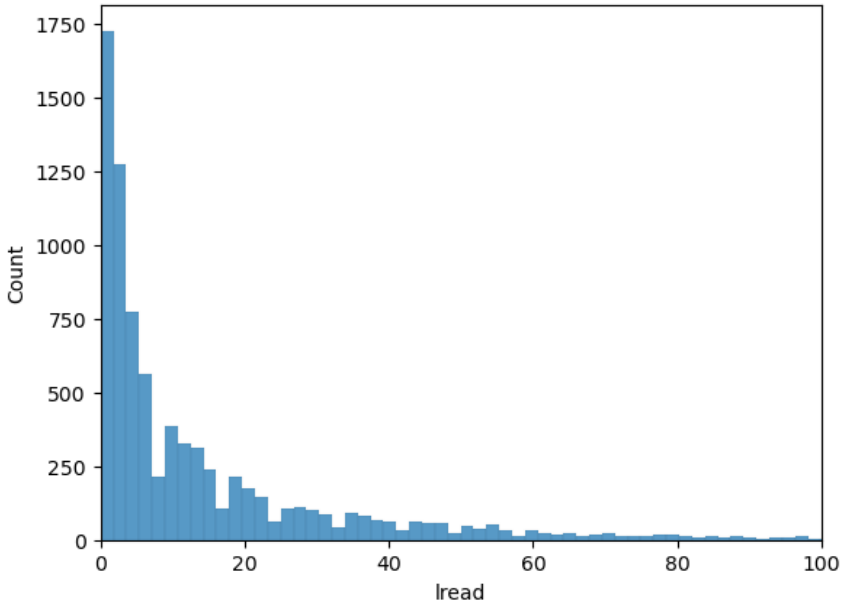
  dtypes: float64(13), int64(8), object(1)
  memory usage: 1.4+ MB

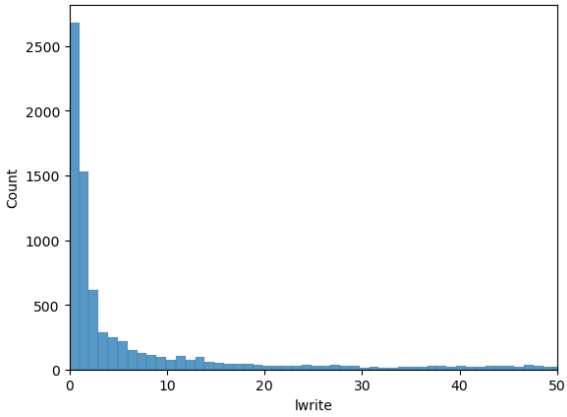
- *These are the number of unique values present in the data set wrt their columns-*

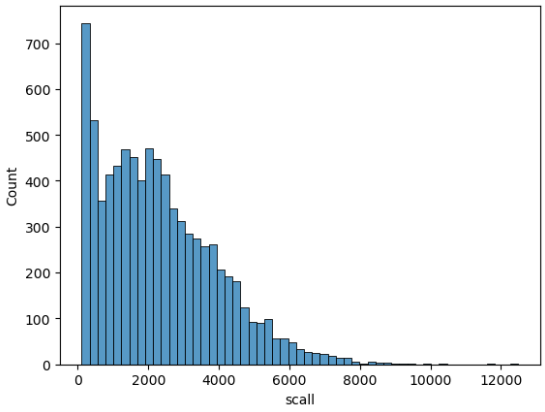
  lread 235
  lwrite 189
  scall 4115

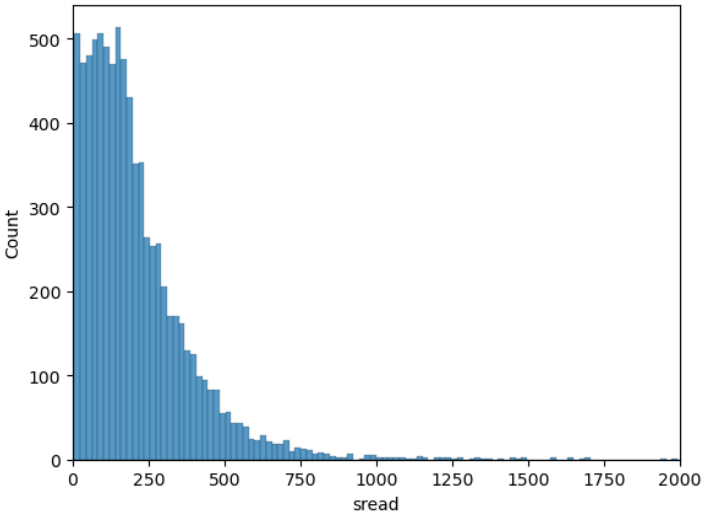
sread	794
swrite	640
fork	228
exec	386
rchar	7898
wchar	7925
pgout	404
ppgout	774
pgfree	1070
pgscan	1202
atch	253
pgin	832
ppgin	1072
pflt	2987
vflt	3799
runqsz	2
freemem	3165
freeswap	7658
usr	56

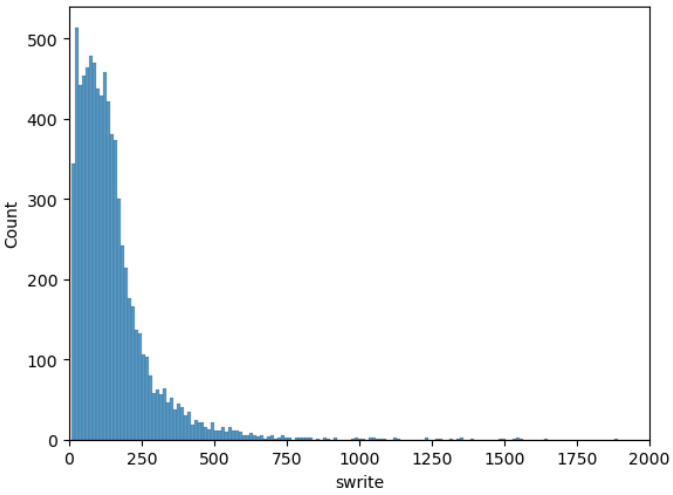
Univariate Analysis

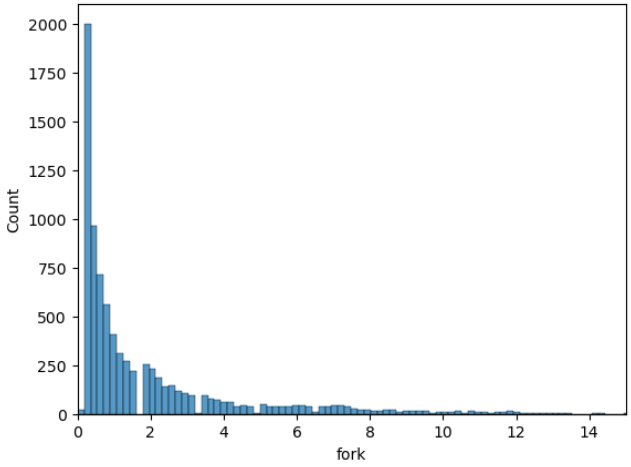


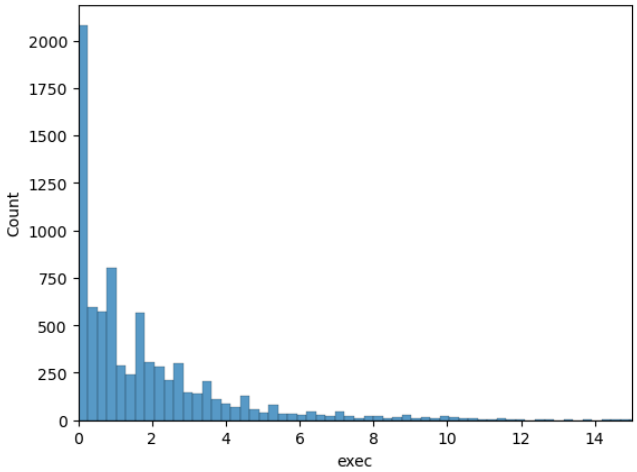

Count
lwrite

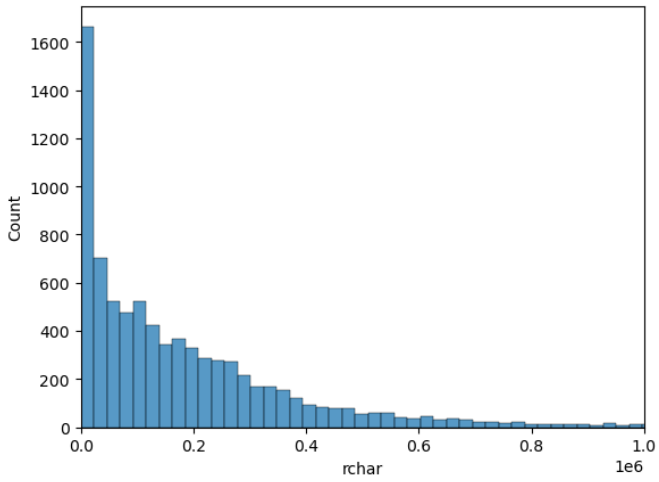

Count
scall

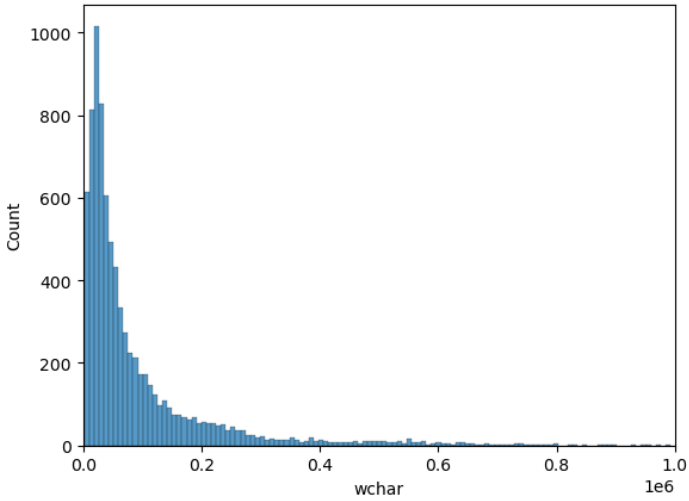

Count
sread

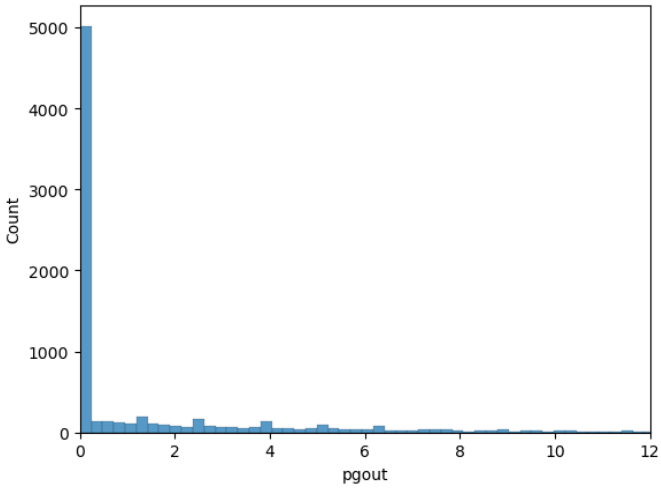

Count
swrite
0
100
200
300
400
500
0
250
500
750
1000
1250
1500
1750
2000

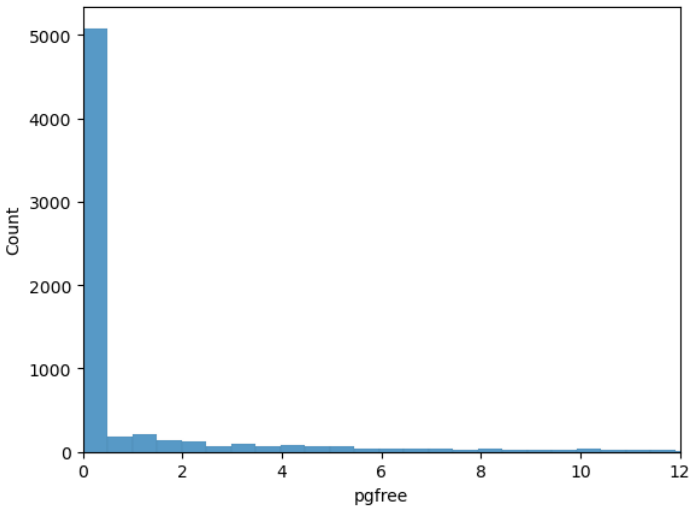

Count
fork
0
250
500
750
1000
1250
1500
1750
2000
0
2
4
6
8
10
12
14

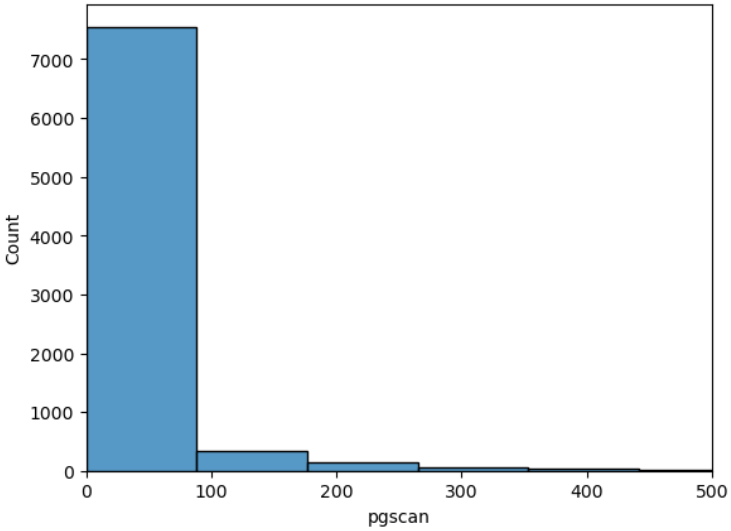

Count
exec
0
250
500
750
1000
1250
1500
1750
2000
0
2
4
6
8
10
12
14

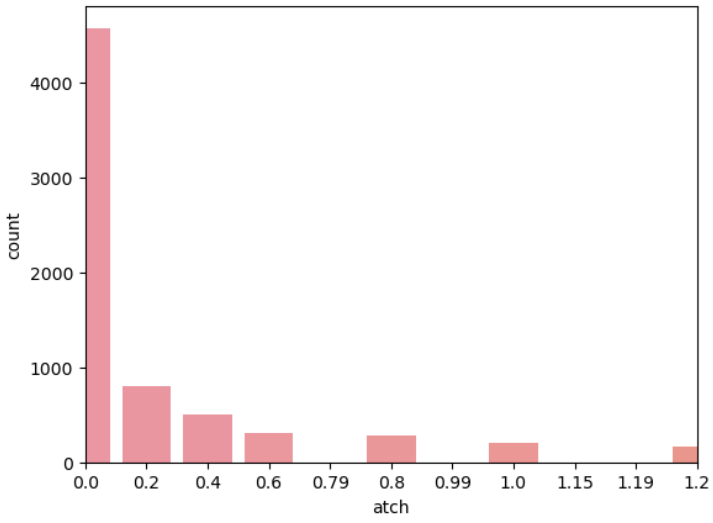

Count
rchar
1e6
0
200
400
600
800
1000
1200
1400
1600
0.0
0.2
0.4
0.6
0.8
1.0

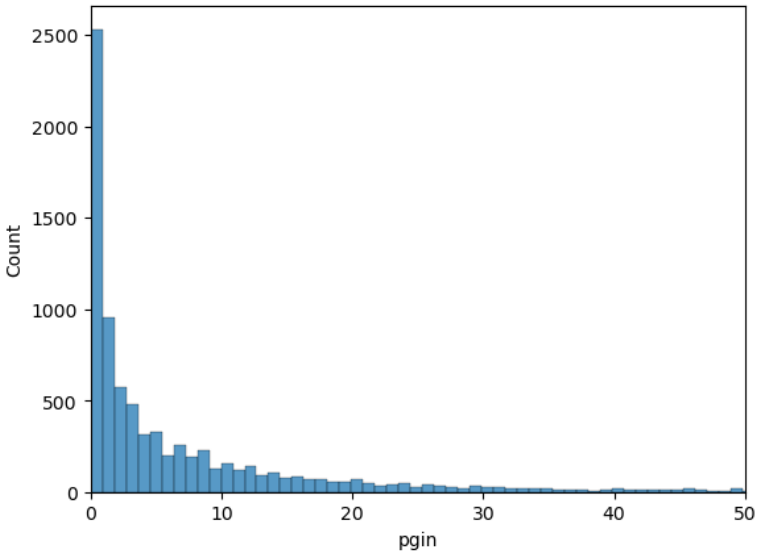

Count
wchar
1e6
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0

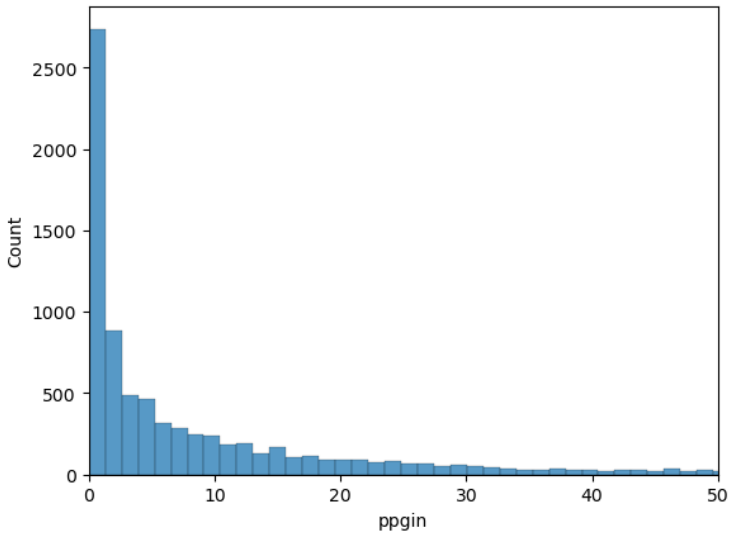

Count
pgout
0
1000
2000
3000
4000
5000
0
2
4
6
8
10
12

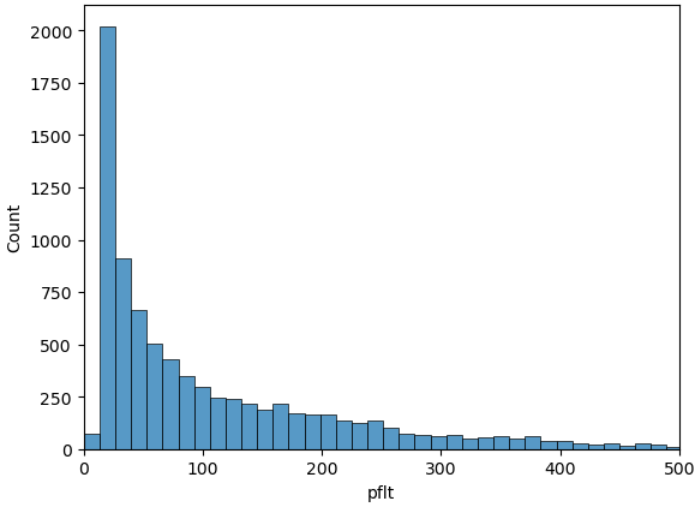

5000
4000
3000
2000
1000
0
Count
0
2
4
6
8
10
12
pgfree

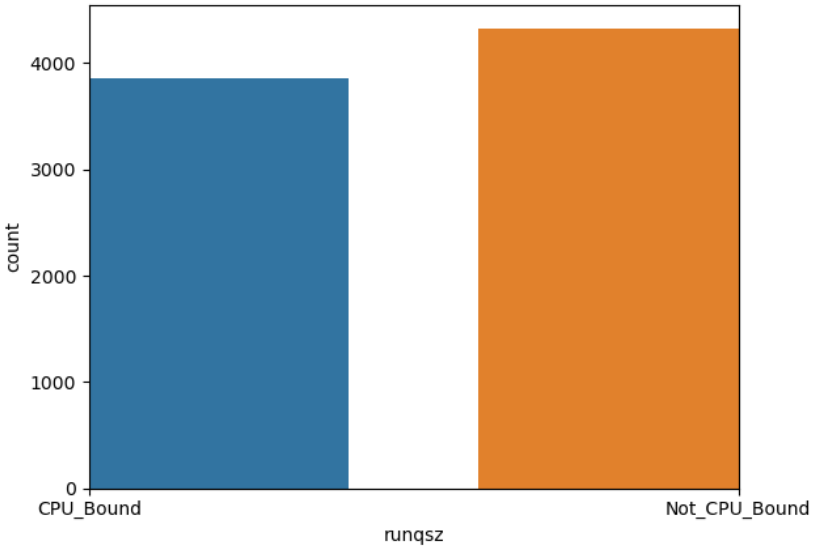

7000
6000
5000
4000
3000
2000
1000
0
Count
0
100
200
300
400
500
pgscan

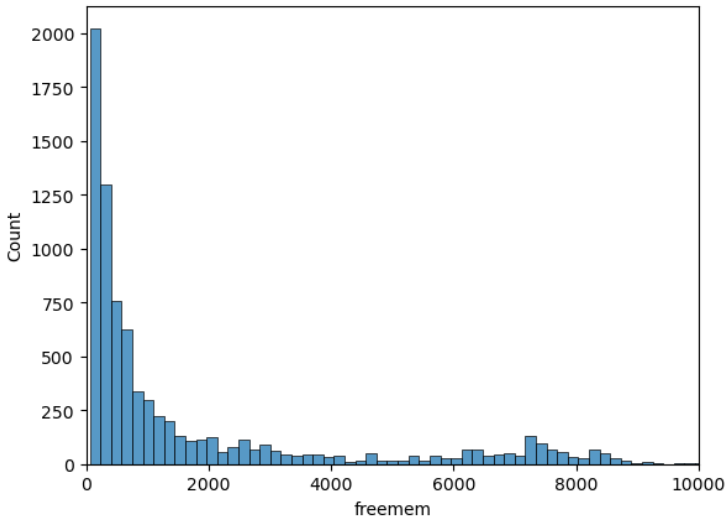

4000
3000
2000
1000
0
count
0.0
0.2
0.4
0.6
0.79
0.8
0.99
1.0
1.15
1.19
1.2
atch

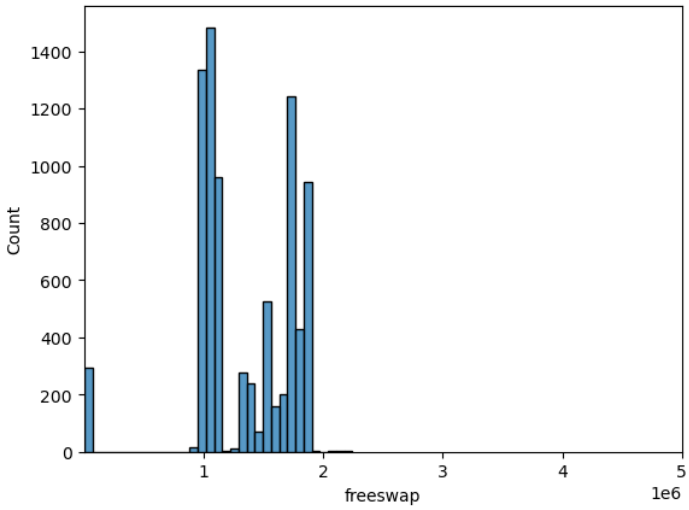

Count
2500
2000
1500
1000
500
0
0
10
20
30
40
50
pgin

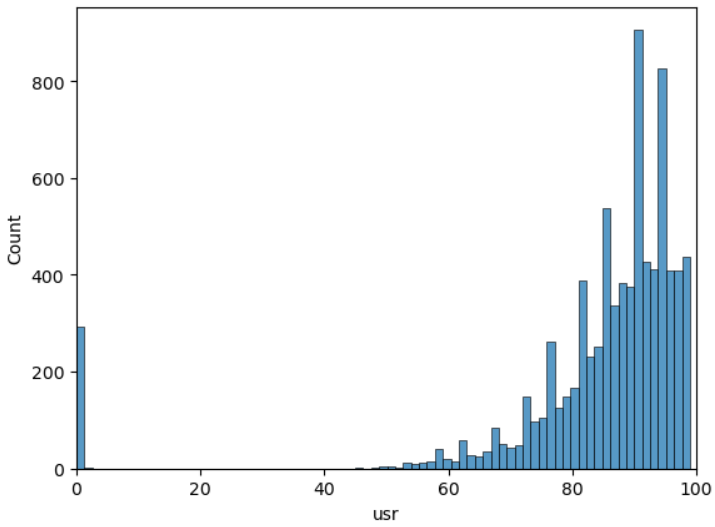

Count
2500
2000
1500
1000
500
0
0
10
20
30
40
50
ppgin


Count
2000
1750
1500
1250
1000
750
500
250
0
0
100
200
300
400
500
pflt

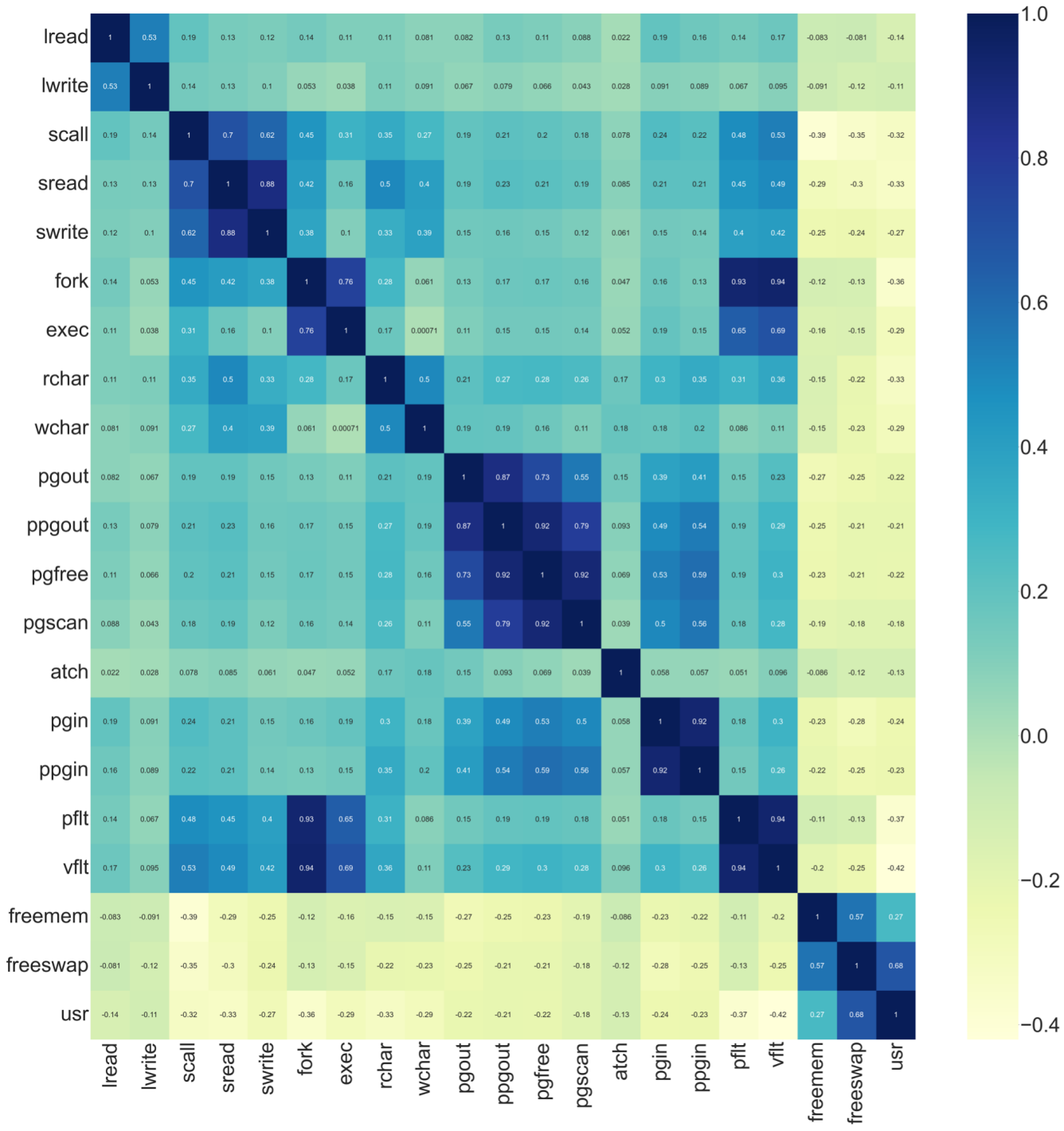

4000
3000
2000
1000
0
count
CPU_Bound
Not_CPU_Bound
runqsz


2000
1750
1500
1250
1000
750
500
250
0
Count
0
2000
4000
6000
8000
10000
freemem

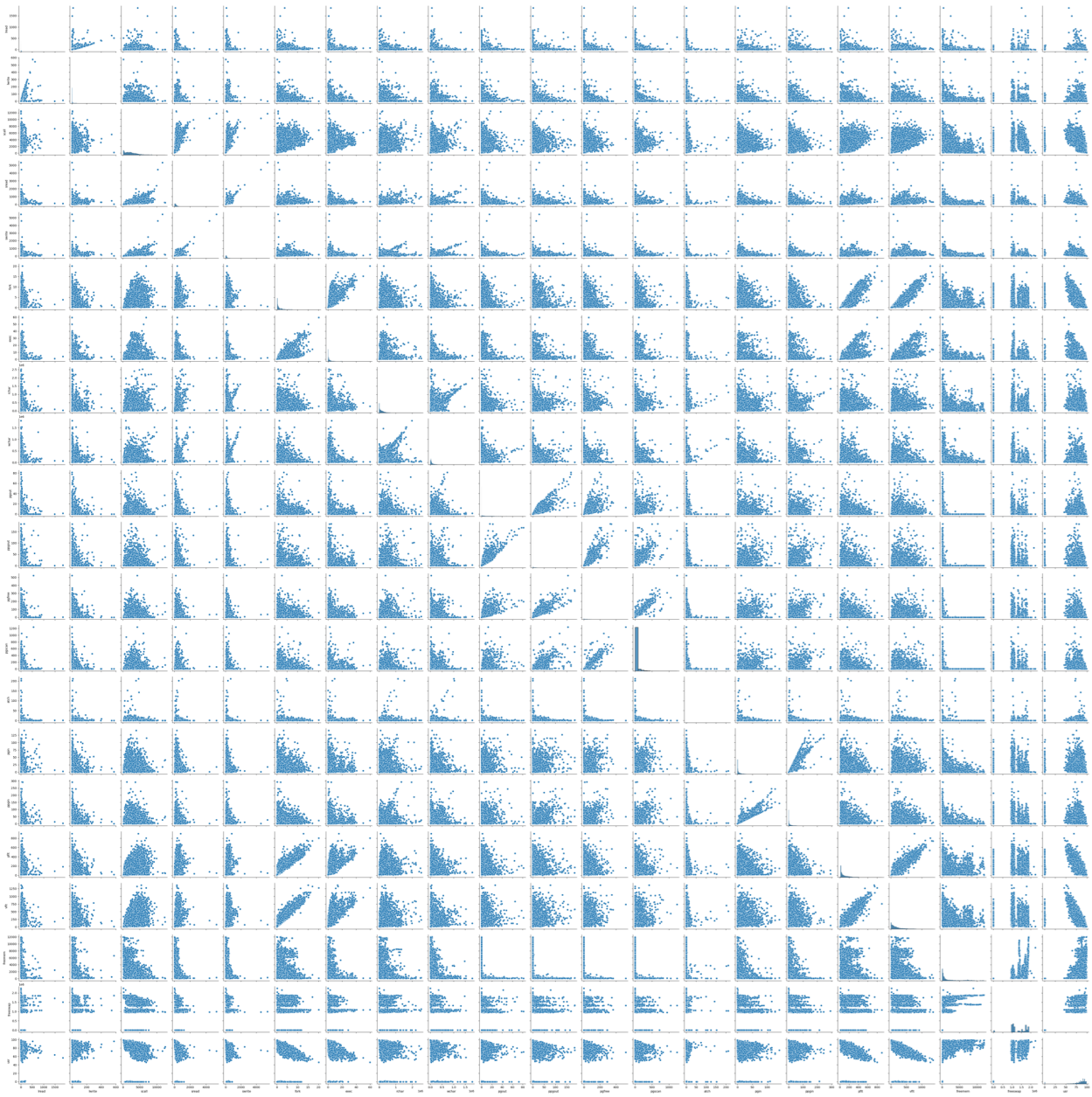

1400
1200
1000
800
600
400
200
0
Count
1
2
3
4
5
1e6
freeswap


800
600
400
200
0
Count
0
20
40
60
80
100
usr

Bivariate Analysis



Multivariate Analysis



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.

- *Checking for null values and treating it -*

  We have found null values in 2 variables rchar and wchar.

  rchar 104
  wchar 15

  So, we will be replacing the null values with the median.

  median of rchar is: 125473.5
  median of wchar is: 46619.0

  And the results are

  rchar 0
  wchar 0

- *Checking for the values which are equal to zero -*

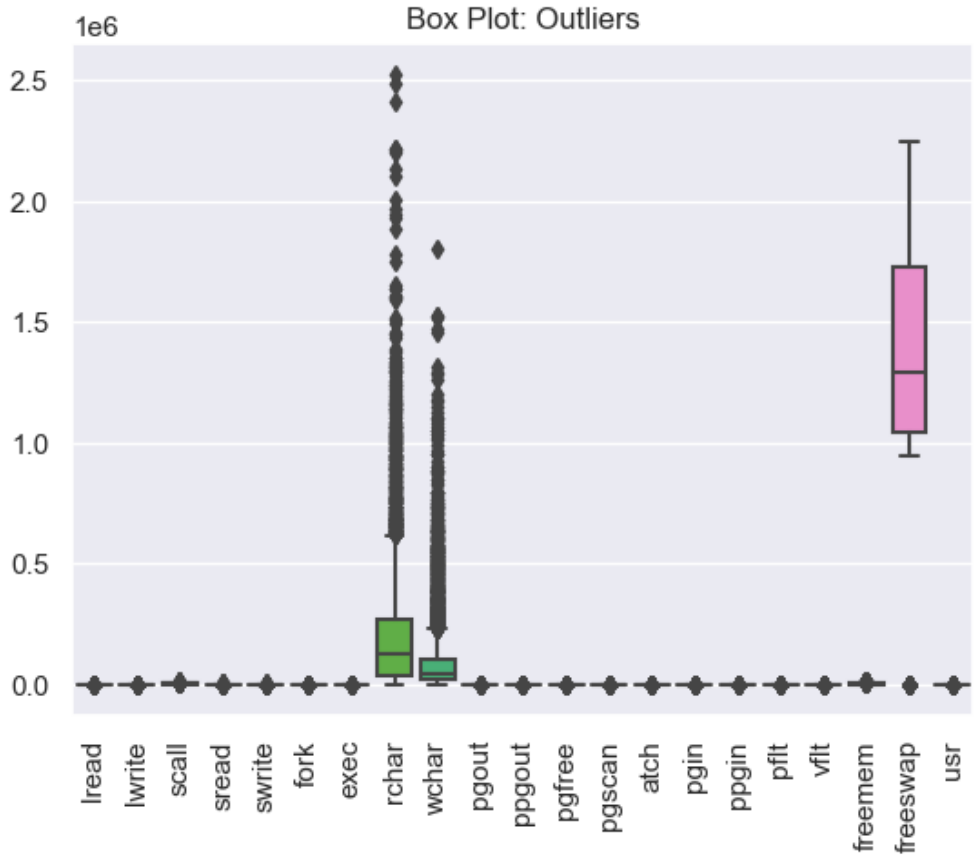
  lread 0
  lwrite 0
  scall 0
  sread 0
  swrite 0
  fork 0
  exec 0
  rchar 104
  wchar 15
  pgout 0
  ppgout 0
  pgfree 0
  pgscan 0
  atch 0

pgin 0
ppgin 0
pflt 0
vflt 0
runqsz 0
freemem 0
freeswap 0
usr 0

As these are the attributes referring to the systems and most of it specifically measured in per second, I will keep the 0s as it is because it makes sense to this dataset.

- *There are 0 duplicates in the given dataset.*

- *There are existing outliers and as it is a system related data given in per second, we will not treat the outliers.*



1.3 Encode the data (having string values) for Modelling. Split the data into train and test (70:30). Apply Linear regression using scikit learn. Perform checks for significant variables using appropriate method from statsmodel. Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare. Compare these models and select the best one with appropriate reasoning.

- *Using get_dummies, we have encoded the variable* - runqsz
- *Further, we have split the dataset into training and testing in the size of 70:30 respectively.*
- *The following is the model summary.*

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.627
Model:	OLS	Adj. R-squared:	0.626
Method:	Least Squares	F-statistic:	458.0
Date:	Sat, 01 Jul 2023	Prob (F-statistic):	0.00
Time:	19:06:22	Log-Likelihood:	-21862.
No. Observations:	5734	AIC:	4.377e+04
Df Residuals:	5712	BIC:	4.391e+04
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
lread	-0.0208	0.003	-6.009	0.000	-0.028	-0.014
lwrite	0.0119	0.006	2.008	0.045	0.000	0.023
scall	0.0010	0.000	6.783	0.000	0.001	0.001
sread	-0.0014	0.002	-0.736	0.462	-0.005	0.002
swrite	-0.0002	0.002	-0.106	0.916	-0.005	0.004
fork	-1.8731	0.251	-7.450	0.000	-2.366	-1.380
exec	-0.0281	0.051	-0.553	0.581	-0.128	0.071
rchar	-3.572e-06	8.65e-07	-4.128	0.000	-5.27e-06	-1.88e-06
wchar	-9.981e-06	1.39e-06	-7.156	0.000	-1.27e-05	-7.25e-06
pgout	-0.2311	0.062	-3.750	0.000	-0.352	-0.110
ppgout	0.1255	0.035	3.544	0.000	0.056	0.195
pgfree	-0.0800	0.019	-4.292	0.000	-0.117	-0.043
pgscan	0.0103	0.005	1.924	0.054	-0.000	0.021
atch	-0.0955	0.033	-2.921	0.003	-0.160	-0.031
pgin	0.0522	0.029	1.812	0.070	-0.004	0.109
ppgin	-0.0350	0.019	-1.868	0.062	-0.072	0.002
pflt	-0.0366	0.004	-8.563	0.000	-0.045	-0.028
vflt	0.0203	0.003	6.077	0.000	0.014	0.027
freemem	-0.0016	7.51e-05	-21.053	0.000	-0.002	-0.001
freeswap	3.212e-05	4.57e-07	70.358	0.000	3.12e-05	3.3e-05
runqsz_CPU_Bound	44.5624	0.745	59.808	0.000	43.102	46.023
runqsz_Not_CPU_Bound	52.3619	0.702	74.637	0.000	50.987	53.737

Omnibus:	1598.133	Durbin-Watson:	2.026
Prob(Omnibus):	0.000	Jarque-Bera (JB):	5302.353
Skew:	-1.397	Prob(JB):	0.00

- After trial and error of dropping the columns and checking for the R-squared value, we get the following table.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    usr   R-squared (uncentered):                   0.967
Model:                            OLS   Adj. R-squared (uncentered):              0.967
Method:                 Least Squares   F-statistic:                          1.869e+04
Date:                Sat, 01 Jul 2023   Prob (F-statistic):                        0.00
Time:                        15:30:50   Log-Likelihood:                         -23897.
No. Observations:                5734   AIC:                                  4.781e+04
Df Residuals:                    5725   BIC:                                  4.787e+04
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
lread         -0.0190      0.005     -3.955      0.000      -0.028      -0.010
lwrite         0.0672      0.008      8.094      0.000       0.051       0.083
rchar       5.411e-06   1.06e-06      5.091      0.000    3.33e-06    7.49e-06
wchar       8.729e-06   1.74e-06      5.008      0.000    5.31e-06    1.21e-05
pgout          0.1360      0.040      3.372      0.001       0.057       0.215
pflt          -0.0177      0.002     -9.383      0.000      -0.021      -0.014
freemem       -0.0033   9.87e-05    -33.808      0.000      -0.004      -0.003
freeswap    5.938e-05   3.07e-07    193.145      0.000    5.88e-05       6e-05
Not_CPU_Bound    14.9163      0.404     36.928      0.000      14.124      15.708
==============================================================================
Omnibus:                       24.060   Durbin-Watson:                   1.966
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               18.192
Skew:                          -0.019   Prob(JB):                     0.000112
Kurtosis:                       2.727   Cond. No.                     2.75e+06
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 2.75e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

The following is the final equation for the linear regression

usr = c - 0.017353 * lread + 0.072186 * lwrite + 0.000006 * rchar + 0.000007 * wchar + 0.124091 * pgout - 0.019502 * pflt - 0.003389 * freemem + 0.000059 * freeswap + 15.002238 * Not_CPU_Bound.

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

The major steps which were involved in building this linear model is as follows -

- Initially, we cleaned the data by treating the null values and performed the EDA.
- Further, we encoded the string values.
- Later, we split the data into training data and test data into 70% and 30% respectively.
- Later we build the linear regression model by trial and error method by dropping columns and checking for their R-squared values and p-values.
- Finally, we built a model to predict 'usr'(Portion of time (%) that cpus run in user mode) and to find out how each attribute affects the system to be in 'usr' mode using a list of system attributes.

Problem 2: Logistic Regression, LDA and CART

You are a statistician at the Republic of Indonesia Ministry of Health and you are provided with a data of 1473 females collected from a Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of the survey.

The problem is to predict do/don't they use a contraceptive method of choice based on their demographic and socio-economic characteristics.

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.

- *Checking for the basic details using .head(), .info(), .shape and .duplicated().*

	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	Expose
1	45.0	Uneducated	Secondary	10.0	Scientology	No	3	Very High	Expose
2	43.0	Primary	Secondary	7.0	Scientology	No	3	Very High	Expose
3	42.0	Secondary	Primary	9.0	Scientology	No	3	High	Expose
4	36.0	Secondary	Secondary	8.0	Scientology	No	3	Low	Expose

<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

(1473, 10) *is the shape of the given dataset.*

- Performing the descriptive analysis -

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_e
count	1326.000000	1393	1393	1372.000000	1393	1393	1393.000000	1393	
unique	NaN	4	4	NaN	2	2	NaN	4	
top	NaN	Tertiary	Tertiary	NaN	Scientology	No	NaN	Very High	
freq	NaN	515	827	NaN	1186	1043	NaN	618	
mean	32.557315	NaN	NaN	3.290816	NaN	NaN	2.174444	NaN	
std	8.289259	NaN	NaN	2.399697	NaN	NaN	0.854590	NaN	
min	16.000000	NaN	NaN	0.000000	NaN	NaN	1.000000	NaN	
25%	26.000000	NaN	NaN	1.000000	NaN	NaN	1.000000	NaN	
50%	32.000000	NaN	NaN	3.000000	NaN	NaN	2.000000	NaN	
75%	39.000000	NaN	NaN	5.000000	NaN	NaN	3.000000	NaN	
max	49.000000	NaN	NaN	16.000000	NaN	NaN	4.000000	NaN	

Here, we can view some basic statistical details like percentile, mean, std, etc. of the data frame.

- *Checking for any null-values.*

  We have found the existence of null-values in 2 columns, namely -

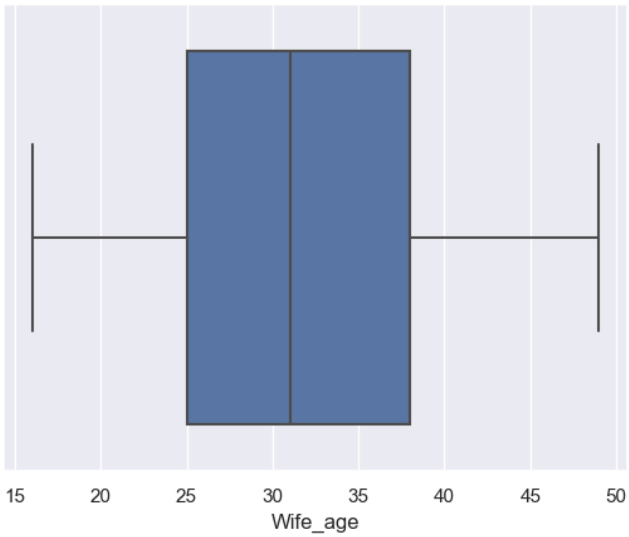
  Wife_age 67
  No_of_children_born 21

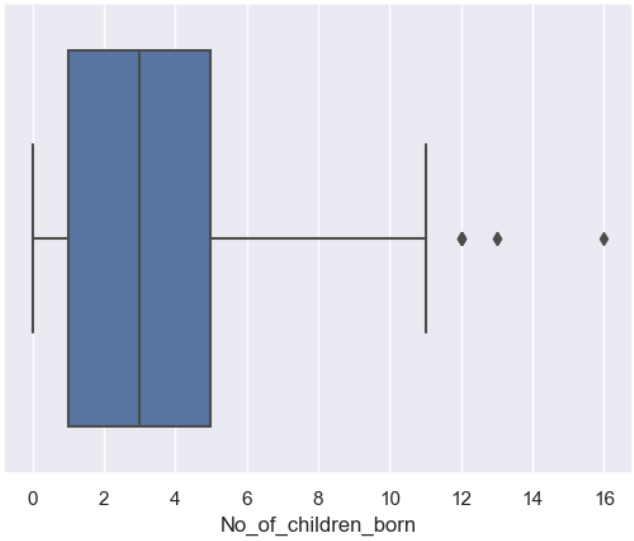
  Hence, to treat these null values, we have filled these NaN with the mode of the respective column.

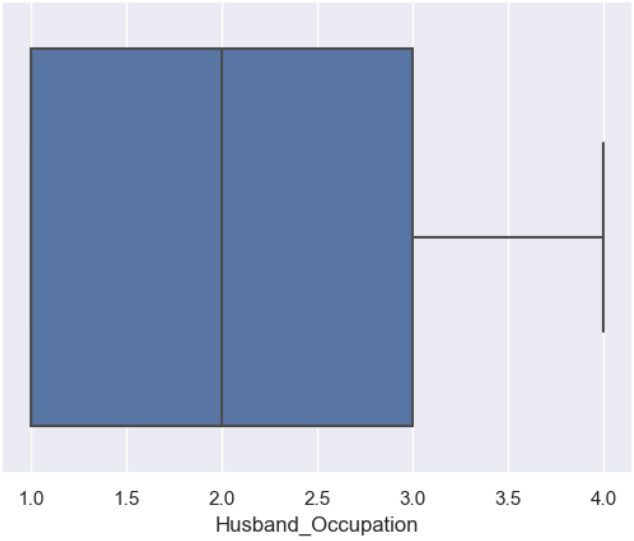
- *Checking for duplicate values -*

  There are 80 duplicated rows in the given data, so we will drop them to clean the data.

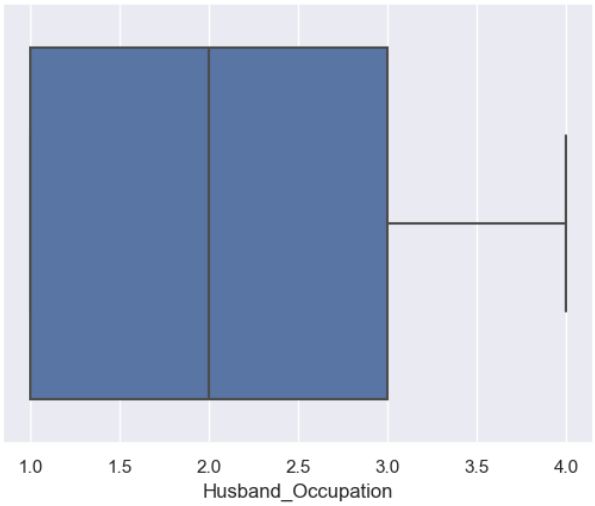
- *Using boxplots, we will check for outliers -*


15
20
25
30
35
40
45
50
Wife_age


0
2
4
6
8
10
12
14
16
No_of_children_born

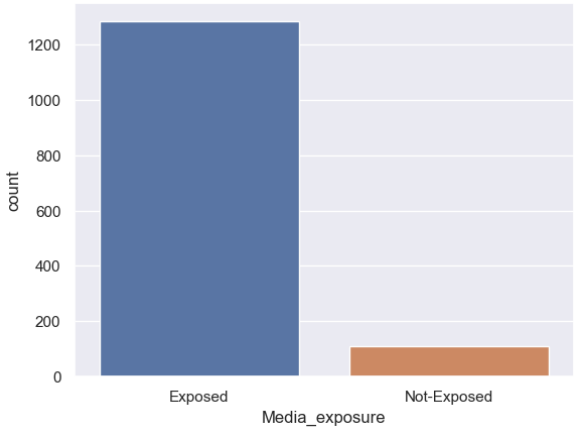

1.0
1.5
2.0
2.5
3.0
3.5
4.0
Husband_Occupation

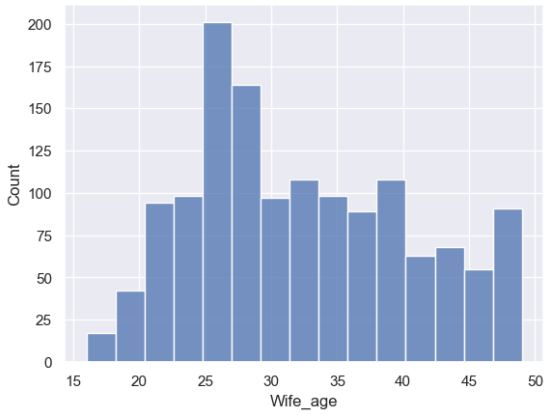
Here, we have observed outliers present in "No_of_children_born" and hence we will be treating them.

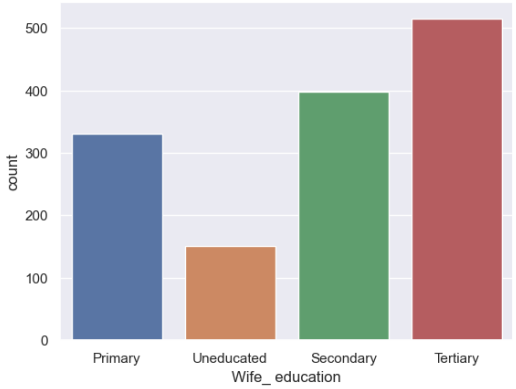


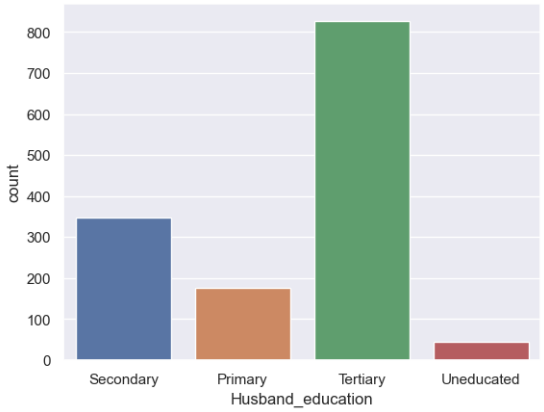
The above is the boxplot after treating the outliers.

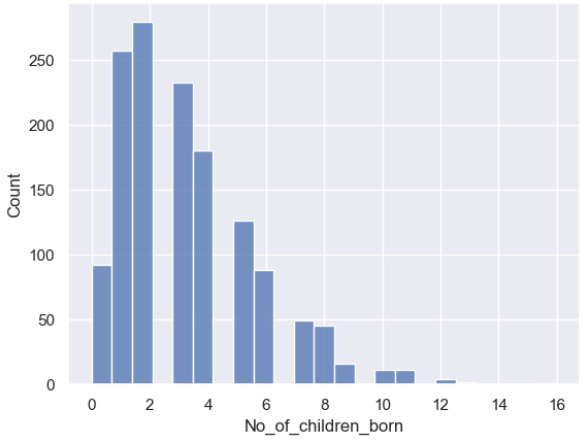
- Performing univariate, bivariate and multivariate analysis -

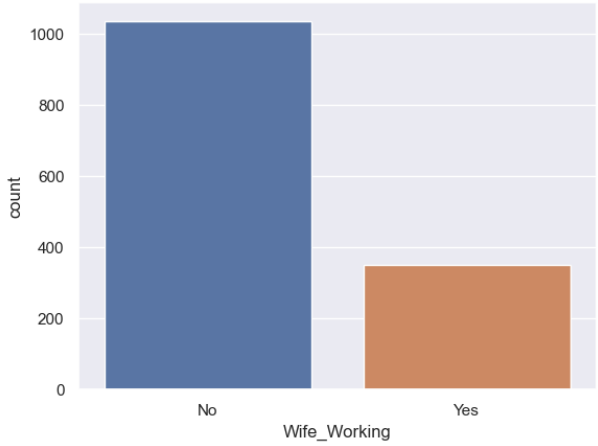



Count
Wife_age
0
25
50
75
100
125
150
175
200
15
20
25
30
35
40
45
50

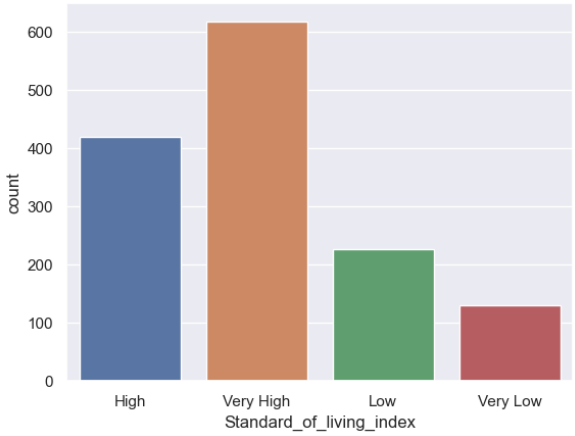

count
Wife_ education
0
100
200
300
400
500
Primary
Uneducated
Secondary
Tertiary

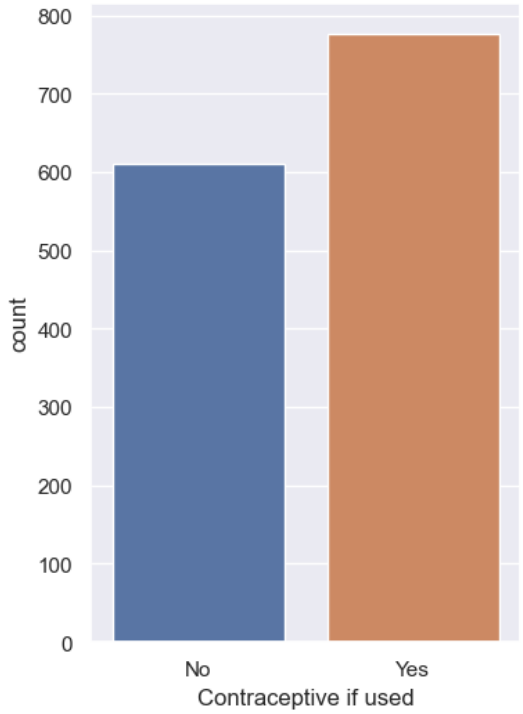

count
Husband_education
0
100
200
300
400
500
600
700
800
Secondary
Primary
Tertiary
Uneducated


Count
No_of_children_born
0
50
100
150
200
250
0
2
4
6
8
10
12
14
16

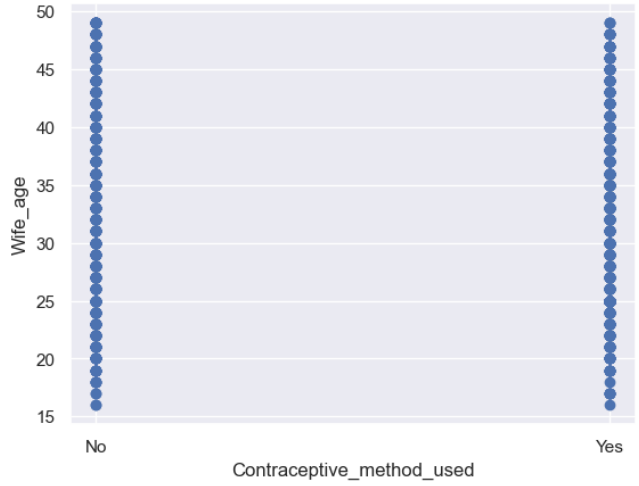

count
Wife_Working
No
Yes
0
200
400
600
800
1000

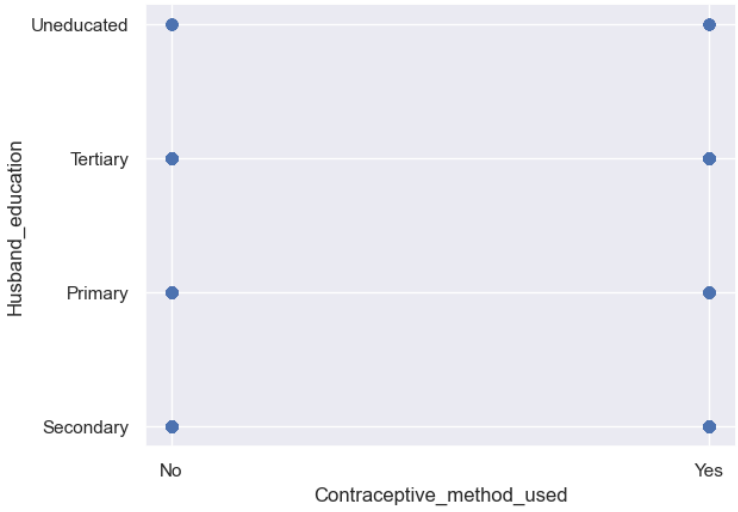

count
Husband_Occupation
1
2
3
4
0
100
200
300
400
500

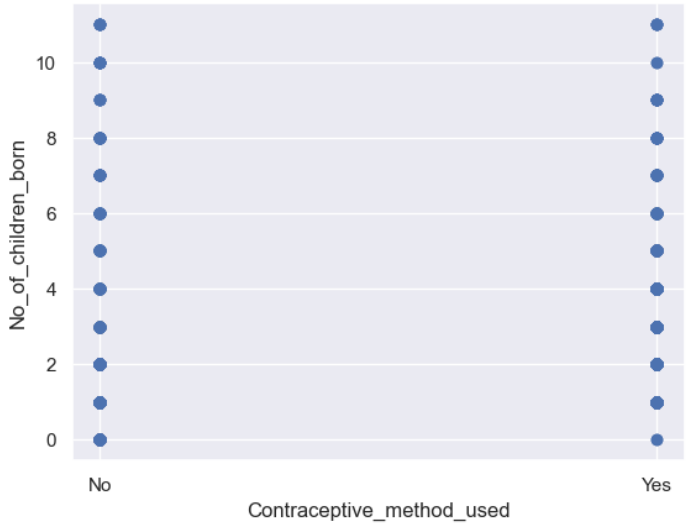

600
500
400
300
200
100
0
count
High
Very High
Low
Very Low
Standard_of_living_index

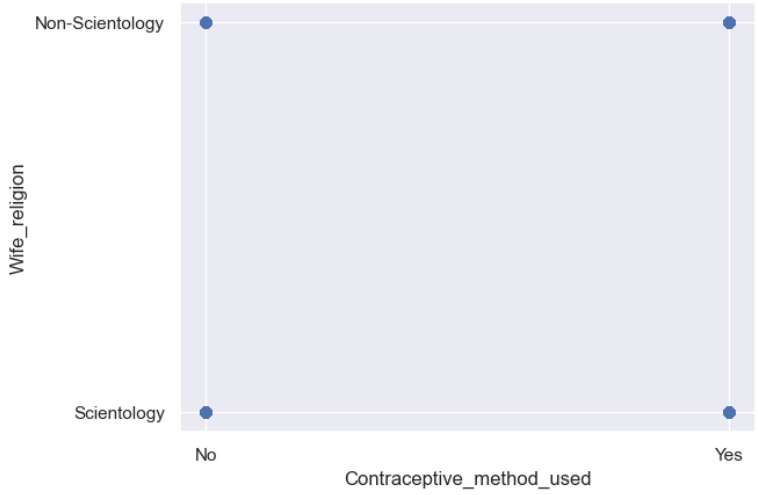

800
700
600
500
400
300
200
100
0
count
No
Yes
Contraceptive if used

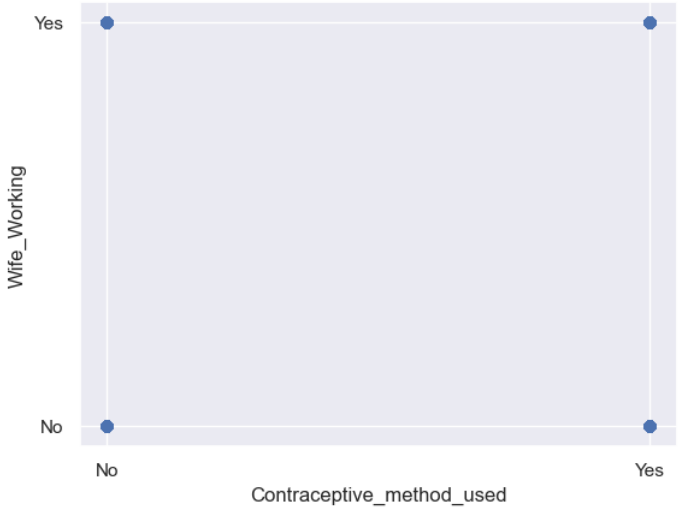
Bivariate Analysis -

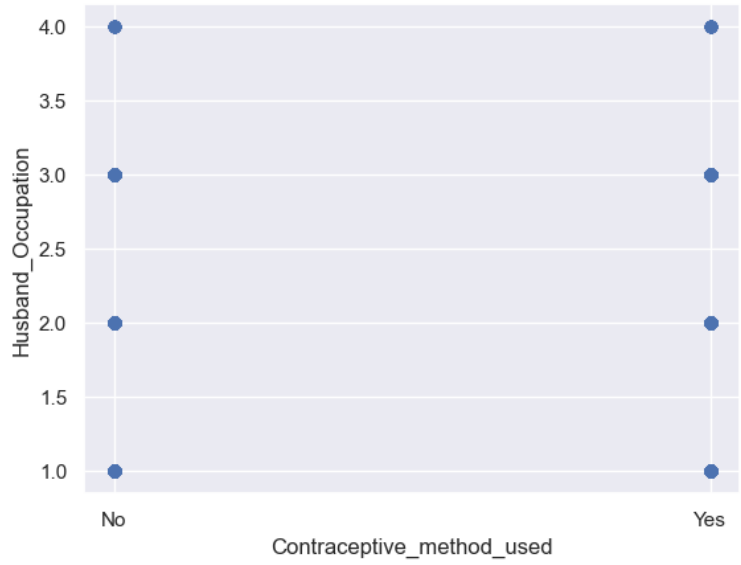


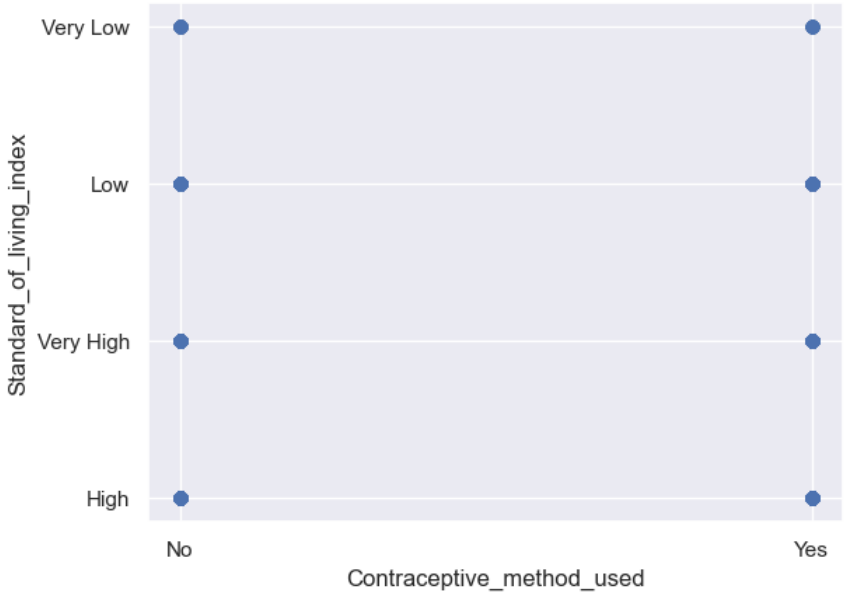



No_of_children_born
10
8
6
4
2
0
No
Yes
Contraceptive_method_used

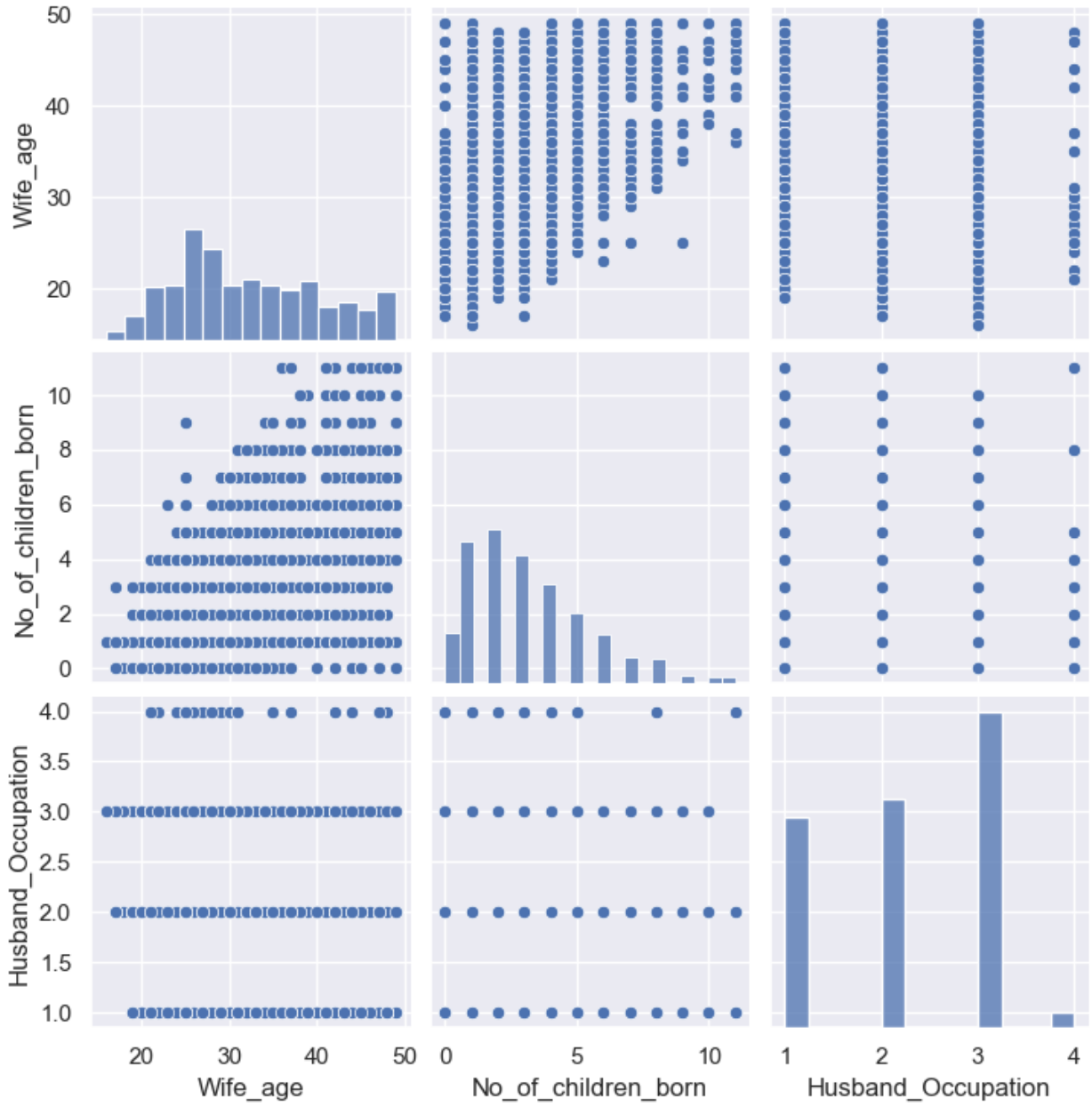

Wife_religion
Non-Scientology
Scientology
No
Yes
Contraceptive_method_used


Wife_Working
Yes
No
No
Yes
Contraceptive_method_used


4.0
3.5
3.0
2.5
2.0
1.5
1.0
Husband_Occupation
No
Yes
Contraceptive_method_used


Very Low
Low
Very High
High
Standard_of_living_index
No
Yes
Contraceptive_method_used

Multivariate analysis -



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.

Logistic Regression -

- Encoding the data which have string value -

Using "pd.get_dummies" we will encode the string variables.

Now, the shape of the data has changed to (1386, 16).

- Using the library "train_test_split" we split the data into a 70:30 ratio.

- Now, we import the library "metrics" and "logesticRegression" to perform Logistic Regression.

- The following is the model score -

```
0.6586538461538461
[[ 89  94]
 [ 48 185]]
              precision    recall  f1-score   support

          No       0.65      0.49      0.56       183
         Yes       0.66      0.79      0.72       233

    accuracy                           0.66       416
   macro avg       0.66      0.64      0.64       416
weighted avg       0.66      0.66      0.65       416
```

- The accuracy here is 0.66 and we get the confusion matrix as well.

LDA (linear discriminant analysis) -

- **For LDA, I have encoded all the string values to numbers.**

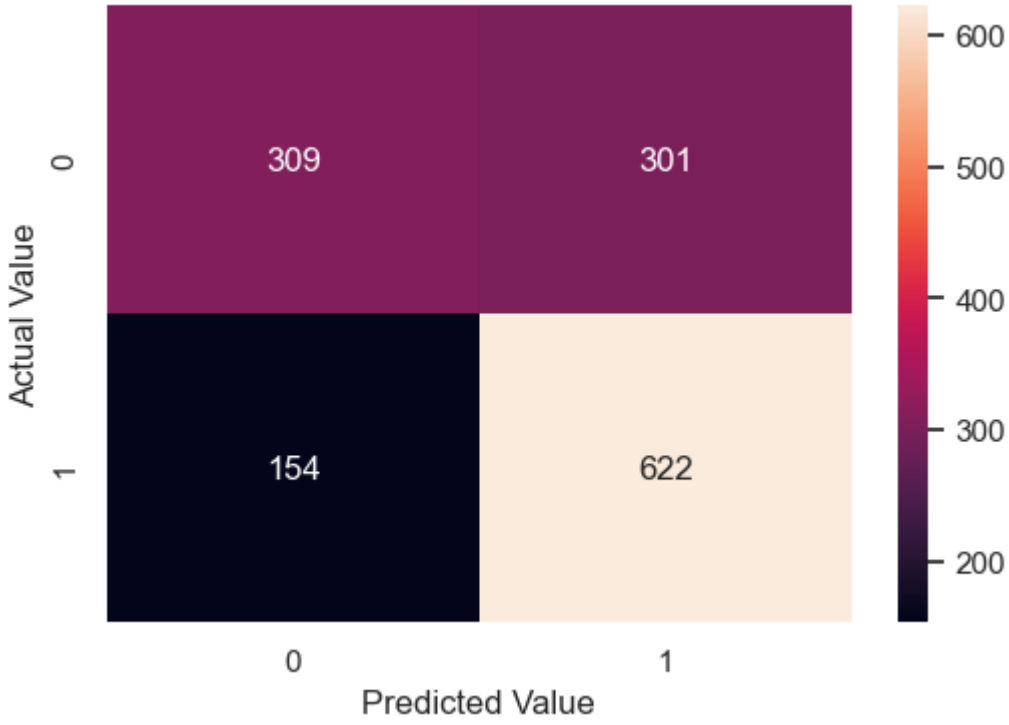
For eg - Wife_religion has 2 classes - 'Scientology', 'Non-Scientology' where, 'Scientology' has been assigned the value 1 and 'Non-Scientology' has been assigned the value 0.

The same is done for Wife_ education, Husband_education, Wife_Working, Standard_of_living_index, Media_exposure.

- **Now, from sklearn.preprocessing we import scale and StandardScaler.**
- **From sklearn.discriminant_analysis we import LinearDiscriminantAnalysis.**
- **We fit both X and y to build the model.**
- **We build a confusion matrix with the help of the library confusion_matrix.**

  [[309, 301],

  [154, 622]])



The following is the classification report.

	precision	recall	f1-score	support
No	0.67	0.51	0.58	610
Yes	0.67	0.80	0.73	776
accuracy			0.67	1386
macro avg	0.67	0.65	0.65	1386
weighted avg	0.67	0.67	0.66	1386

- *Model.intercept_ helps us find the intercept value.*
- *The final LDA equation is*

Contraceptive_method_used=0.280+ X1*1.137 + X2*(-0.463) + X3*(0.833) + X4*(-1.083) + X5*3.803 + X6*(-0.531) + X7*0.361 + X8*5.959 + X9*0.144 + X10*2.30.

- **So from the above equation the following things can be summarized as**

-The coefficient of the X8 predictor is largest in magnitude thus it helps in discriminating the target the best.

-The coefficient of the X4 predictor is smallest in magnitude thus it helps in discriminating the target the least.

-All the DS can be computed for each row using the above f(x) which will aid in classification.

The predictive analysis -

1386 rows classified as 1 (Contraceptive method used)

0 rows classified as 0 (Contraceptive method not used)

923 rows classified as 1 (Contraceptive method used)

463 rows classified as 0 (Contraceptive method not used)

CART -

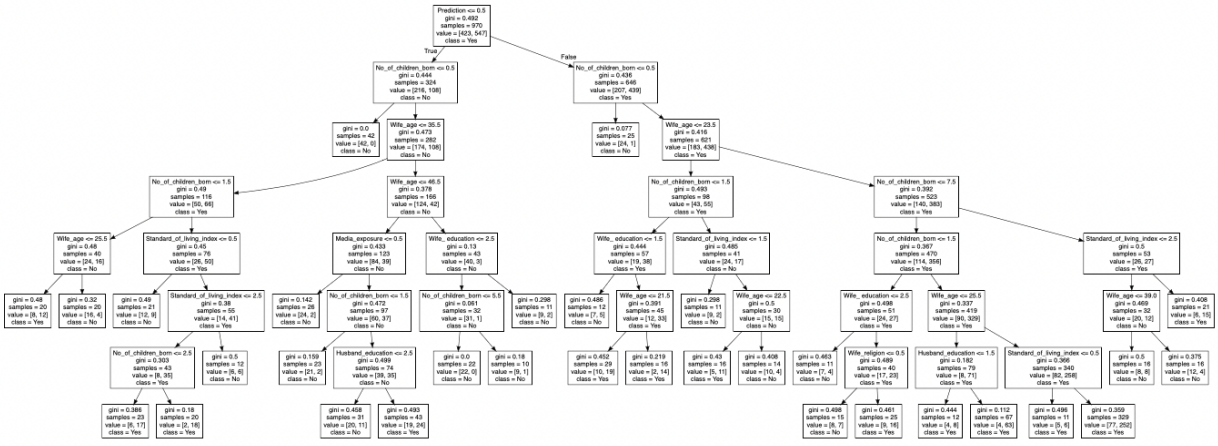
- *For CART we will import DecisionTreeClassifier from sklearn.tree*
- *Then we will classify X and y,*

  X = data.drop("Contraceptive_method_used" , axis=1)

  y = data.pop("Contraceptive_method_used")

- *The following is the trimmed decision tree, with the following restrictions -*

  criterion = 'gini', max_depth = 7,min_samples_leaf=10,min_samples_split=30

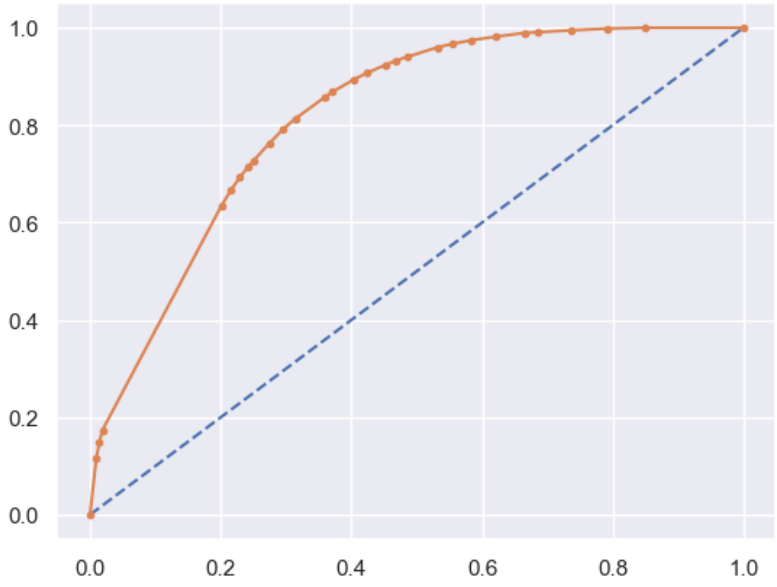


- **The following are the important variables' scores -**

	Imp
Wife_age	0.289241
Wife_ education	0.074335
Husband_education	0.056992
No_of_children_born	0.196237
Wife_religion	0.025790
Wife_Working	0.039673
Husband_Occupation	0.096939
Standard_of_living_index	0.095202
Media_exposure	0.012817
Prediction	0.112773

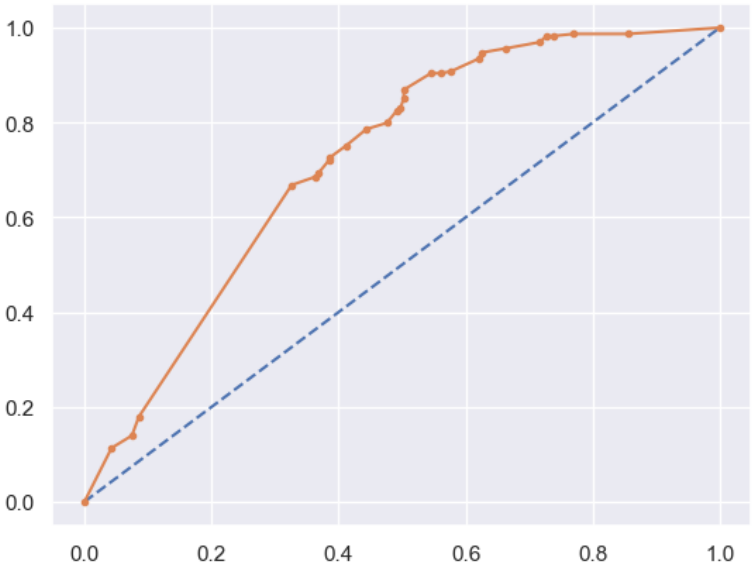
- **AUC and ROC curve for the train data -**

AUC: 0.818



- **AUC and ROC curve for the test data -**

  AUC: 0.721



2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model Final Model: Compare Both the models and write inference which model is best/optimized.

- *The confusion matrix for the training data -*

array([[266, 157],

[72, 475]])

- *The confusion matrix for the testing data -*

array([[94, 93],

[39, 190]])

- Finally we calculate the accuracy score for the models built for both the training data and testing data -

  Model accuracy score for training data - 0.7639175257731958
  Model accuracy score for Testing data - 0.6826923076923077

  - **The model built here is neither underfit nor overfit, hence we consider it as a valid model.**

2.4 Inference

By performing Logistic Regression, LDA and CART, we conclude that usage of contraceptive methods are influenced by their demographic and socio-economic characteristics.