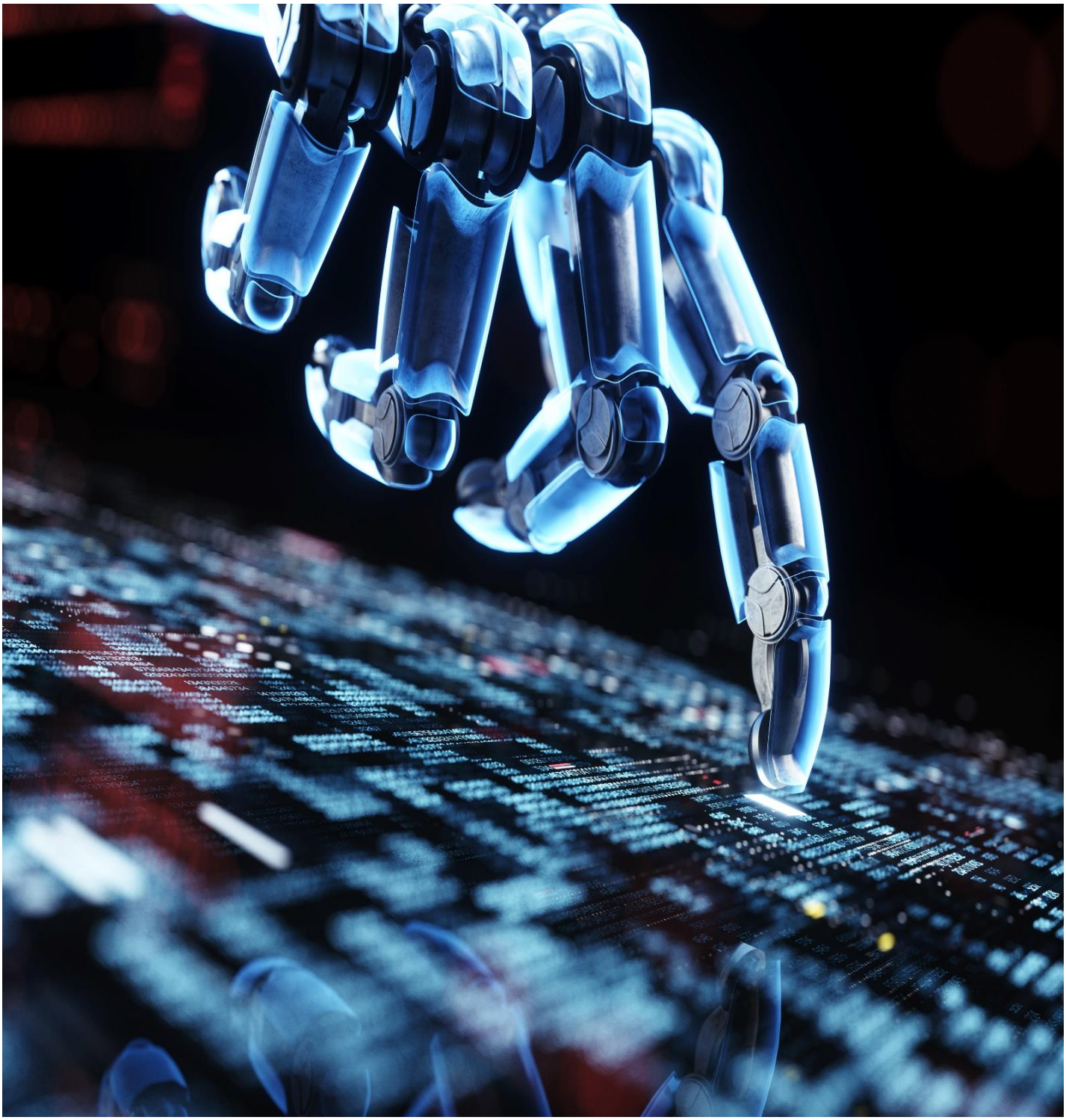
Predictive Modelling

Student Name: Suneel Kumar Pentapalli

Date: 10/July/2023



Contents

Linear Regression

Problem 1: Linear Regression

The comp-activ databases is a collection of a computer systems activity measures . The data was collected from a Sun Sparc station 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.

Dataset for Problem 1: compactiv.xlsx

DATA DICTIONARY:

System measures used:

- lread - Reads (transfers per second) between system memory and user memory
- lwrite - writes (transfers per second) between system memory and user memory
- scall - Number of system calls of all types per second
- sread - Number of system read calls per second .
- swrite - Number of system write calls per second .
- fork - Number of system fork calls per second.
- exec - Number of system exec calls per second.
- rchar - Number of characters transferred per second by system read calls
- wchar - Number of characters transfreed per second by system write calls
- pgout - Number of page out requests per second
- ppgout - Number of pages, paged out per second
- pgfree - Number of pages per second placed on the free list.
- pgscan - Number of pages checked if they can be freed per second
- atch - Number of page attaches (satisfying a page fault by reclaiming a page in memory) per second
- pgin - Number of page-in requests per second
- ppgin - Number of pages paged in per second
- pflt - Number of page faults caused by protection errors (copy-on-writes).
- vflt - Number of page faults caused by address translation .
- runqsz - Process run queue size (The number of kernel threads in memory that are waiting for a CPU to run.

Typically, this value should be less than 2. Consistently higher values mean that the system might be CPU-bound.)

- freemem - Number of memory pages available to user processes
- freeswap - Number of disk blocks available for page swapping.

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

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

The given dataset "compactiv.xlsx" has

- 8192 records and 22 columns.
- 1.2% and 0.18% Nulls in rchar and wchar respectively.
- No duplicate records.
- Of all 22 columns, only runqsz is of object type and rest all are of numeric type.

Below is the sample data,

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	ppgout	pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
0	1	0	2147	79	68	0.2	0.2	40671.0	53995.0	0.0	0.0	0.0	0.0	0.0	1.6	2.6	16.00	26.40	CPU_Bound	4670	1730946	95
1	0	0	170	18	21	0.2	0.2	448.0	8385.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.63	16.83	Not_CPU_Bound	7278	1869002	97
2	15	3	2162	159	119	2.0	2.4	NaN	31950.0	0.0	0.0	0.0	0.0	1.2	6.0	9.4	150.20	220.20	Not_CPU_Bound	702	1021237	87
3	0	0	160	12	16	0.2	0.2	NaN	8670.0	0.0	0.0	0.0	0.0	0.0	0.2	0.2	15.60	16.80	Not_CPU_Bound	7248	1863704	98
4	5	1	330	39	38	0.4	0.4	NaN	12185.0	0.0	0.0	0.0	0.0	0.0	1.0	1.2	37.80	47.60	Not_CPU_Bound	633	1760253	90

Data types and column names:

```
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
```

5 Point summary of compactiv.xlsx:

	count	mean	std	min	25%	50%	75%	max
lread	8192.0	19.560	53.354	0.0	2.0	7.0	20.000	1845.00
lwrite	8192.0	13.106	29.892	0.0	0.0	1.0	10.000	575.00
scall	8192.0	2306.318	1633.617	109.0	1012.0	2051.5	3317.250	12493.00
sread	8192.0	210.480	198.980	6.0	86.0	166.0	279.000	5318.00
swrite	8192.0	150.058	160.479	7.0	63.0	117.0	185.000	5456.00
fork	8192.0	1.885	2.479	0.0	0.4	0.8	2.200	20.12
exec	8192.0	2.792	5.212	0.0	0.2	1.2	2.800	59.56
rchar	8088.0	197385.728	239837.494	278.0	34091.5	125473.5	267828.750	2526649.00
wchar	8177.0	95902.993	140841.708	1498.0	22916.0	46619.0	106101.000	1801623.00
pgout	8192.0	2.285	5.307	0.0	0.0	0.0	2.400	81.44
ppgout	8192.0	5.977	15.215	0.0	0.0	0.0	4.200	184.20
pgfree	8192.0	11.920	32.364	0.0	0.0	0.0	5.000	523.00
pgscan	8192.0	21.527	71.141	0.0	0.0	0.0	0.000	1237.00
atch	8192.0	1.128	5.708	0.0	0.0	0.0	0.600	211.58
pgin	8192.0	8.278	13.875	0.0	0.6	2.8	9.765	141.20
ppgin	8192.0	12.389	22.281	0.0	0.6	3.8	13.800	292.61
pflt	8192.0	109.794	114.419	0.0	25.0	63.8	159.600	899.80
vflt	8192.0	185.316	191.001	0.2	45.4	120.4	251.800	1365.00
freemem	8192.0	1763.456	2482.105	55.0	231.0	579.0	2002.250	12027.00
freeswap	8192.0	1328125.960	422019.427	2.0	1042623.5	1289289.5	1730379.500	2243187.00
usr	8192.0	83.969	18.402	0.0	81.0	89.0	94.000	99.00

As per above summary,

1. values in each columns seems to be of different scales.
2. 13 attributes have values starting with 0.
3. there is considerable difference between mean and median for almost all the attributes which indicates skewness in data.

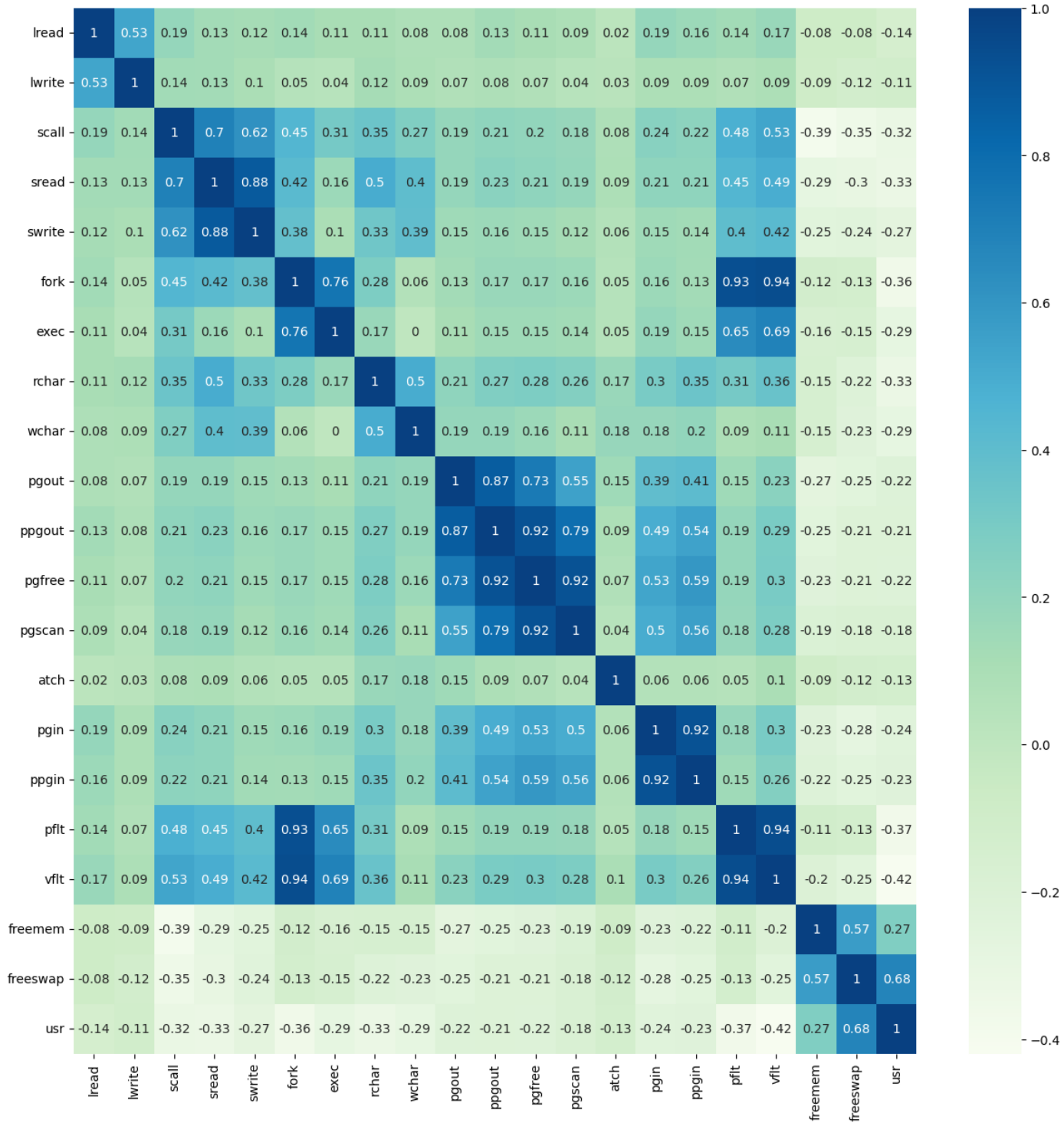
Correlation Plot:

From the correlation plot, we can see that,

There is high amount of positive correlation between fork-pfit, fork-vfit, vfit-pfit, pgin-ppgin, pgscan-pgfree, ppgout,pgfree, sread-swrite etc.,

There is high amount of negative correlation between usr-vfit,usr-pfit etc.,

Correlation values near to 1 or -1 are highly positively correlated and highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.



Univariate Analysis:

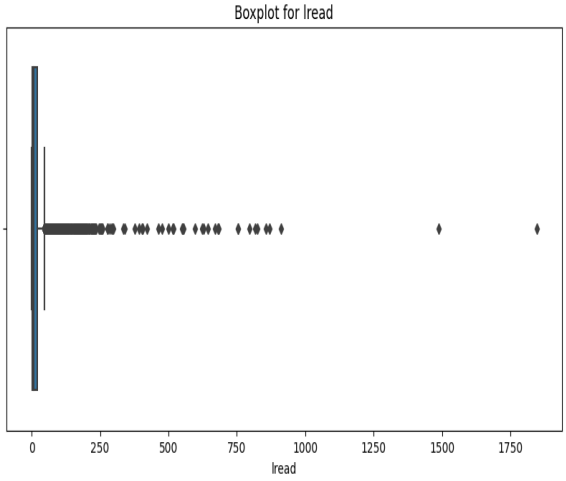
Univariate analysis is the simplest form of analyzing data. "Uni" means "one", so in other words your data has only one variable. It doesn't deal with causes or relationships and it's major purpose is to describe; It takes data, summarizes that data and finds patterns in the data.

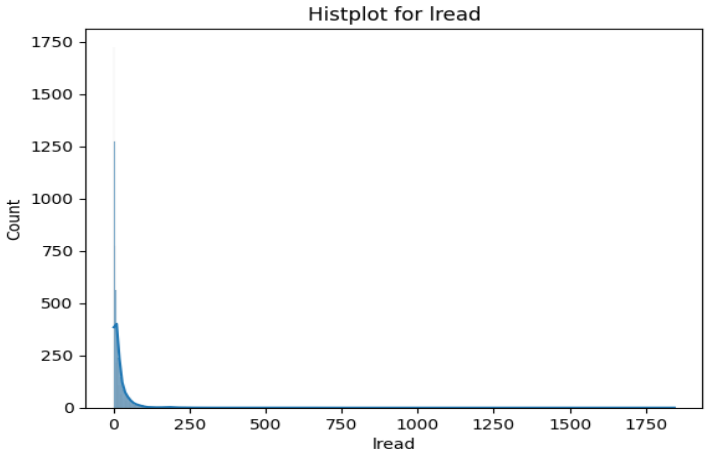
Below is the visual representation of univariate analysis of all the numerical attributes,

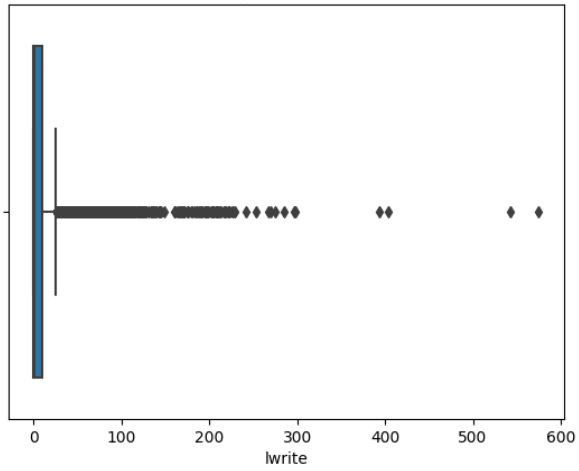
Graph on the left side is a box plot, the ponts in the box plot represents outliers.

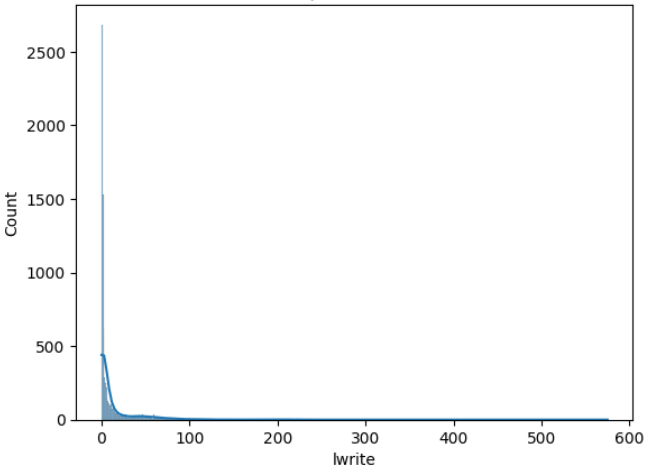
Similarly graph to the right shows a histplot for visualizing distribution of data in the corresponding column. Higher Skewness in this graph represents outliers.

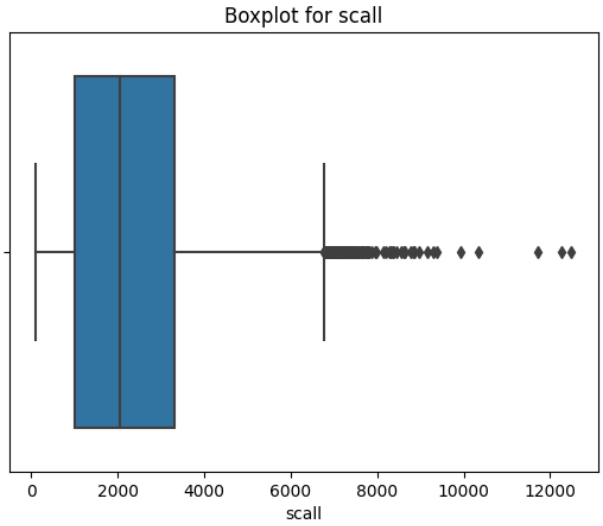
As per both the graphs, it is evident that most of the attributes are highly skewed. Also distribution of data indicates that mode lies near lower numbers.

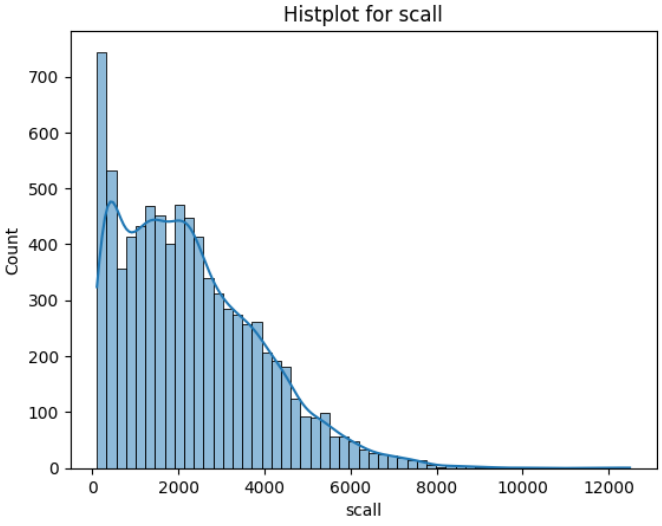


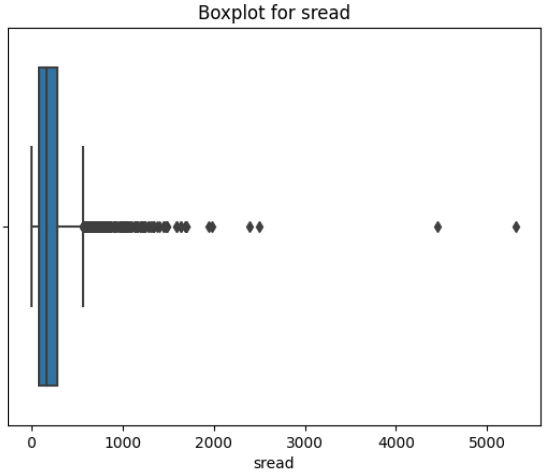


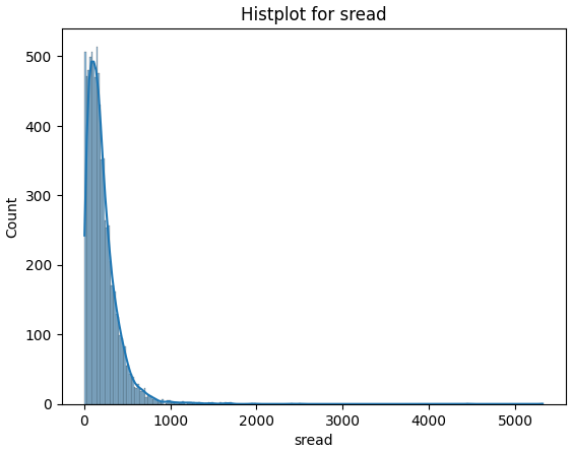


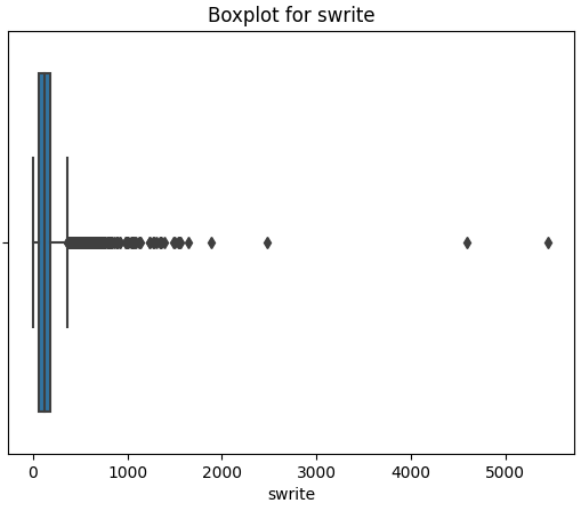


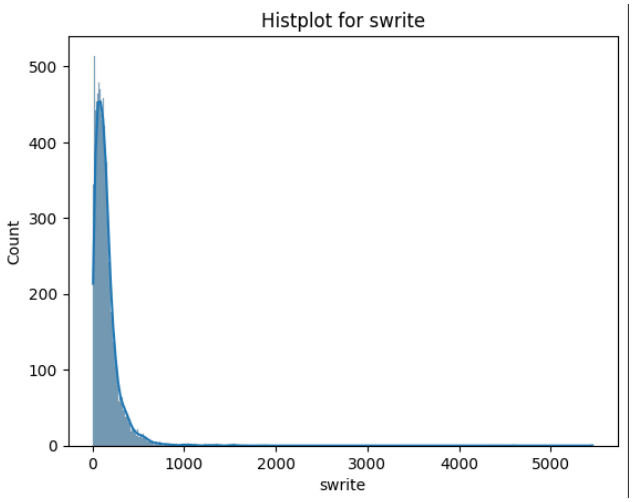


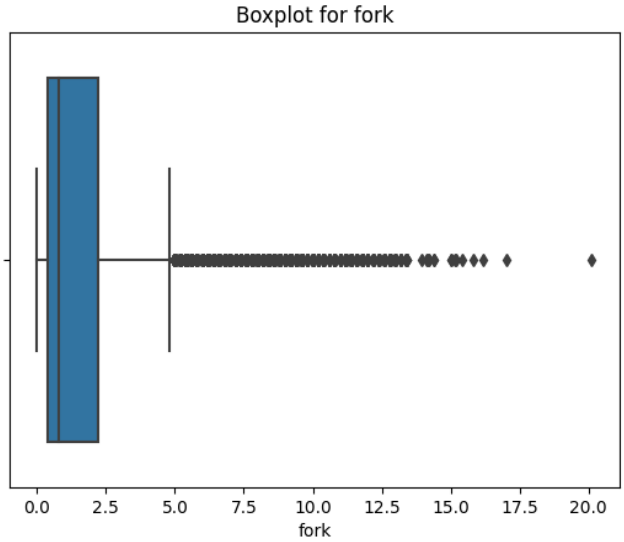


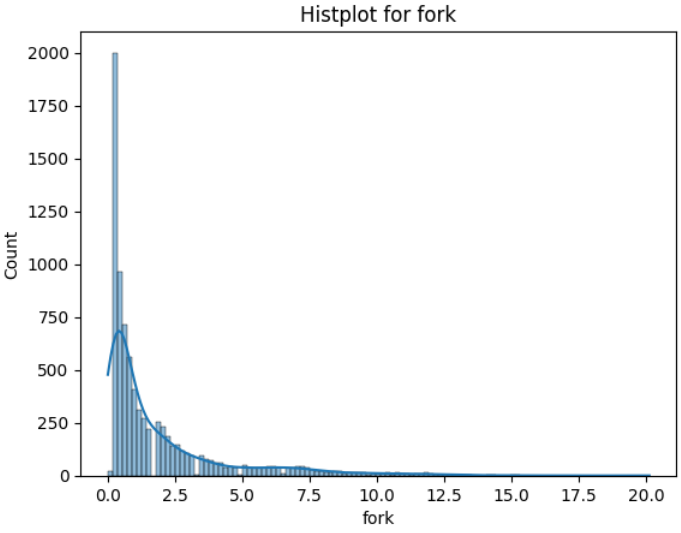

Boxplot for sread
0
1000
2000
3000
4000
5000
sread

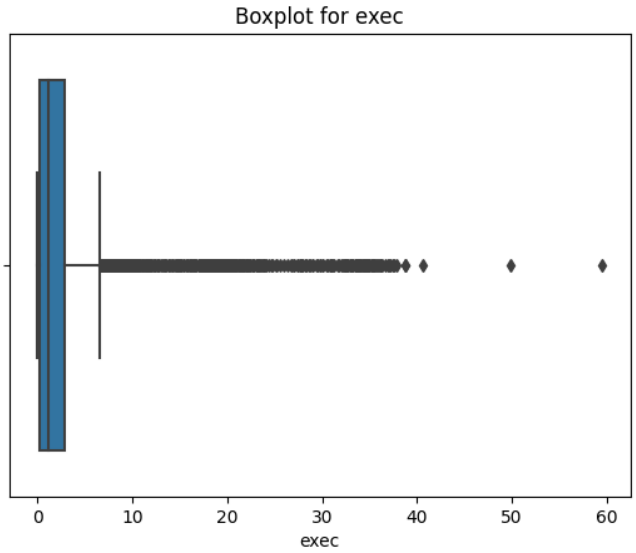

Histplot for sread
Count
0
100
200
300
400
500
0
1000
2000
3000
4000
5000
sread

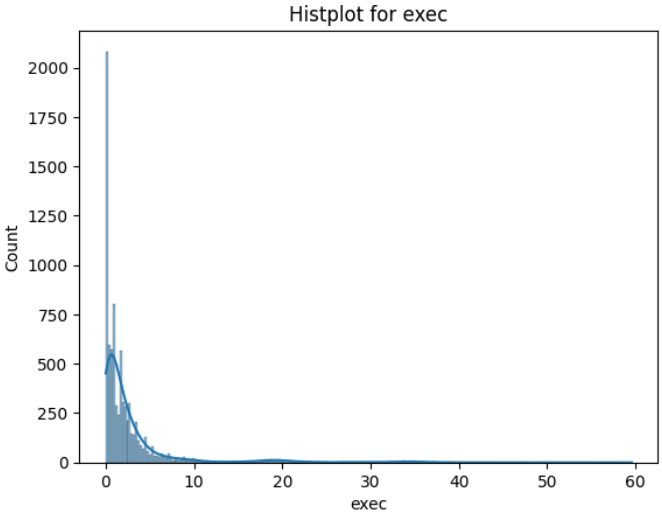

Boxplot for swrite
0
1000
2000
3000
4000
5000
swrite

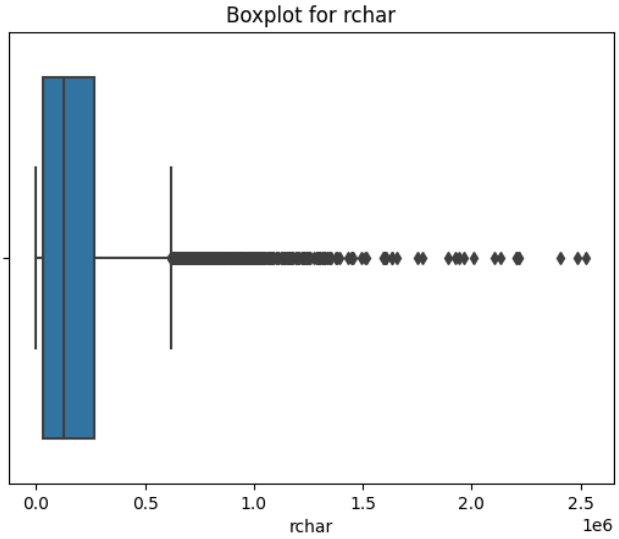

Histplot for swrite
Count
0
100
200
300
400
500
0
1000
2000
3000
4000
5000
swrite

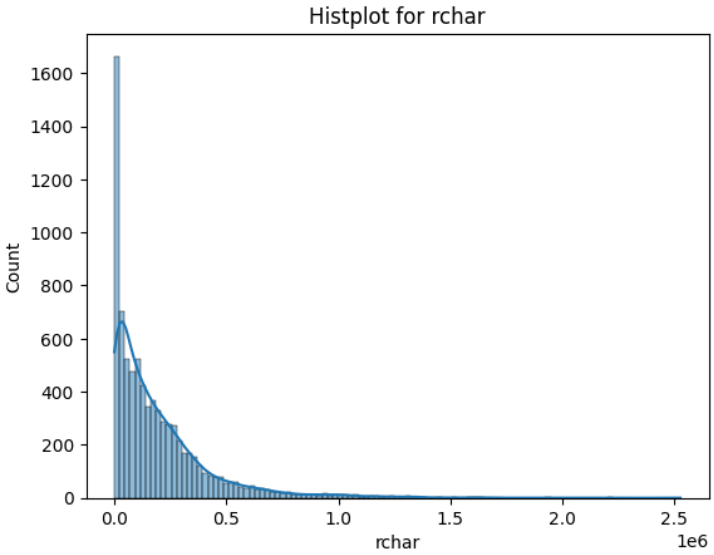

Boxplot for fork
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
fork

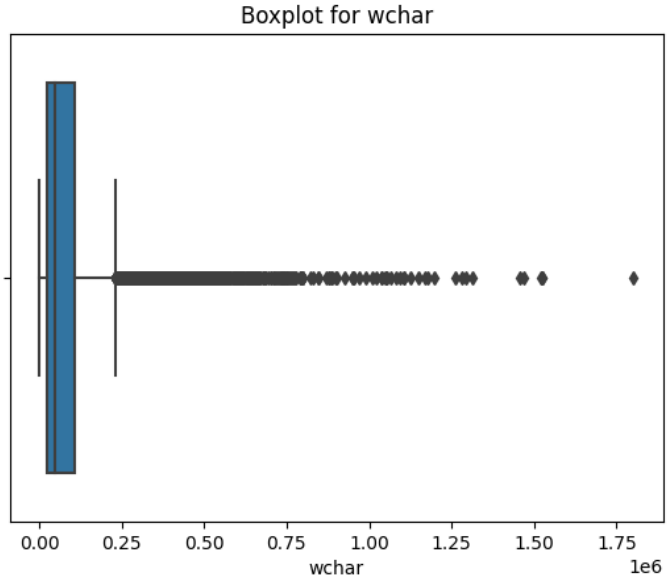

Histplot for fork
Count
0
250
500
750
1000
1250
1500
1750
2000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
fork

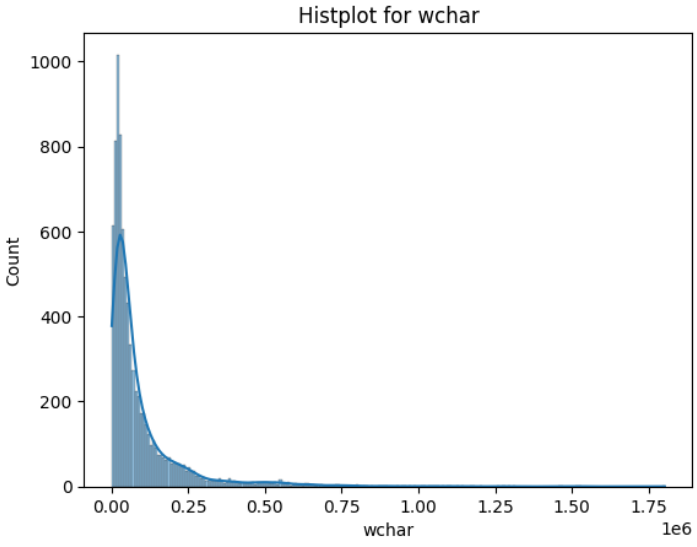

Boxplot for exec
0
10
20
30
40
50
60
exec

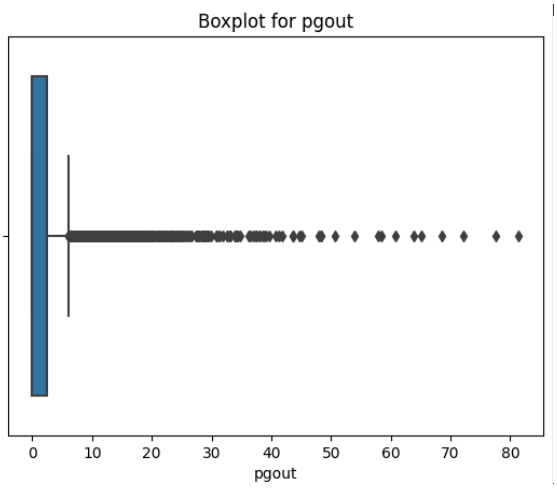

Histplot for exec
2000
1750
1500
1250
1000
750
500
250
0
Count
0
10
20
30
40
50
60
exec

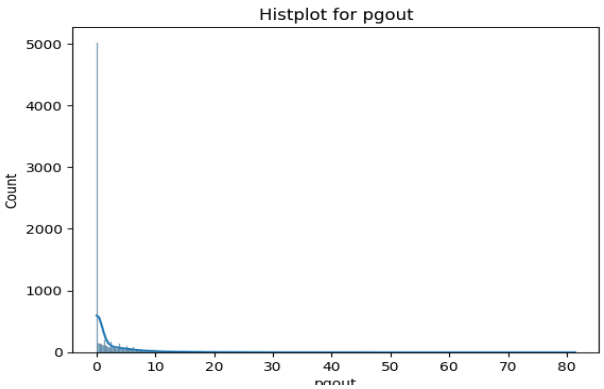

Boxplot for rchar
0.0
0.5
1.0
1.5
2.0
2.5
1e6
rchar

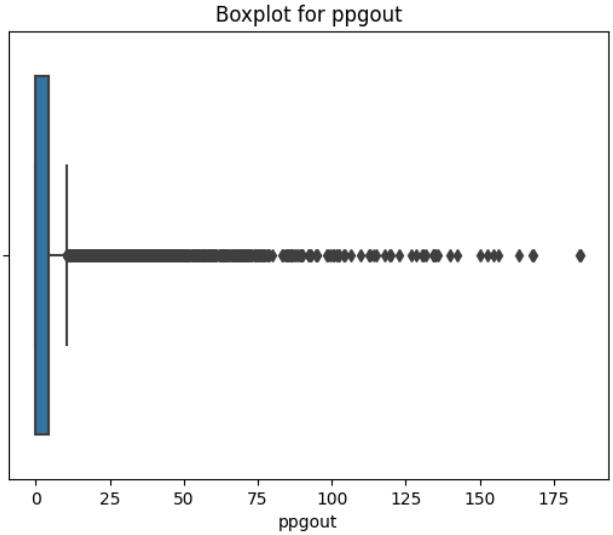

Histplot for rchar
1600
1400
1200
1000
800
600
400
200
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
1e6
rchar

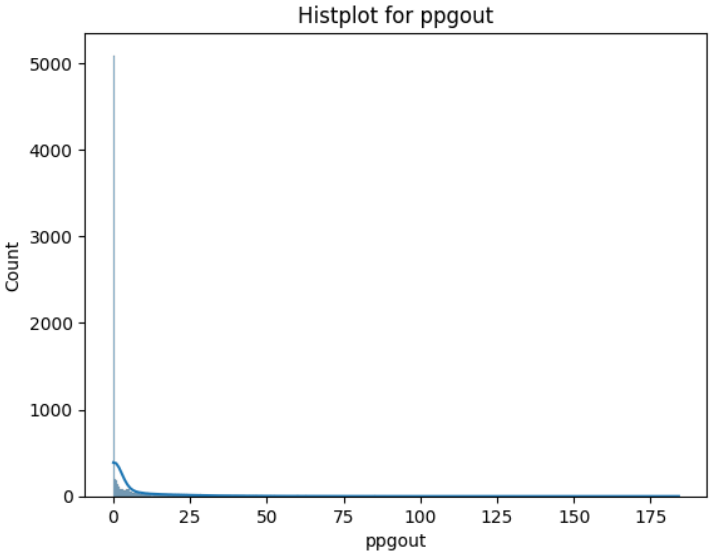

Boxplot for wchar
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
1e6
wchar

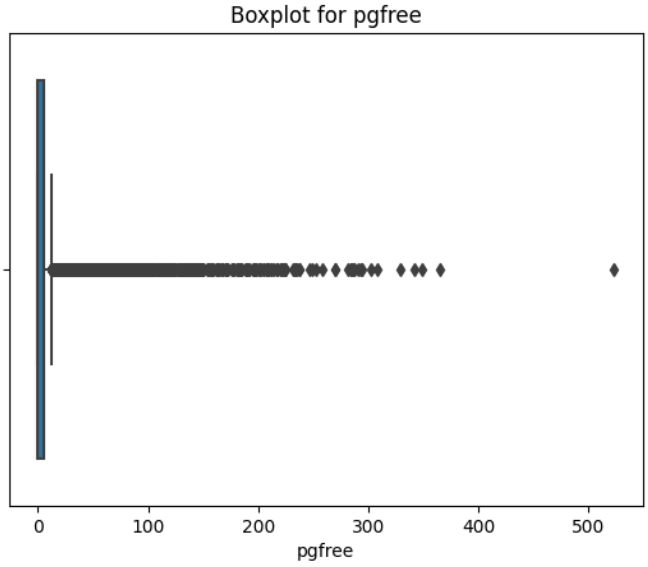

Histplot for wchar
1000
800
600
400
200
0
Count
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
1e6
wchar

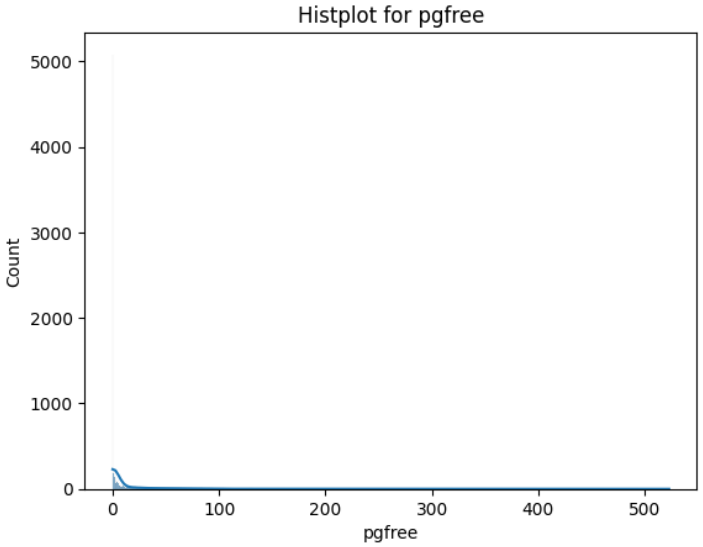

Boxplot for pgout
0
10
20
30
40
50
60
70
80
pgout

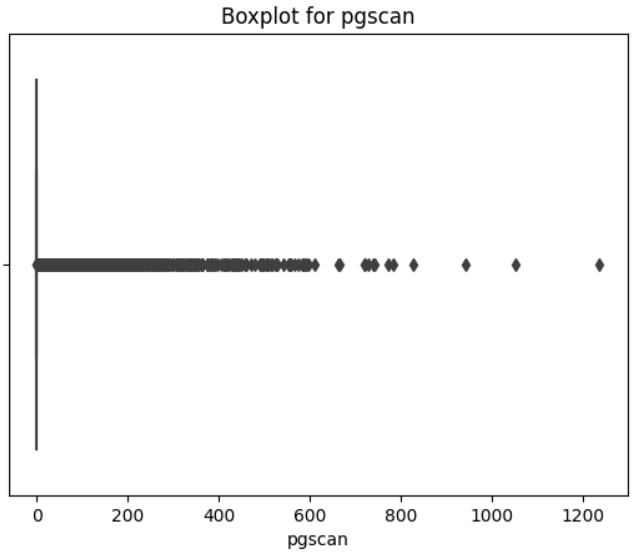

Histplot for pgout
5000
4000
3000
2000
1000
0
Count
0
10
20
30
40
50
60
70
80

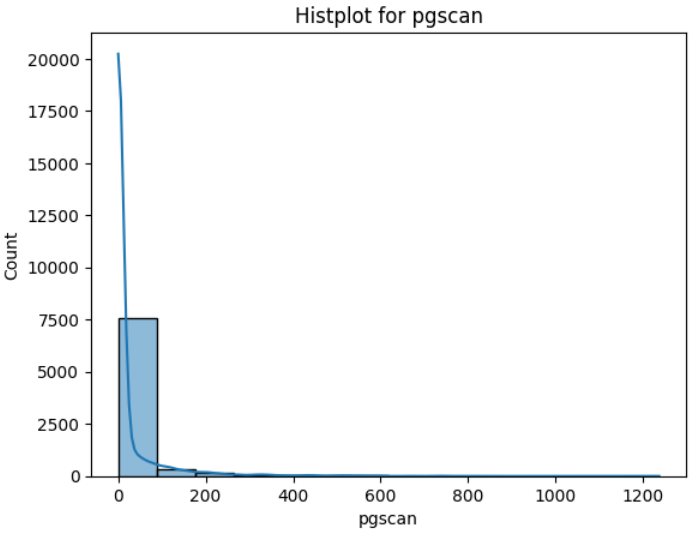

Boxplot for ppgout
0
25
50
75
100
125
150
175
ppgout

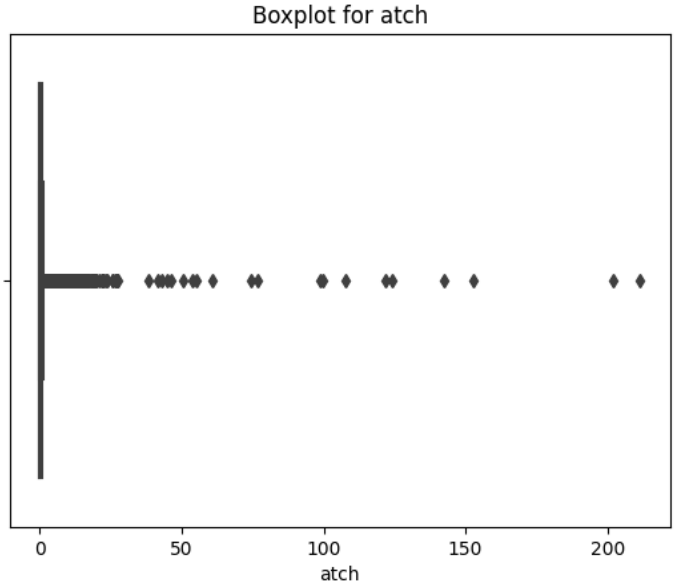

Histplot for ppgout
5000
4000
3000
2000
1000
0
Count
0
25
50
75
100
125
150
175
ppgout

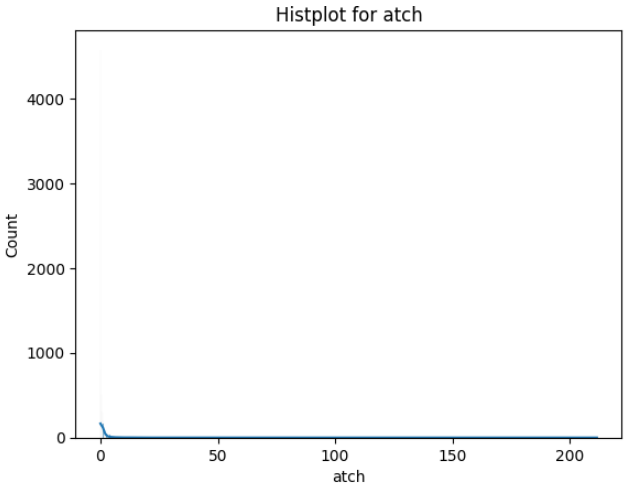

Boxplot for pgfree
0
100
200
300
400
500
pgfree

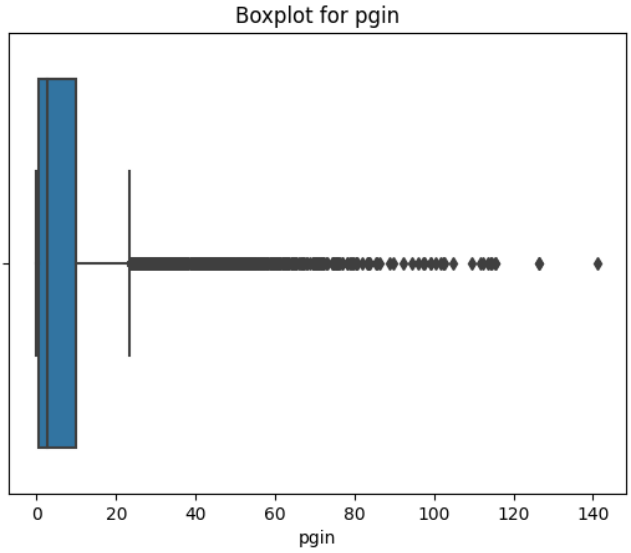

Histplot for pgfree
5000
4000
3000
2000
1000
0
Count
0
100
200
300
400
500
pgfree

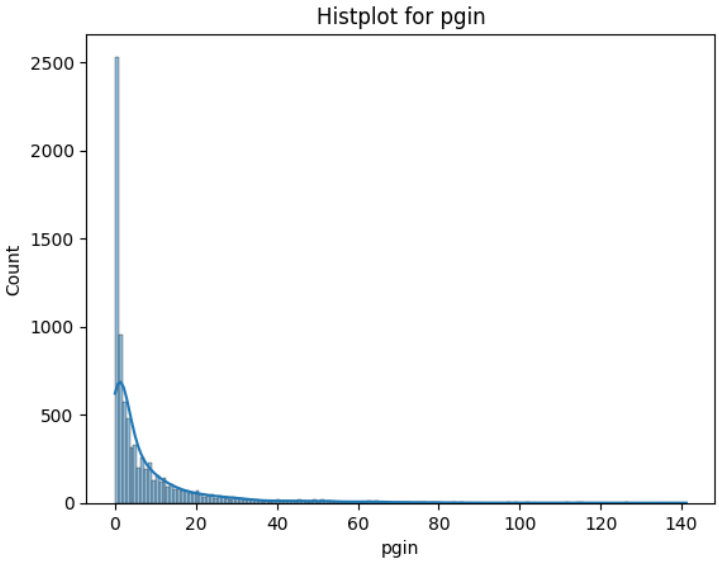

Boxplot for pgscan
0
200
400
600
800
1000
1200
pgscan

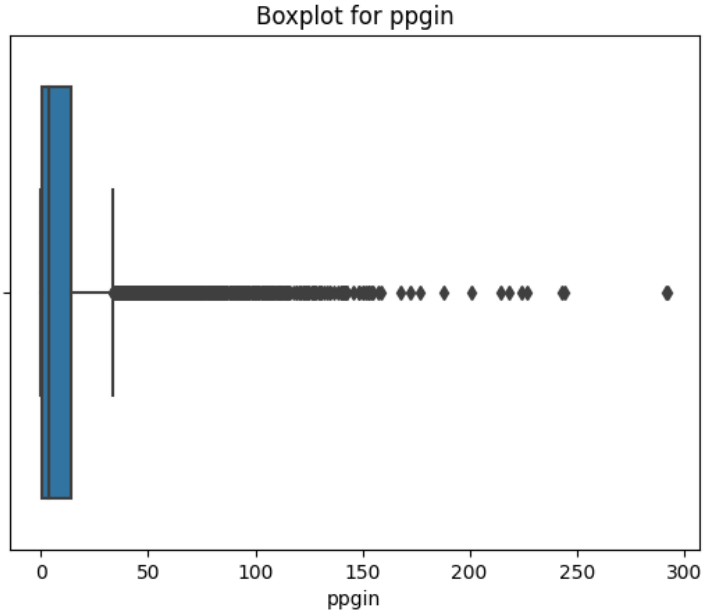

Histplot for pgscan
20000
17500
15000
12500
10000
7500
5000
2500
0
Count
0
200
400
600
800
1000
1200
pgscan

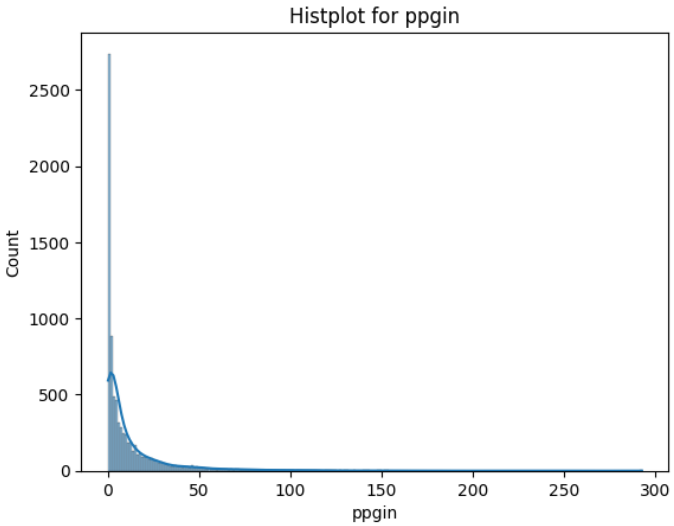

Boxplot for atch
0
50
100
150
200
atch

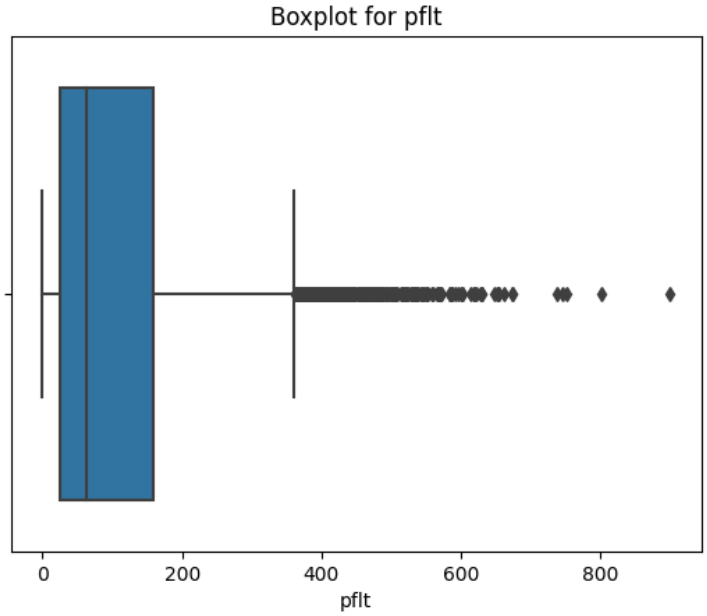

Histplot for atch
4000
3000
2000
1000
0
Count
0
50
100
150
200
atch

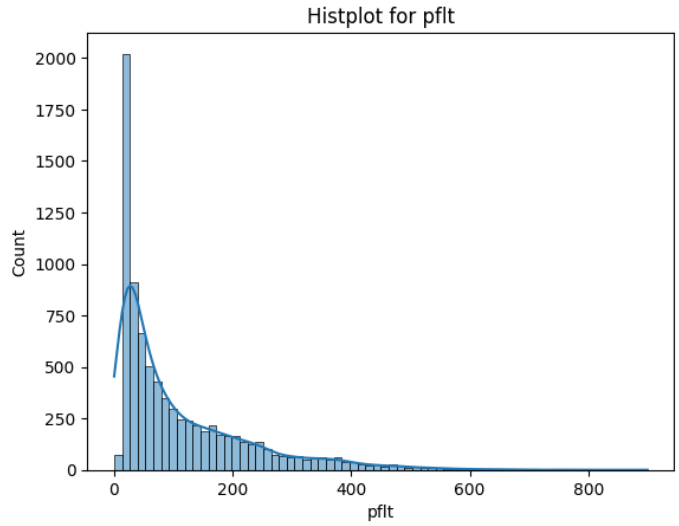

Boxplot for pgin
0
20
40
60
80
100
120
140
pgin

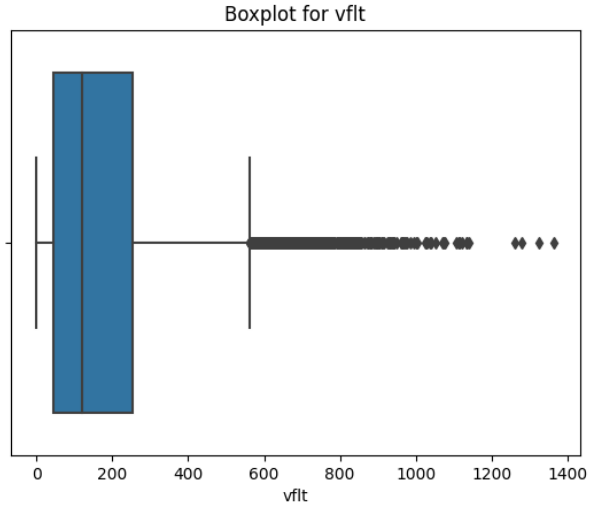

Histplot for pgin
2500
2000
1500
1000
500
0
Count
0
20
40
60
80
100
120
140
pgin

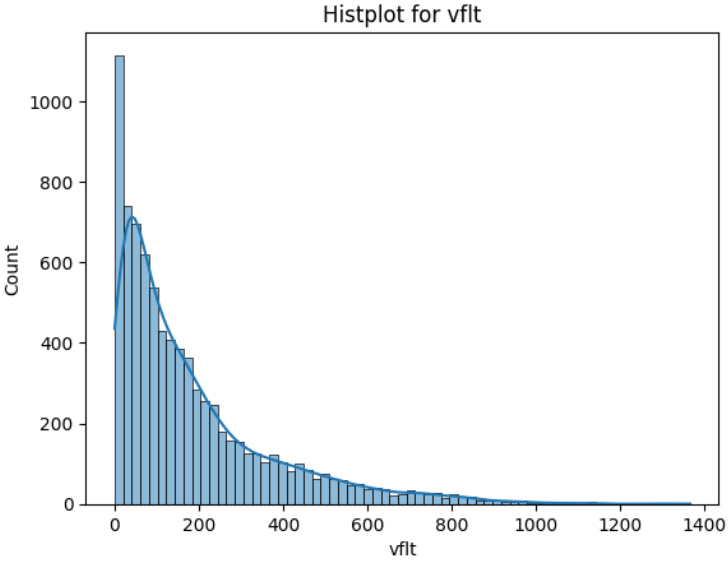

Boxplot for ppgin
0
50
100
150
200
250
300
ppgin

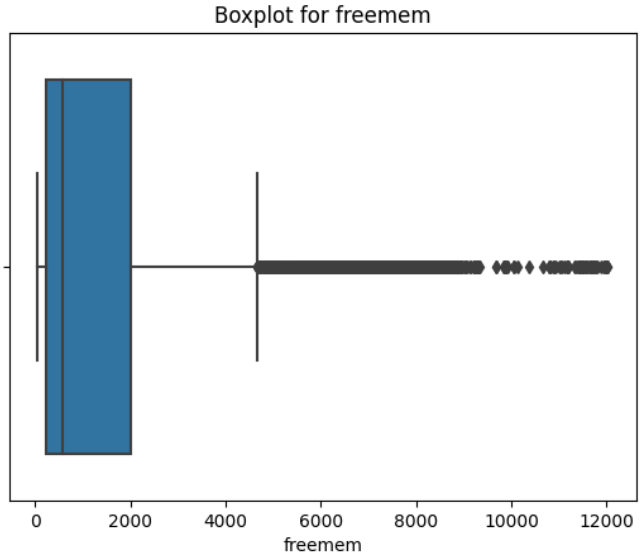

Histplot for ppgin
Count
0
500
1000
1500
2000
2500
0
50
100
150
200
250
300
ppgin

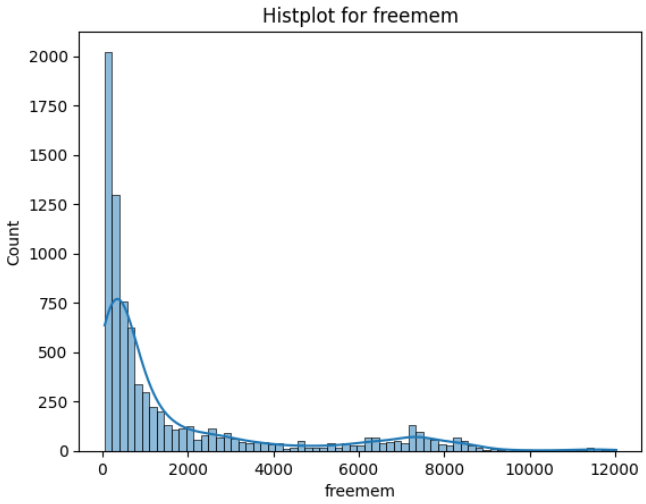

Boxplot for pflt
0
200
400
600
800
pflt

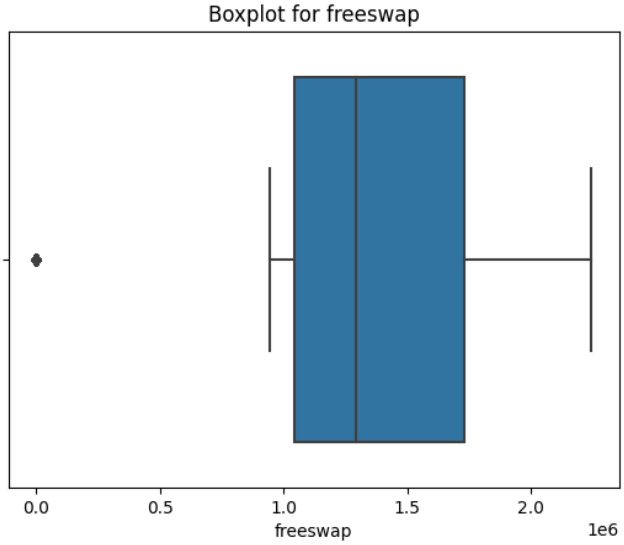

Histplot for pflt
Count
0
250
500
750
1000
1250
1500
1750
2000
0
200
400
600
800
pflt

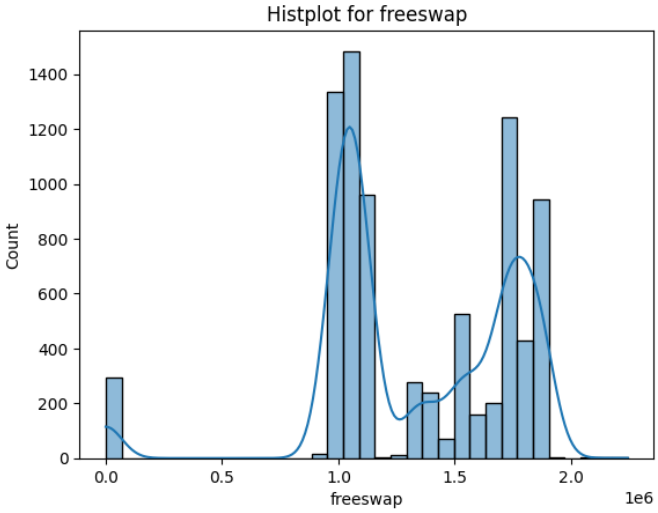

Boxplot for vflt
0
200
400
600
800
1000
1200
1400
vflt

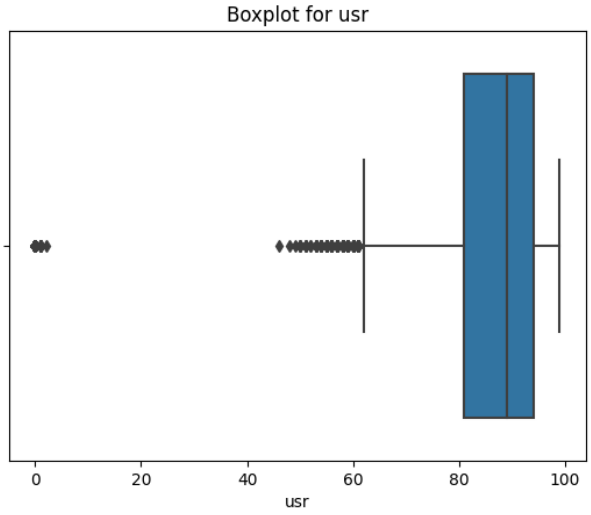

Histplot for vflt
Count
0
200
400
600
800
1000
0
200
400
600
800
1000
1200
1400
vflt

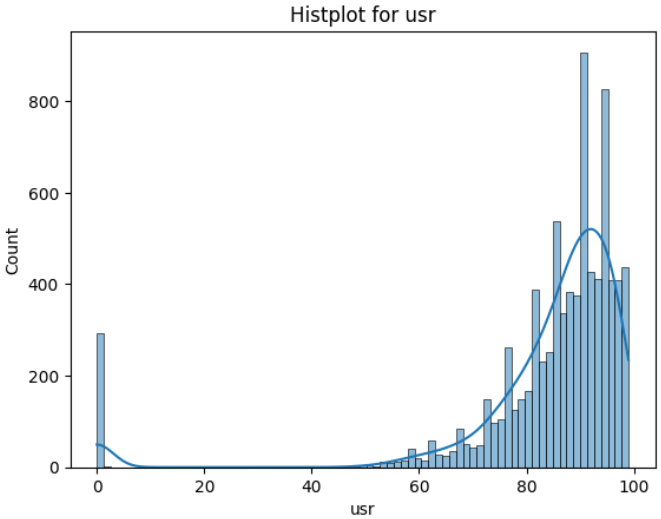










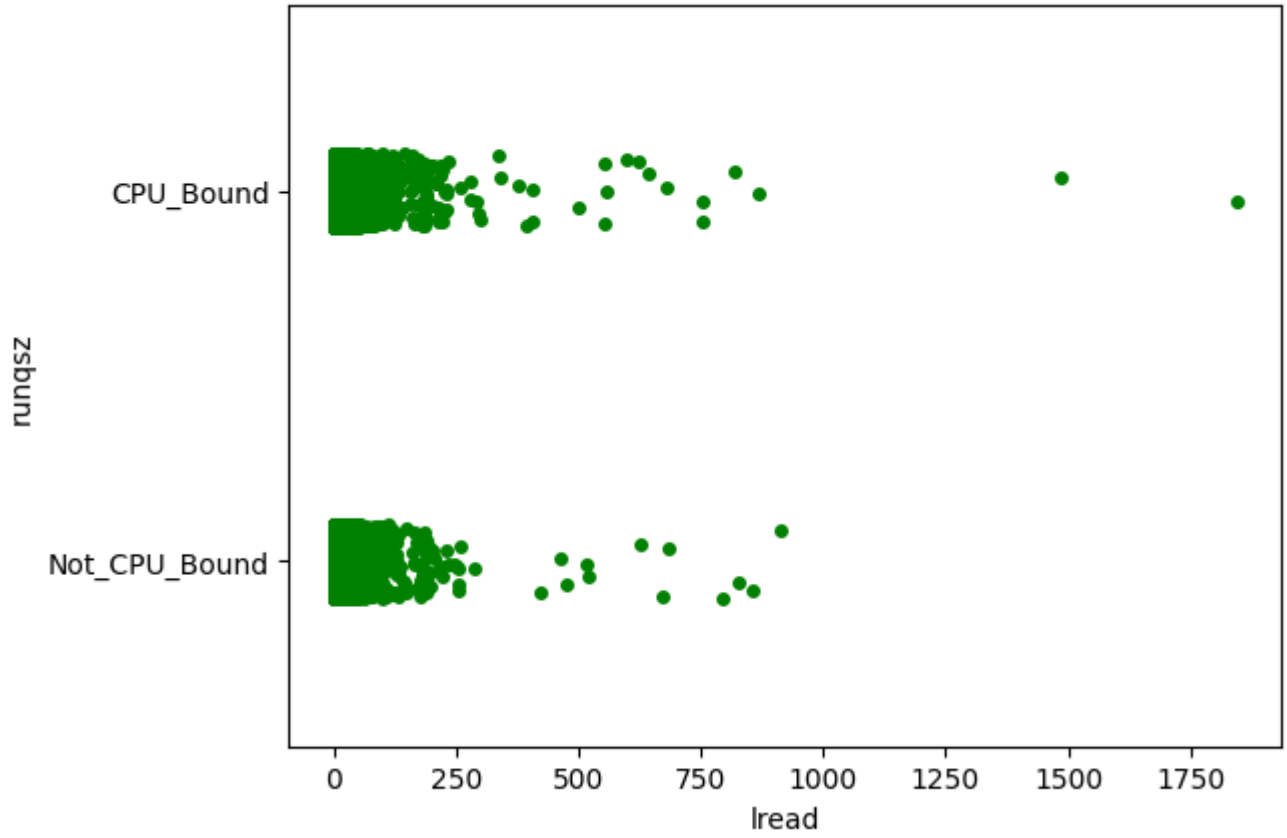


Bivariate Analysis:

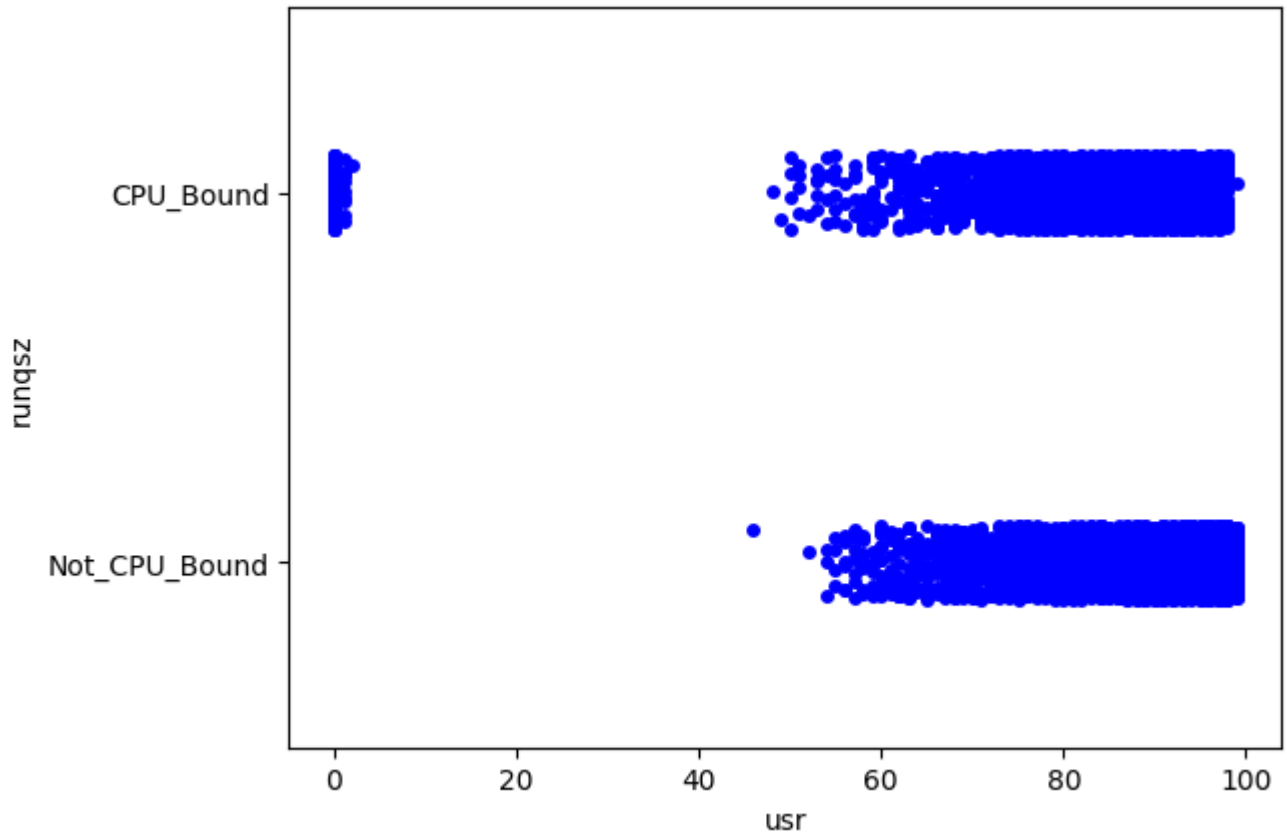
In bivariate analysis we study distribution and patterns between two variables as shown below,

Here we have used stripplot to look at the datapoints against each runqsz type. Stripplot of lread against runqsz shows more lread(Reads (transfers per second) between

system memory and user memory) in CPU_Bound systems than in Not_CPU_Bound systems.



Stripplot between usr(Portion of time (%) that cpus run in user mode) and runqsz shows that portion of systems with CPU_Bound are staying idle.

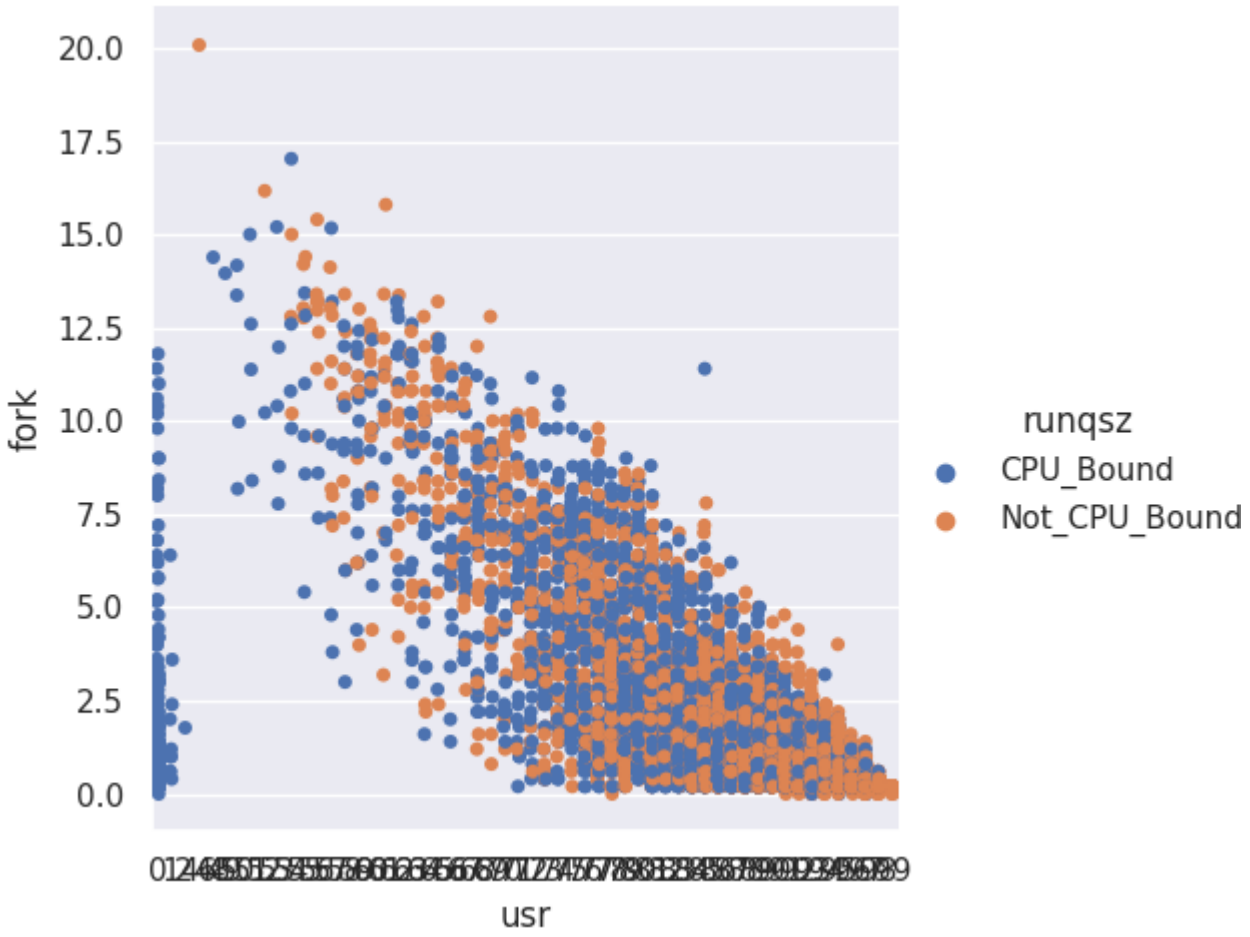


Multivariate Analysis:

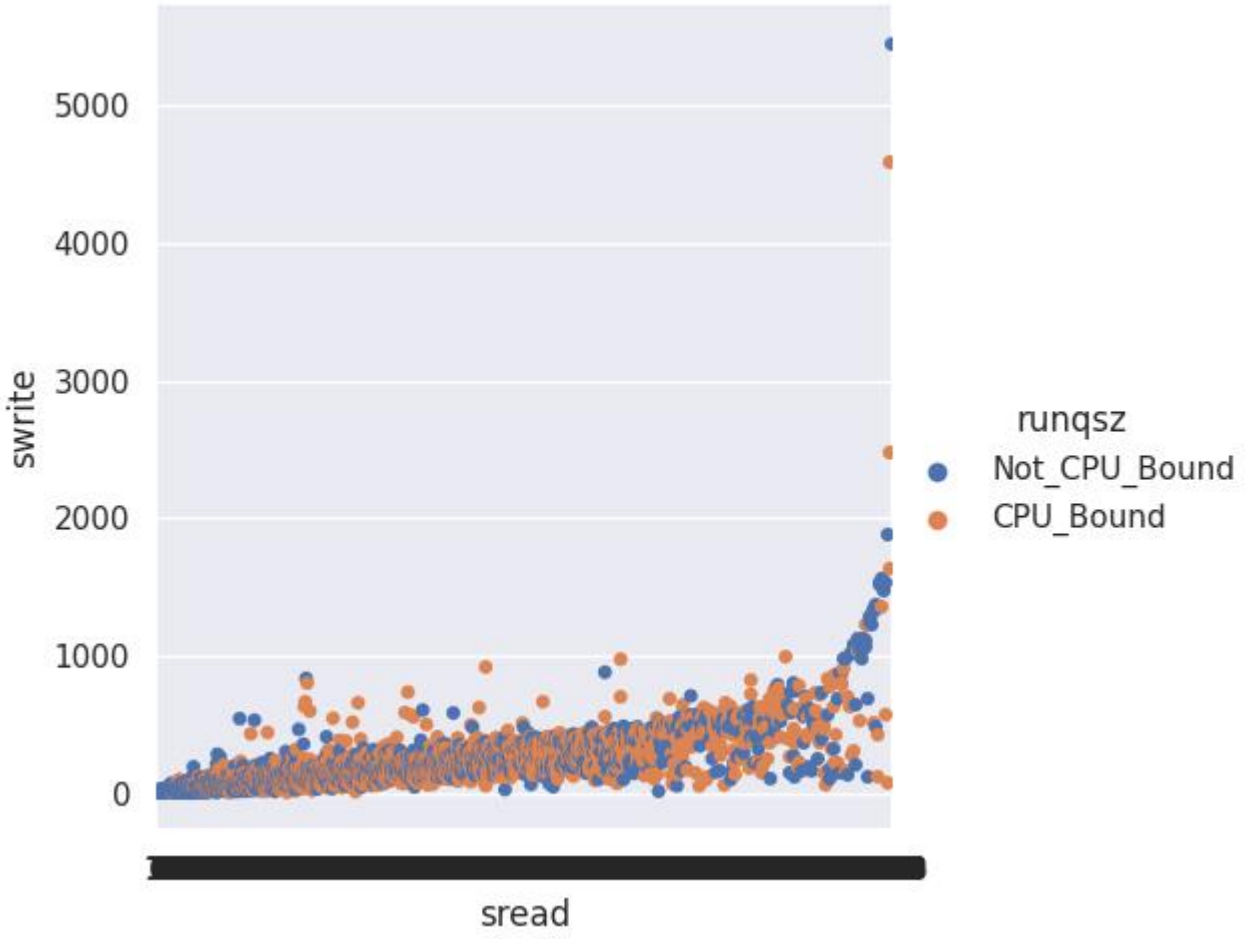
Multivariate analysis is a set of techniques to study data with more than one variable and their relationships

Below graph shows the correlation of fork and usr against runqsz.

fork and user are varying inversely due to their negative correlation.



attributes sread and lread are almost evenly distributed and also varying in same direction due to high correlation.



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.

Handling Null Values:

Fetching null percentages in null columns.

```
rchar    1.269531
wchar    0.183105
```

Null percentage in above columns is very less which corresponds to around 1.2% and 0.18% in rchar and wchar respectively.

Replaced all the null values with the corresponding attribute mean values.

```
print("Number of null columns after imputing with mean values: ",len(df.isnull().sum()[df.isnull().sum()>0]))
```

```
Number of null columns after imputing with mean values:  0
```

Checking for zeroes:

Below are the percentages of zeroes in each column,

```
lread        8.239746
lwrite      32.763672
scall        0.000000
sread        0.000000
swrite       0.000000
fork         0.256348
exec         0.256348
rchar        0.000000
wchar        0.000000
pgout       59.545898
ppgout      59.545898
pgfree      59.436035
pgscan      78.710938
atch        55.847168
pgin        14.892578
ppgin       14.892578
pflt         0.036621
vflt         0.000000
runqsz       0.000000
freemem      0.000000
freeswap     0.000000
usr          3.454590
dtype: float64
```

six attributes have zeroes more than 50%, but there were no rules available on how each attribute is related or calculated. Hence we are not changing/imputing those zeroes.

As already shown in boxplots in 1.1,

Outliers are present in all the attributes(lread, lwrite, scall, sread, swrite, fork, exec, rchar, wchar, pgout, ppgout, pgfree, pgscan, atch, pgin, ppgin, pfit, vfit, freemem, freeswap, usr)

Only freeswap columns has less outliers

But as these outliers seems to be actual values, we will proceed with further steps without treating them.

Duplicates:

As shown below the given dataset has no duplicates.

```
print("compactiv dataset has "+str(df.duplicated().sum())+" duplicates")
```

```
compactiv dataset has 0 duplicates
```

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.

Encoding String values in runqsz:

As seen above, the only object type column in the dataset is runqsz and it has values 'CPU_Bound', 'Not_CPU_Bound' only.

```
Not_CPU_Bound    4331
CPU_Bound        3861
```

Before building our linear regression model its better to change the object datatype to int by decoding the values to 0 and 1 as shown below.

Note: Here we have hardcoded value "Not_CPU_Bound" of type object to the value 1 of type int, similary modified "CPU_Bound" to 0 of int type.

```
1    4331
0    3861
```

Now we have out dataset full of numerical and categorical values.

Splitting the dataset in to test and train data:

Before splitting the data and test and train sets, lets divide our features into predictor/features and target splits as shown below.

```
x= df.drop('usr',axis=1)
y= df['usr']
print("Feature dataset has {0} rows and {1} columns".format(x.shape[0],x.shape[1]))
print("Shape of target dataset: ",y.shape)
```

```
Feature dataset has 8192 rows and 21 columns
Shape of target dataset:  (8192,)
```

Now that we have created two datasets with all predictors in one dataframe(x) and target variable(usr) in one data frame(y)

Now we will split x and y into x_train,x_test and y_train and y_test dataframes with test sample size of 30% and train sample size of 70%

```
Shape of training predictor dataset(x_train):  (5734, 21)
Shape of test predictor dataset(x_test):  (2458, 21)
Shape of training target dataset(y_train):  (5734,)
Shape of test target dataset(y_test):  (2458,)
```

Now we have created a model and fitted the training values in the model.

Here are some of the metrics of model on training and test datasets,

R^2 Value: The most common interpretation of r-squared is how well the regression model explains observed data. For example, an r-squared of 60% reveals that 60% of the variability observed in the target variable is explained by the regression model. Higher the r-squared value, better the model is. This means, our model is almost 64% accurate.

The coefficient of determination R^2 of the prediction on Train set 0.6363661657685671

The coefficient of determination R^2 of the prediction on Test set 0.6468867961739946

RMSE: Root mean square error shows us the deviation of predicted values from actual values. Lesser the deviation, better the model.

The Root Mean Square Error (RMSE) of the model for training set is: 10.991906841867554

The Root Mean Square Error (RMSE) of the model for testing set is: 11.168605637051579

Adjusted R^2 Value: Adjusted R-squared is a modified version of R-squared that adjusts for the number of predictors in a regression model. R-squared measures the proportion of the variation in the dependent variable explained by the independent variables for a linear regression model. Adjusted R-squared compares the explanatory power of regression models that contain different numbers of predictors. It increases only if the new term improves the model more than would be expected by chance.

Adjusted R square of the model for training set is: 0.6350292766721279

Adjusted R square of the model for testing set is: 0.643842716830667

As shown above, our model is only 64% accurate, lets explore more and find out if it can be improved,

Variance Inflation Factor(VIF):

This metric shows us the multicollinearity in predictors.

Generally, a VIF above 3 indicates that multicollinearity might exist, and further investigation is required. When VIF is higher than 10, there is significant multicollinearity that needs to be corrected.

vflt	34.701887
fork	27.240158
pflt	21.633703
pgfree	20.026026
ppgout	16.974238
sread	14.547175
ppgin	10.655991
pgin	10.368163
swrite	10.124971
pgscan	8.295372
scall	6.876673
pgout	6.391462
freeswap	5.505457
exec	3.913269
rchar	3.319048
freemem	2.454773
wchar	2.286181
runqsz	2.080664
lwrite	1.699056
lread	1.690906
atch	1.132194

Upon verifying by recreating the model removing each column at a time from the above list didn't really increase our model accuracy.

Hence lets proceed further and check for the feature importance by checking the pvalue of each attribute and remove the variables that are not adding any value to the model.

Used ols method to create model and fetch the below data,

	coef	std err	t	P>|t|	[0.025	0.975]
const	43.4974	0.744	58.447	0.000	42.038	44.956
lread	-0.0218	0.003	-7.095	0.000	-0.028	-0.016
lwrite	0.0068	0.006	1.207	0.228	-0.004	0.018
scall	0.0011	0.000	7.760	0.000	0.001	0.001
sread	0.0004	0.002	0.210	0.834	-0.003	0.004
swrite	-0.0021	0.002	-1.051	0.293	-0.006	0.002
fork	-1.9818	0.250	-7.920	0.000	-2.472	-1.491
exec	-0.0165	0.048	-0.344	0.731	-0.111	0.077
rchar	-3.203e-06	8.66e-07	-3.697	0.000	-4.9e-06	-1.5e-06
wchar	-1.091e-05	1.3e-06	-8.367	0.000	-1.35e-05	-8.35e-06
pgout	-0.2141	0.067	-3.212	0.001	-0.345	-0.083
ppgout	0.1082	0.039	2.780	0.005	0.032	0.185
pgfree	-0.0713	0.019	-3.772	0.000	-0.108	-0.034
pgscan	0.0098	0.005	1.786	0.074	-0.001	0.021
atch	-0.0513	0.024	-2.165	0.030	-0.098	-0.005
pgin	0.0305	0.028	1.091	0.276	-0.024	0.085
ppgin	-0.0223	0.018	-1.238	0.216	-0.058	0.013
pflt	-0.0387	0.004	-9.068	0.000	-0.047	-0.030
vflt	0.0222	0.003	6.737	0.000	0.016	0.029
runqsz	7.7659	0.307	25.320	0.000	7.165	8.367
freemem	-0.0016	7.59e-05	-21.349	0.000	-0.002	-0.001
freeswap	3.272e-05	4.55e-07	71.870	0.000	3.18e-05	3.36e-05

Values in the columns "P>|t|" represents p value, if its less than assumed significance level(0.05) then that means the attribute is important and we shouldn't delete it. If its greater than 0.05 then we can try deleting the column and calculate if its impacting our R2 value or RMSE.

Upon repeating above steps, we have identified that the columns 'lwrite','sread','swrite','exec','pgscan','pgin','ppgin' are adding no values to the model and can be removed.

Final model score still remains the same. This means the given accuracy is due to the nature of data.

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

Final model score still remains the same. This means the given accuracy is due to the nature of data.

Linear equation to calculate "usr" variable is as follows,

usr = (-0.02)*lread+(0.001)*scall+(-2.072)*fork+(-0.0)*rchar+(-0.0)*wchar+(-0.24)*pgout+(0.106)*ppgout+(-0.05)*pgfree+(-0.051)*atch+(-0.039)*pflt+(0.023)*vflt+(7.749)*runqsz+(-0.002)*freemem+(0.0)*freeswap+43.589

Note that out of 22 columns given, only above columns are identified as important and they alone were able to predict the value of usr variable with 63% accuracy.

Though the variables rchar, wchar and freeswap have very less coefficient values, we cannot deleted those as dropping those columns is reducing the our model accuracy and increasing RSME value. Hence we choose not to drop them.

Basically linear regression model assumes to perform better when the data has below characteristics,

1. Linear relationship: There exists a linear relationship between the independent variable, x, and the dependent variable, y.

2. Independence: The residuals are independent. In particular, there is no correlation between consecutive residuals in time series data.

3. Homoscedasticity: The residuals have constant variance at every level of x.

4. Normality: The residuals of the model are normally distributed.

But unfortunately our dataset has violated all the above, Hence this might have caused our low model score.

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.

Dataset for Problem 2: Contraceptive_method_dataset.xlsx
Data Dictionary:
1. Wife's age (numerical)
2. Wife's education (categorical) 1=uneducated, 2, 3, 4=tertiary
3. Husband's education (categorical) 1=uneducated, 2, 3, 4=tertiary
4. Number of children ever born (numerical)
5. Wife's religion (binary) Non-Scientology, Scientology
6. Wife's now working? (binary) Yes, No
7. Husband's occupation (categorical) 1, 2, 3, 4(random)
8. Standard-of-living index (categorical) 1=verlow, 2, 3, 4=high
9. Media exposure (binary) Good, Not good
10. Contraceptive method used (class attribute) No,Yes

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

The given dataset "compactiv.xlsx" has

- 1473 records and 10 columns.
- 71 nulls in column Wife_age and 21 nulls in No_of_children_born.
- 80 duplicate records.
- Of all 10 columns, Wife_age, N.

Below is the sample datao_of_children_born and Husband_Occupation are of numeric type and remaining 7 columns are of object type,

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

Datatypes and Column names:

```
 #   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
```

Five point summary after removing duplicates:

	count	mean	std	min	25%	50%	75%	max
Wife_age	1326.0	32.557	8.289	16.0	26.0	32.0	39.0	49.0
No_of_children_born	1372.0	3.291	2.400	0.0	1.0	3.0	5.0	16.0
Husband_Occupation	1393.0	2.174	0.855	1.0	1.0	2.0	3.0	4.0

As per above summary,

Minimum age of the wife from the given data set is 16 and maximum is 49. Max age seems legible but min age 16 is concerning.

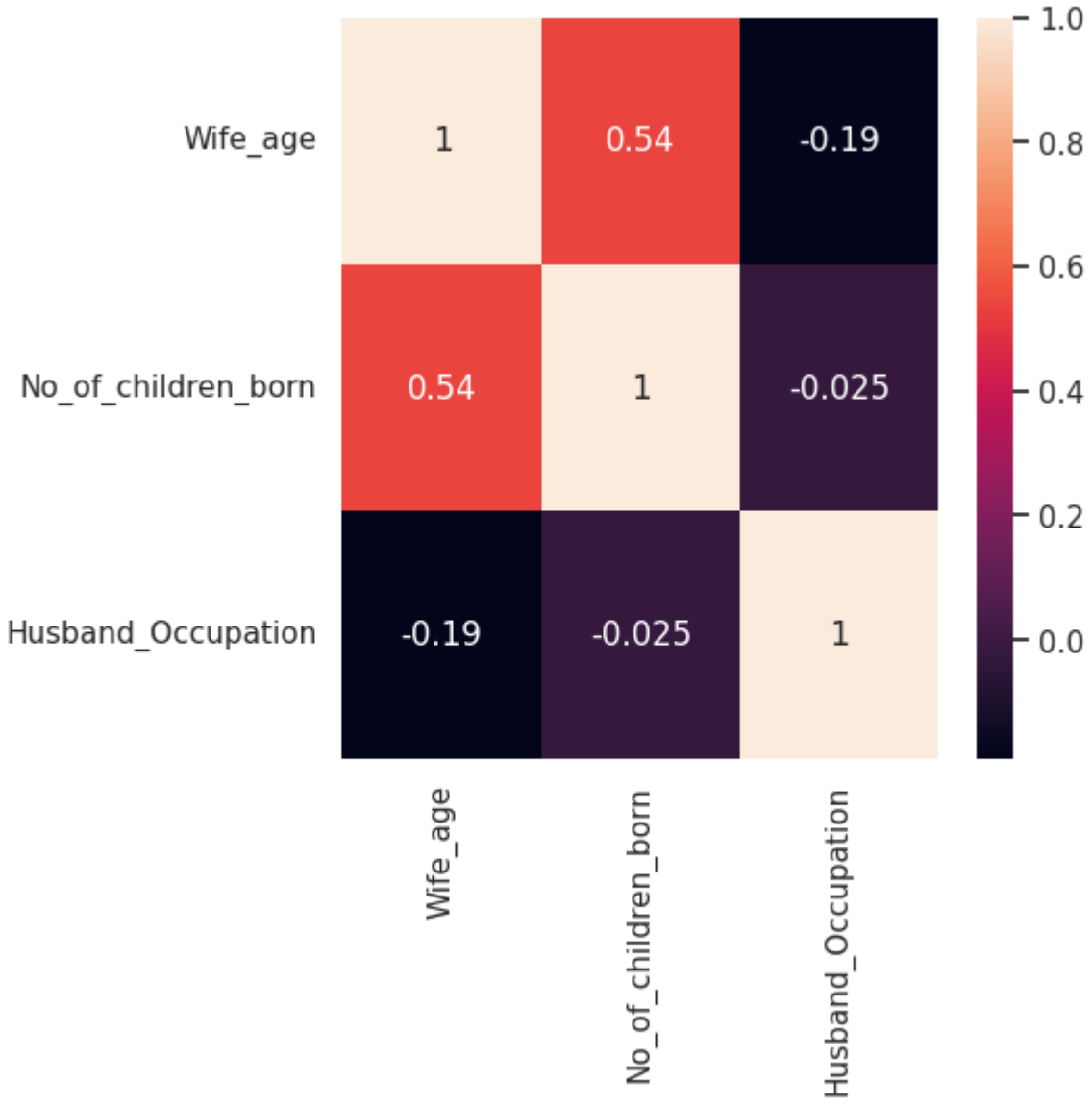
Max Number of children is 16, which might be incorrect given the age of wife. This could be a outlier data.

Correlation Plot:

From the correlation plot, we can see that,

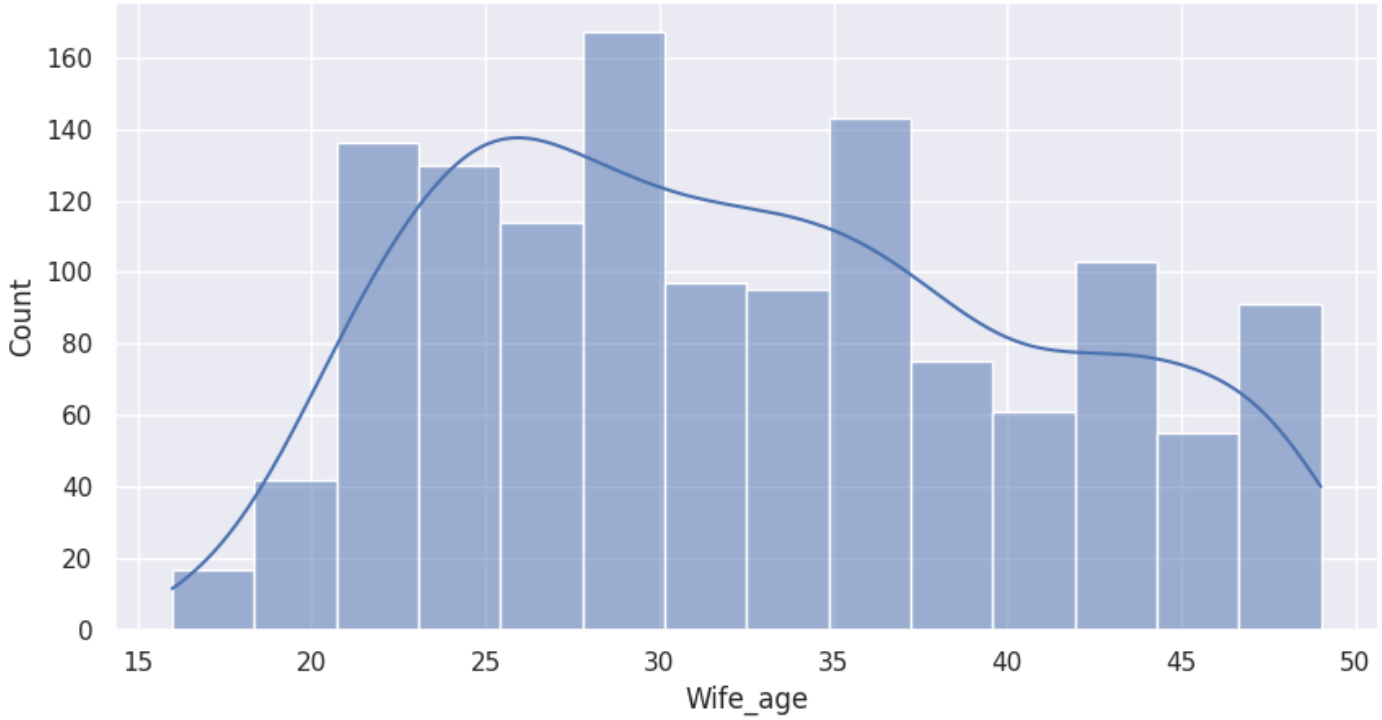
There is a positive correlation between wife age and number of children which is obvious. Remaining columns don't have much correlation.

Correlation values near to 1 or -1 are highly positively correlated and highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.

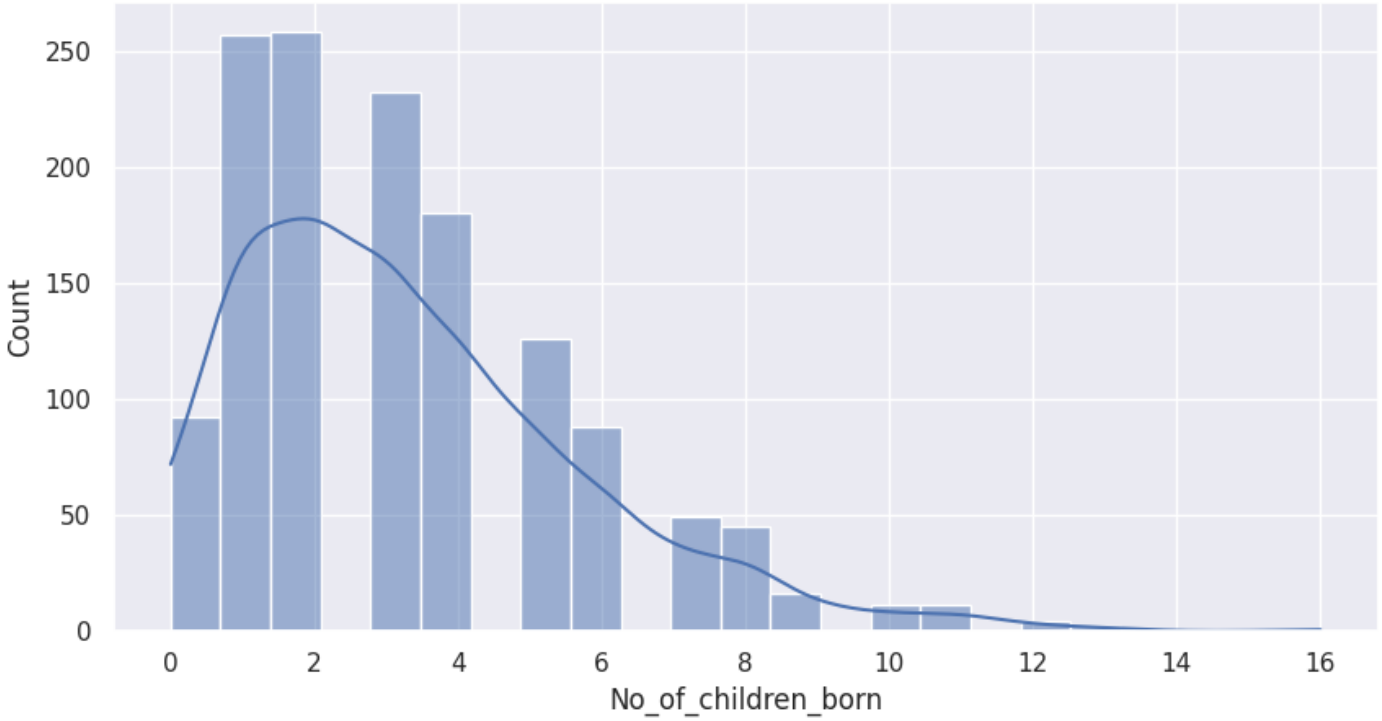


Univariate Analysis:

Checking Distribution of Wife_age, No_of_children_born and Husband_Occupation



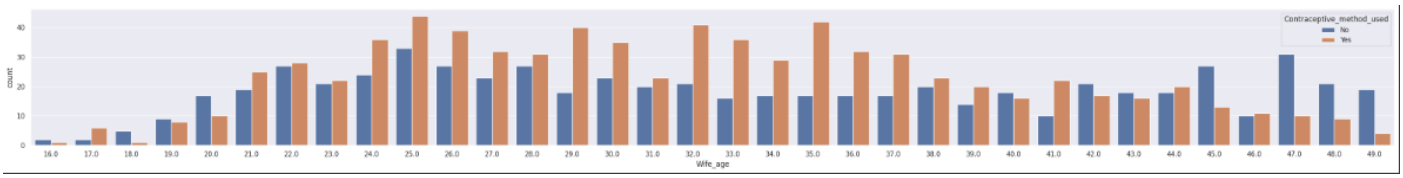
Values in Wife_age column are almost uniformly distributed, with mode value between 25 to 30.



When we look at distribution of No_of_children_born, it is evident that the abnormal child count near 16 is a outlier and its causing skewness in data. Most of the people are giving births to minimum 2 kids.

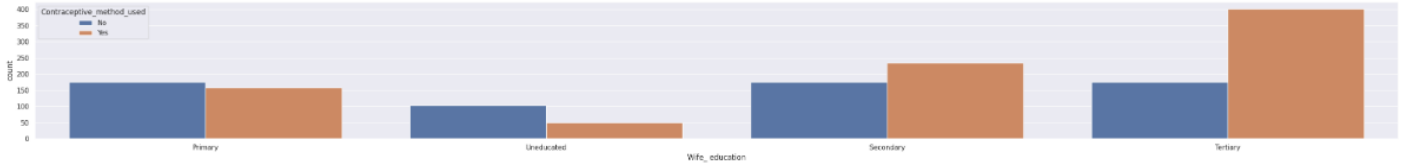
From the distribution of Husband_Occupation, it shows that the given dataset has more records with occupation type 3 and less records with occupation type 4.

Bivariate Analysis:



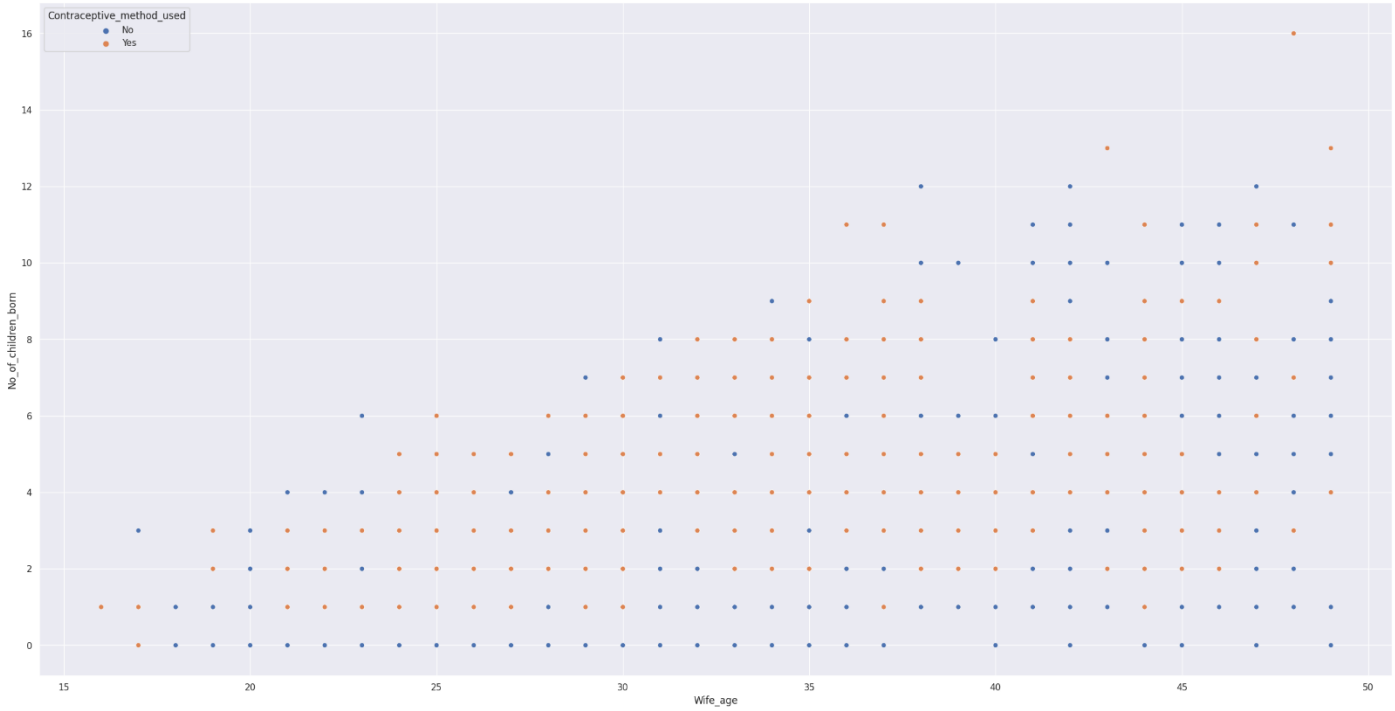
Above graph shows counts of females who did take contraceptives and who didn't among different age groups. Age groups 25 to 35 seems to have taken more people who took contraceptives.

Above chart shows the distribution of females who took/didn't took contraceptives among working and not working wives. From the given data, female who are not working seems to be taking more contraceptives.



Similarly above graph shows distribution of contraceptives among different wife_education categories. Females with education type Tertiary have taken more contraceptives.

Multivariate Analysis:



Above is the visual representation of change of contraceptives used in accordance with age and number of children.

There is an increase in females with contraceptives use with increase in number of children.

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.

Before splitting our data into predictor and predicted set,

Lets treat the data by removing nulls that we observed in Wife_age and No_of_children_born.

Treat outliers present in No_of_children_born.

Handling Nulls:

Below are the counts on nulls in each column,

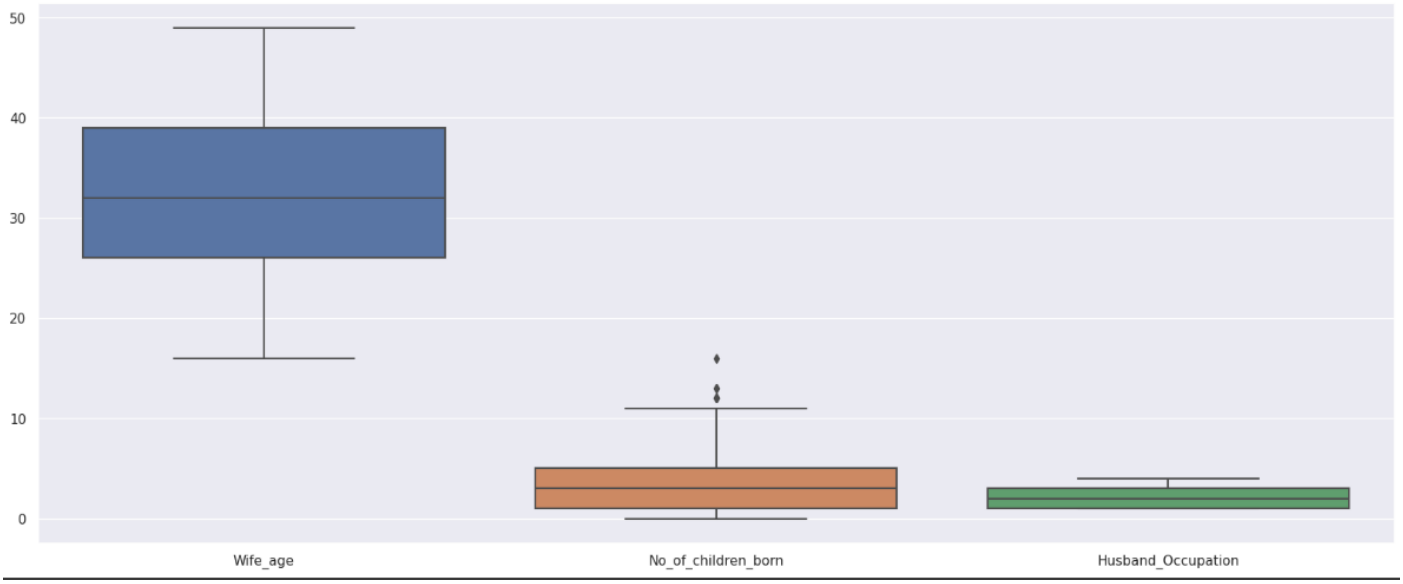
```
Wife_age                    67
Wife_ education               0
Husband_education             0
No_of_children_born          21
Wife_religion                 0
Wife_Working                  0
Husband_Occupation            0
Standard_of_living_index      0
Media_exposure                0
Contraceptive_method_used     0
dtype: int64
```

Imputed nulls in wife_age column with mean age of the column and the nulls in No_of_children_born are replaced with zeroes.

Here is the count of nulls after treatment.

```
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
```

Handling Outliers:

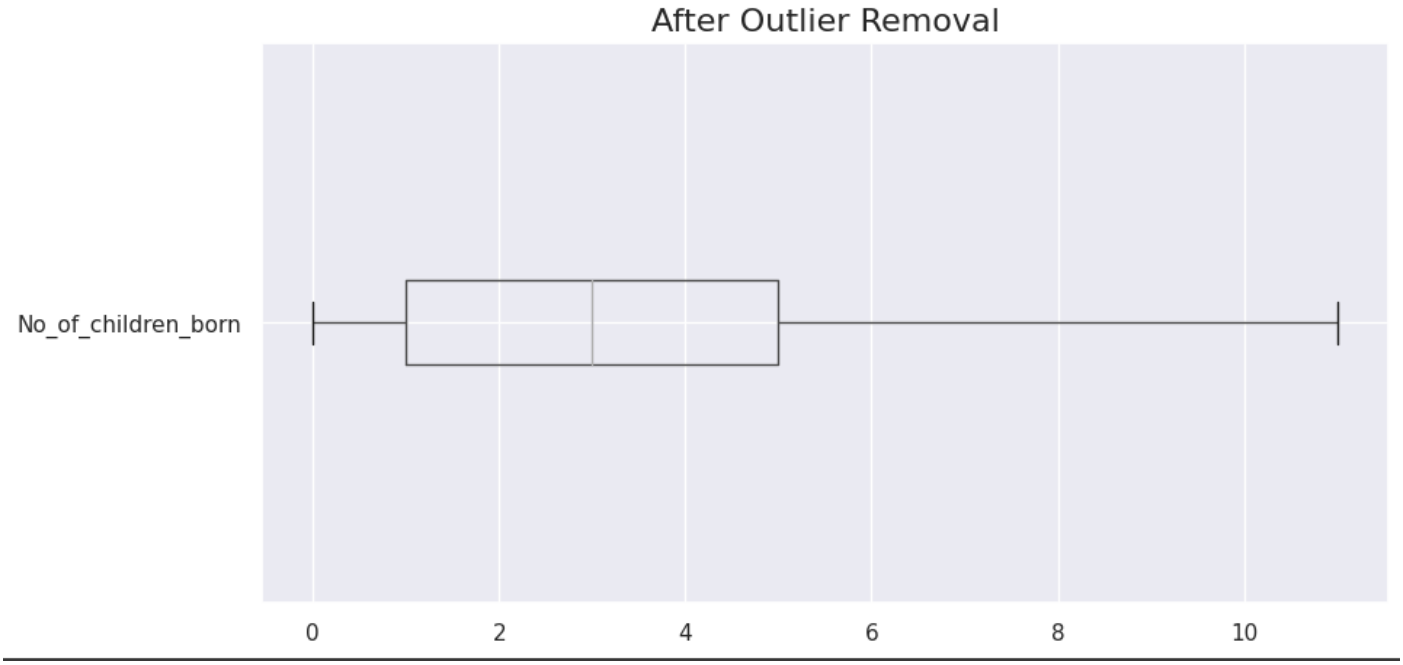


As we have already observed some radical values in No_of_children_born column like number of children as more than 8. We will treat them using Inter Quartile Range method.

This method says any value less than (Q1-1.5IQR) or value greater than (Q3+1.5IQR) is treated as outlier and we can replace them with 25th and 75th percentiles of the total columns values respectively.

Where IQR = 75th percentile of (No_of_children_born) – 25th percentile of (No_of_children_born)

After removing outliers, below is the new boxplot, where we can observe that there are no dots/datapoints outside our right whisker.



Now that we have removed irregularities in our data,

We also need decode the categorical values in the columns as shown below, so that we can pass the numerical values to our model.

{

'Wife_education': {'Primary': 0, 'Uneducated': 1, 'Secondary': 2, 'Tertiary': 3},

'Husband_education': {'Secondary': 0, 'Primary': 1, 'Tertiary': 2, 'Uneducated': 3},

'Wife_religion': {'Scientology': 0, 'Non-Scientology': 1},

'Wife_Working': {'No': 0, 'Yes': 1},

'Standard_of_living_index': {'High': 0, 'Very High': 1, 'Low': 2, 'Very Low': 3},

'Media_exposure': {'Exposed': 0, 'Not-Exposed': 1},

'Contraceptive_method_used': {'No': 0, 'Yes': 1}}

Here is the summary of values in each attribute after decoding,

	count	mean	std	min	25%	50%	75%	max
Wife_age	1393.0	32.557315	8.087308	16.0	26.0	32.0	38.0	49.0
Wife_education	1393.0	1.788227	1.175213	0.0	1.0	2.0	3.0	3.0
Husband_education	1393.0	1.407753	0.896351	0.0	1.0	2.0	2.0	3.0
No_of_children_born	1393.0	3.231874	2.379285	0.0	1.0	3.0	5.0	11.0
Wife_religion	1393.0	0.148600	0.355822	0.0	0.0	0.0	0.0	1.0
Wife_Working	1393.0	0.251256	0.433891	0.0	0.0	0.0	1.0	1.0
Husband_Occupation	1393.0	2.174444	0.854590	1.0	1.0	2.0	3.0	4.0
Standard_of_living_index	1393.0	1.047380	0.912427	0.0	0.0	1.0	2.0	3.0
Media_exposure	1393.0	0.078248	0.268658	0.0	0.0	0.0	0.0	1.0
Contraceptive_method_used	1393.0	0.559225	0.496658	0.0	0.0	1.0	1.0	1.0

Now we split data into

predictor variables(x) with columns ['Wife_age', 'Wife_education', 'Husband_education', 'No_of_children_born', 'Wife_religion', 'Wife_Working', 'Husband_Occupation', 'Standard_of_living_index', 'Media_exposure'] and

predicted dataset(y) with column ['Contraceptive_method_used']

Now we can split our data into training and testing datasets in 70:30 ratio respectively. Also we used stratify method while splitting to ensure the distribution of target variable remains almost similary in training and test datasets.

Here are the record counts and column counts after splitting data. Training data has 975 records and testing data has 418 records.

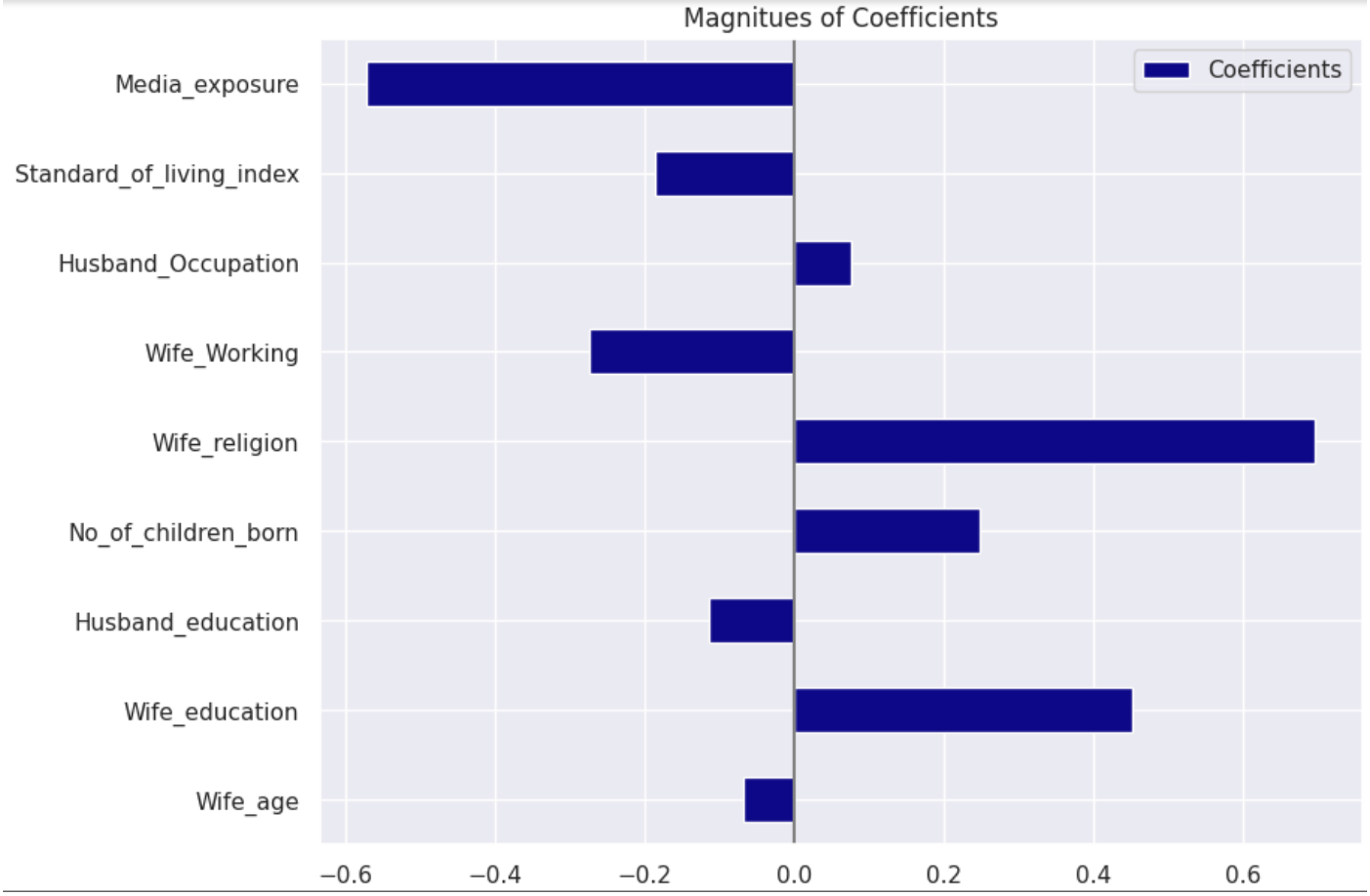
```
Shape of training dataset(x_train):  (975, 9)
Shape of testing dataset(x_test):  (418, 9)
Shape of target training dataset(y_train):  (975,)
Shape of target testing dataset(y_test):  (418,)
```

Building Logistic Regression(LR) Model:

Upon building logistic regression model and post passing our training set for fitting, these are the scores for training and testing datasets.

```
LR_model_score_train_data -->  0.6564102564102564
LR_model_score_test_data -->  0.631578947368421
```

Below graph shows the magnitude of each coefficient for the linear equation we can use from this model.

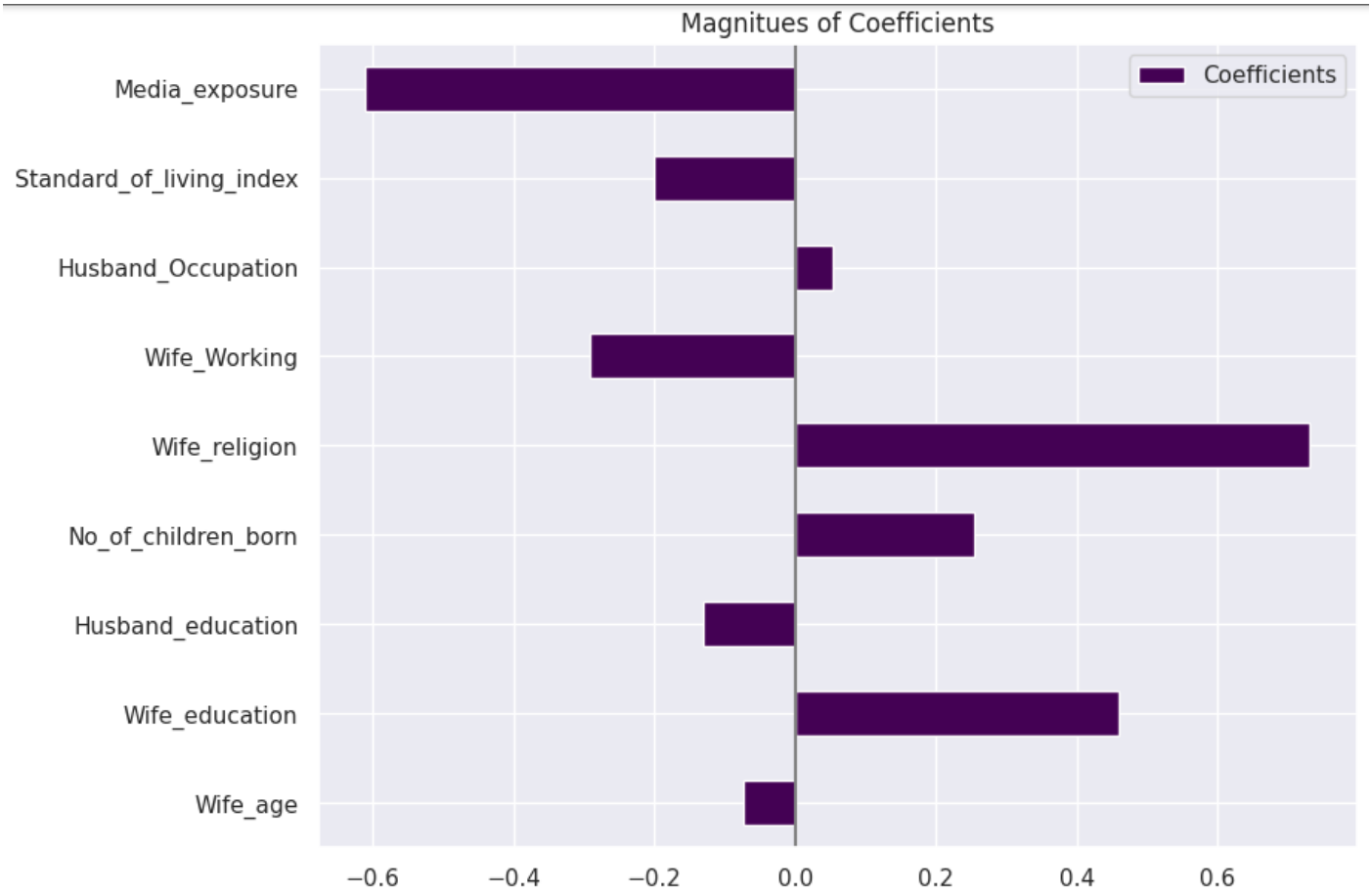


Building Linear Discriminant Analysis(LDA) Model:

Upon building LDA model and post passing our training set for fitting, these are the scores for training and testing datasets.

```
LDA_model_score_train_data -->  0.6564102564102564
LDA_model_score_test_data -->  0.6196172248803827
```

For the linear equation that LDA model has predicted, below are the values for each coefficient.



Building Classification and Regression Tree(CART) Model:

Instead of directly building of CART model, lets find out the best possible parameters to pass for into our DecisionClassifier method using GridSeachCV algorithm.

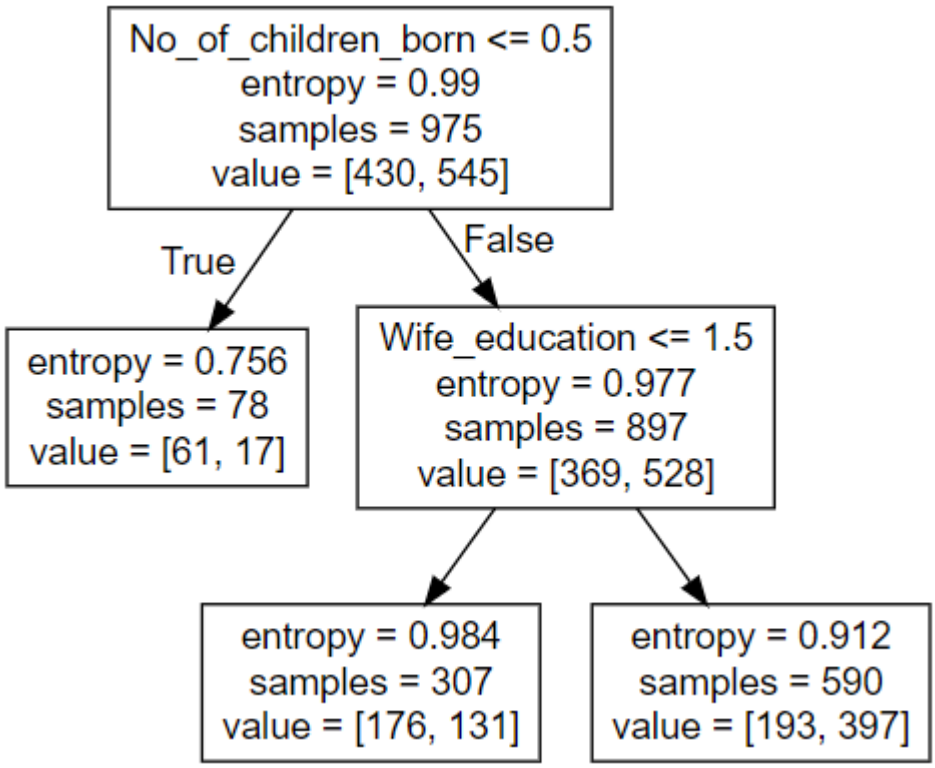
The output we got is as follows,

```
Fitting 5 folds for each of 450 candidates, totalling 2250 fits
DecisionTreeClassifier(ccp_alpha=0.01, criterion='entropy', max_depth=5,
                       max_features='auto', min_samples_leaf=5,
                       random_state=1024)
```

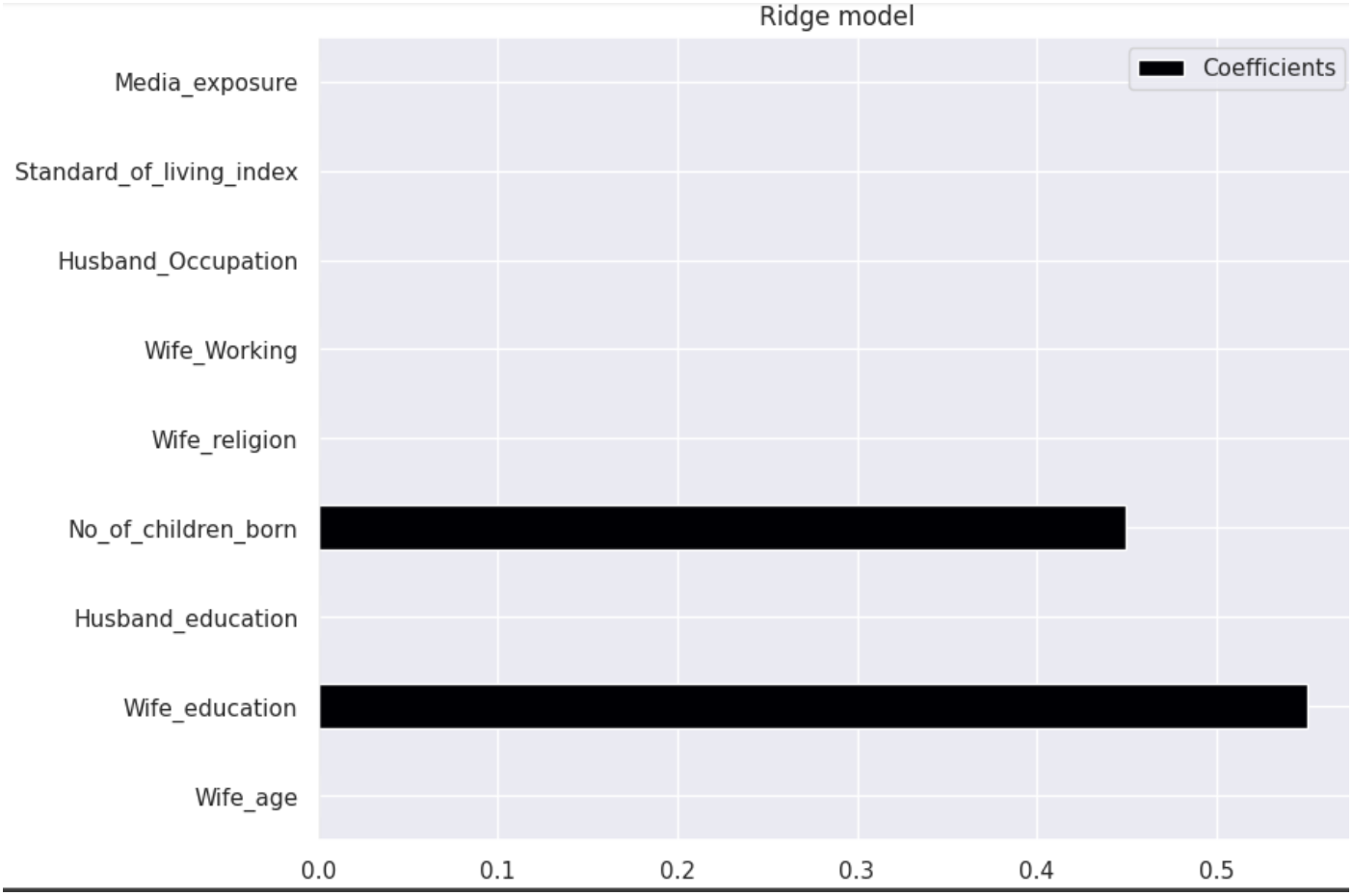
Upon building CARTmodel and post passing our training set for fitting with the above parameters, these are the scores for training and testing datasets.

```
CART_model_score_train_data -->  0.6502564102564102
CART_model_score_test_data -->  0.631578947368421
```

Below is the decision tree used by the model to predict contraceptive use,



As this is a decision tree model, unlike regression model we do not get any linear equation. Hence we wont get coefficients as above, instead we can see the features with utmost importance for this model, they are as follows,

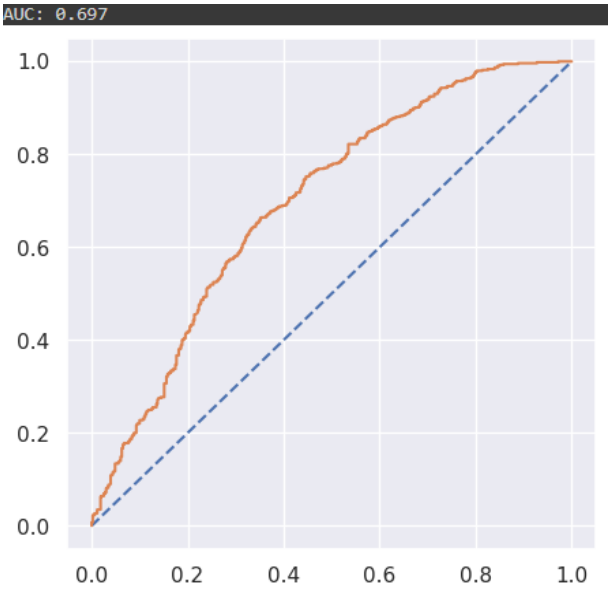


This model just used two columns to predict the target variable, we will see how each of this model is performing in below sections.

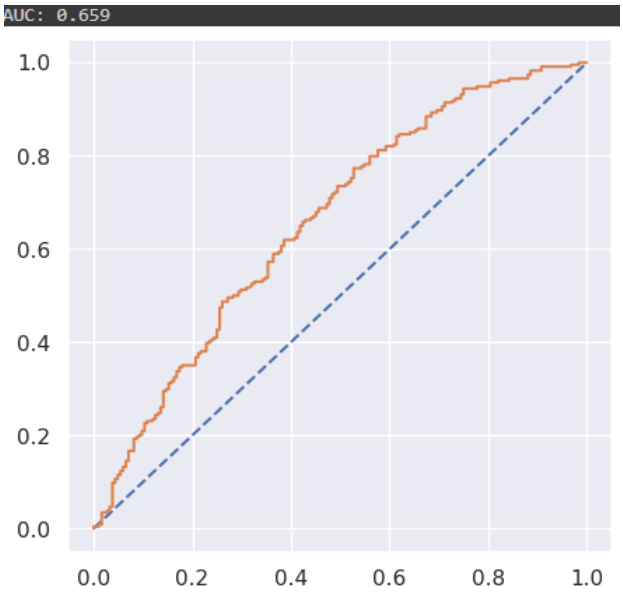
2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model Final Model: Compare Both the models and write inference which model is best/optimized.

Logistic Regression:

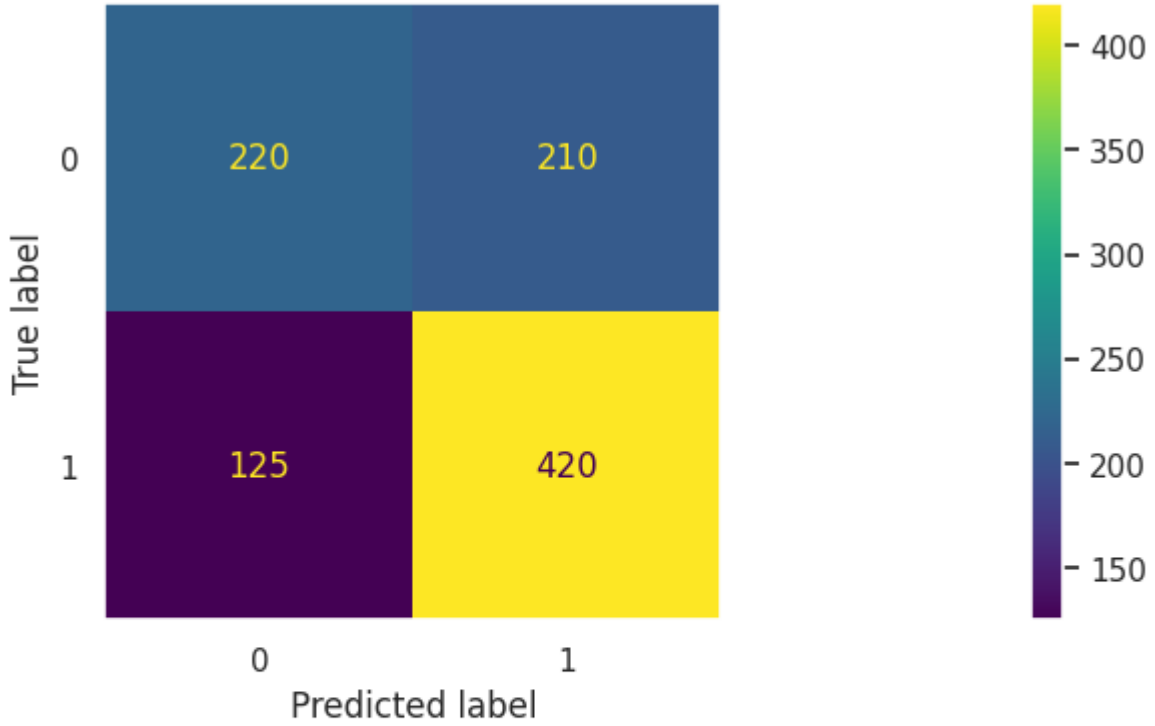
AUC and ROC for the training data(69.7)



AUC and ROC for the testing data(65.9)

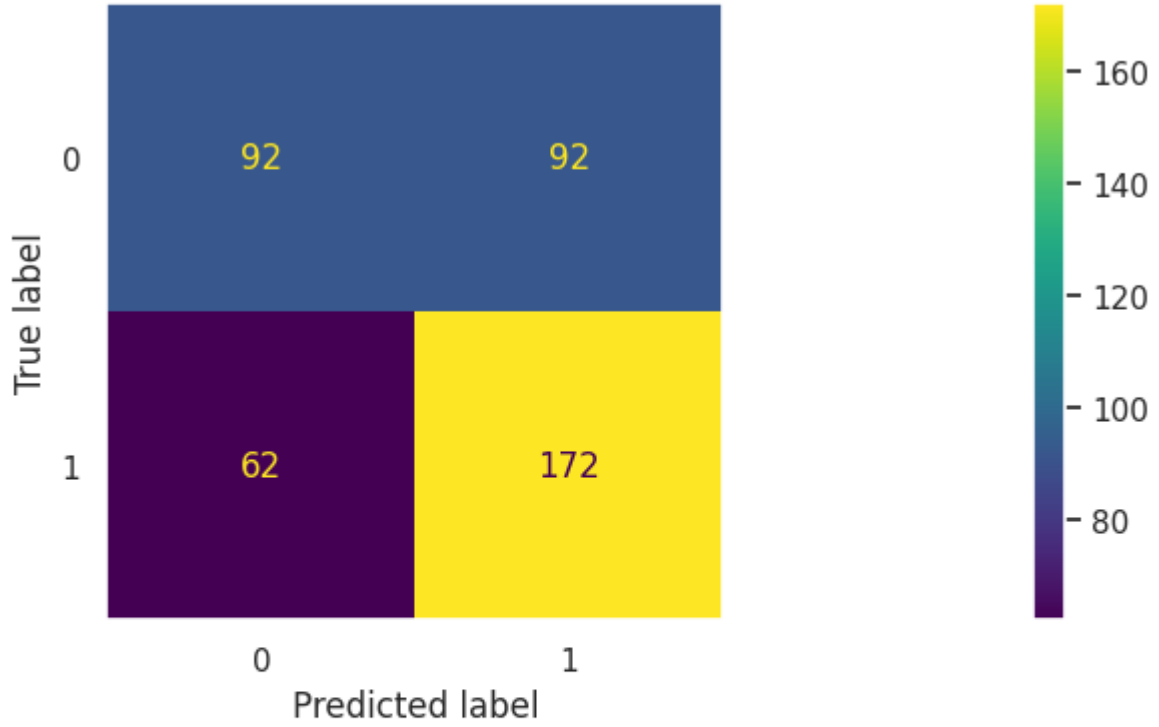


Confusion Matrix and Classification Report for training Data



	precision	recall	f1-score	support
0	0.64	0.51	0.57	430
1	0.67	0.77	0.71	545
accuracy			0.66	975
macro avg	0.65	0.64	0.64	975
weighted avg	0.65	0.66	0.65	975

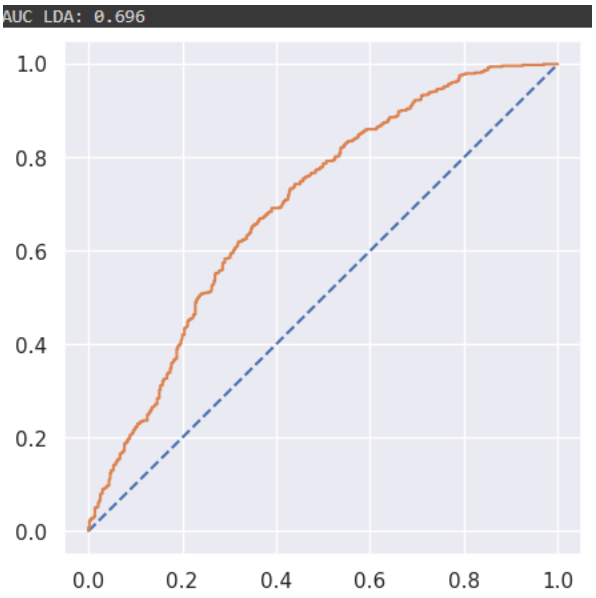
Confusion Matrix and Classification Report for testing Data



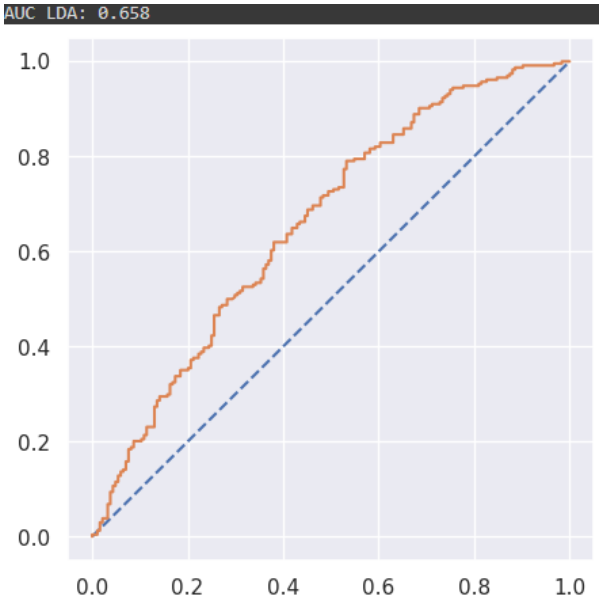
	precision	recall	f1-score	support
0	0.60	0.50	0.54	184
1	0.65	0.74	0.69	234
accuracy			0.63	418
macro avg	0.62	0.62	0.62	418
weighted avg	0.63	0.63	0.63	418

Linear Discriminant Analysis(LDA):

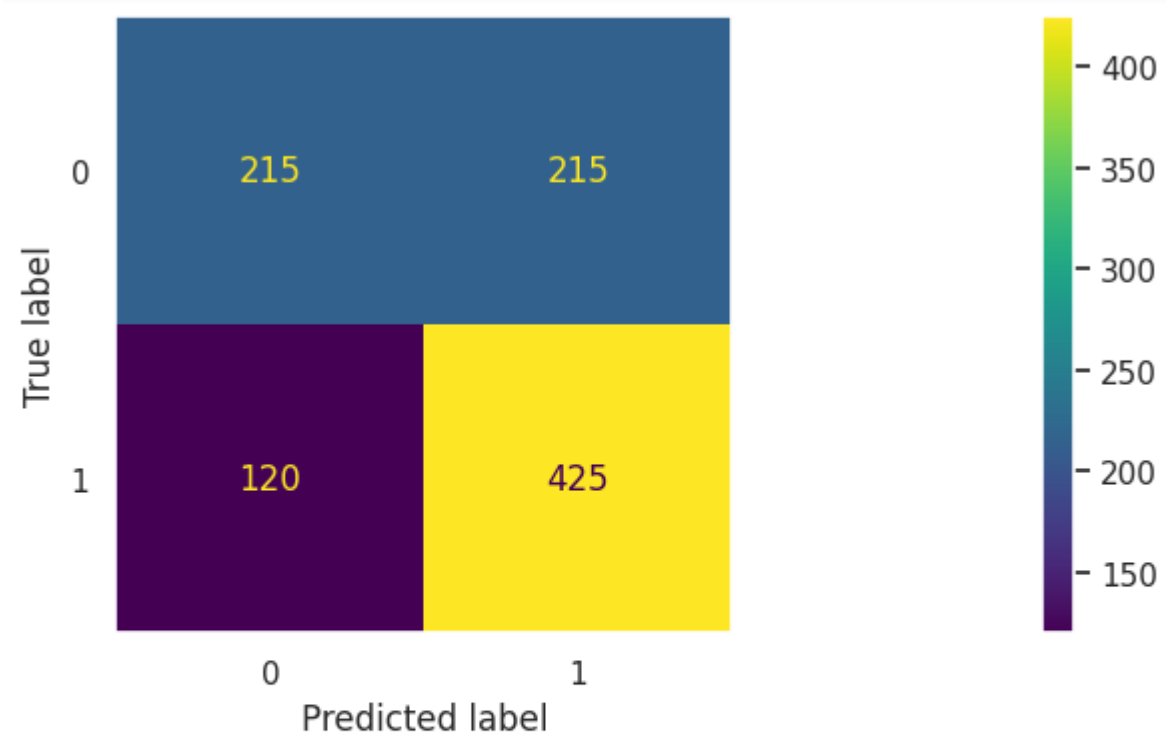
AUC and ROC for the training data(69.6)



AUC and ROC for the testing data(65.8)

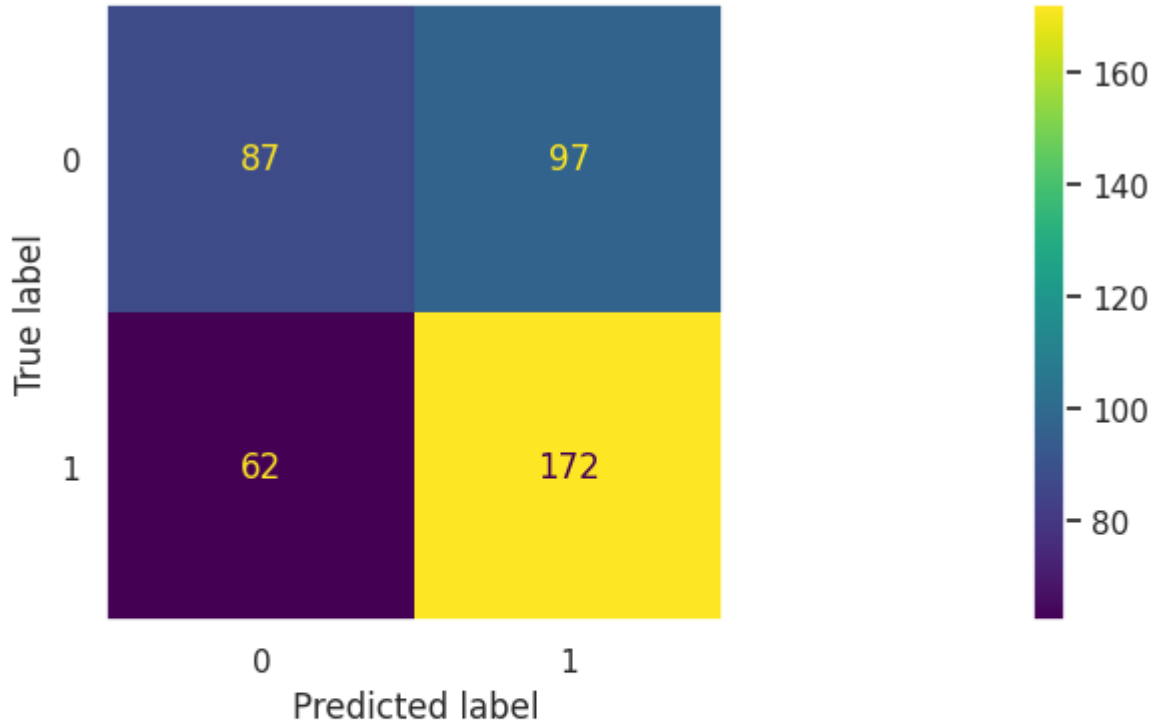


Confusion Matrix and Classification Report for training Data



	precision	recall	f1-score	support
0	0.64	0.50	0.56	430
1	0.66	0.78	0.72	545
accuracy			0.66	975
macro avg	0.65	0.64	0.64	975
weighted avg	0.65	0.66	0.65	975

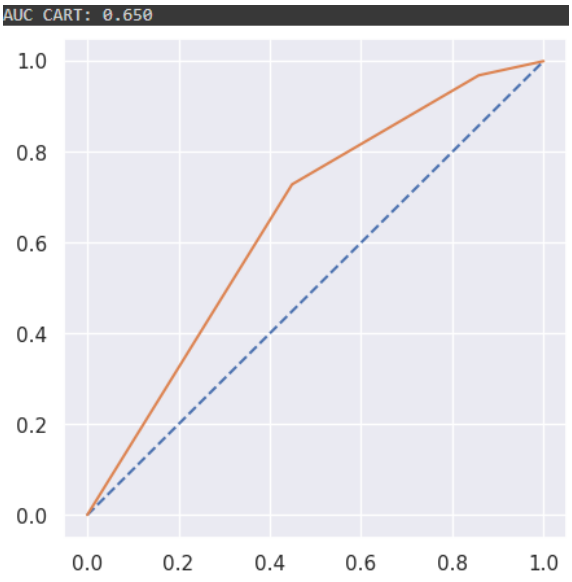
Confusion Matrix and Classification Report for testing Data



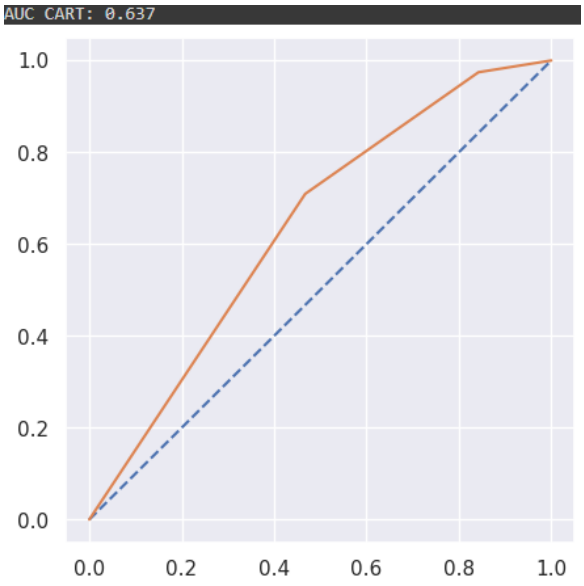
	precision	recall	f1-score	support
0	0.58	0.47	0.52	184
1	0.64	0.74	0.68	234
accuracy			0.62	418
macro avg	0.61	0.60	0.60	418
weighted avg	0.61	0.62	0.61	418

Classification and Regression Tree(CART):

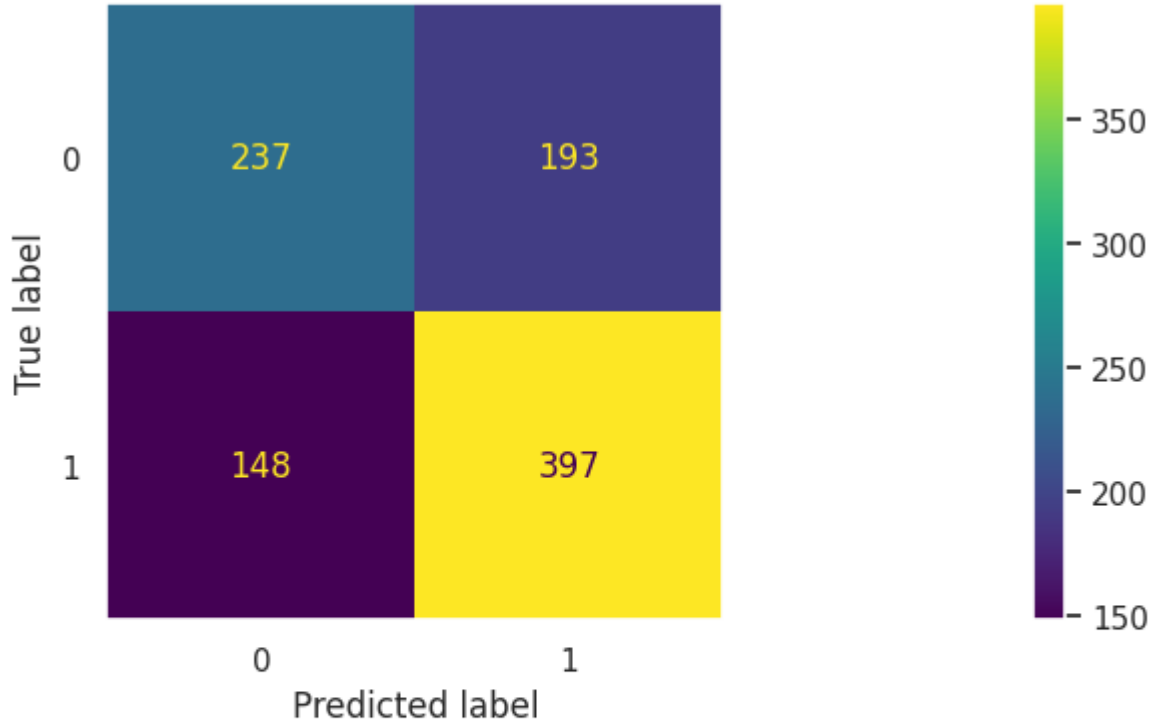
AUC and ROC for the training data(65.0)



AUC and ROC for the testing data(63.7)

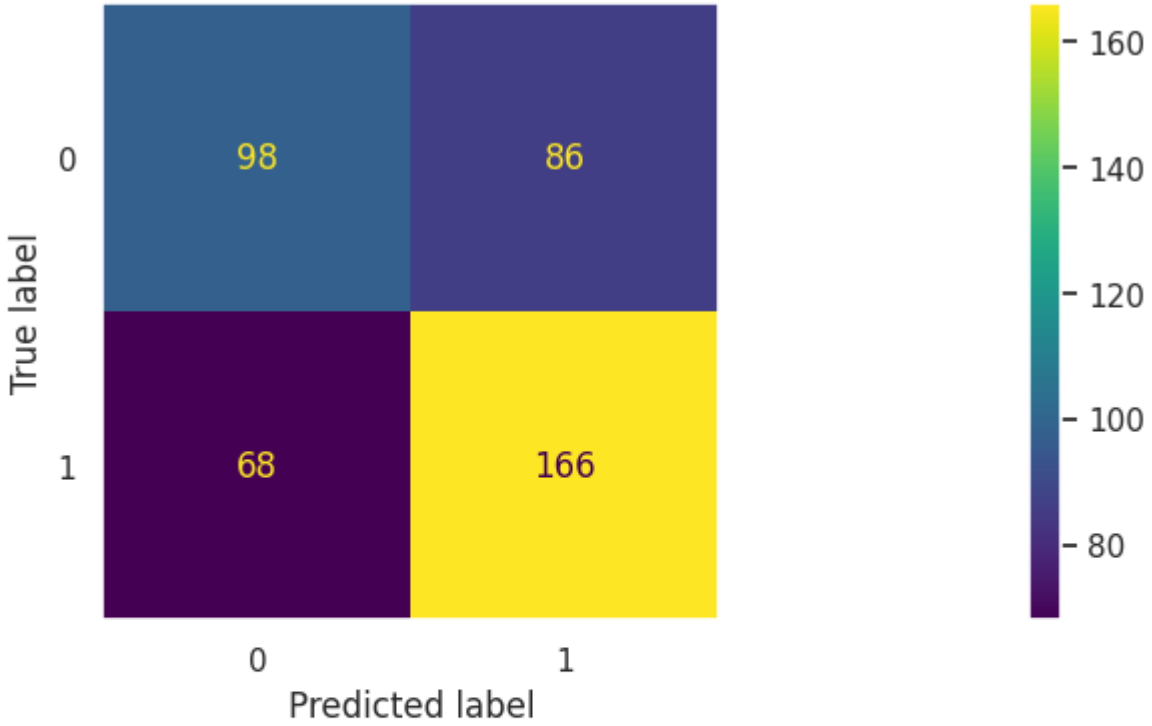


Confusion Matrix and Classification Report for training Data



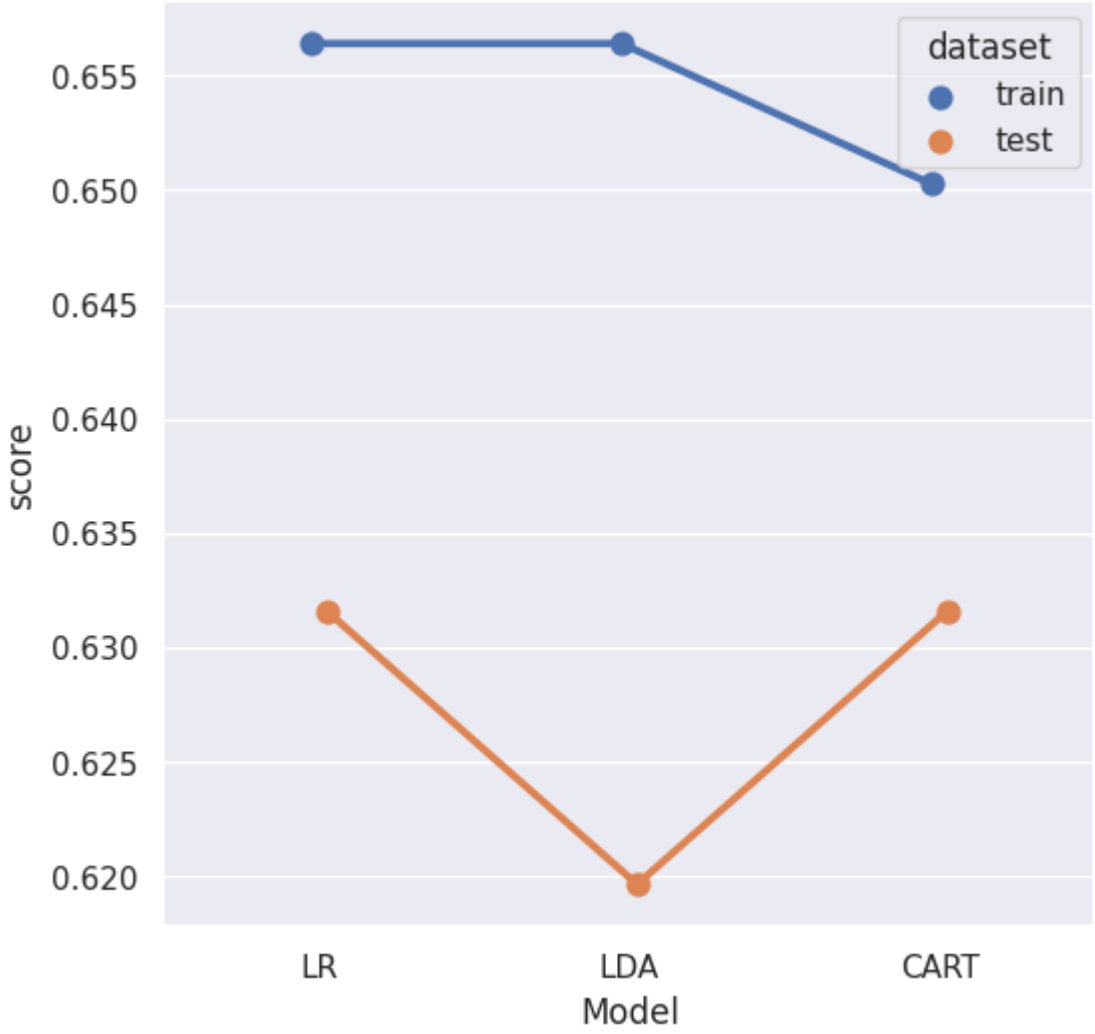
	precision	recall	f1-score	support
0	0.62	0.55	0.58	430
1	0.67	0.73	0.70	545
accuracy			0.65	975
macro avg	0.64	0.64	0.64	975
weighted avg	0.65	0.65	0.65	975

Confusion Matrix and Classification Report for testing Data



	precision	recall	f1-score	support
0	0.59	0.53	0.56	184
1	0.66	0.71	0.68	234
accuracy			0.63	418
macro avg	0.62	0.62	0.62	418
weighted avg	0.63	0.63	0.63	418

Comparison of score from all the models:



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

Logistic Regression:

Inferences- from Logistic Regression Model :

Contraceptive method used : **0 indicates No, 1 indicated Yes**

For predicting that person do not use contraceptive method (Label-0):

Precision (60%) – 60% of females predicted are not using contraceptives out of all females that were predicted as not using contraceptives.

Recall (50%) – Out of all the females who didnt take contraceptives, 50% of females have been predicted correctly .

For predicting that a person takes contraceptive(Label-1):

Precision (65%) – Of all the females predicted to use contraceptives, 65% of females actually use contraceptives.

Recall (74%) – Out of all the females actually taking contraceptives , 74% of females have been predicted correctly .

Linear Equation from Logistic Regression to find Contraceptive_method_used:

Contraceptive_method_used = 1.038 + Wife_age*(-0.068) + Wife_education*(0.452) + Husband_education*(-0.114) + No_of_children_born*(0.248) + Wife_religion*(0.697) + Wife_Working*(-0.273) + Husband_Occupation*(0.076) + Standard_of_living_index*(-0.185) + Media_exposure*(-0.571)

Linear Discriminant Analysis:

Inferences- from Linear Discriminant Analsyis(LDA) :

Contraceptive method used : 0 indicates No, 1 indicated Yes

For predicting that person do not use contraceptive method (Label-0):

Precision (58%) – 58% of females predicted are not using contraceptives out of all females that were predicted as not using contraceptives.

Recall (47%) – Out of all the females who didnt take contraceptives, 47% of females have been predicted correctly .

For predicting that a person takes cotraceptive(Label-1):

Precision (64%) – Of all the females predicted to use contraceptives, 64% of females actually use contraceptives.

Recall (74%) – Out of all the females actually taking contraceptives , 74% of females have been predicted correctly .

Linear Equation from Linear Discriminant Analysis to find Contraceptive_method_used:

Contraceptive_method_used = 1.26 + Wife_age*(-0.072) + Wife_education*(0.461) + Husband_education*(-0.13) + No_of_children_born*(0.254) + Wife_religion*(0.732) + Wife_Working*(-0.291) + Husband_Occupation*(0.053) + Standard_of_living_index*(-0.201) + Media_exposure*(-0.61)

CART:

Contraceptive method used : 0 indicates No, 1 indicated Yes

For predicting that person do not use contraceptive method (Label-0):

Precision (59%) – 59% of females predicted are not using contraceptives out of all females that were predicted as not using contraceptives.

Recall (53%) – Out of all the females who didnt take contraceptives, 53% of females have been predicted correctly .

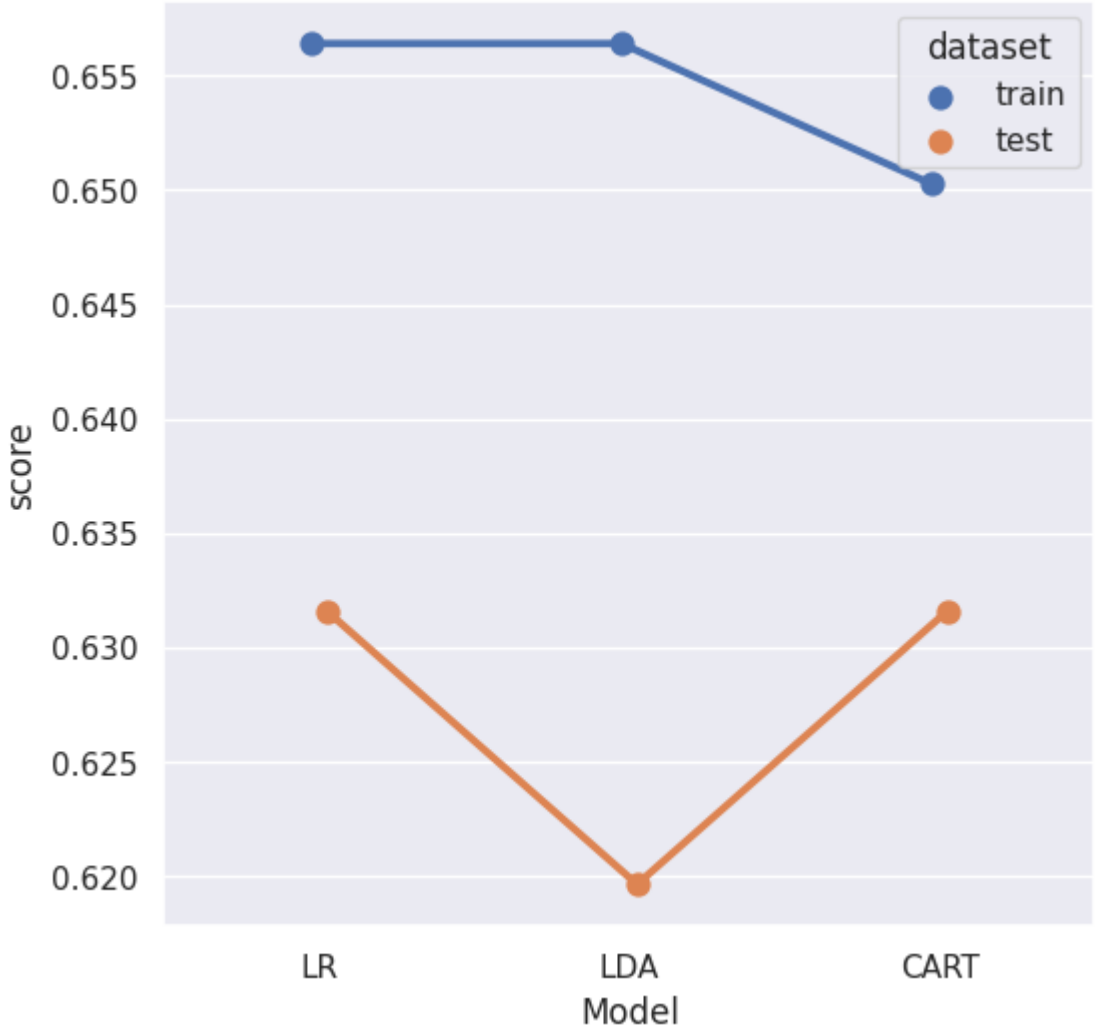
For predicting that a person takes cotraceptive(Label-1):

Precision (66%) – Of all the females predicted to use contraceptives, 66% of females actually use contraceptives.

Recall (71%) – Out of all the females actually taking contraceptives , 71% of females have been predicted correctly .

As per CART, only Wife_education and No_of_children_born are enough to predict the target column.

Conclusion:



Though LR and LDA models are giving 74% Recall value for contraceptive taken(Yes), as per above graph, LDA model score has been reduced in testing set. CART model gives us the Recall value of 71% which is relatively less.

Hence it is suggested to use LR model for prediction for this dataset.

Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

Below are the steps performed while doing the predictions in this project,

1. Data Collection
2. Exploratory Data Analysis
3. Data Cleansing
    a. Null handling
    b. Outlier Removal
    c. Duplicate record Handling
    d. Datatype management
4. Data Split into training and testing sets
5. Model Building
6. Testing Model Predictions
7. Checking and Comparing model performance obtained by passing different parameters each time, also used below to quantify performance
    a. AUC

    b. ROC
    c. RSME
    d. R2 value
    e. Adjusted R2 value
8. Documenting Inferences