Predictive Modelling Business Report

Yedupati Venkata Yamini

Table of Contents

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.

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.

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.

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.

1.4 Inference: Basis on these predictions, what are the business insights and recommendations. Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

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.

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.

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.

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.

2.4 Inference: Basis on these predictions, what are the insights and recommendations.
Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

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.

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.

Ans:

- The given data set has 8192 rows and 22 columns including dependent and independent variables.
- Viewing the basic information of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8192 entries, 0 to 8191
Data columns (total 22 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   lread     8192 non-null   int64
 1   lwrite    8192 non-null   int64
 2   scall     8192 non-null   int64
 3   sread     8192 non-null   int64
 4   swrite    8192 non-null   int64
 5   fork      8192 non-null   float64
 6   exec      8192 non-null   float64
 7   rchar     8088 non-null   float64
 8   wchar     8177 non-null   float64
 9   pgout     8192 non-null   float64
 10  ppgout    8192 non-null   float64
 11  pgfree    8192 non-null   float64
 12  pgscan    8192 non-null   float64
 13  atch      8192 non-null   float64
 14  pgin      8192 non-null   float64
 15  ppgin     8192 non-null   float64
 16  pflt      8192 non-null   float64
 17  vflt      8192 non-null   float64
 18  runqsz    8192 non-null   object
 19  freemem   8192 non-null   int64
 20  freeswap  8192 non-null   int64
 21  usr       8192 non-null   int64
dtypes: float64(13), int64(8), object(1)
memory usage: 1.4+ MB
```

Insights:

1. All the columns present in the dataset are either integers or decimal values.
2. Variable 'usr' is the dependent variable and all others are independent variables.
3. There are missing/null values for the columns 'rchar' and 'wchar' which will need to be revisited to decide whether to drop (or) impute the missing values.

- Viewing the first 5 rows of the dataset:

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

ows × 22 columns

- Viewing the last 5 rows of the dataset:

read	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
16	12	3009	360	244	1.6	5.81	405250.0	85282.0	8.02	...	55.11	0.6	35.87	47.90	139.28	270.74	CPU_Bound	387	986647	80
4	0	1596	170	146	2.4	1.80	89489.0	41764.0	3.80	...	0.20	0.8	3.80	4.40	122.40	212.60	Not_CPU_Bound	263	1055742	90
16	5	3116	289	190	0.6	0.60	325948.0	52640.0	0.40	...	0.00	0.4	28.40	45.20	60.20	219.80	Not_CPU_Bound	400	969106	87
32	45	5180	254	179	1.2	1.20	62571.0	29505.0	1.40	...	18.04	0.4	23.05	24.25	93.19	202.81	CPU_Bound	141	1022458	83
2	0	985	55	46	1.6	4.80	111111.0	22256.0	0.00	...	0.00	0.2	3.40	6.20	91.80	110.00	CPU_Bound	659	1756514	94

- Viewing the summary of the dataset:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
lread	8192.0	NaN	NaN	NaN	19.559692	53.353799	0.0	2.0	7.0	20.0	1845.0
lwrite	8192.0	NaN	NaN	NaN	13.106201	29.891726	0.0	0.0	1.0	10.0	575.0
scall	8192.0	NaN	NaN	NaN	2306.318237	1633.617322	109.0	1012.0	2051.5	3317.25	12493.0
sread	8192.0	NaN	NaN	NaN	210.47998	198.980146	6.0	86.0	166.0	279.0	5318.0
swrite	8192.0	NaN	NaN	NaN	150.058228	160.47898	7.0	63.0	117.0	185.0	5456.0
fork	8192.0	NaN	NaN	NaN	1.884554	2.479493	0.0	0.4	0.8	2.2	20.12
exec	8192.0	NaN	NaN	NaN	2.791998	5.212456	0.0	0.2	1.2	2.8	59.56
rchar	8088.0	NaN	NaN	NaN	197385.728363	239837.493526	278.0	34091.5	125473.5	267828.75	2526649.0
wchar	8177.0	NaN	NaN	NaN	95902.992785	140841.707911	1498.0	22916.0	46619.0	106101.0	1801623.0
pgout	8192.0	NaN	NaN	NaN	2.285317	5.307038	0.0	0.0	0.0	2.4	81.44
ppgout	8192.0	NaN	NaN	NaN	5.977229	15.21459	0.0	0.0	0.0	4.2	184.2
pgfree	8192.0	NaN	NaN	NaN	11.919712	32.36352	0.0	0.0	0.0	5.0	523.0
pgscan	8192.0	NaN	NaN	NaN	21.526849	71.14134	0.0	0.0	0.0	0.0	1237.0
atch	8192.0	NaN	NaN	NaN	1.127505	5.708347	0.0	0.0	0.0	0.6	211.58
pgin	8192.0	NaN	NaN	NaN	8.27796	13.874978	0.0	0.6	2.8	9.765	141.2
ppgin	8192.0	NaN	NaN	NaN	12.388586	22.281318	0.0	0.6	3.8	13.8	292.61
pflt	8192.0	NaN	NaN	NaN	109.793799	114.419221	0.0	25.0	63.8	159.6	899.8
vflt	8192.0	NaN	NaN	NaN	185.315796	191.000603	0.2	45.4	120.4	251.8	1365.0
runqsz	8192	2	Not_CPU_Bound	4331	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freemem	8192.0	NaN	NaN	NaN	1763.456299	2482.104511	55.0	231.0	579.0	2002.25	12027.0
freeswap	8192.0	NaN	NaN	NaN	1328125.959839	422019.426957	2.0	1042623.5	1289289.5	1730379.5	2243187.0
usr	8192.0	NaN	NaN	NaN	83.968872	18.401905	0.0	81.0	89.0	94.0	99.0

Insights:

1. Almost 75% of the data have less than or equal to 20 reads between system and user memory. Also almost 75% of the data have less than or equal to 10 writes between system and user memory.
2. This indicates that in most of the cases where cpu mostly runs in user mode, the number of reads and writes between system and user memory stay below or equal to 20 and 10 respectively.
3. On an average, 1633 system calls of all types happen per second. 210 system read calls happen per second. 150 system write calls happen per second.
4. The average number of characters transferred per second by system read calls is pretty higher than that of system write calls ('rchar' and 'wchar' values)
5. More than half of the given data has zeroes for pgout, ppgout, pgfree, pgscan, acth.
6. On average, for the given dataset, 83% of the time CPU runs is user mode. Highest - 99% of the time CPU runs in user mode.

- Checking for null values in the data:

Rows having null values for rchar:

lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
15	3	2162	159	119	2.0	2.4	NaN	31950.0	0.0	...	0.0	1.2	6.0	9.4	150.2	220.2	Not_CPU_Bound	702	1021237	87
0	0	160	12	16	0.2	0.2	NaN	8670.0	0.0	...	0.0	0.0	0.2	0.2	15.6	16.8	Not_CPU_Bound	7248	1863704	98
5	1	330	39	38	0.4	0.4	NaN	12185.0	0.0	...	0.0	0.0	1.0	1.2	37.8	47.6	Not_CPU_Bound	633	1760253	90
0	0	1201	65	61	0.4	0.4	NaN	58703.0	0.0	...	0.0	0.0	0.0	0.0	28.4	34.4	Not_CPU_Bound	6854	1877461	96
1	0	5744	168	190	0.2	0.2	NaN	189975.0	6.0	...	0.0	4.4	0.6	0.6	27.4	28.6	Not_CPU_Bound	312	1013458	89
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
81	80	1086	134	81	0.4	0.4	NaN	NaN	3.6	...	18.2	1.6	2.0	2.2	35.4	102.8	Not_CPU_Bound	129	991770	92
7	2	2342	125	74	2.2	4.2	NaN	NaN	10.6	...	152.2	1.0	70.0	75.8	146.4	293.2	Not_CPU_Bound	175	1058875	85
1	0	645	120	69	0.4	0.6	NaN	NaN	0.0	...	0.0	0.6	2.0	3.8	37.2	56.0	CPU_Bound	1092	1728875	94
3	2	1388	68	49	0.2	0.2	NaN	NaN	0.0	...	0.0	0.0	5.4	5.4	16.0	25.4	CPU_Bound	5245	1675056	96
56	59	5526	776	598	6.2	1.6	NaN	NaN	3.8	...	0.0	0.4	3.2	5.4	294.8	454.0	CPU_Bound	292	1545926	72

ows × 22 columns

Rows having null values for wchar:

lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
14	9	3448	370	261	2.0	2.0	NaN	NaN	5.8	...	74.8	3.8	5.2	16.6	130.60	235.60	CPU_Bound	158	1026549	82
95	120	3749	281	194	1.4	1.2	NaN	NaN	0.0	...	0.0	0.4	11.0	11.0	68.80	225.60	CPU_Bound	1428	1307952	86
1	0	1084	91	63	0.2	0.2	NaN	NaN	0.0	...	0.0	0.0	1.4	1.8	32.60	25.20	CPU_Bound	622	1016328	95
37	7	1889	126	126	0.6	0.6	NaN	NaN	5.6	...	287.4	1.8	25.8	41.2	62.80	217.40	Not_CPU_Bound	238	1064165	88
42	47	1476	172	103	2.8	2.8	NaN	NaN	0.0	...	0.0	0.0	1.4	2.4	172.40	256.00	CPU_Bound	1426	1070854	86
81	80	1086	134	81	0.4	0.4	NaN	NaN	3.6	...	18.2	1.6	2.0	2.2	35.40	102.80	Not_CPU_Bound	129	991770	92
7	2	2342	125	74	2.2	4.2	NaN	NaN	10.6	...	152.2	1.0	70.0	75.8	146.40	293.20	Not_CPU_Bound	175	1058875	85
1	0	645	120	69	0.4	0.6	NaN	NaN	0.0	...	0.0	0.6	2.0	3.8	37.20	56.00	CPU_Bound	1092	1728875	94
3	2	1388	68	49	0.2	0.2	NaN	NaN	0.0	...	0.0	0.0	5.4	5.4	16.00	25.40	CPU_Bound	5245	1675056	96
56	59	5526	776	598	6.2	1.6	NaN	NaN	3.8	...	0.0	0.4	3.2	5.4	294.80	454.00	CPU_Bound	292	1545926	72
1	0	3294	271	172	1.0	2.6	569257.0	NaN	4.0	...	0.0	2.4	17.6	20.0	90.80	222.60	Not_CPU_Bound	369	1017470	85

- There are no duplicated rows in the dataset.

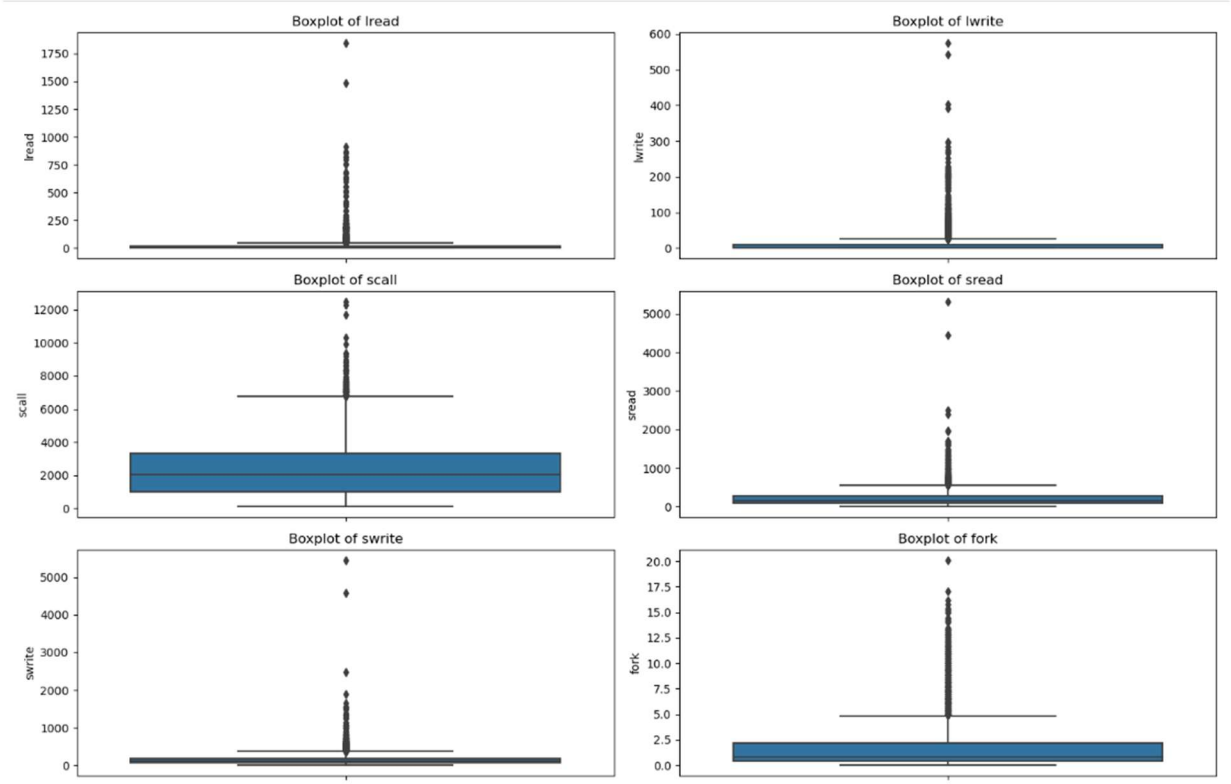
- Unique value counts for the variable ‘runqsz’ are shown below:

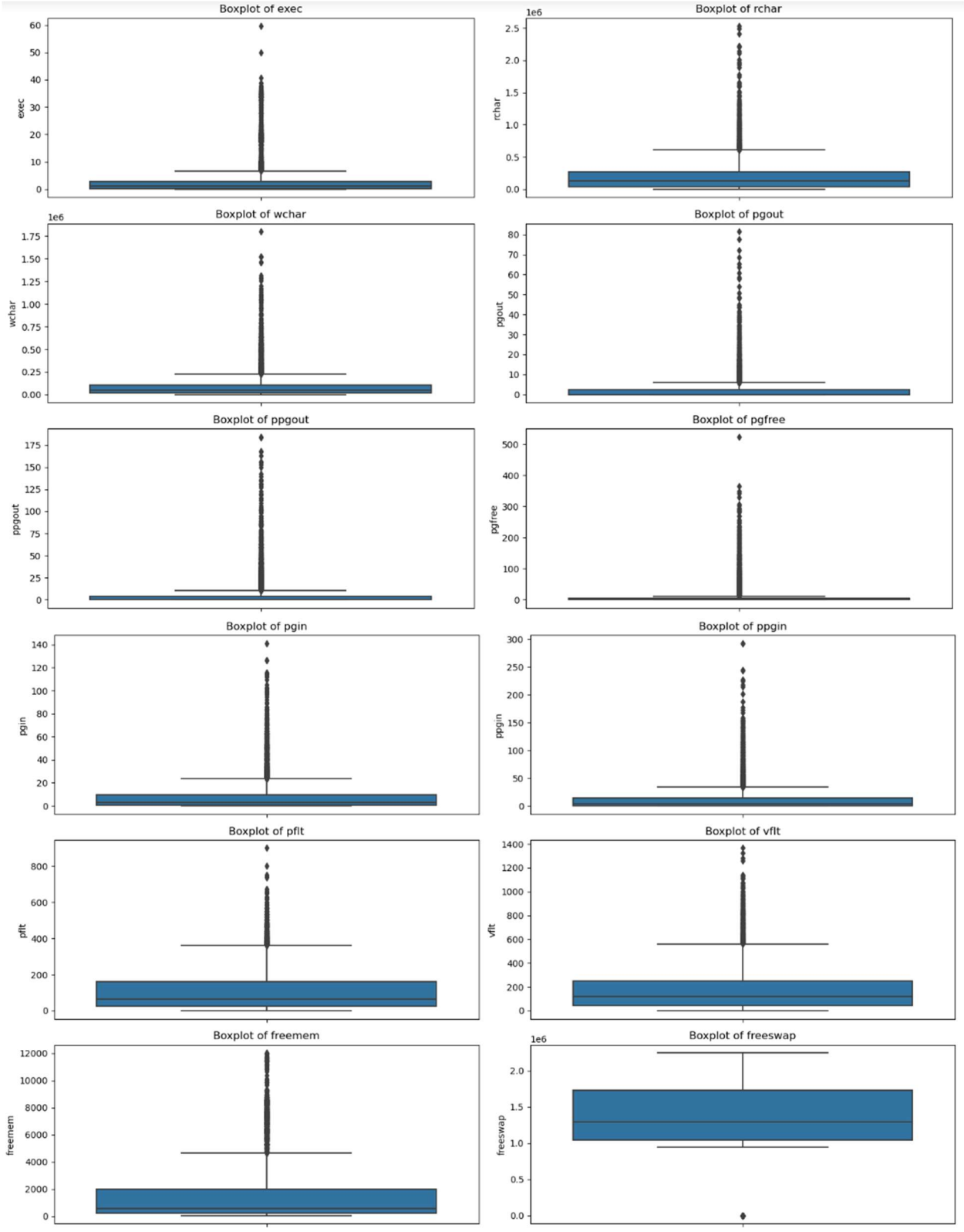
```
Not_CPU_Bound    4331
CPU_Bound        3861
Name: runqsz, dtype: int64
```

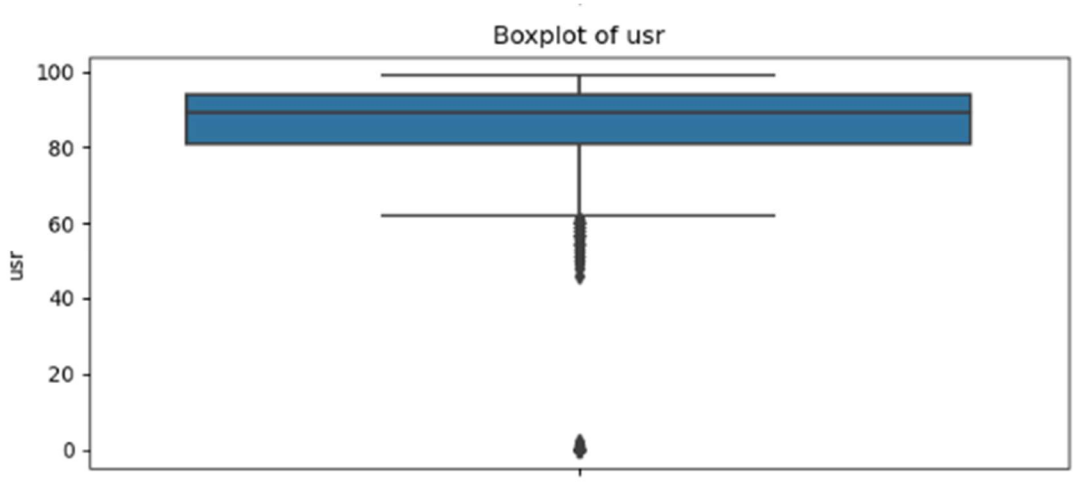
- There are no duplicated rows in the dataset.

Univariate analysis:

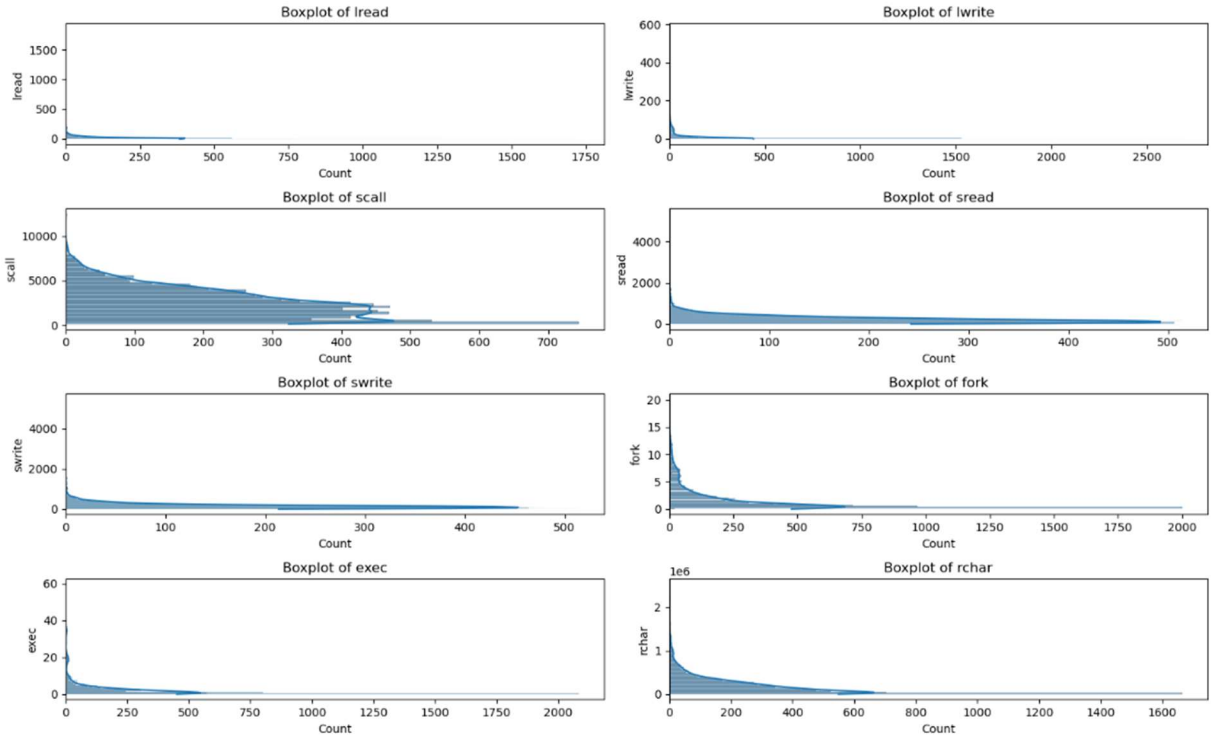
Since all the variables in the data set are numerical variables, examining the boxplots for all the variables:

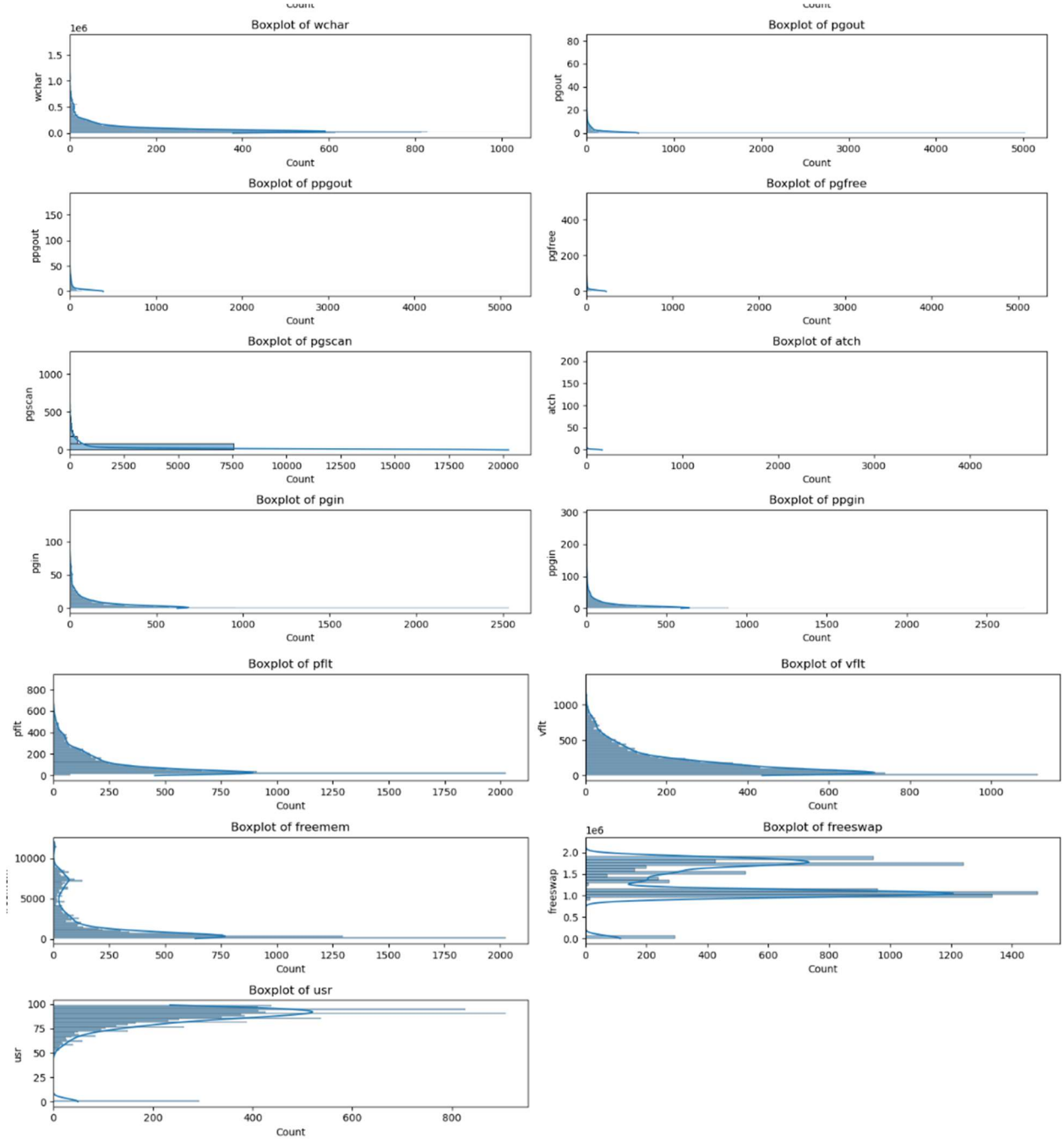



Boxplot of exec
exec
Boxplot of rchar
1e6
rchar
Boxplot of wchar
1e6
wchar
Boxplot of pgout
pgout
Boxplot of ppgout
ppgout
Boxplot of pgfree
pgfree
Boxplot of pgin
pgin
Boxplot of ppgin
ppgin
Boxplot of pflt
pflt
Boxplot of vflt
vflt
Boxplot of freemem
freemem
Boxplot of freeswap
1e6
freeswap

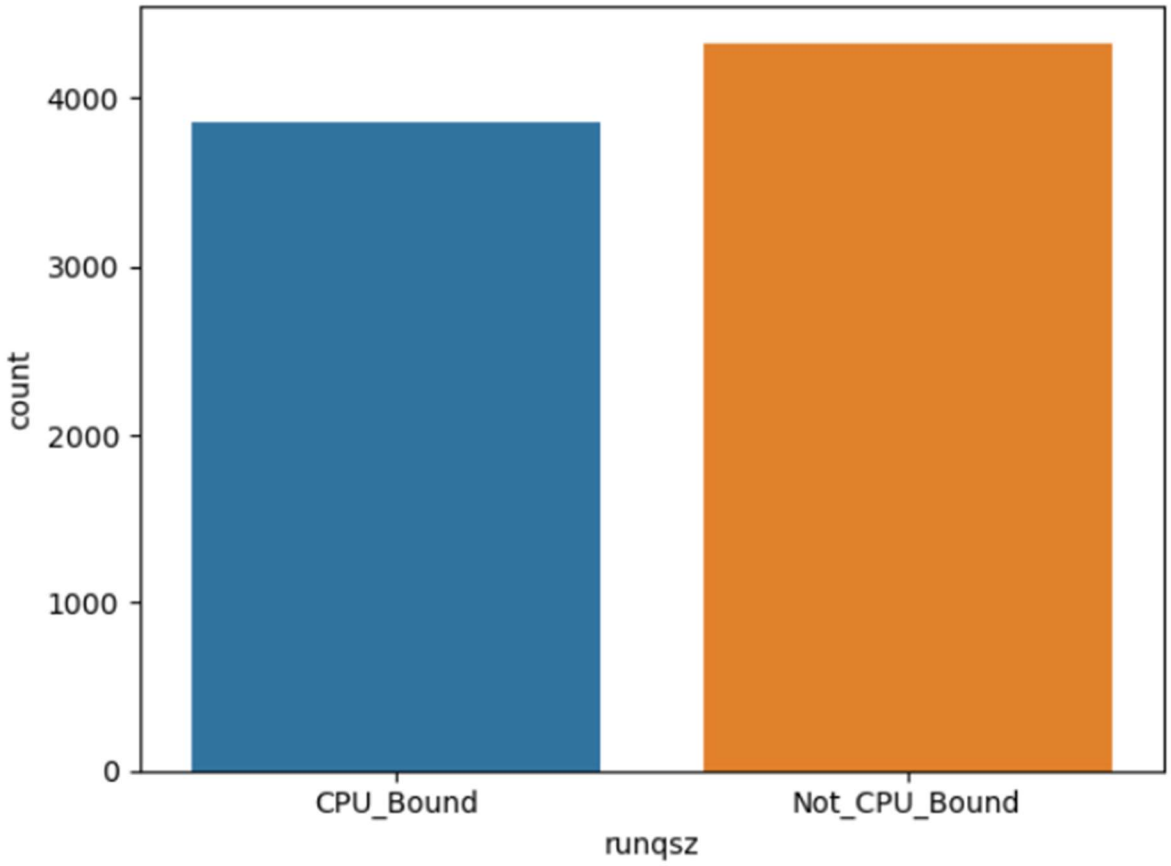


Plotting the distributions of all variables with kernel density estimate:




Boxplot of wchar
1e6
wchar
Count
Boxplot of pgout
pgout
Count
Boxplot of ppgout
ppgout
Count
Boxplot of pgfree
pgfree
Count
Boxplot of pgscan
pgscan
Count
Boxplot of atch
atch
Count
Boxplot of pgin
pgin
Count
Boxplot of ppgin
ppgin
Count
Boxplot of pflt
pflt
Count
Boxplot of vflt
vflt
Count
Boxplot of freemem
Count
Boxplot of freeswap
1e6
freeswap
Count
Boxplot of usr
usr
Count

Analyzing the counts of categorical variable ‘runqsz’:

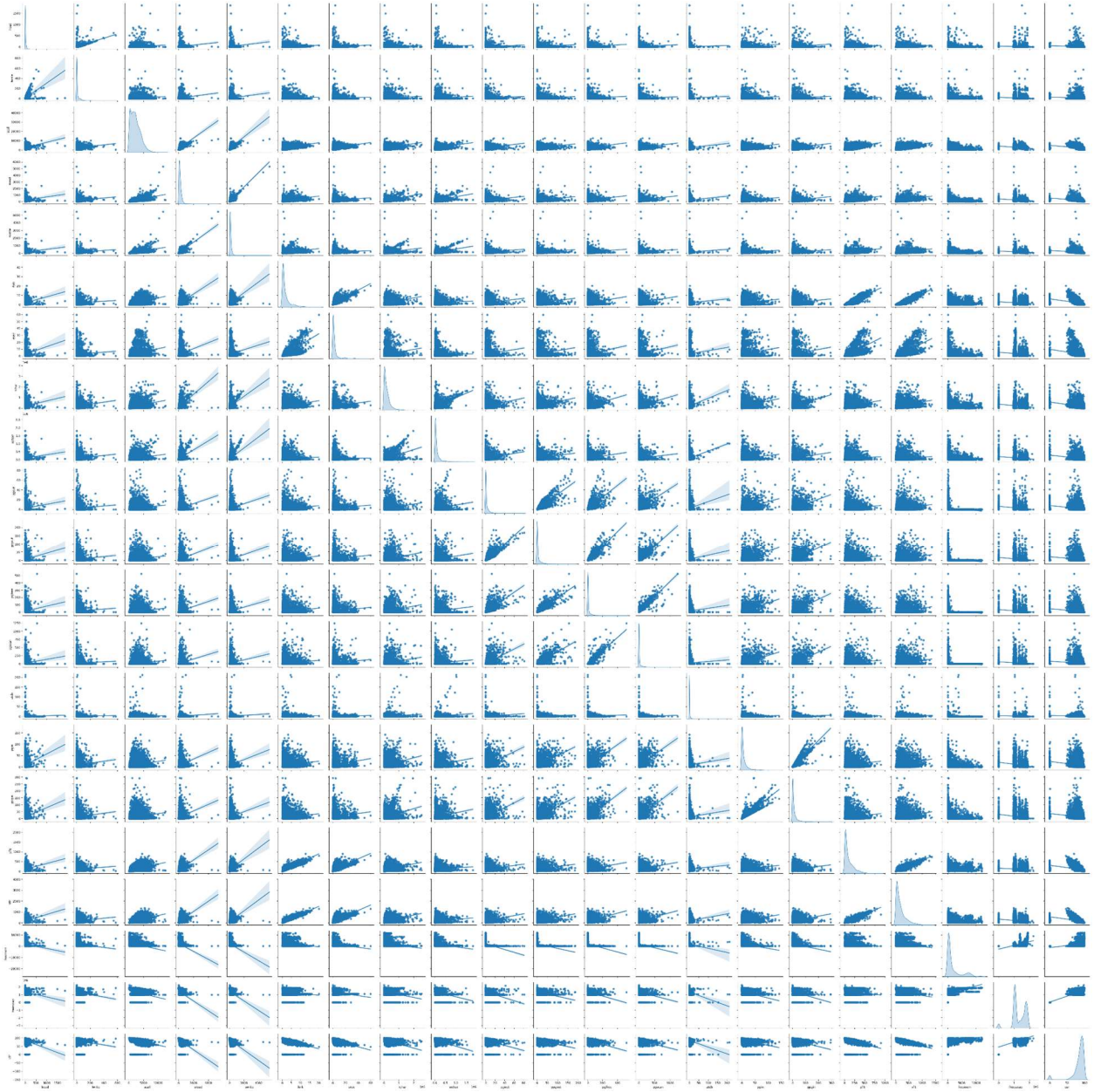


Insights:

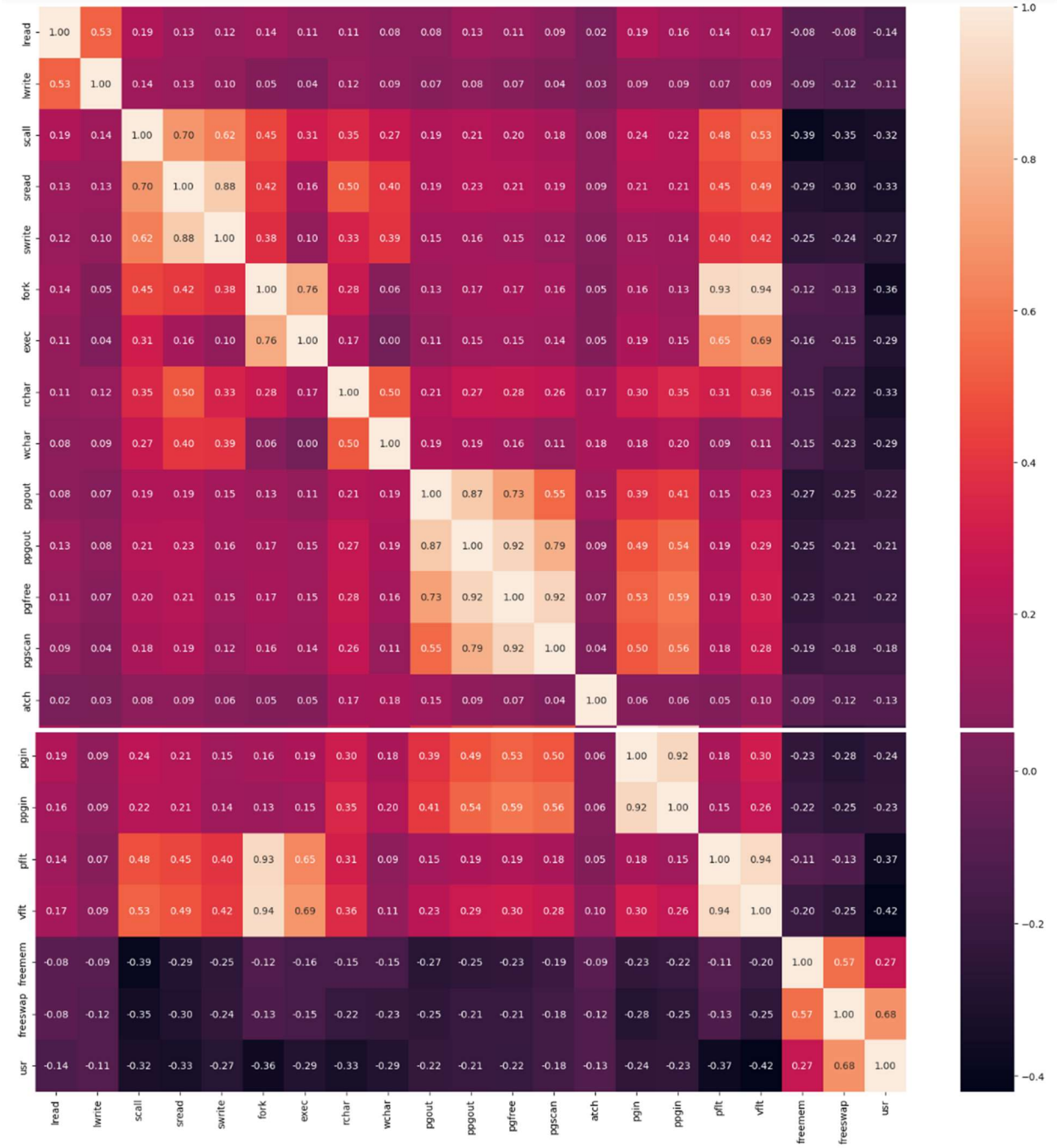
1. From the boxplots, most of the numeric variables seem to have outliers.
2. Most of the independent variables are skewed.
3. Average transfers between system user memory and system memory lie in the range 13 - 19. 13 write calls and 19 read calls.
4. Maximum number of system calls per second are 12493 with average being around 2300 per second.
5. Number of characters transferred per second by system read calls on an average are 197385+ and by write calls are 95902+
6. More than half of the page out requests, pageouts, pages free, pages scanned, pages attached per second are zero.
7. On an average from the given data, 83% of time cpu runs is user mode with 97% of the values lying between 3 standard deviations from the mean.
8. Based on the queue size, system is often not CPU Bound than CPU Bound/

Bivariate analysis:

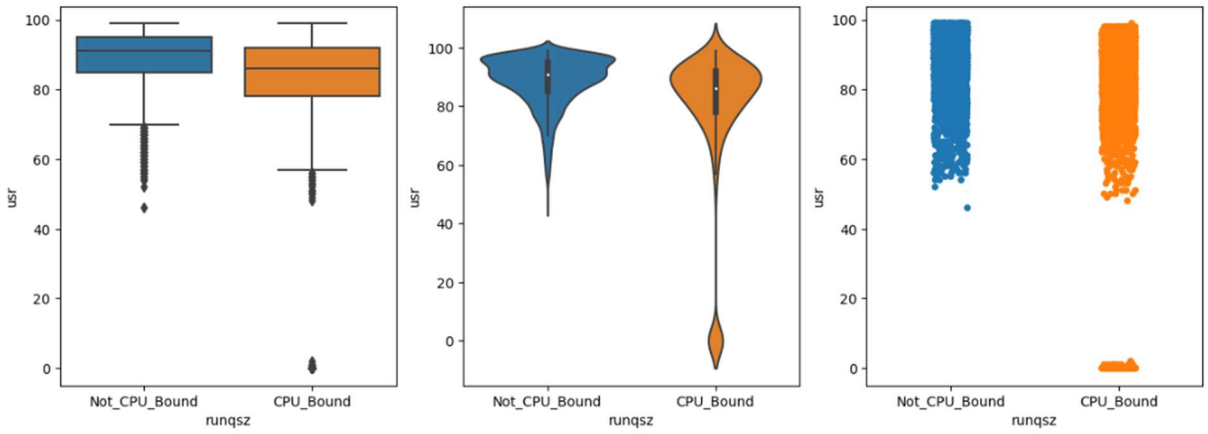
Analyzing the pair plots of all numeric variables:



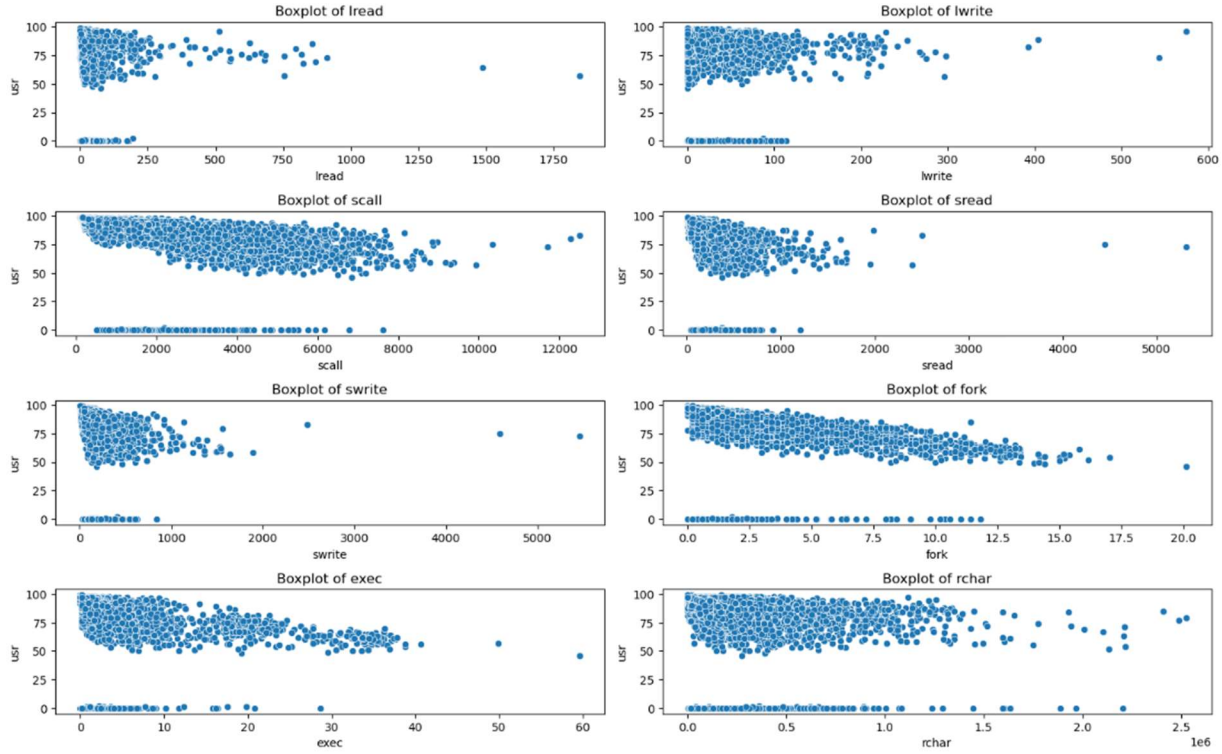
Correlation maps for all numeric variables:

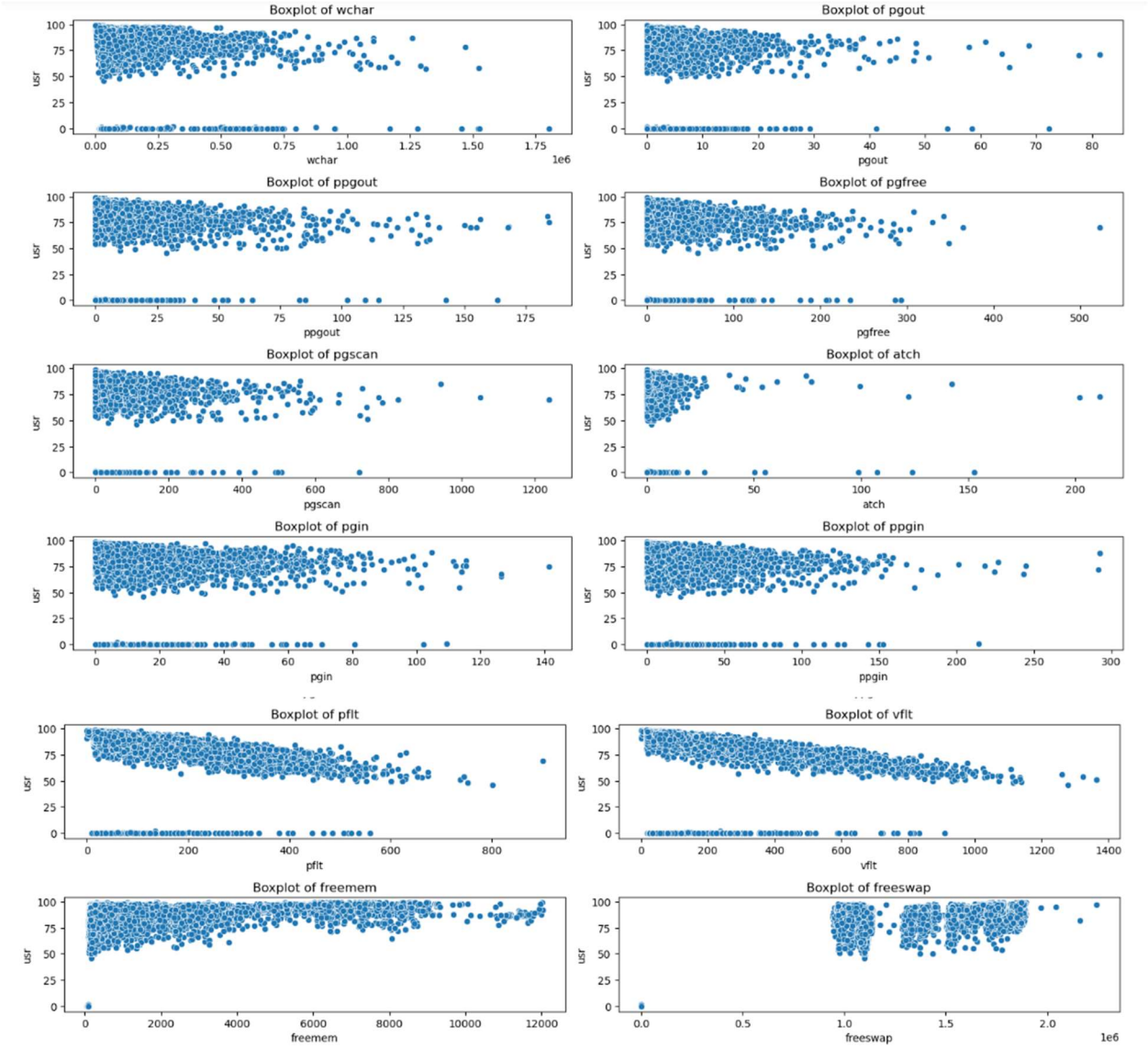

lread lwrite scall sread swrite fork exec rchar wchar pgout ppgout pgfree pgscan atch pgin ppgin pflt vflt freemem freeswap usr
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4

Analyzing the percent of time cpu runs in user mode for different runqsz categories:



Analyzing correlations of independent variables with dependent variable:





Insights:

1. CPU Bound characteristic leads the system to run for less amount of time in user mode than the Non-CPU Bound characteristic.
2. From the correlation map and the plots, the variables which tend to have high correlation with the dependent variable are:

Weak negative correlation: (< 0 and >- 0.20):

- Read, write, pgscan, acth,

Moderate Negative correlation: (< - 0.20 and > -0.40)

- Scall, sread, swrite, fork, exec, rchar, wchar, pgout, ppgout, pgfree, pgin ppgin, pflt

Strong negative correlation: (< -0.40)

- Vflt

Moderate positive correlation: Freemem

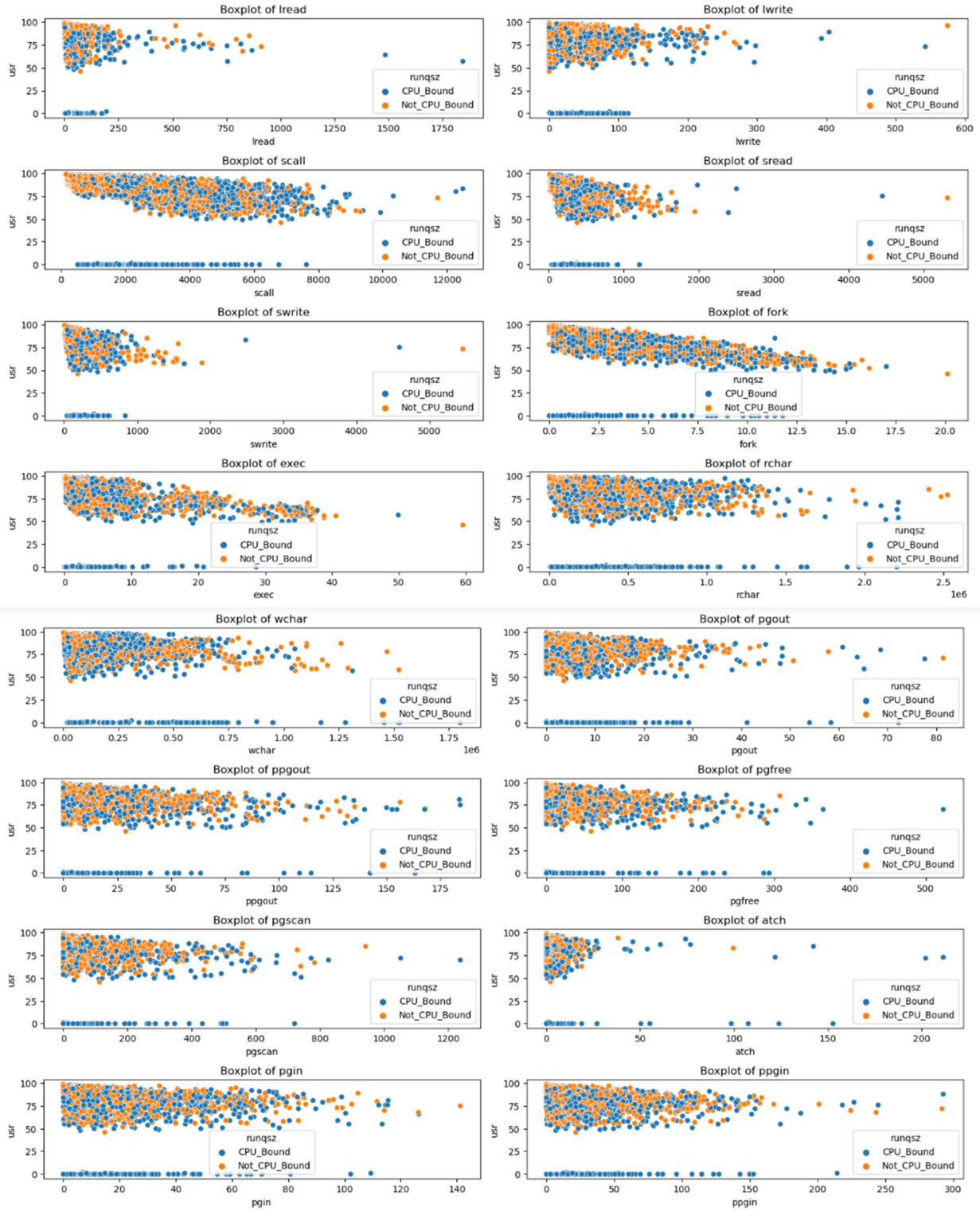
Strong positive correlation: freeswap

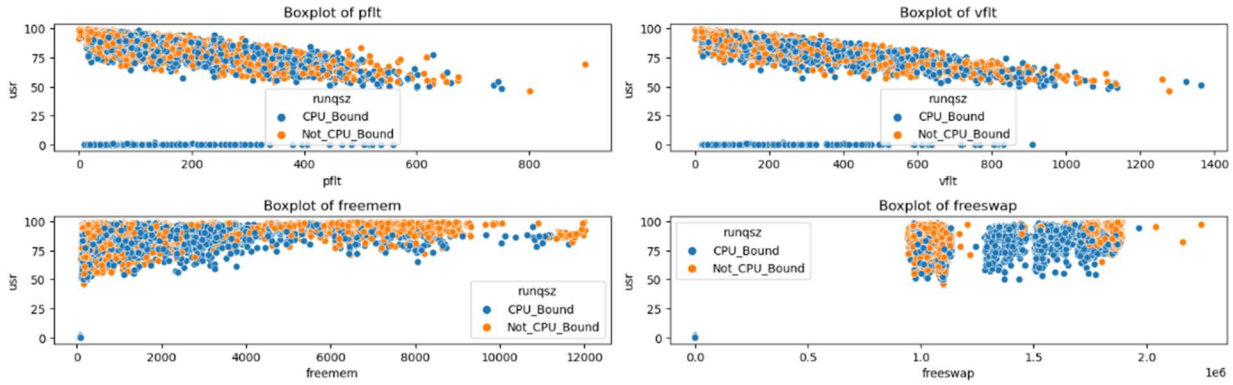
3. Correlations among independent variables:

High positive correlations: (derived from heatmap shown above)

lRead with lwrite
Scall with sread
Scall with swrite
Scall with fork (moderate)
Scall with vflt, pflt
sread with swrite (0.88)
Fork with exec (0.76)
Fork with pflt and vflt
Exec with pflt and vflt
Rchar with wchar
pgout with ppgout and free
Ppgout with pgfree and pgscan
Pgfree with pgscan
Pgin with ppgin
Vflt with pflt
Freemem with freeswap

Multivariate analysis:





Insights:

1. There is no clear separation for the independent variables based on queue size
2. Similar correlations are exhibited by system characteristics which are CPU Bound and which are not CPU Bound
3. Scall, fork, exec, pflt and vflt are the variables showing negative correlation with the target variable.

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.

Ans:

Checking for null values to impute if present:

- There are 104 null values present in rchar and 15 in wchar columns.
- Rchar is the number of characters transferred by read calls per second
- Wchar is the number of characters transferred by write calls per second.

As the number of system read and write calls is not zero in the missing value cases and most of the sread and swrite calls we cannot assume that the number of characters transferred per second by these calls to be zero.

Hence, imputing the missing values of rchar and wchar values with the mean values of their respective columns.

Checking for number of zeros in all columns:

```
Count of zeros for  lread is  675
Count of zeros for  lwrite is  2684
Count of zeros for  scall is  0
Count of zeros for  sread is  0
Count of zeros for  swrite is  0
Count of zeros for  fork is  21
Count of zeros for  exec is  21
Count of zeros for  rchar is  0
Count of zeros for  wchar is  0
Count of zeros for  pgout is  4878
Count of zeros for  ppgout is  4878
Count of zeros for  pgfree is  4869
Count of zeros for  pgscan is  6448
Count of zeros for  atch is  4575
Count of zeros for  pgin is  1220
Count of zeros for  ppgin is  1220
Count of zeros for  pflt is  3
Count of zeros for  vflt is  0
Count of zeros for  freemem is  0
Count of zeros for  freeswap is  0
Count of zeros for  usr is  283
```

There are a total of 8192 rows. The columns pgout, pggout, pgfree, pgscan, acth have zeroes for more than half of their observations.

It means that more than half of observations have

Number of pageout requests 0 : It means that most of the time the page is stored on the RAM instead of on the harddisk.

Number paged out pages 0: This makes sense to have more zeroes as most there are no page out requests for most of the observations.

Number of pages placed on the free list 0: most of the times there are no free pages placed idle in the memory per second.

Number of pages checked if they can be scanned to place on free list 0: most of the time there are no free pages to be scanned to be placed idle in the memory per second.

Number of page attaches per second 0: less number of page attaches to memory.

These columns can be chosen to be dropped. However, these columns if insignificant result in their co-efficients to be zero. We will examine these columns in built in linear regression models.

Checking for the possibility of creating new features if required:

Since there are high correlations between some pairs of individual variables, we can add a couple of columns into a single column, drop the two independent columns and build a regression model on the modified dataset.

Scall has high correlations with other variables, hence this variable can be dropped from our dataset and build the model.

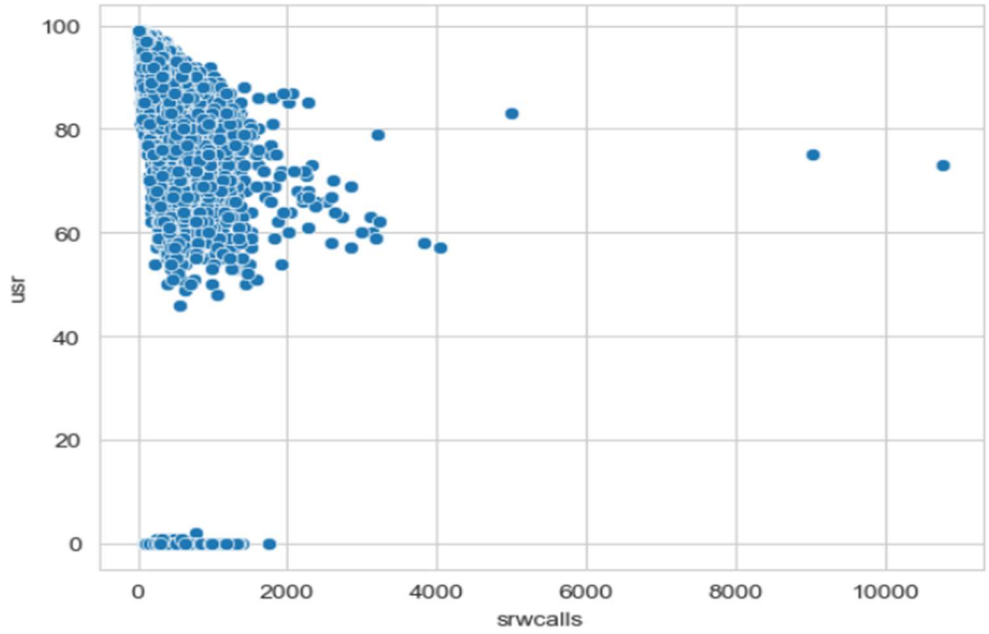
sread and swrite have highest correlation with each other. They can be added into a single column and fed into the model.

Other independent variables have high correlations with vflt and pflt. Vflt and pflt are faults that can be added together as a single column and fed into the model to reduce multi-collinearity to some extent.

This is the feature-engineered dataset from which the model to be built for evaluation:

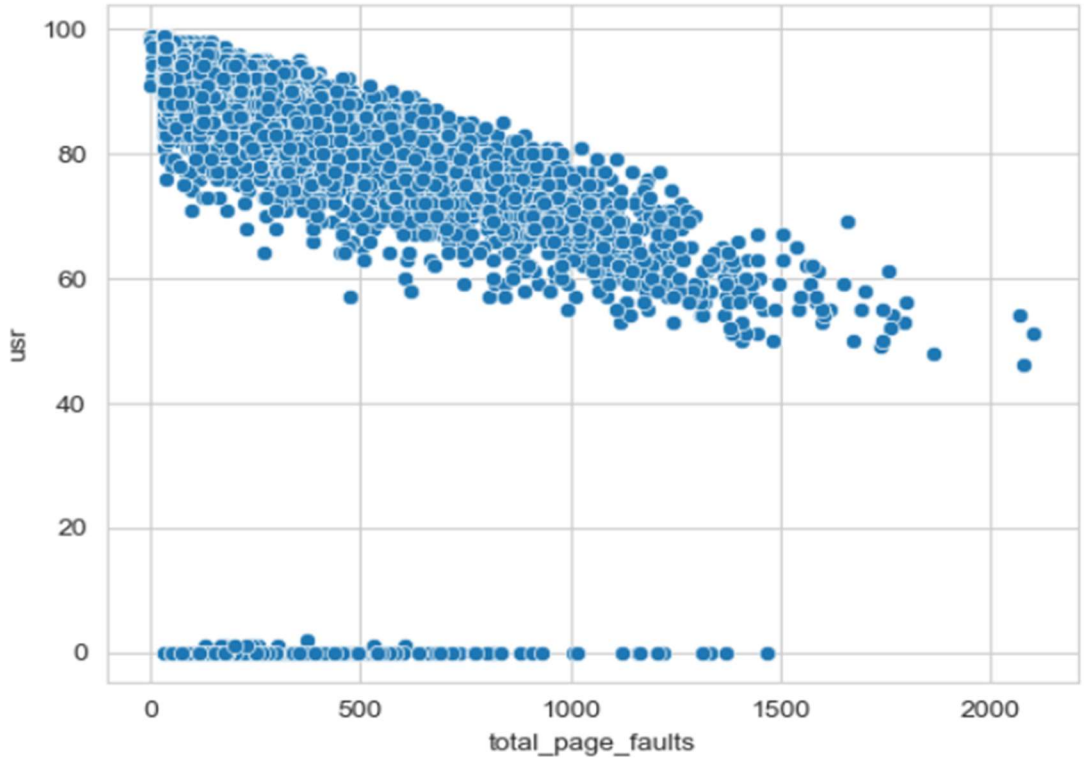
	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
0	1	0	2147	79	68	0.2	0.2	40671.00	53995.0	1.6	2.6	16.00	26.40	CPU_Bound	4670	1730946	95
1	0	0	170	18	21	0.2	0.2	448.00	8385.0	0.0	0.0	15.63	16.83	Not_CPU_Bound	7278	1869002	97
2	15	3	2162	159	119	2.0	2.4	197385.73	31950.0	6.0	9.4	150.20	220.20	Not_CPU_Bound	702	1021237	87
3	0	0	160	12	16	0.2	0.2	197385.73	8670.0	0.2	0.2	15.60	16.80	Not_CPU_Bound	7248	1863704	98
4	5	1	330	39	38	0.4	0.4	197385.73	12185.0	1.0	1.2	37.80	47.60	Not_CPU_Bound	633	1760253	90

Distribution of systemreadwrite calls with respect to ‘usr’ dependent variable:



Very mild correlation exists with independent variable.

Distribution of totalpagefaults with dependent variable:



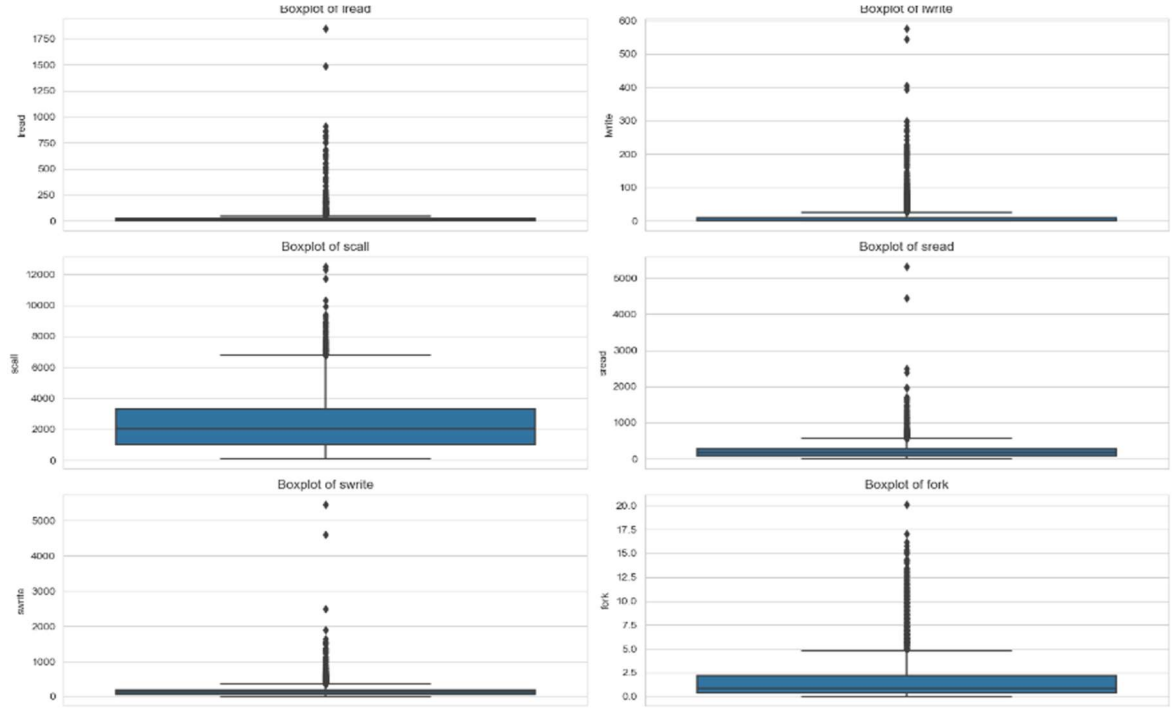
The variable total_page_faults show moderate to strong negative correlation.

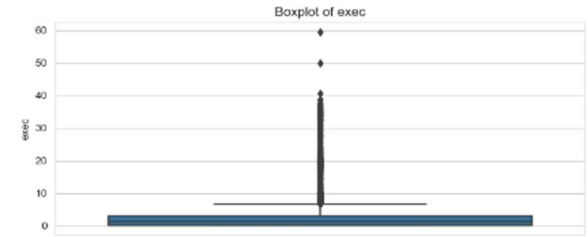
Checking for outliers:

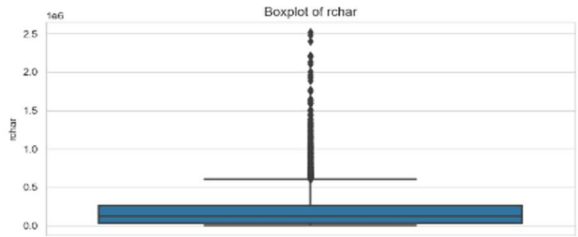
From the box plots depicted in univariate analysis, it is evident that there are outliers existing for most of the independent variables. Since linear regression is sensitive to outliers, we shall remove the outliers by capping them to lower quartile and upper quartile values. However, we will build linear regression models by feeding the data with and without outliers.

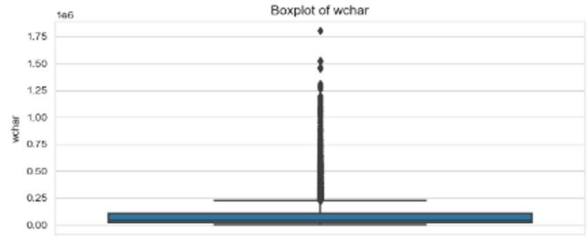
Outlier treatment is done by capping the values to minimum and maximum values.

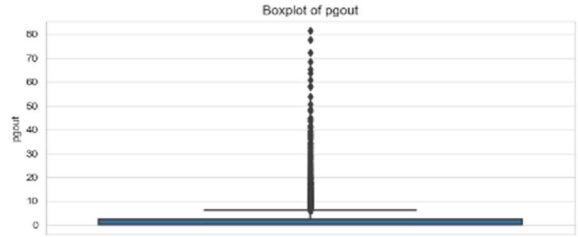
Boxplots of the original independent variables before outlier treatment:

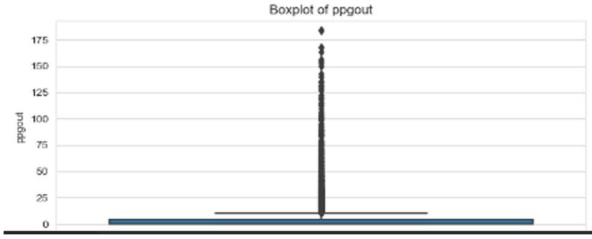


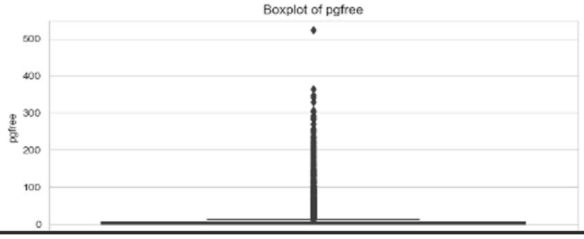

Boxplot of exec
exec

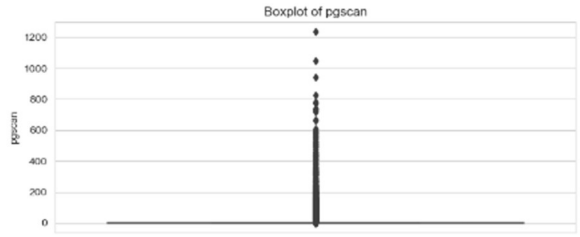

Boxplot of rchar
1e6
rchar

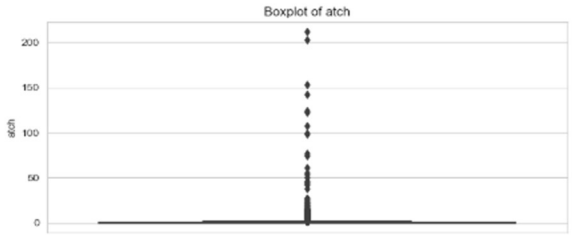

Boxplot of wchar
1e6
wchar

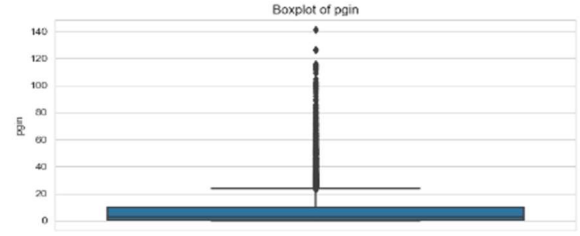

Boxplot of pgout
pgout

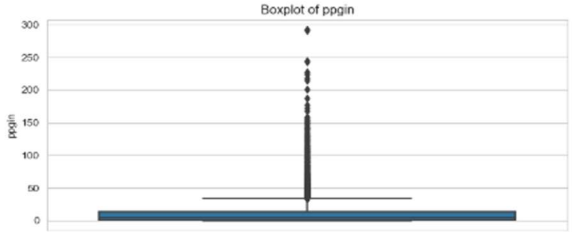

Boxplot of ppgout
ppgout

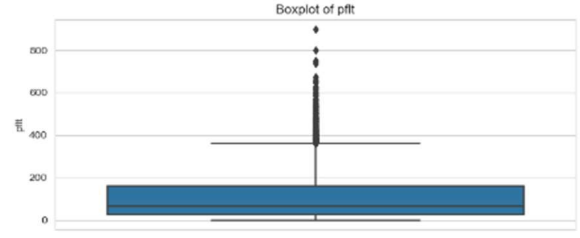

Boxplot of pgfree
pgfree

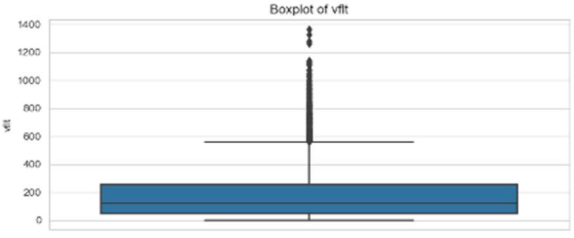

Boxplot of pgscan
pgscan

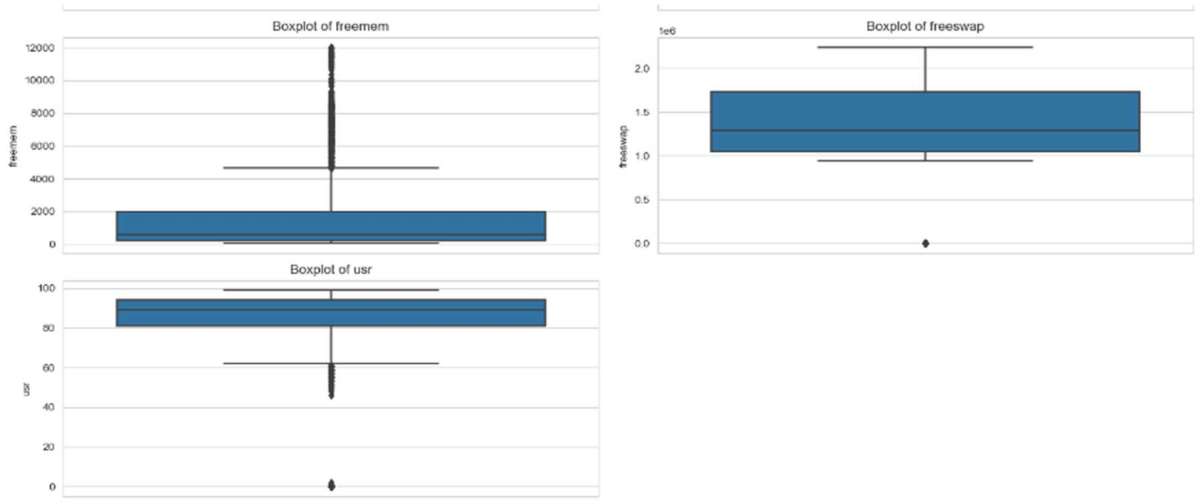

Boxplot of atch
atch


Boxplot of pgin
pgin

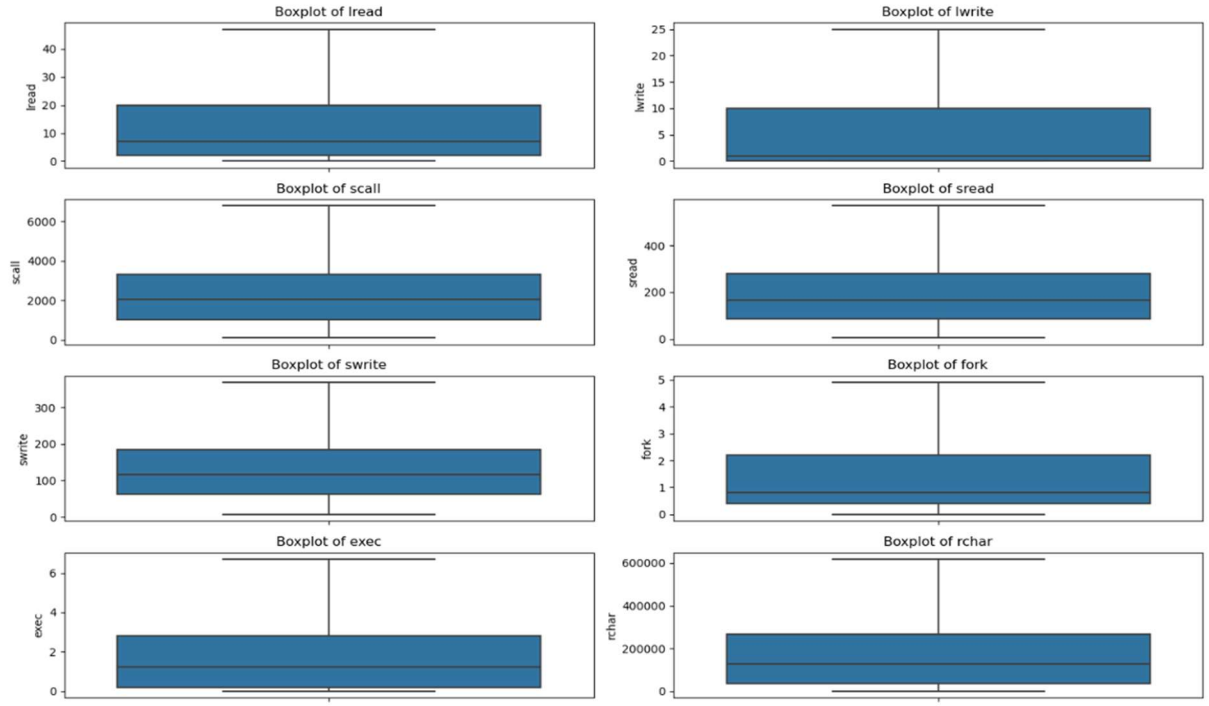

Boxplot of ppgin
ppgin

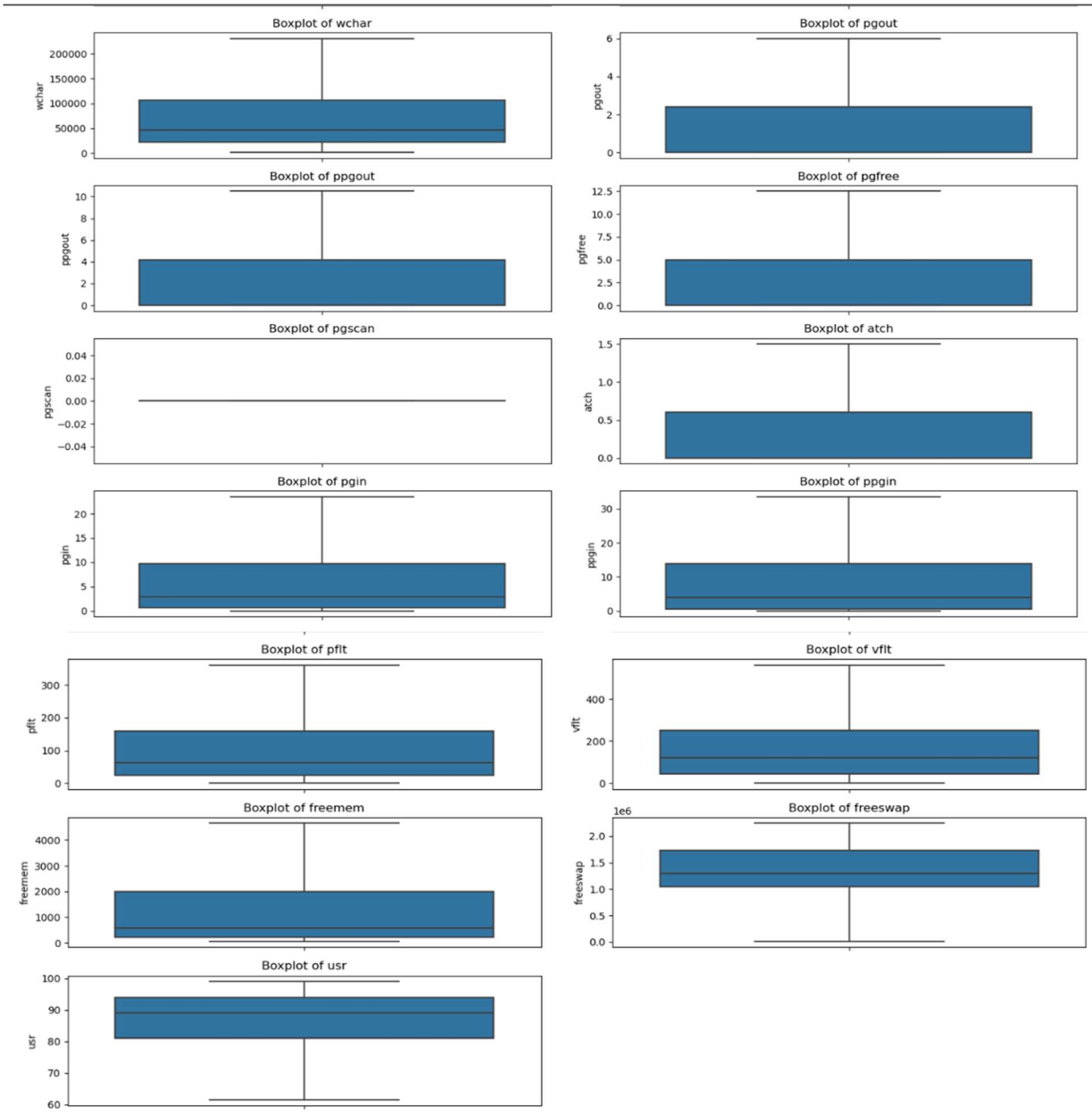

Boxplot of pflt
pflt


Boxplot of vflt
vflt

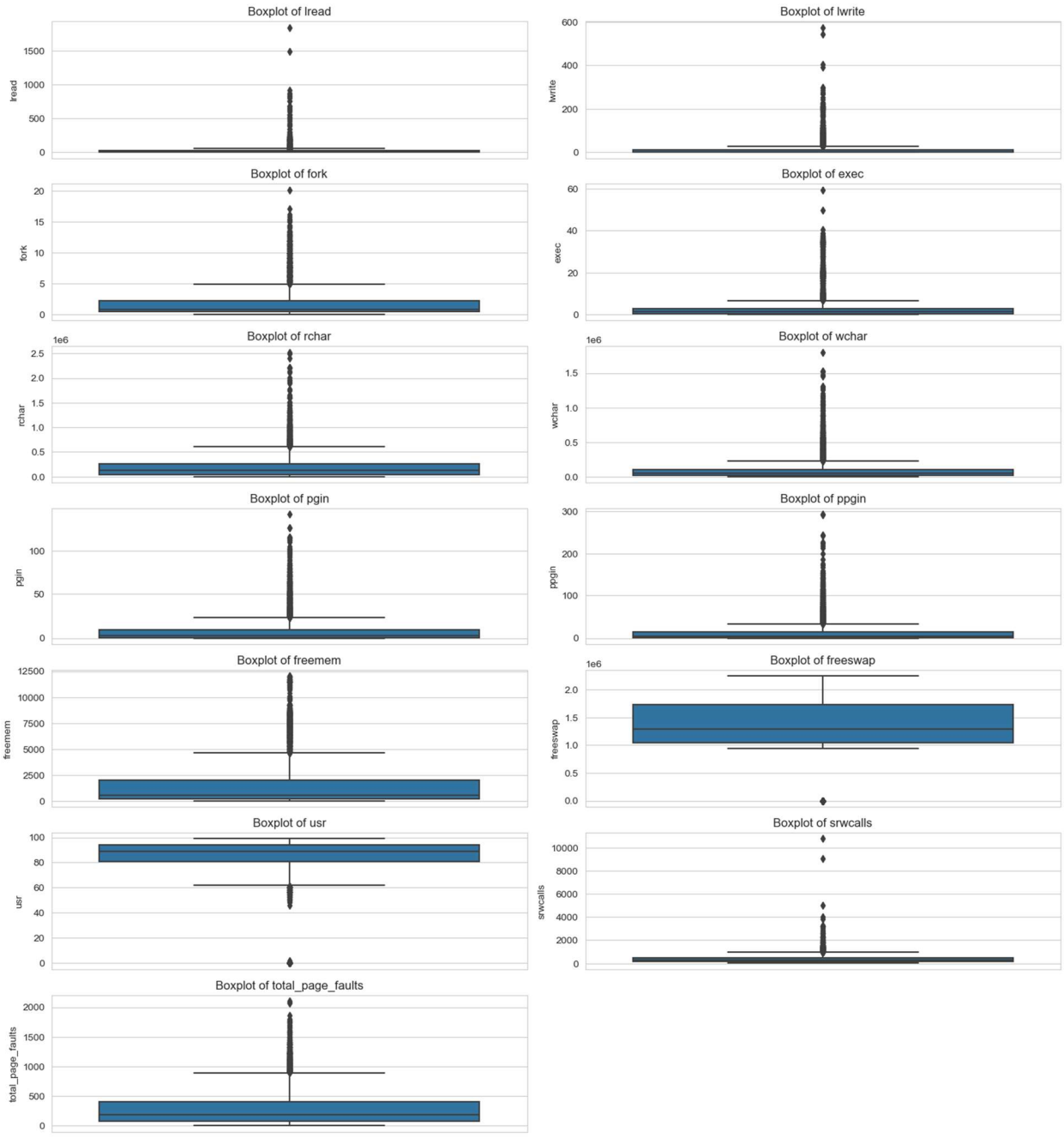


Boxplots of the original independent variables before outlier treatment:

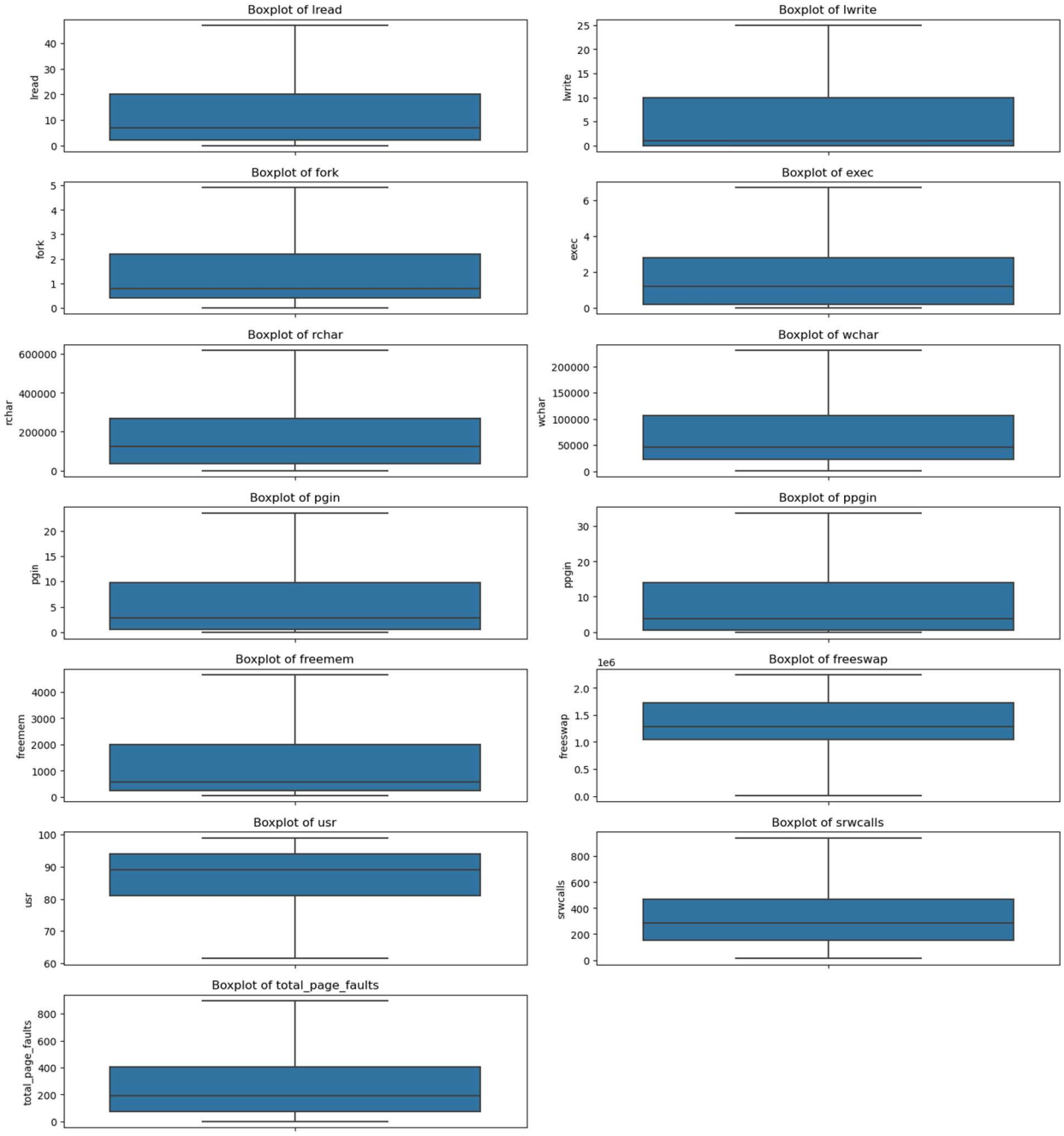




Boxplots for the feature engineered dataset before treating outliers:



Boxplots for the feature engineered dataset after treating outliers:



Checking for duplicated rows:

There are no duplicates present in original dataset as well as feature engineered dataset.

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.

1.3.1 Encoding 'runqsz' column since it has categorical values to be able to pass it to the model:

According to business understanding:

- CPU_Bound indicates that typically the process queue size is more than 2
- Therefore, to get a better business sense, encoding CPU_Bound category with label 2 and Non_CPU_Bound category with label 1.

Value counts for 'runqsz' before encoding:

```
Not_CPU_Bound    4331
CPU_Bound        3861
Name: runqsz, dtype: int64
```

Value counts for 'runqsz' after encoding:

```
1    4331
2    3861
Name: runqsz, dtype: int64
```

Encoded the categorical variable in all the 3 dataframes.

1.3.2 Split the data into train and test (70:30)

The dataset has been split into train and test datasets with 70% of the data in the training set and 30% of the data in the testing set. Random state specified is 1.

Displaying the first 5 rows of independent variables of training set:

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	ppgout	pgfree	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap
694	1	1	1345	223	192	0.6	0.6	198703.0	293578.0	0.60	6.20	23.40	3.80	7.40	28.20	56.60	2	121	1375446
5535	1	1	1429	87	67	0.2	0.2	7163.0	24842.0	0.00	0.00	0.00	1.60	1.60	15.77	30.74	1	1476	1021541
4244	49	71	3273	225	180	0.6	0.4	83246.0	53705.0	5.39	7.19	7.19	3.99	4.59	59.88	74.05	2	82	18
2472	13	8	4349	300	191	2.8	3.0	96009.0	70467.0	0.00	0.00	0.00	2.80	3.20	129.00	236.80	2	772	993909
7052	17	23	225	13	13	0.4	1.6	17132.0	12514.0	0.00	0.00	0.00	0.00	0.00	19.80	23.80	1	4179	1821682

Displaying the first 5 rows of the predictor variable in training dataset:

	usr
694	91
5535	94
4244	0
2472	83
7052	94

Displaying the first 5 rows of independent variables in test data set

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	ppgout	pgfree	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap
3894	27	39	1252	53	118	0.2	0.2	26592.0	54394.0	0.0	0.0	0.0	0.4	0.6	19.44	20.04	1	7762	1875466
4276	1	0	996	85	55	0.4	0.4	16667.0	36431.0	0.0	0.0	0.0	1.0	1.4	35.53	52.10	1	2979	1010114
3414	9	7	1530	247	135	0.4	0.4	14513.0	61905.0	13.8	19.2	30.4	14.8	18.4	26.80	186.20	2	89	11
4165	32	4	3243	182	140	5.2	5.6	337517.0	94832.0	0.8	1.0	1.0	4.6	7.0	250.60	420.20	2	1300	1535309
7385	16	3	5017	259	249	2.8	1.4	73537.0	237547.0	0.0	0.0	0.0	5.6	5.8	142.80	276.20	2	2114	988600

Displaying first 5 rows of dependent variables in test data set:

	usr
3894	95
4276	95
3414	0
4165	80
7385	79

1.3.3 Apply Linear regression using scikit learn

The dataset has been imputed to remove null values.
Dropped the ‘usr’ variable into another dataframe and named ‘y’
Dataframe having all the independent variables is names ‘X’

Applying linear regression on original processed model:

R-squared obtained on the trained model for training data: 0.643
R-squared obtained on the trained model for test data: 0.631

Insights:

- Accuracy of the model on training and test dataset is decent but can be improved.
- The built model explains 64% of the variance in the data.
- The accuracies of training and test are similar. This indicates that the model has not overfitted.

Root mean squared error (RMSE) for the training data: 10.81
Root mean squared error (RMSE) for test data: 11.59

Insights:

- The difference between predicted values and actual values is high.
- The residuals differ more in case of testing data than on trained data.

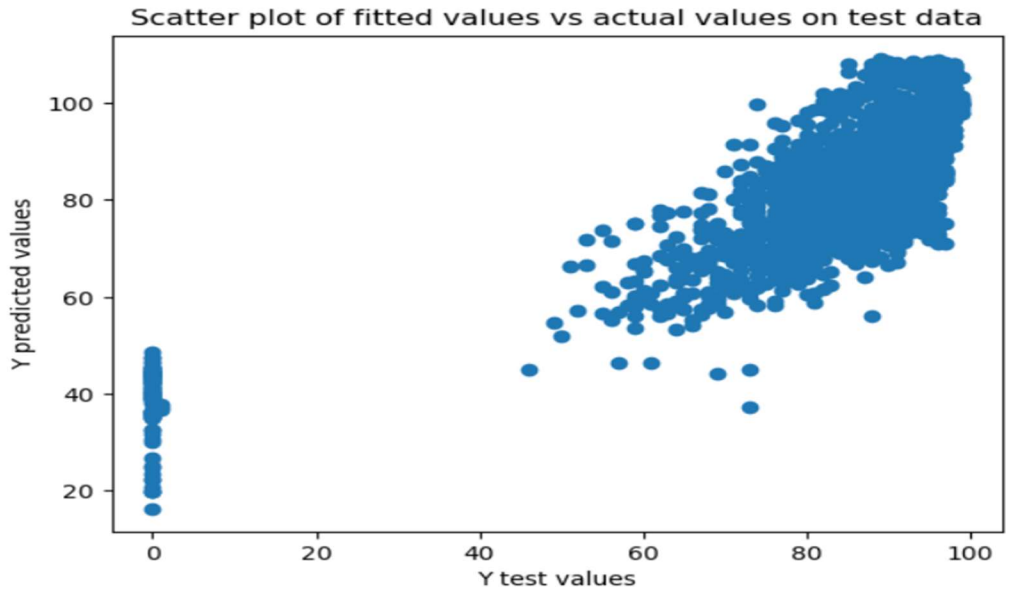
Mean absolute error for training data: 7.77
Mean absolute error for test data: 8.17

Adjusted r-sqaured on the trained model for training data: 0.64
Adjusted r-sqaured on the trained model for test data: 0.63

Insights:

- Adjusted r-sqaured value comes into picture when there is multicollinearity in the independent variables.
- In this case, adjusted r-sqaured value is also similar for training and test data but less accurate.

Scatter plot showing the distribution of actual values vs predicted values:



Coefficients of independent variables for the regression model:

[-1.98982426e-02, 4.82254950e-03, 1.00783287e-03,
-4.29251108e-04, -2.07850528e-03, -1.72163526e+00,
-8.96257233e-02, -4.11424988e-06, -1.16031000e-05,
-1.74144052e-01, 9.89642463e-02, -7.02837829e-02,
 8.61101010e-03, -7.82968598e-02, 9.13688023e-02,
-5.93593716e-02, -4.15026113e-02, 2.22821368e-02,
-7.78836881e+00, -1.61663832e-03, 3.21908454e-05]

These coefficients are mapped to the columns considered in the trained data correspondingly.

Intercept for the above regression model: 60.22

Summary: From the regression model built using sklearn, the accuracy turned out to be 64% which can be improved by evaluating the model with other outlier treated dataframes and other regression approaches

1.3.4 Perform checks for significant variables using appropriate method from statsmodel**.**

Linear regression model for the same dataset using statsmodel has been applied on the dataset.

Displaying the summary of the model:

OLS Regression Results

Dep. Variable:	usr	**R-squared:**	0.643
Model:	OLS	**Adj. R-squared:**	0.642
Method:	Least Squares	**F-statistic:**	489.6
Date:	Sat, 29 Apr 2023	**Prob (F-statistic):**	0.00
Time:	19:35:43	**Log-Likelihood:**	-21787.
No. Observations:	5734	**AIC:**	4.362e+04
Df Residuals:	5712	**BIC:**	4.377e+04
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	60.2236	0.776	77.637	0.000	58.703	61.744
lread	-0.0199	0.003	-6.217	0.000	-0.026	-0.014
lwrite	0.0048	0.006	0.799	0.424	-0.007	0.017
scall	0.0010	0.000	7.449	0.000	0.001	0.001
sread	-0.0004	0.002	-0.234	0.815	-0.004	0.003
swrite	-0.0021	0.002	-1.037	0.300	-0.006	0.002
fork	-1.7216	0.244	-7.050	0.000	-2.200	-1.243
exec	-0.0896	0.048	-1.879	0.060	-0.183	0.004
rchar	-4.114e-06	8.29e-07	-4.961	0.000	-5.74e-06	-2.49e-06
wchar	-1.16e-05	1.28e-06	-9.091	0.000	-1.41e-05	-9.1e-06
pgout	-0.1741	0.064	-2.721	0.007	-0.300	-0.049
ppgout	0.0990	0.037	2.702	0.007	0.027	0.171
pgfree	-0.0703	0.020	-3.505	0.000	-0.110	-0.031
pgscan	0.0086	0.006	1.361	0.174	-0.004	0.021
atch	-0.0783	0.027	-2.939	0.003	-0.131	-0.026

pgin	0.0914	0.029	3.107	0.002	0.034	0.149
ppgin	-0.0594	0.019	-3.127	0.002	-0.097	-0.022
pflt	-0.0415	0.004	-9.696	0.000	-0.050	-0.033
vflt	0.0223	0.003	6.665	0.000	0.016	0.029
runqsz	-7.7884	0.303	-25.684	0.000	-8.383	-7.194
freemem	-0.0016	7.53e-05	-21.482	0.000	-0.002	-0.001
freeswap	3.219e-05	4.53e-07	70.984	0.000	3.13e-05	3.31e-05

Omnibus:	1507.116	**Durbin-Watson:**	2.057
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4767.078
Skew:	-1.333	**Prob(JB):**	0.00
Kurtosis:	6.584	**Cond. No.**	7.80e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.8e+06. This might indicate that there are strong multicollinearity or other numerical problems.

Interpreting the summary of linear regression model and the significant variables:

- R-sqaured from the above table is obtained as: 0.643
- This indicates that the linear regression model built using ols was able to explain 64% of the variance in the data. In other words, accuracy is 64%
- Adjusted r-squared: The intention of this metric is similar to r-squared with the exception that the adjusted r-sqaured value gets adjusted with the multi-collinear independent variables.
- F-statistic is calculated using the formula: (SSR/DFssr)/(SSE/DFsse)
    - The following represents the hypothesis test for the linear regression model:
    - H0: $\beta1 = \beta2 = \beta3 = 0$
    - Ha: At least one of the coefficients is not equal to zero.
    - The probability of f-statistic is 0 which is less than 0.05, with 95% confidence we can say reject the null hypothesis. That means, we can say that at least one of the coefficients is not equal to 0.
- Lower AIC, BIC scores indicate a better model. These scores indicate the loss in data while training the model.

- The coefficients of the linear regression equation obtained from the model can be interpreted as below:

(60.224) * const + (-0.02) * lread + (0.005) * lwrite + (0.001) * scall + (-0.0) * sread + (-0.002) * swrite + (-1.722) * fork + (-0.09) * exec + (-0.0) * rchar + (-0.0) * wchar + (-0.174) * pgout + (0.099) * ppgout + (-0.07) * pgfree + (0.009) * pgscan + (-0.078) * atch + (0.091) * pgin + (-0.059) * ppgin + (-0.042) * pflt + (0.022) * vflt + (-7.788) * runqsz + (-0.002) * freemem + (0.0) * freeswap

- The values in the columns [0.025] and [0.975] indicate that the actual values of coefficients lie between the values in the respective columns with 95% confidence.
    - For example:
        - The coefficient of lread will lie in the interval [-0.026, -0.014] with 95% confidence.
- Interpreting significant coefficients using the model:
    - The coefficients tell us how one unit change in X can affect y.
    - The sign of the coefficient indicates if the relationship is positive or negative.
    - In this data set, for example
        - An increase in 1 unit of fork is going to decrease the value of usr by -1.653.
        - An increase in 2 unit of ppgout is going to increase the value of use by 0.102
- However we observe that not all variables have very significant influence on the predictor variable. Value of cond from the above summary is high which indicates that there are also possible signs of multi collinearity
- Multicollinearity occurs when predictor variables in a regression model are correlated. This correlation is a problem because predictor variables should be independent. If the collinearity between variables is high, we might not be able to trust the p-values to identify independent variables that are statistically significant.
- When we have multicollinearity in the linear model, the coefficients that the model suggests are unreliable.

<u>Interpreting significant coefficients:</u>

Null hypothesis: Predictor variable is not significant
Alternate hypothesis: predictor variable is significant

(P > |t|) gives the p-value for each predictor variable to check the null hypothesis.

If the level of significance is set to 5% (0.05), the p-values greater than 0.05 would indicate that the corresponding predictor variables are not significant.

However, due to the presence of multicollinearity in our data, the p-values will also change.

We need to ensure that there is no multicollinearity to interpret the p-values.

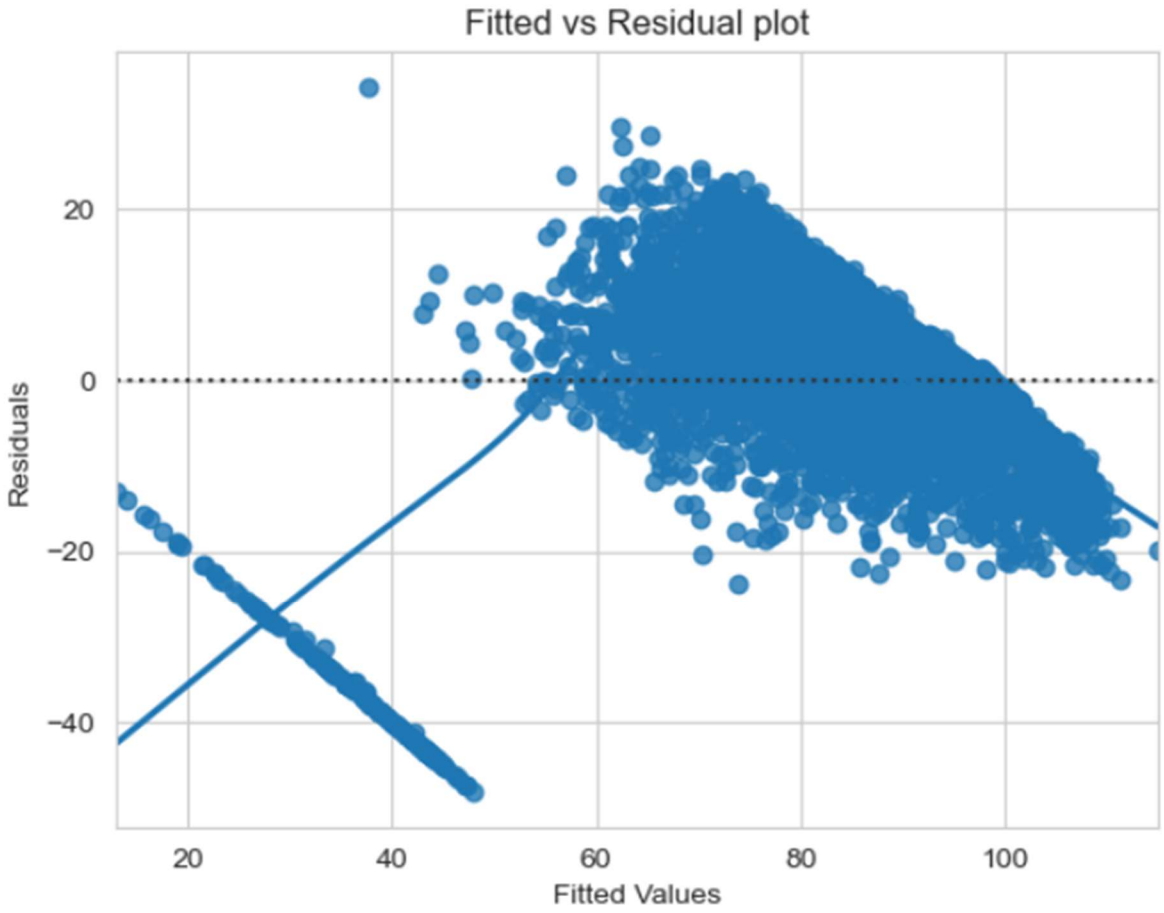
Determining if each variable is significant:

- H0:coefficient of lread = 0, probability of lread being -0.0199 is 0 - reject null hypothesis - variable significant
- H0: lwrite = 0, probability of lwrite being 0.006 is 0.42 - fail to reject null hypothesis - variable not significant
- H0: scall = 0, probability of scall being 0.001 is 0.0 - reject null hypothesis - variable significant
- H0: sread = 0, probability of sread being is -0.0004 is 0.81 - fail to reject null hypothesis - variable not significant
- H0: swrite = 0, probability of swrite being -0.0021 is 0.3 - fail to reject null hypothesis - variable not significant
- H0: fork = 0, probability of fork being -1.7216 is 0 - reject null hypothesis - variable significant
- H0: exec = 0, probability of exec being -0.0896 is 0.06 - fail to reject null hypothesis - variable not significant.
- H0: pgscan = 0, probability of pgscan being 0.0086 is 0.174 - fail to reject null hypothesis - variable not significant.
- All the probabilities for the remaining variables turn out to be less then 0.5. This indicates that all the remaining variables are significant.
- Insignificant variables turned out to be lwrite, sread, swrite, exec, pgscan

Checking the dataset has satisfied assumptions of linear regression with the built model:

1. Linearity assumption:

Plot a graph between fitted values and residuals. The obtained plot should not show any pattern.

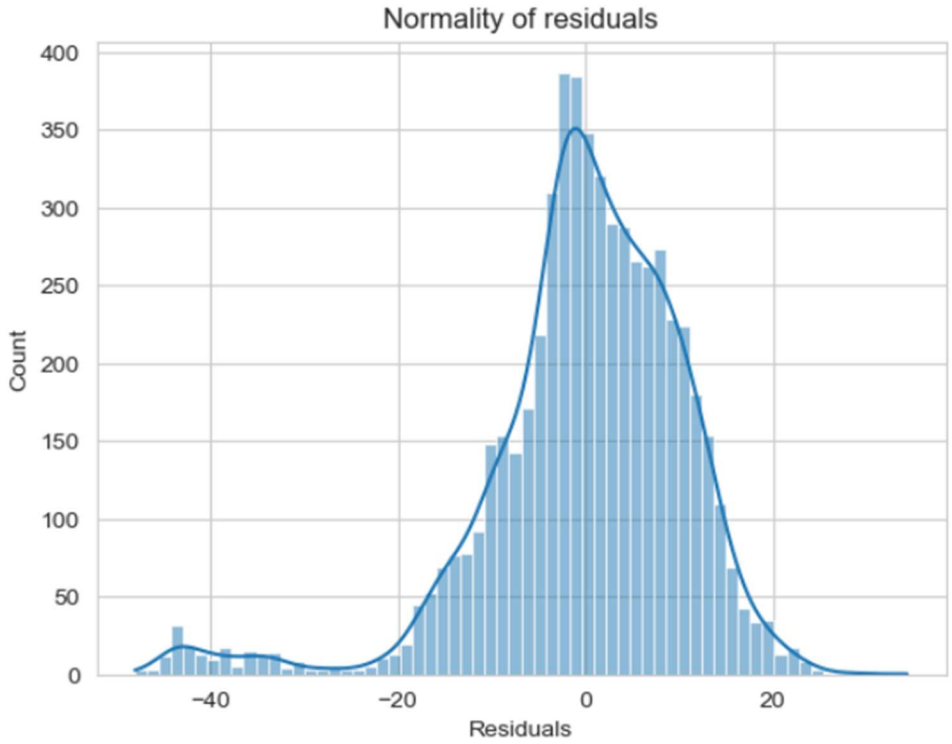


However, for the plot we can see some patterns. We will use feature engineered dataset for building regression model (or) other models that come from stepwise regression.

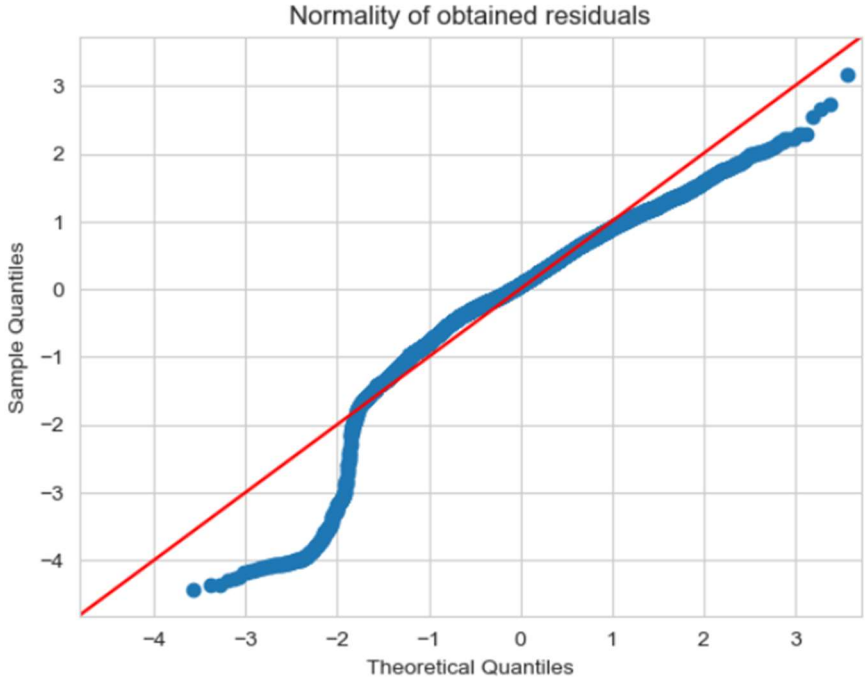
2. Testing for independence of residuals:

The above plot can be used to interpret if the residuals are independent. If there is no pattern observed, the residuals are said to be independent.

3. Testing the normality of residuals



- The graph for the residuals distribution shows that the residuals are normally distributed.



Above plot is the QQ plot for the obtained residuals

The Shapiro-Wilk test can also be used for checking the normality. The null and alternate hypotheses of the test are as follows:

Null hypothesis - Data is normally distributed.
Alternate hypothesis - Data is not normally distributed.

The obtained p-value for the shapiro test is 0.0. SInce this is less than 0.05, the null hypothesis that the data is normally distributed is rejected. It means residuals are not normally distributed.

4. Test for homoscedasticity:

- Residuals should be having equal variances.
- We can use goldfeldquandt test to predict homoscedasticity.

- The null and alternate hypotheses of the goldfeldquandt test are as follows:

  - Null hypothesis : Residuals are homoscedastic
  - Alternate hypothesis : Residuals have hetroscedasticity

- The probability that the residuals are homoscedastic is less than 0.05. [('F statistic', 1.1191911669084869), ('p-value', 0.0013407520853184723)]

- With 95% confidence we can say that the residuals are not homoscedastic. However, we will continue with the further rounds of linear regression.

5. Test for multi-collinearity using variance inflation factor:

- Variance inflation factor indicates the extent of collinearity an independent variable exhibits in relation with all other independent variables.

- High variance inflation factor indicates that the variable is highly correlated with other independent variables. If such variables are included in building the model, then the interpretation of co-efficients becomes tricky.

- The VIF values for the given data set are:
  const ---> 29.396674024351302
  lread ---> 1.4726232048968209
  lwrite ---> 1.4059261869416075
  scall ---> 2.414380871062771
  sread ---> 6.835552281610552
  swrite ---> 5.3194637012489245

fork ---> 18.210272008017036
exec ---> 3.0598443621331946
rchar ---> 1.9733948315385506
wchar ---> 1.552883396109173
pgout ---> 5.776039832734953
ppgout ---> 15.906860655416306
pgfree ---> 20.437658507680652
pgscan ---> 9.237219859065991
atch ---> 1.087544971161994
pgin ---> 8.074840311169055
ppgin ---> 8.670888402691117
pflt ---> 11.834193634585576
vflt ---> 20.230202165729885
runqsz ---> 1.1190316847699355
freemem ---> 1.677344709491036
freeswap ---> 1.7612101175022241

Basis the above VIF values, the variables that can show high multicollinearity with independent variables are: fork, pgfree, vflt, ppgout

1.3.5 Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare.

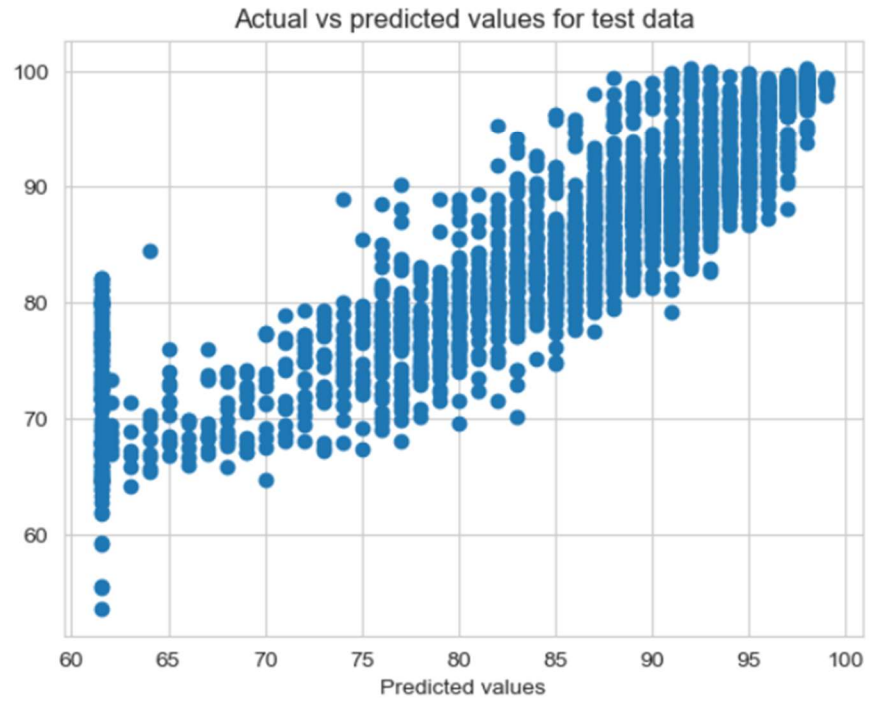
Model2: Running linear regression on outlier treated dataset

Basic information of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8192 entries, 0 to 8191
Data columns (total 22 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   lread     8192 non-null   float64
 1   lwrite    8192 non-null   float64
 2   scall     8192 non-null   float64
 3   sread     8192 non-null   float64
 4   swrite    8192 non-null   float64
 5   fork      8192 non-null   float64
 6   exec      8192 non-null   float64
 7   rchar     8192 non-null   float64
 8   wchar     8192 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  freemem   8192 non-null   float64
 19  freeswap  8192 non-null   float64
 20  usr       8192 non-null   float64
 21  runqsz    8192 non-null   int32
dtypes: float64(21), int32(1)
memory usage: 1.3 MB
```

Statistics obtained for the model built on outlier treated dataset:

```
r-squared for training data: 0.7961565330412979
r-squared for test data: 0.7676695029876079
RMSE for training data: 4.41901667552372
RMSE for test data: 4.652920160978209
MAE for training data: 3.2853840717691467
MAE for test data: 3.3811292738881704
adjusted r-sqaured for training data: 0.7954071085304203
adjusted r-squared for training data: 0.7675749058797039
Coefficients of model are: [[-6.34099732e-02  4.80183861e-02 -6.64352337e-04  3.38587579e-04
  -5.45988181e-03  2.96329957e-02 -3.21063250e-01 -5.21187916e-06
  -5.34633546e-06 -3.66852298e-01 -7.86092008e-02  8.52582013e-02
  -2.22044605e-16  6.30438035e-01  1.97538559e-02 -6.71537211e-02
  -3.35919897e-02 -5.46492175e-03 -4.57661401e-04  8.82932042e-06
  -1.61373724e+00]]
Intercept of the model is: [87.3589129]
```



Linear regression is sensitive to outliers. Hence removing the outliers improved the performance of the model.

Model3: Using RidgeRegression on the initial outlier treated dataset:

Basic info of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8192 entries, 0 to 8191
Data columns (total 22 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   lread     8192 non-null   float64
 1   lwrite    8192 non-null   float64
 2   scall     8192 non-null   float64
 3   sread     8192 non-null   float64
 4   swrite    8192 non-null   float64
 5   fork      8192 non-null   float64
 6   exec      8192 non-null   float64
 7   rchar     8192 non-null   float64
 8   wchar     8192 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  freemem   8192 non-null   float64
 19  freeswap  8192 non-null   float64
 20  usr       8192 non-null   float64
 21  runqsz    8192 non-null   int32
dtypes: float64(21), int32(1)
memory usage: 1.3 MB
```

Supplied parameters: {'alpha':[0.1, 0.3, 0.9,2,5,10]}

Parameter chosen by RidgeRegression (GridSearchCV) to fit to the model: 0.10

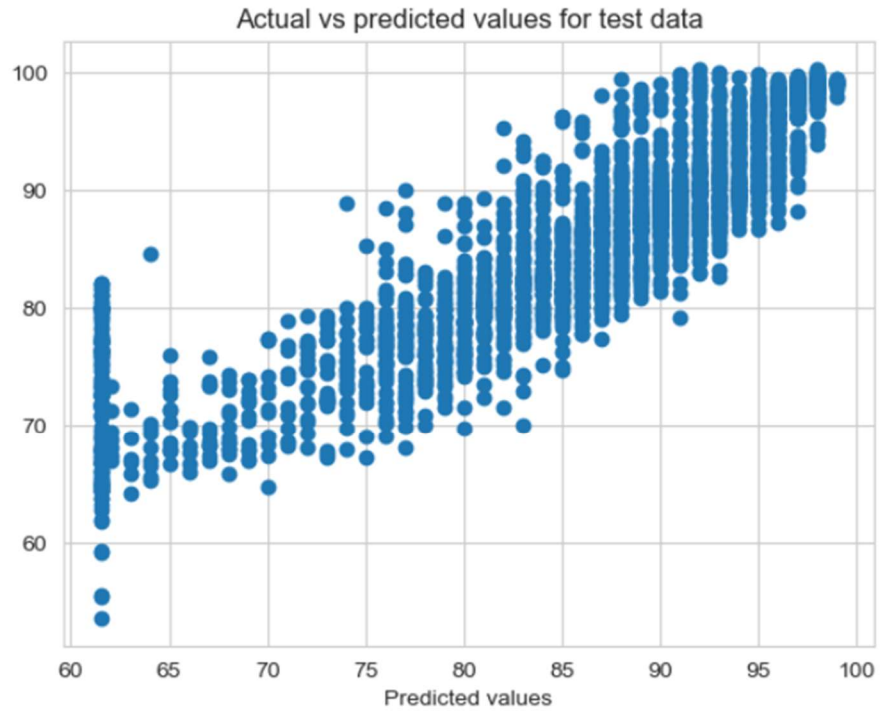
```
Metrics for ridge regression on outlier treated data are as follows:
Train_R_squared = 0.7961560625318421
Test_R_squared = 0.7676340089467395
RMSE_train = 4.4190217754858665
RMSE_test =  4.653275569831691
MAPE_train = 4.07420489455903
MAPE_test =  4.212657002437359
coefficients: [[-6.34925693e-02  4.81214007e-02 -6.65073197e-04  3.36966691e-04
  -5.45554857e-03  2.89028623e-02 -3.20530850e-01 -5.21484284e-06
  -5.36332166e-06 -3.64367367e-01 -7.90216221e-02  8.50837414e-02
   0.00000000e+00  6.23768033e-01  1.96785056e-02 -6.71021446e-02
  -3.35952578e-02 -5.45732301e-03 -4.57415785e-04  8.82755644e-06
  -1.60086257e+00]]
intercept: [87.34539859]
```

From all the models, considering the best accuracy scores for training and test data. The equation built by ridge regression proves to explain maximum variance in the data which inturn is going to increase the accuracy of the given model.

Model4: Selecting ideal features using stepwise regression to get the set of independent variables with lowest AIC score.

- Consider outlier treated dataset
- Drop the dependent variable from the dataframe
- Split the data into training and test data with 70:30 ratio.
- Apply stepwise regression algorithm on the dataset.
- The set of independent variables obtained after applying stepwise regression are:
- ['vflt', 'Lread', 'lwrite', 'scall', 'swrite', 'exec', 'rchar', 'wchar', ‘pgout', 'atch', 'ppgin', 'pflt', 'freemem', 'freeswap', 'runqsz', 'pgfree']
- Applying linear regression on the above selected independent variables.
- Metrics of the data are:

```
r-squared for training data: 0.7960962386216612
r-squared for test data: 0.7677279175447309
RMSE for training data: 4.419670172911013
RMSE for test data: 4.652335185349423
MAE for training data: 3.286162439718979
MAE for test data: 3.3800684324361923
adjusted r-sqaured for training data: 0.7955255791530494
adjusted r-squared for training data: 0.7676333442212556
Coefficients of model are: [[-5.19023169e-03 -6.36683762e-02  4.81569479e-02 -6.59779758e-04
  -5.11186705e-03 -3.19002386e-01 -5.17158692e-06 -5.40492946e-06
  -4.23749368e-01  6.29653335e-01 -5.51537567e-02 -3.35239622e-02
  -4.59067883e-04  8.81290149e-06 -1.61097005e+00  4.63958548e-02]]
Intercept of the model is: [87.39360546]
```



Updated AIC Score for the data is around 33348 which has been reduced from 46320 which is the AIC score for the initial model. AIC BIC scores indicated the loss of information in the model.

Low AIC BIC scores indicate that model has preserved most of the training data.

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

1. Initial description and analysis regarding shape and size of the data, basic information about the data, 5-point summary etc. are performed.
2. Performed univariate analysis on the data
3. Performed bivariate analysis on the data
4. Performed multivariate analysis on the data
5. Performed basic pre-processing of the data.

    a. Checked for null values
    b. Checked for duplicates
    c. Checked for zeros present in the data
    d. Checked for outliers - we have observed outliers in the data and treated them using min-max scaling.
6. Encoded the categorical variable present in the data to be able to pass it to the regression model.
7. Split the data into training and test set in 70:30 ratio.
8. Applied linear regression using scikit learn on the pre-processed dataset.
    a. Accuracy of the model is 0.64
9. Analyzed the rsqaured and other metrics and the coefficients
10. Checked if the trained model is satisfying linear regression assumptions.
11. Applied linear regression on the same data set using OLS regression method
12. Analyzed the summary of the data along with the significance of variables. (By predicting significance of their respective coefficients)
13. Fed the model with different variants of data
    a. Data after removing outliers
    b. Accuracy for this model is 0.79
14. Also applied ridge regression using GridSearchCV to the data after removing outliers.
    a. Ridge regression adds a constant of the sum of squared weights to the linear equation built by the model. GridSearchCV helps in finding the best parameters to be fed to the model to achieve high accuracy.
    b. Accuracy for this model is 0.8
15. Applied stepwise regression on the outlier treated data to identify the optimal independent variables. Built a regression model from the obtained variables.
    a. Accuracy for this model is 0.8
16. Model that gave the best r squared value and least mean squared error value is chosen to be the best model for prediction. We choose the model built using Ridge Regression.

Model selection: Models fed with treated outliers data have given best accuracy We will choose the model with good accuracy on training and test data.

We may choose either of models built using ridge regression or stepwise regression.

Final equation of the model is:

[87.39360546 + (-0.01* vflt) + (-0.06 * lread) + (0.05 * lwrite) + (-0.01 * swrite) + (-0.32 * exec) + (-0.42 * pgout) + (0.63 * atch) + (-0.06 * ppgin) + (-0.03 * pflt) + (runqsz * -1.61) + (pgfree * 0.05)]

Insights:

- Most of the variables indicate negative correlation with the dependent variable.
- An increase in vflt (number of page faults caused by address translation) decreases the portion of time cpu runs in user mode by 0.01 i.e., 1% keeping all other predictor variables constant.
- An increase in exec calls decreases the time portion of time cpu runs in user mode by 0.32 (32%) keeping all other predictor variables constant.
- The intercept of the model is 87.4
- Pgout, acth and runqsz affect the portion of time cpu runs in user mode to a greater extent compared to other independent variables

Actionable insights:

1. System needs to be less CPU Bound i.e., run queue size should be less than 2 for CPU to run in user mode in a better amount of time.
2. Variables rchar, wchar, pgscan do not affect the portion of time CPU runs in user mode to a greater extent. It means that there is no influence of number of characters read or written and number of pages scanned to be freed have least effect on the dependent variable
3. Number of system exec calls per second, number page attaches need to be taken care to increase which can increase the portion of time CPU runs in user mode.
4. Queue size, number of pageout requests per second need to be minimzed to increase the time CPU runs in user mode.

Problem2:

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.

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.**

Solution:

The dataset contains 1473 observations and 10 features.

Viewing the first 5 rows of the dataset:

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

Media_exposure	Contraceptive_method_used
Exposed	No
Exposed	No
Exposed	No
Exposed	No
Exposed	No

Viewing the last 5 rows of the dataset:

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

Media_exposure	Contraceptive_method_used
Exposed	Yes
Exposed	Yes
Exposed	Yes
Exposed	Yes
Exposed	Yes

Viewing the information of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1473 entries, 0 to 1472
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Wife_age                    1402 non-null   float64
 1   Wife_ education             1473 non-null   object
 2   Husband_education           1473 non-null   object
 3   No_of_children_born         1452 non-null   float64
 4   Wife_religion               1473 non-null   object
 5   Wife_Working                1473 non-null   object
 6   Husband_Occupation          1473 non-null   int64
 7   Standard_of_living_index    1473 non-null   object
 8   Media_exposure              1473 non-null   object
 9   Contraceptive_method_used   1473 non-null   object
dtypes: float64(2), int64(1), object(7)
memory usage: 115.2+ KB
```

Insights:

There are 8 independent variables, 1 dependent variable(Contraceptive_method_used)

1. wife_age seems to be a number but is float 64. Need to change datatype for wife_age
2. wife_age has some null values.
3. no_of_children born is also a float type which ideally has to be integer.
4. wife_religion also has some null values.
5. Husband_occupation is a random categorical variable which is interpreted as integer. Need to change it as well.

Viewing the summary of the data:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Wife_age	1402.0	NaN	NaN	NaN	32.606277	8.274927	16.0	26.0	32.0	39.0	49.0
Wife_ education	1473	4	Tertiary	577	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Husband_education	1473	4	Tertiary	899	NaN	NaN	NaN	NaN	NaN	NaN	NaN
No_of_children_born	1452.0	NaN	NaN	NaN	3.254132	2.365212	0.0	1.0	3.0	4.0	16.0
Wife_religion	1473	2	Scientology	1253	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Wife_Working	1473	2	No	1104	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Husband_Occupation	1473.0	NaN	NaN	NaN	2.137814	0.864857	1.0	1.0	2.0	3.0	4.0
Standard_of_living_index	1473	4	Very High	684	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Media_exposure	1473	2	Exposed	1364	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Contraceptive_method_used	1473	2	Yes	844	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Insights:

1. Average wife_age is around 32 years with 25% of women in between 39 and 49 years.
2. Most of the wives and husbands are having education level 'Tertiary'
3. Maximum number of children born is 16 which might need to be reviewed to be an outlier.
4. Average number of children born are 3 this might indicate that women who have children have also used contraceptive method.
5. Husband occupation is better understood after converting it to categorical.
6. most of the families are media_exposed.
7. Around half of the people in the given dataset have used contraceptive method and half haven't used.

Checking for null values:

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

There are 71 null values in Wife_age and 21 null values in no_of_children.

Replacing null values for Wife_age with median of the data to balance the age values present in the dataset.

And no_of_children with 0 as it makes more sense for the column.

Performing null check after imputing null values:

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

We can see that the null values have been imputed.

Husband_occupation column has random categories labeled as 1,2,3,4 but the data type is displayed as integer. Converting the datatype for the column to object

Number of children born can also be considered as a categorical variable as the numbers indicate categories of women with the specified number of children. Converting the No_of_children to categorical column.

Viewing the value counts of No_of_children born:

```
2     274
1     273
3     255
4     192
5     131
0     118
6      90
7      49
8      46
9      16
10     11
11     11
12      4
13      2
16      1
Name: No_of_children_born, dtype: int64
```

It shows that there are around 27 women with 10 or more children. Combining the No_of_children values greater than 10 to 10 as the number of children being beyond 10 is a rare case and such values can be grouped together into a single category. Updated the column.

Viewing the updated information of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1473 entries, 0 to 1472
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Wife_age                   1402 non-null   float64
 1   Wife_ education            1473 non-null   object
 2   Husband_education          1473 non-null   object
 3   No_of_children_born        1452 non-null   object
 4   Wife_religion              1473 non-null   object
 5   Wife_Working               1473 non-null   object
 6   Husband_Occupation         1473 non-null   object
 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(1), object(9)
memory usage: 115.2+ KB
```

All the null values have been imputed. There are no missing values or inconsistencies.

Checking for duplicates in the data: There are 83 duplicates.
Removed the duplicates in the data.

After removing duplicates we have 1390 observations and 10 columns in the dataset.

Viewing the column names of the dataset:

```
Index(['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'],
      dtype='object')
```

Column names Wife_education and Media_exposure have unnecessary spaces in their names. Removing them to easily access them for further analysis.

Updated column name list:

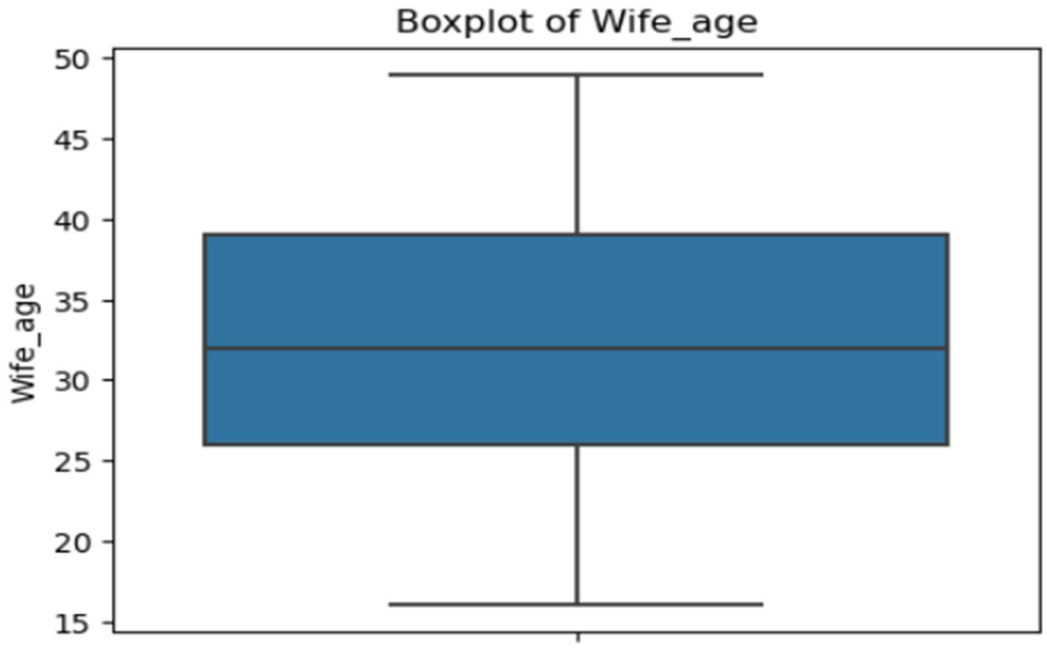
```
Index(['Wife_age', 'Wife_education', 'Husband_education',
       'No_of_children_born', 'Wife_religion', 'Wife_Working',
       'Husband_Occupation', 'Standard_of_living_index', 'Media_exposure',
       'Contraceptive_method_used'],
      dtype='object')
```

Viewing the updated summary of the data:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Wife_age	1326.0	NaN	NaN	NaN	32.557315	8.289259	16.0	26.0	32.0	39.0	49.0
Wife_education	1393	4	Tertiary	515	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Husband_education	1393	4	Tertiary	827	NaN	NaN	NaN	NaN	NaN	NaN	NaN
No_of_children_born	1372.0	15.0	2.0	258.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Wife_religion	1393	2	Scientology	1186	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Wife_Working	1393	2	No	1043	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Husband_Occupation	1393.0	4.0	3.0	570.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Standard_of_living_index	1393	4	Very High	618	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Media_exposure	1393	2	Exposed	1284	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Contraceptive_method_used	1393	2	Yes	779	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Checking for outliers:

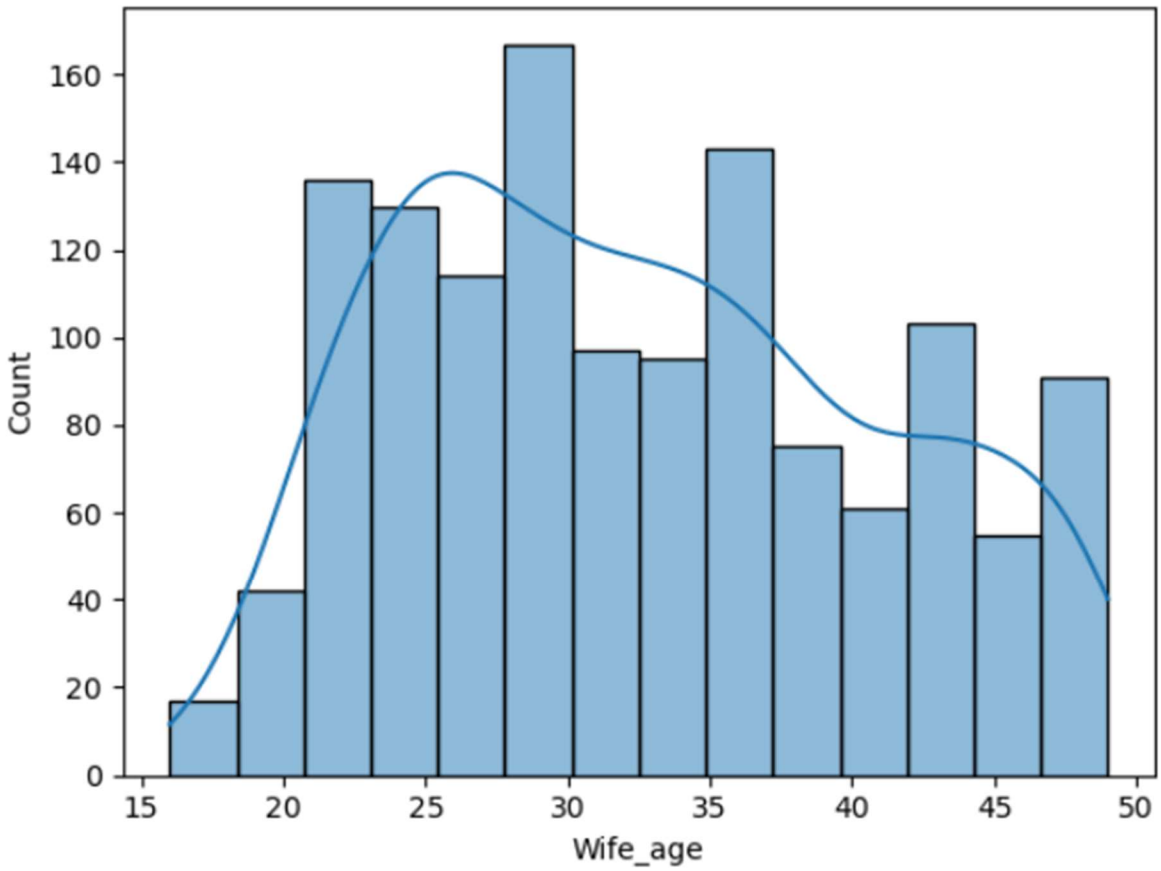
We have a single numeric variable Age. Checking for outliers in Age:



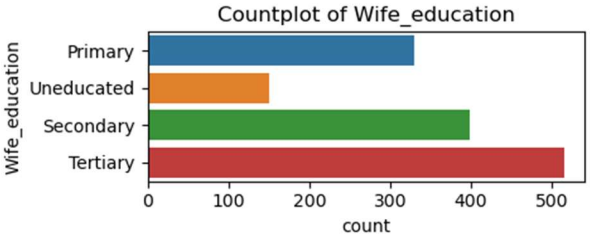
We do not have any outliers.

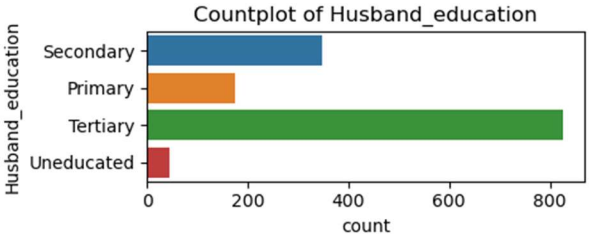
Univariate analysis:

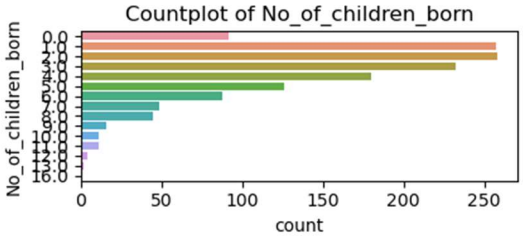
Plotting the continuous distribution of the only numeric variable present in the dataset Wife_age. The values follow slightly normal distribution with average lying at around 25. Age of women in the given sample ranges from 15 to 25.

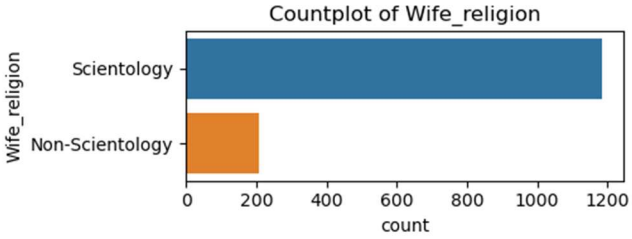


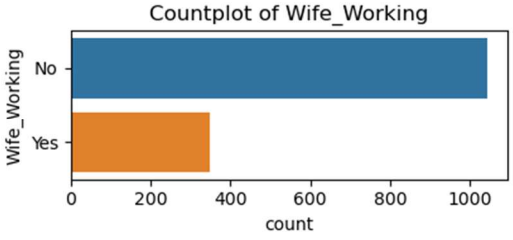
Univariate analysis on categorical variables:

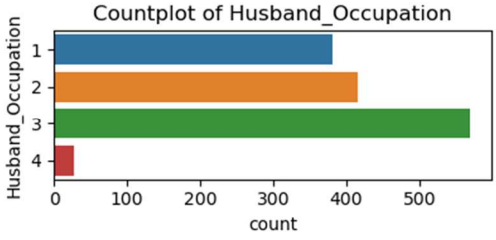

Countplot of Wife_education
Wife_education
Primary
Uneducated
Secondary
Tertiary
0
100
200
300
400
500
count

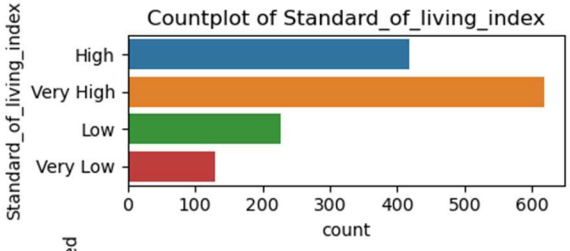

Countplot of Husband_education
Husband_education
Secondary
Primary
Tertiary
Uneducated
0
200
400
600
800
count

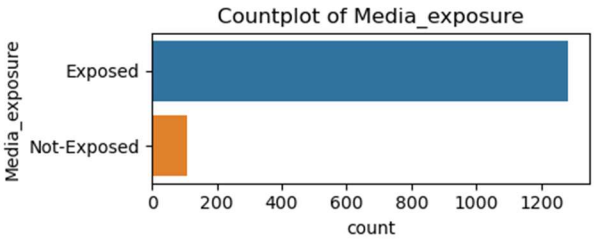

Countplot of No_of_children_born
No_of_children_born
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
16.0
0
50
100
150
200
250
count

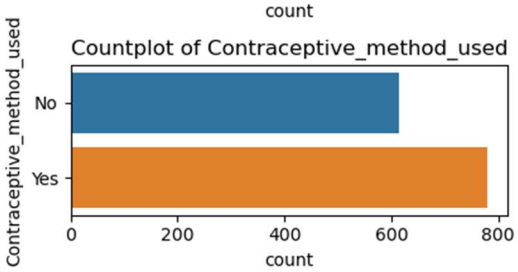

Countplot of Wife_religion
Wife_religion
Scientology
Non-Scientology
0
200
400
600
800
1000
1200
count


Countplot of Wife_Working
Wife_Working
No
Yes
0
200
400
600
800
1000
count


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


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


Countplot of Media_exposure
Media_exposure
Exposed
Not-Exposed
0
200
400
600
800
1000
1200
count


Countplot of Contraceptive_method_used
Contraceptive_method_used
No
Yes
0
200
400
600
800
count

Value counts in terms of percentage for each categorical variable:

```
value counts for  Wife_education
 Tertiary      0.368345
Secondary     0.286331
Primary       0.237410
Uneducated    0.107914
Name: Wife_education, dtype: float64

value counts for  Husband_education
 Tertiary      0.592806
Secondary     0.249640
Primary       0.125899
Uneducated    0.031655
Name: Husband_education, dtype: float64

value counts for  No_of_children_born
 1     0.184892
2     0.184892
3     0.166187
4     0.128777
5     0.090647
0     0.081295
6     0.063309
7     0.035252
8     0.032374
10    0.020863
9     0.011511
Name: No_of_children_born, dtype: float64

value counts for  Wife_religion
 Scientology        0.851799
Non-Scientology    0.148201
Name: Wife_religion, dtype: float64

value counts for  Wife_Working
 No     0.74964
Yes    0.25036
Name: Wife_Working, dtype: float64
```

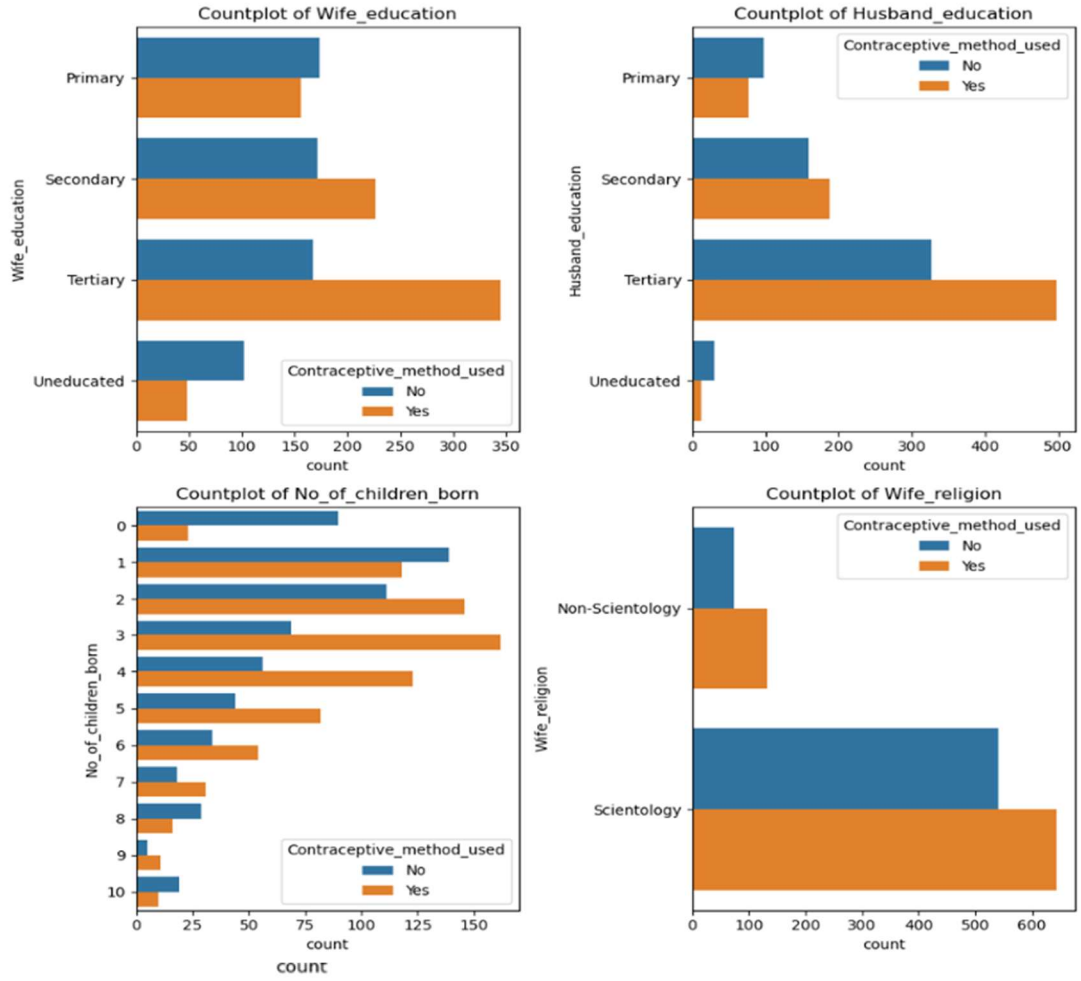
```
value counts for  Husband_Occupation
 3    0.410072
2    0.297842
1    0.272662
4    0.019424
Name: Husband_Occupation, dtype: float64

value counts for  Standard_of_living_index
 Very High    0.442446
High         0.301439
Low          0.163309
Very Low     0.092806
Name: Standard_of_living_index, dtype: float64

value counts for  Media_exposure
 Exposed        0.921583
Not-Exposed    0.078417
Name: Media_exposure, dtype: float64

value counts for  Contraceptive_method_used
 Yes    0.558273
No     0.441727
Name: Contraceptive_method_used, dtype: float64
```

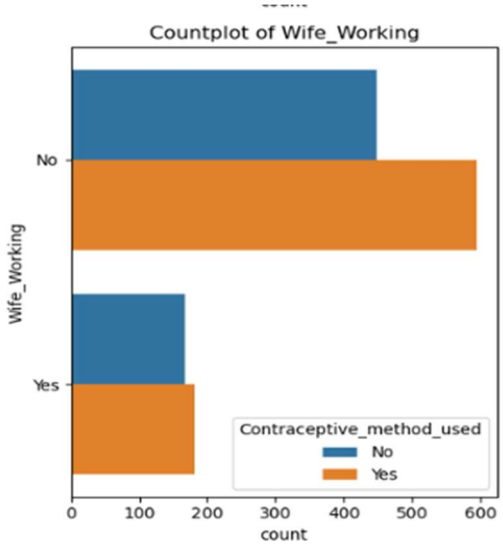
Insights:

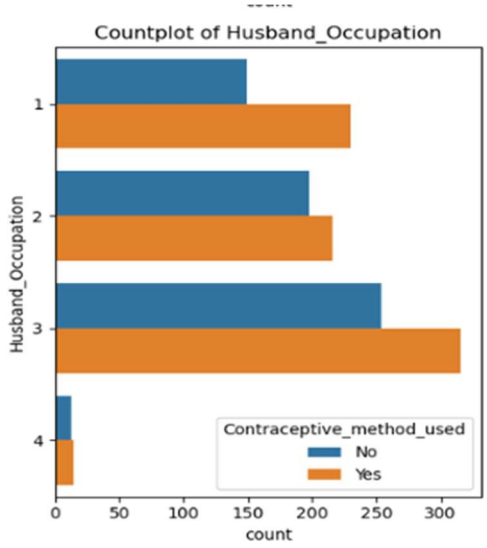
1. Maximum number of wives and husbands belong to the education level ‘Tertiary’ followed by ‘Secondary’. This indicates that most of people are well educated and possibly living with good standard of life
2. Non working wives dominate the data. Most of the women possess a high to very high standard of living which was expected earlier.
3. Most of the women(92%) are exposed to the media. Detailed analysis on this variable needs to be done about the usage of contraceptive methods due to media exposure.
4. 55% of the women have used contraceptive methods.

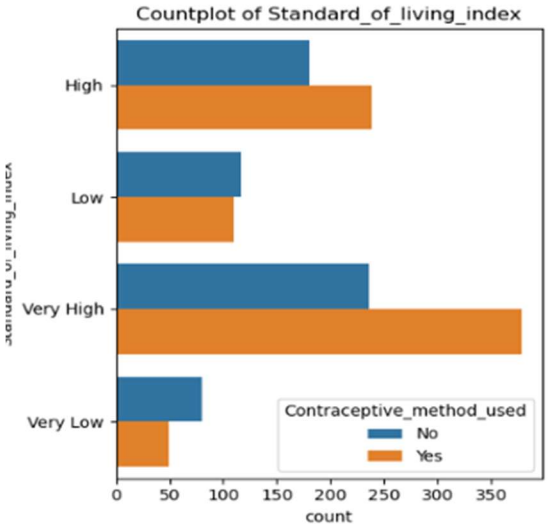
Bivariate analysis:

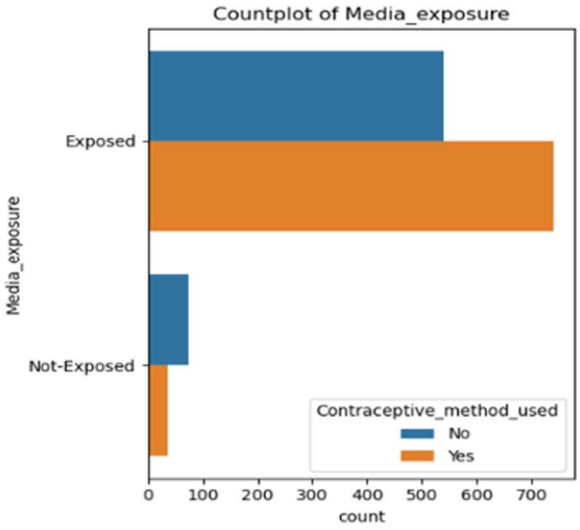
Analyzing the usage of contraceptive methods with other independent variables:

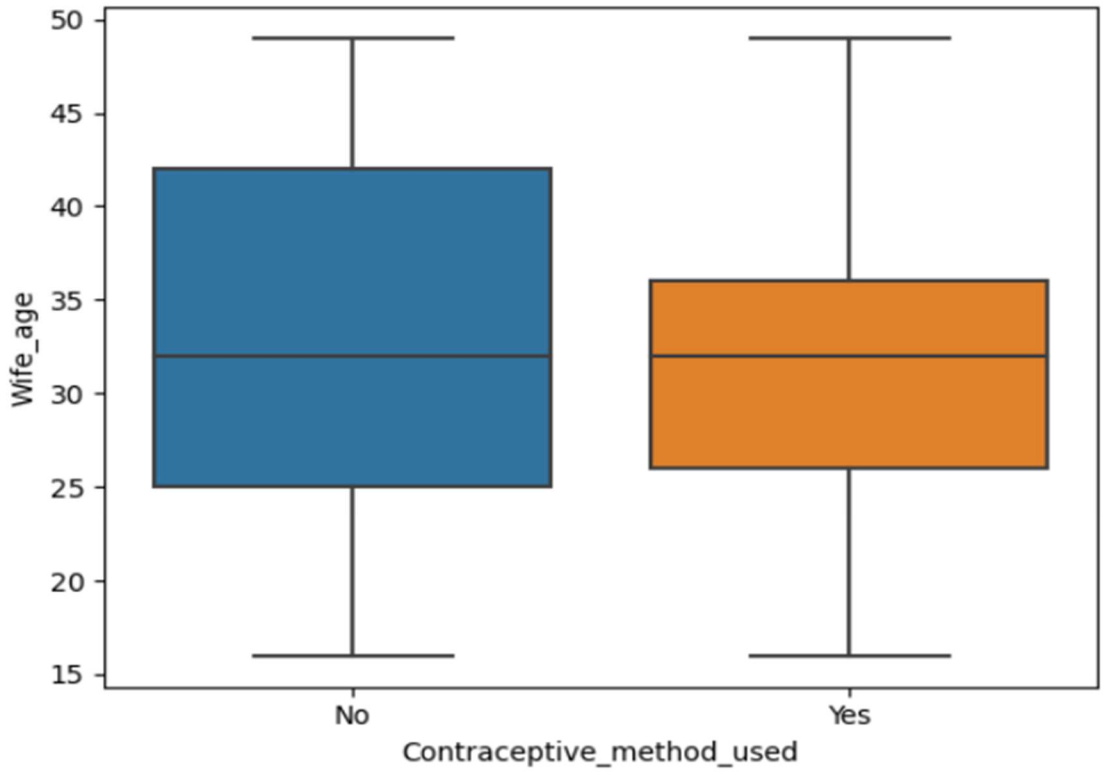



Countplot of Wife_Working
Wife_Working
No
Yes
0
100
200
300
400
500
600
count
Contraceptive_method_used
No
Yes


Countplot of Husband_Occupation
Husband_Occupation
1
2
3
4
0
50
100
150
200
250
300
count
Contraceptive_method_used
No
Yes

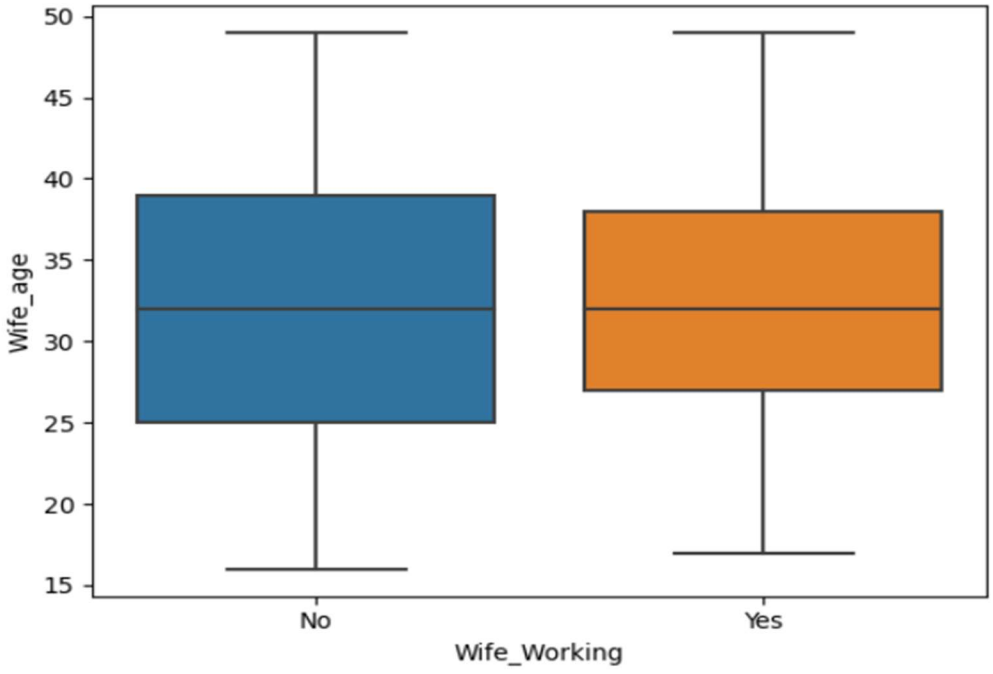

Countplot of Standard_of_living_index
High
Low
Very High
Very Low
0
50
100
150
200
250
300
350
count
Contraceptive_method_used
No
Yes


Countplot of Media_exposure
Media_exposure
Exposed
Not-Exposed
0
100
200
300
400
500
600
700
count
Contraceptive_method_used
No
Yes



Above graph shows the distribution of age for women using and not using contraceptive methods.

Boxplot depicting the working trends of wives with age:



Insights:

1. Wives having tertiary education with husbands also having tertiary education have resorted to using contraceptive methods the most.
2. Non working wives have used contraceptive methods in comparison to working wives.
3. Women living in high standard of index and exposed to media tend to use contraceptive methods the most.

Multivariate analysis:

Education metrics of wife and husband who have resorted to using contraceptive methods:

Wife_education Husband_education	Primary	Secondary	Tertiary	Uneducated	All
Primary	0.055412	0.014175	0.001289	0.028351	0.099227
Secondary	0.074742	0.118557	0.030928	0.018041	0.242268
Tertiary	0.065722	0.155928	0.411082	0.009021	0.641753
Uneducated	0.005155	0.003866	0.001289	0.006443	0.016753
All	0.201031	0.292526	0.444588	0.061856	1.000000

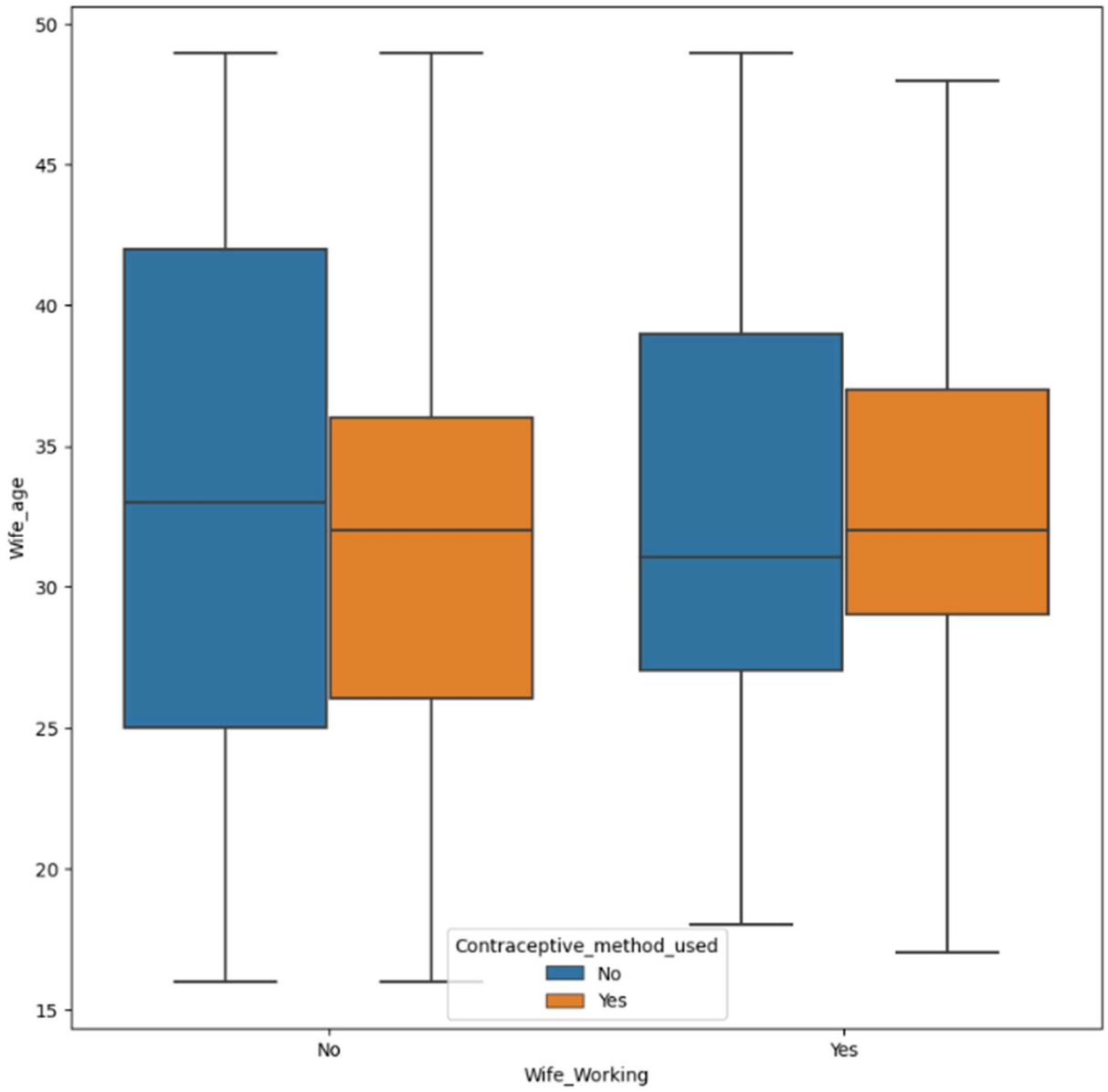
Working trends of wife and husband who have used contraceptive methods:

Husband_Occupation Wife_Working	1	2	3	4	All
No	0.220361	0.207474	0.326031	0.012887	0.766753
Yes	0.076031	0.070876	0.081186	0.005155	0.233247
All	0.296392	0.278351	0.407216	0.018041	1.000000

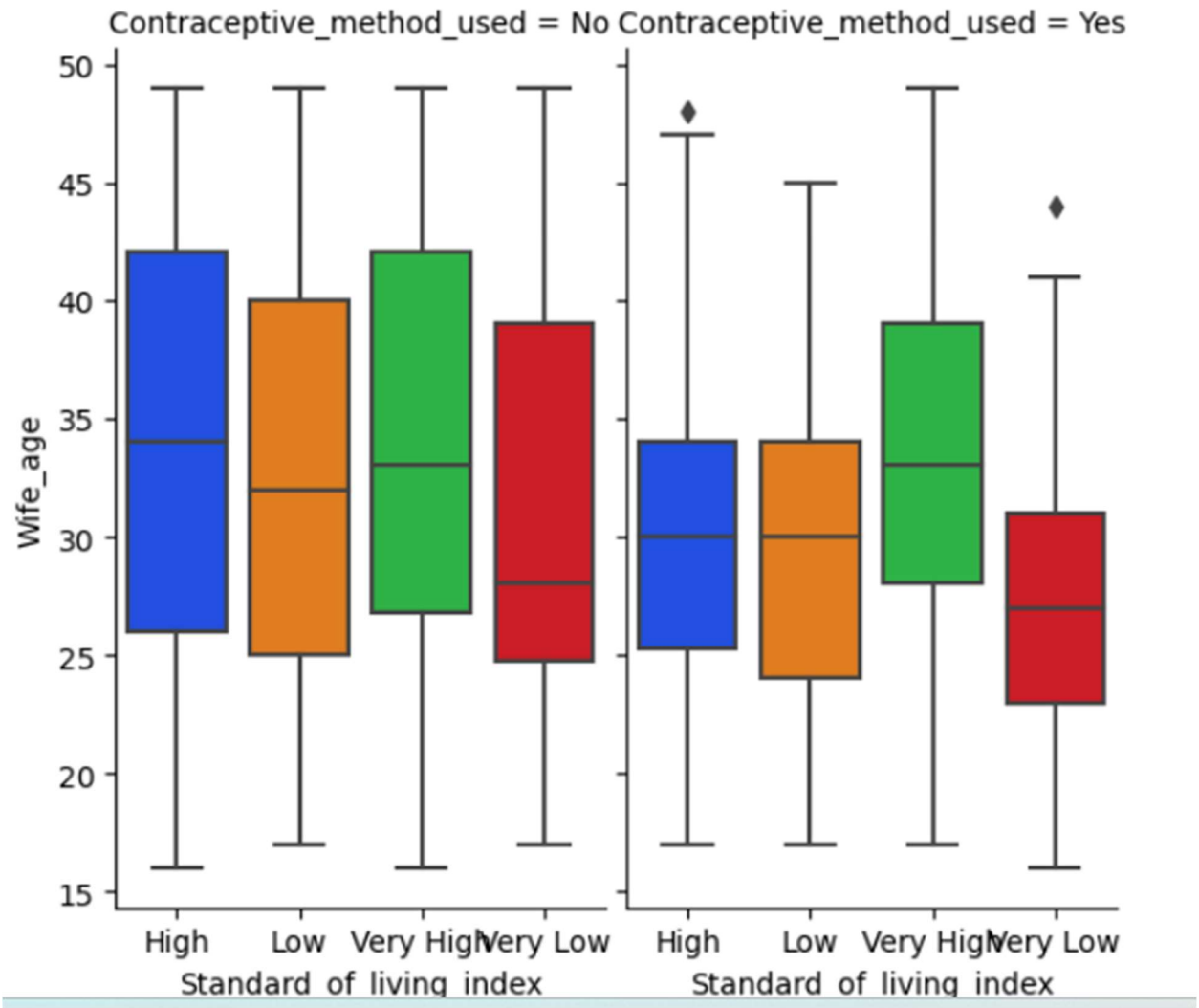
Dsistribution of people using contraceptive methods among various standards of living and media exposure levels:

Media_exposure Standard_of_living_index	Exposed	Not-Exposed	All
High	0.300258	0.006443	0.306701
Low	0.122423	0.019330	0.141753
Very High	0.478093	0.010309	0.488402
Very Low	0.054124	0.009021	0.063144
All	0.954897	0.045103	1.000000

Boxplots of age distribution of wives who have used contraceptive methods:



Boxplot showing age distribution of wives having different number of children:



Insights:

1. Women in the age group 25 - 35 who are not working tend to use contraceptive methods whereas working women within age 28-38 tend to use contraceptive methods the most.
2. The median age of non working wives using contraceptive methods is around 32 whereas that of working wives is slightly higher
3. Median age of wives using contraceptive methods is 30-35.
4. Women with age below 30 tend to use contraceptive pills when they have 3 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.**

Solution:

Encoding the data using one hot encoding to be able to pass to logistic regression and LDA models. One hot encoding will create n-1 columns for a column having categories of 'n' levels.

Basic info of the dataset after performing one hot encoding:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1390 entries, 0 to 1472
Data columns (total 27 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Wife_age                              1390 non-null   int64
 1   Wife_education_Secondary              1390 non-null   uint8
 2   Wife_education_Tertiary               1390 non-null   uint8
 3   Wife_education_Uneducated             1390 non-null   uint8
 4   Husband_education_Secondary           1390 non-null   uint8
 5   Husband_education_Tertiary            1390 non-null   uint8
 6   Husband_education_Uneducated          1390 non-null   uint8
 7   No_of_children_born_1                 1390 non-null   uint8
 8   No_of_children_born_2                 1390 non-null   uint8
 9   No_of_children_born_3                 1390 non-null   uint8
 10  No_of_children_born_4                 1390 non-null   uint8
 11  No_of_children_born_5                 1390 non-null   uint8
 12  No_of_children_born_6                 1390 non-null   uint8
 13  No_of_children_born_7                 1390 non-null   uint8
 14  No_of_children_born_8                 1390 non-null   uint8
 15  No_of_children_born_9                 1390 non-null   uint8
 16  No_of_children_born_10                1390 non-null   uint8
 17  Wife_religion_Scientology             1390 non-null   uint8
 18  Wife_Working_Yes                      1390 non-null   uint8
 19  Husband_Occupation_2                  1390 non-null   uint8
 20  Husband_Occupation_3                  1390 non-null   uint8
 21  Husband_Occupation_4                  1390 non-null   uint8
 22  Standard_of_living_index_Low          1390 non-null   uint8
 23  Standard_of_living_index_Very High    1390 non-null   uint8
 24  Standard_of_living_index_Very Low     1390 non-null   uint8
 25  Media_exposure_Not-Exposed            1390 non-null   uint8
 26  Contraceptive_method_used_Yes         1390 non-null   uint8
dtypes: int64(1), uint8(26)
memory usage: 89.3 KB
```

Viewing the first 5 rows of the dataset:

Wife_age	Wife_education_Secondary	Wife_education_Tertiary	Wife_education_Uneducated	Husband_education_Secondary	Husband_education_Tertiary	H
24	0	0	0	1	0	
45	0	0	1	1	0	
43	0	0	0	1	0	
42	1	0	0	0	0	
36	1	0	0	1	0	

Husband_education_Uneducated	No_of_children_born_1	No_of_children_born_2	No_of_children_born_3	...	Wife_religion_Scientology	Wife_Working_Yes
0	0	0	1	...	1	0
0	0	0	0	...	1	0
0	0	0	0	...	1	0
0	0	0	0	...	1	0
0	0	0	0	...	1	0

Husband_Occupation_2	Husband_Occupation_3	Husband_Occupation_4	Standard_of_living_index_Low	Standard_of_living_index_Very High
1	0	0	0	0
0	1	0	0	1
0	1	0	0	1
0	1	0	0	0
0	1	0	1	0

Standard_of_living_index_Very Low	Media_exposure_Not-Exposed	Contraceptive_method_used_Yes
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Logistic regression:

1. Split the data into training and test data in 70:30 ratio
2. Target variable separated out is 'Contraceptive_method_used_Yes'.
3. Using GridSearchCV and logistic regression to find the parameters to be sent to logistic regression.
4. Parameters used for GridSearchCV:

```
GridSearchCV(cv=3, estimator=LogisticRegression(n_jobs=-1), n_jobs=-1,
             param_grid={'penalty': ['l2', 'none'],
                         'solver': ['sag', 'newton-cg']},
             scoring='f1')
```

5. Best model for logistic regression using GridSearchCV turned out to be

```
: LogisticRegression(n_jobs=-1, penalty='none', solver='newton-cg')
```

6. Therefore, building the logistic regression for the given data with the specified parameters.

Linear Discriminant Analysis:

Encoded the data using one hot encoding to be able to pass to logistic regression and LDA models

Basic info of the dataset after performing one hot encoding:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1390 entries, 0 to 1472
Data columns (total 27 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Wife_age                              1390 non-null   int64
 1   Wife_education_Secondary              1390 non-null   uint8
 2   Wife_education_Tertiary               1390 non-null   uint8
 3   Wife_education_Uneducated             1390 non-null   uint8
 4   Husband_education_Secondary           1390 non-null   uint8
 5   Husband_education_Tertiary            1390 non-null   uint8
 6   Husband_education_Uneducated          1390 non-null   uint8
 7   No_of_children_born_1                 1390 non-null   uint8
 8   No_of_children_born_2                 1390 non-null   uint8
 9   No_of_children_born_3                 1390 non-null   uint8
 10  No_of_children_born_4                 1390 non-null   uint8
 11  No_of_children_born_5                 1390 non-null   uint8
 12  No_of_children_born_6                 1390 non-null   uint8
 13  No_of_children_born_7                 1390 non-null   uint8
 14  No_of_children_born_8                 1390 non-null   uint8
 15  No_of_children_born_9                 1390 non-null   uint8
 16  No_of_children_born_10                1390 non-null   uint8
 17  Wife_religion_Scientology             1390 non-null   uint8
 18  Wife_Working_Yes                      1390 non-null   uint8
 19  Husband_Occupation_2                  1390 non-null   uint8
 20  Husband_Occupation_3                  1390 non-null   uint8
 21  Husband_Occupation_4                  1390 non-null   uint8
 22  Standard_of_living_index_Low          1390 non-null   uint8
 23  Standard_of_living_index_Very High    1390 non-null   uint8
 24  Standard_of_living_index_Very Low     1390 non-null   uint8
 25  Media_exposure_Not-Exposed            1390 non-null   uint8
 26  Contraceptive_method_used_Yes         1390 non-null   uint8
dtypes: int64(1), uint8(26)
memory usage: 89.3 KB
```

Steps followed to perform Linear Discriminant Analysis:

1. Split the data into training and test data in 70:30 ratio
2. Target variable separated out is ‘Contraceptive_method_used_Yes’.
3. Applying LinearDiscriminantAnalysis from scikit learn on the training data set.

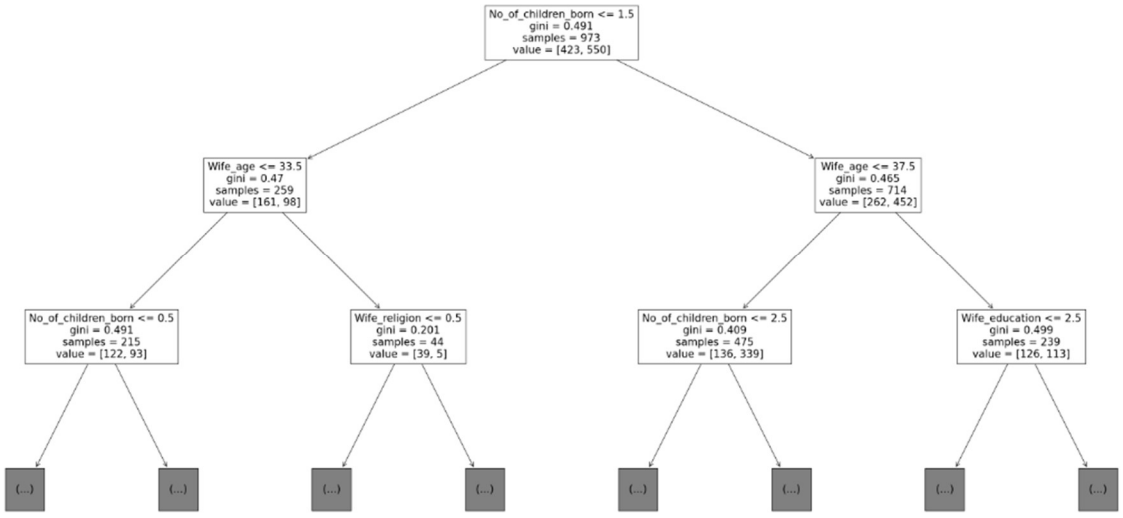
Applying CART:

1. CART model can accept only numerical and categorical data types. Hence, we need to modify the datatypes of the variables in the data set.
2. For the problem, we change all the categorical datatypes which are object to category
3. We also assign codes to each categorical column where codes represent each level of the category pertaining to that column.
4. Basic info of the data set after encoding:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1390 entries, 0 to 1472
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Wife_age                   1390 non-null   int64
 1   Wife_education             1390 non-null   category
 2   Husband_education          1390 non-null   category
 3   No_of_children_born        1390 non-null   category
 4   Wife_religion              1390 non-null   category
 5   Wife_Working               1390 non-null   category
 6   Husband_Occupation         1390 non-null   category
 7   Standard_of_living_index   1390 non-null   category
 8   Media_exposure             1390 non-null   category
 9   Contraceptive_method_used  1390 non-null   category
dtypes: category(9), int64(1)
memory usage: 67.9 KB
```

5. Split the data into training and test data in 70:30 ratio
6. Target variable separated out is ‘Contraceptive_method_used_Yes’.
7. Using DecisionTreeClassifier with criterion being ‘Gini’ and random_state being 1

High level view of the built decision tree classifier in step 7:



Applying regularized decision tree classifier on the data:

Depending on the dataset, in order for the decision tree to generalize instead of overfit, we regularize the model using some parameters in the decision tree classifier.

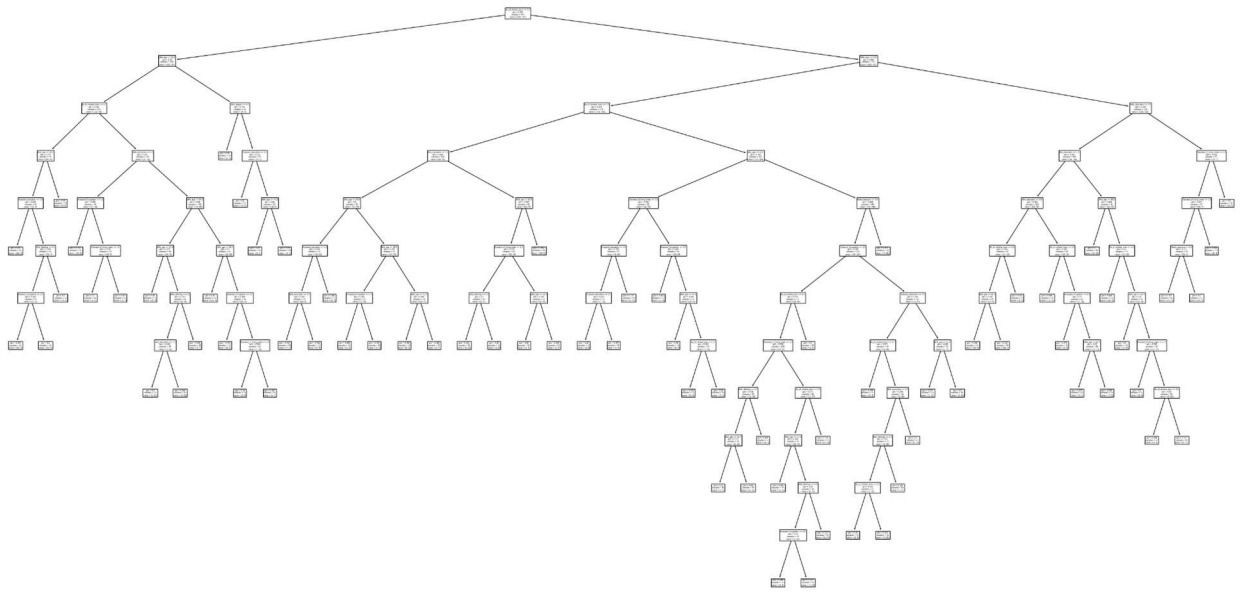
Since we have around 1400 rows and 10 columns, let us set the parameters below accordingly.

maxDepth - 30

Min_samples_leaf = 10,

Min_samples_split = 10

Applying the regularized decision tree classifier on the data, we see the below decision tree.



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**.

Solution:

Evaluating performance metrics for different models built:

Description of different parameters considered in evaluating the performance of a logistic regression/LDA/CART model.

To classify the predictor variable, the algorithm calculates the probabilities of the predictor variable belonging to each class. A threshold needs to be devised to partition the response space into success and failure. Typically, the threshold is set at 50% level.

If the probability of success is 50% or above for a given combination of predictors, the value of response is taken to be 1, otherwise 0.

However, this threshold may be set at some other convenient level.

Classification matrix:

Let P be the total number of successes (positives) in the data and N be the total number of failures (negatives).

If a success is predicted as success, it is an example of True Positive (TP).
If on the other hand a failure is predicted as failure, it is an example of True Negative (TN).

In both cases, classification is correct.

However,

if a success is predicted as a failure, it is an example of False Negative (FN)
if a failure is predicted as a success, it is an example of false positives. These are misclassified.

Table 2 - Example of a confusion (misclassification) matrix

Confusion Matrix		Actual	
		Success (Positive)	Failure (Negative)
Predicted	Success (Positive)	TP	FP
	Failure (Negative)	FN	TN
Total		P	N

Probability of misclassification = $FN+FP\ n$,
where n is the sample size.

For the perfect logistic regression, misclassification probability is 0; i.e. no observation would have been misclassified.
This indicates overfit of the model and not to be recommended, since such a model will not have good predicting power.

A few other performance metrics are equally important.

Precision = $TP\ /TP+FP$, i.e. among all the successes (positives) in the data, how many are identified as positive by the logistic regression.

Specificity = $TN\ /TN+FP$, i.e. among all failures (negatives) in the data, how many are actually identified as negative by the logistic regression

Sensitivity or Recall = $TP/\ TP+FN$, i.e. among all the predicted successes, how many are actually success.

The F-score of the model is defined as $2(Precision*Recall)\ /\ Precision+Recall$.

F is between 0 and 1, and the closer it is to 1, the better is the model.

Among two competing logistic regressions, the one that maximizes all the accuracy measures, is the one of choice. However, it may not be possible to maximize all criteria simultaneously.

1. Performance metrics for Logistic regression model:

Predicted probabilities:

Predicted probabilities by the model on training data:

:

	0	1
0	0.188111	0.811889
1	0.119583	0.880417
2	0.227578	0.772422
3	0.164813	0.835187
4	0.762136	0.237864

Predicted probabilities by the model on test data:

	0	1
0	0.915619	0.084381
1	0.639021	0.360979
2	0.257000	0.743000
3	0.240676	0.759324
4	0.252505	0.747495

Insights: For row 0 in test dataset, the probability of predictor variable belonging to class 0 is 0.9 whereas the probability of predictor variable belonging to class 1 is 0.08

Accuracy:

Accuracy of the logistic regression model on the training data: 0.72

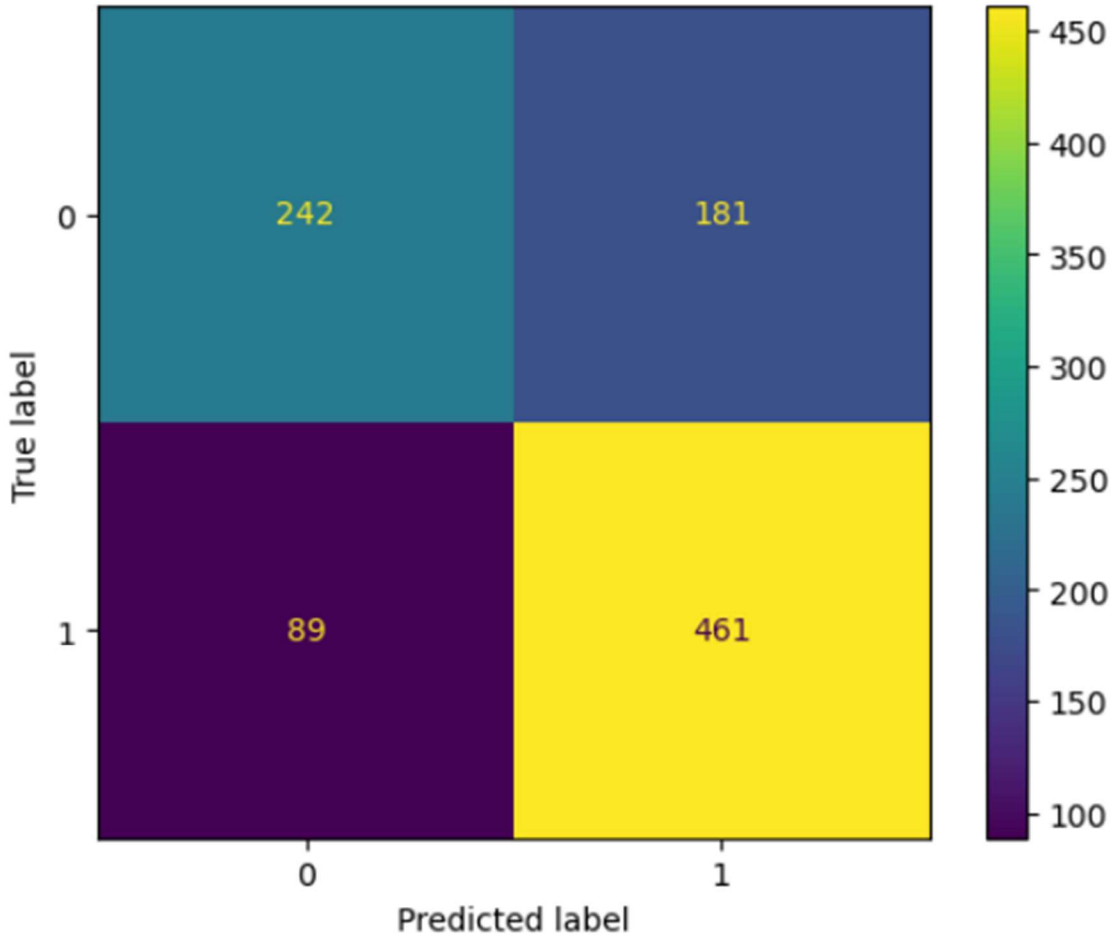
Accuracy of the logistic regression model on the test data: 0.68

Insights: Accuracy for the training data is 72% whereas that of testing data is 68%. Accuracies of training and test data are close to each other which indicates that the model has decently generalized.

Confusion matrix:

Confusion matrix of training data:

```
array([[242, 181],
       [ 89, 461]], dtype=int64)
```



Insights:

There are a total of 973 observations in training data.

Out of which 242 negatives have been correctly classified as negatives

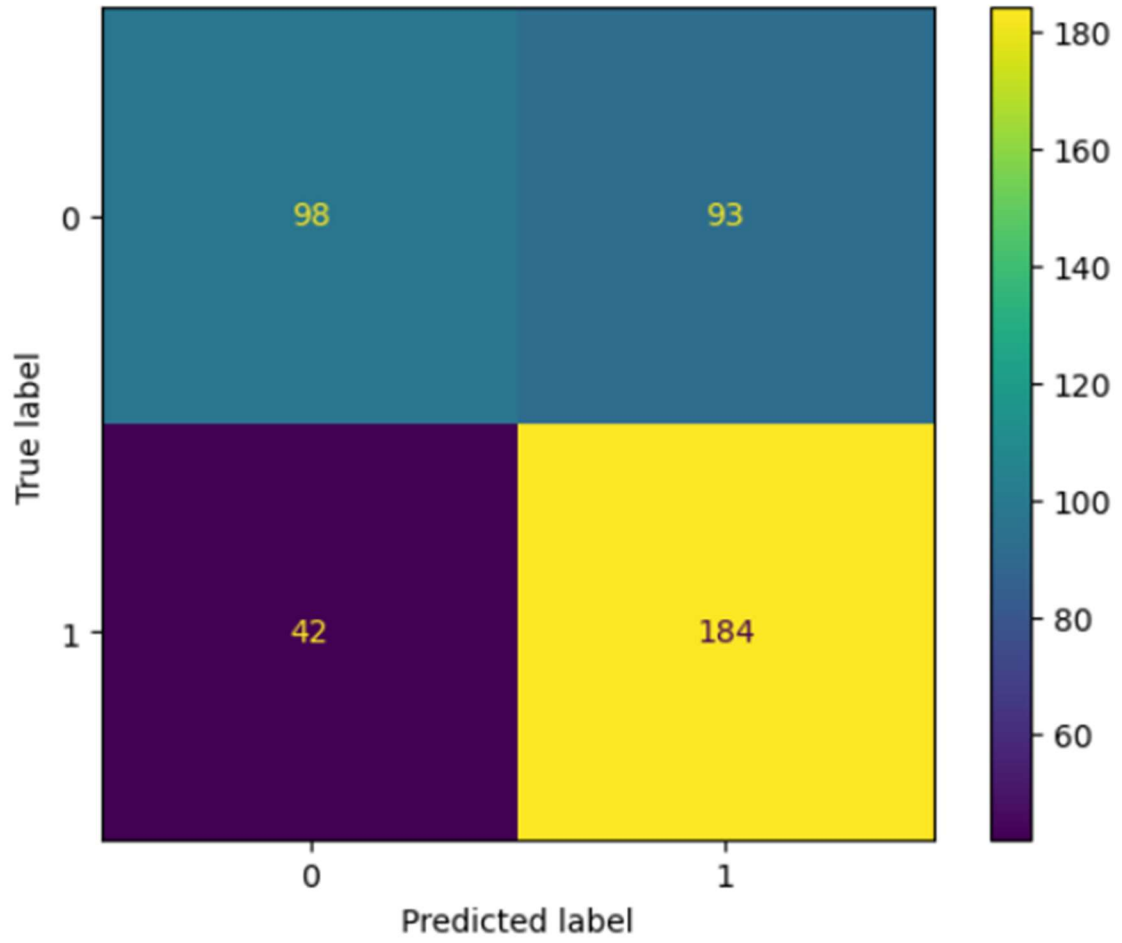
181 observations are actually negative but have been classified as positive

89 observations are positive but have been classified as negative

461 observations are positive and predicted as positive.

Confusion matrix on test data:

```
array([[ 98,  93],
       [ 42, 184]], dtype=int64)
```



Insights:

There are a total of 417 observations in training data.

Out of which 98 negatives have been correctly classified as negatives

93 observations are actually negative but have been classified as positive

42 observations are positive but have been classified as negative

184 observations are positive and predicted as positive.

Classification report for training data:

	precision	recall	f1-score	support
0	0.73	0.57	0.64	423
1	0.72	0.84	0.77	550
accuracy			0.72	973
macro avg	0.72	0.71	0.71	973
weighted avg	0.72	0.72	0.72	973

The above chart shows the precision and recall values. Row with index 0 indicates the precision recall and f-score values if 0 is considered to be positive.

Row with index 1 indicates the precision, recall and f-score values if 1 is considered to be positive.

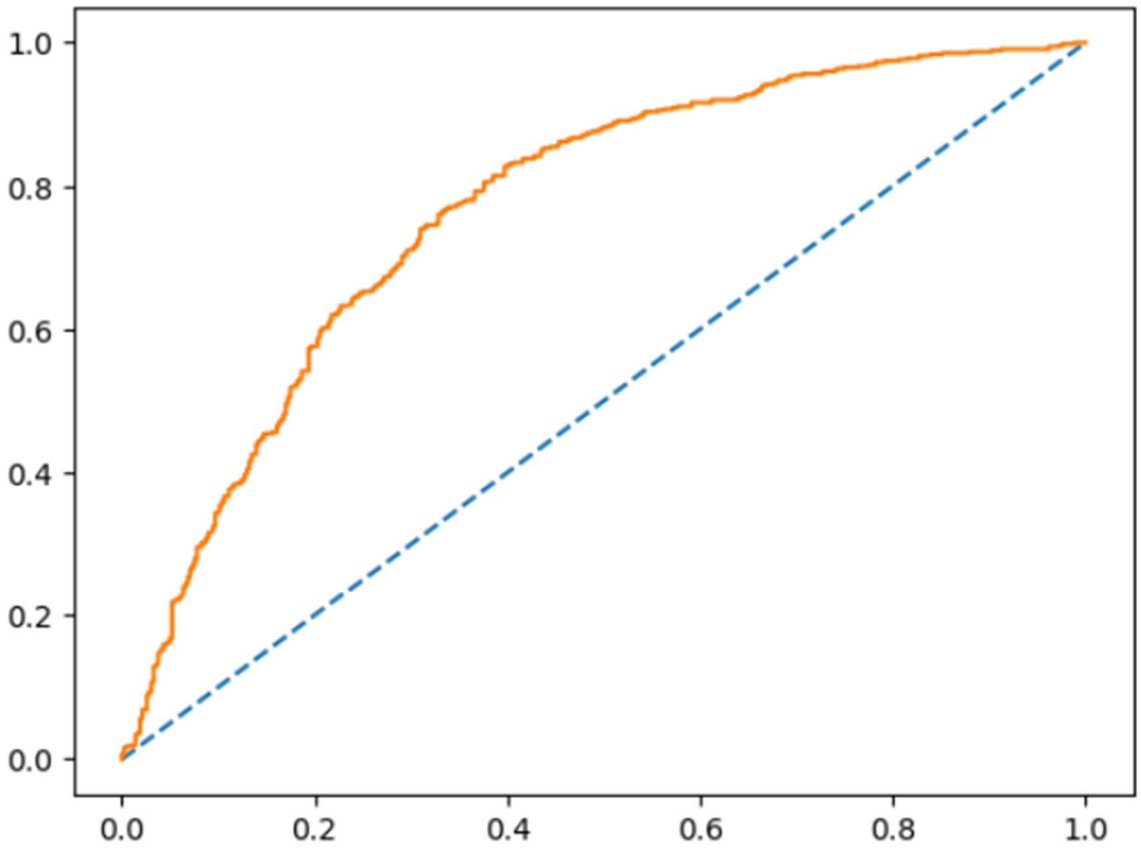
Classification report on testing data:

	precision	recall	f1-score	support
0	0.70	0.51	0.59	191
1	0.66	0.81	0.73	226
accuracy			0.68	417
macro avg	0.68	0.66	0.66	417
weighted avg	0.68	0.68	0.67	417

ROC Curve:

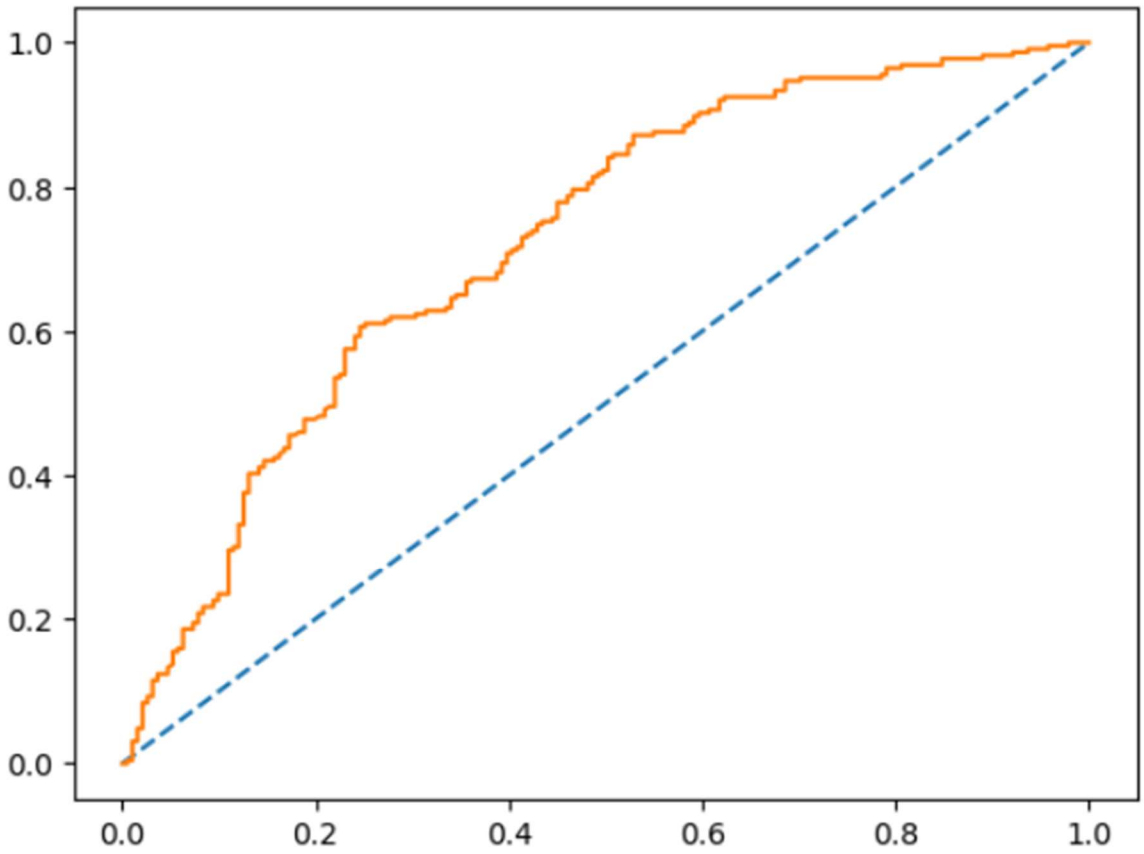
The curve is typically obtained by plotting 1 – specificity (False Positive Rate, FPR) on the x-axis and sensitivity (True Positive Rate, TPR) on the y-axis

ROC Curve for the training data:



Area under the ROC curve: AUC ROC score on training data: 0.76

ROC curve on test data:



AUC ROC score on test data: 0.72

Higher the AUC ROC score tends to 1 better is the model.

2. Performance metrics on Linear Discriminant Analysis:

Coefficients for the obtained linear discriminant analysis:

array([[-0.09, 0.44, 1.07, -0.24, -0. , -0.09, -0.05, 1.13, 1.86,
2.77, 3.15, 3.02, 3.25, 3.39, 2.26, 4.4 , 2.75, -0.32,
-0.09, -0.14, 0.1 , 0.96, -0.25, 0.21, -0.91, -0.31]])

Intercept for the obtained linear discriminant analysis:

array([0.83389084])

Predicted probabilities:

Predicted probabilities by the model on training data:

	0	1
0	0.175636	0.824364
1	0.112941	0.887059
2	0.217452	0.782548
3	0.153660	0.846340
4	0.752638	0.247362

Predicted probabilities by the model on test data:

	0	1
0	0.916088	0.083912
1	0.622876	0.377124
2	0.247417	0.752583
3	0.231228	0.768772
4	0.239941	0.760059

Insights: For row 0 in test dataset, the probability of predictor variable belonging to class 0 is 0.9 whereas the probability of predictor variable belonging to class 1 is 0.08

Accuracy:

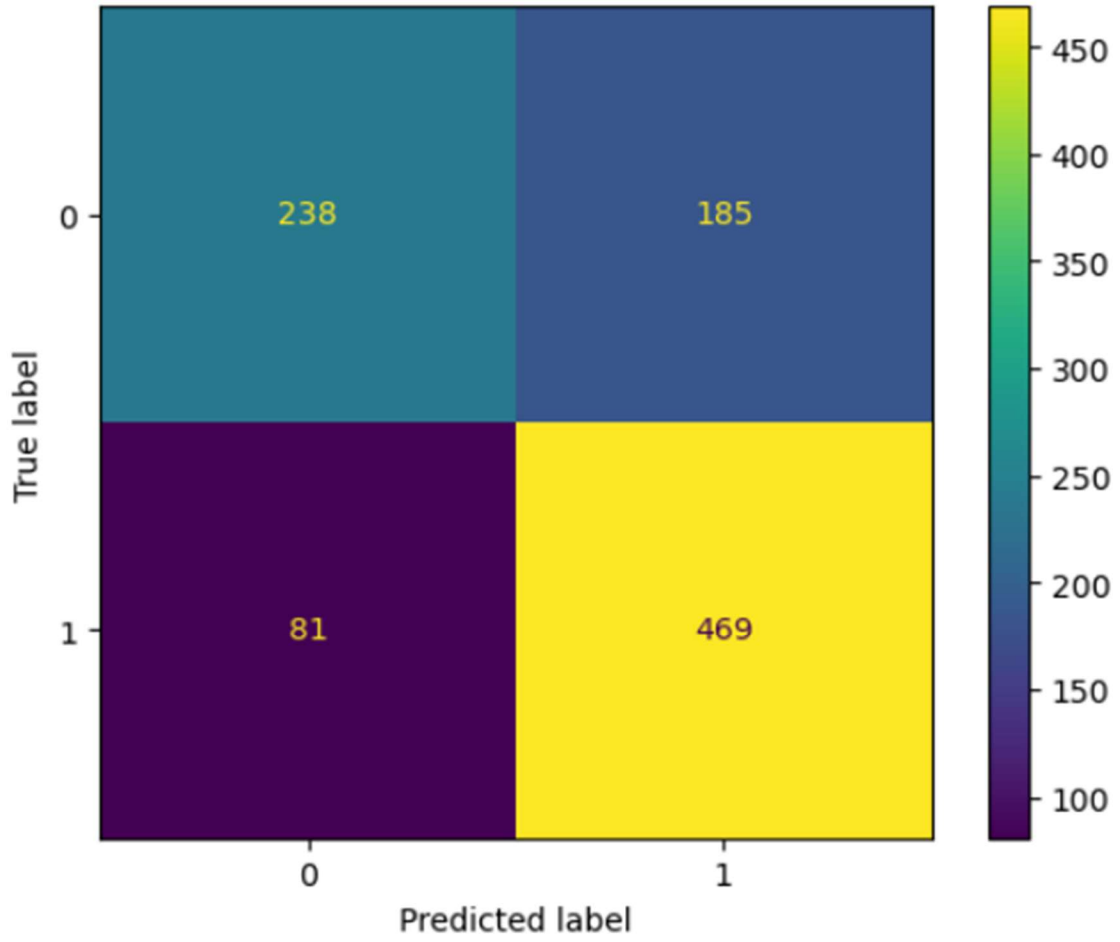
Accuracy of the LDAmodel on the training data: 0.73

Accuracy of the LDAmodel on the test data: 0.68

Insights: Accuracy for the training data is 73% whereas that of testing data is 68%. Accuracies of training and test data are close to each other which indicates that the model has decently generalized.

Confusion matrix:

Confusion matrix of training data:



Insights:

There are a total of 973 observations in training data.

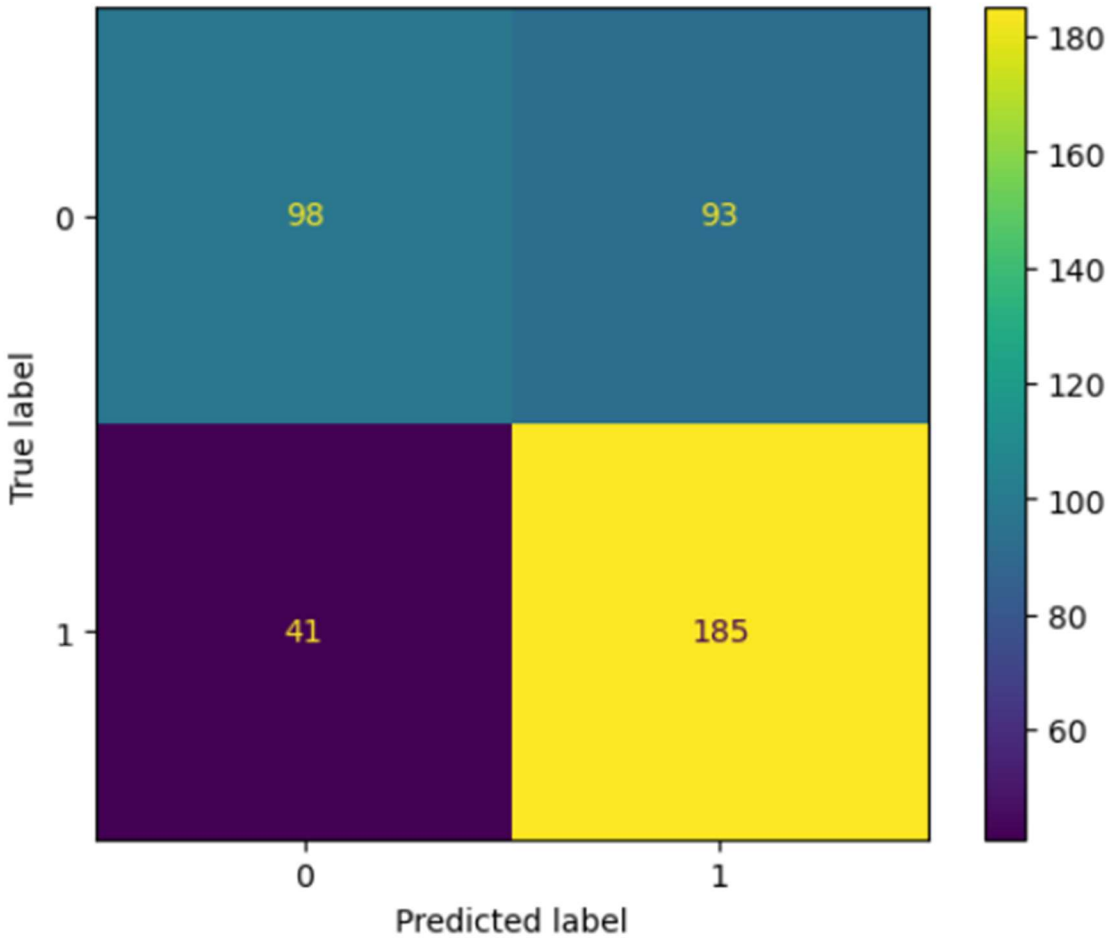
Out of which 238 negatives have been correctly classified as negatives

185 observations are actually negative but have been classified as positive

81 observations are positive but have been classified as negative

469 observations are positive and predicted as positive.

Confusion matrix on test data:



Insights:

There are a total of 417 observations in training data.

Out of which 98 negatives have been correctly classified as negatives

93 observations are negative but have been classified as positive

41 observations are positive but have been classified as negative

185 observations are positive and predicted as positive.

Classification report for training and test data:

	precision	recall	f1-score	support
0	0.75	0.56	0.64	423
1	0.72	0.85	0.78	550
accuracy			0.73	973
macro avg	0.73	0.71	0.71	973
weighted avg	0.73	0.73	0.72	973

Classification report for test data:

	precision	recall	f1-score	support
0	0.71	0.51	0.59	191
1	0.67	0.82	0.73	226
accuracy			0.68	417
macro avg	0.69	0.67	0.66	417
weighted avg	0.68	0.68	0.67	417

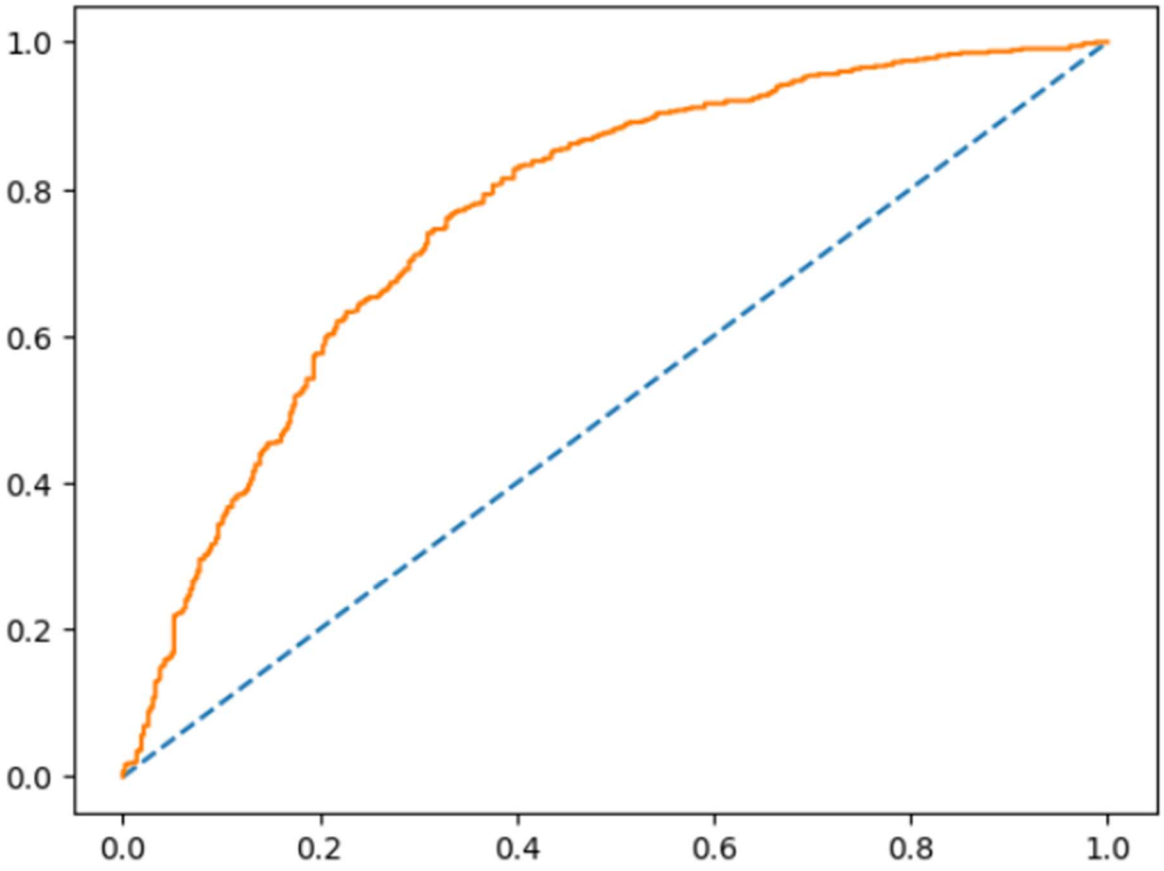
The above chart shows the precision and recall values. Row with index 0 indicates the precision recall and f-score values if 0 is positive.

Row with index 1 indicates the precision, recall and f-score values if 1 is considered to be positive.

ROC Curve:

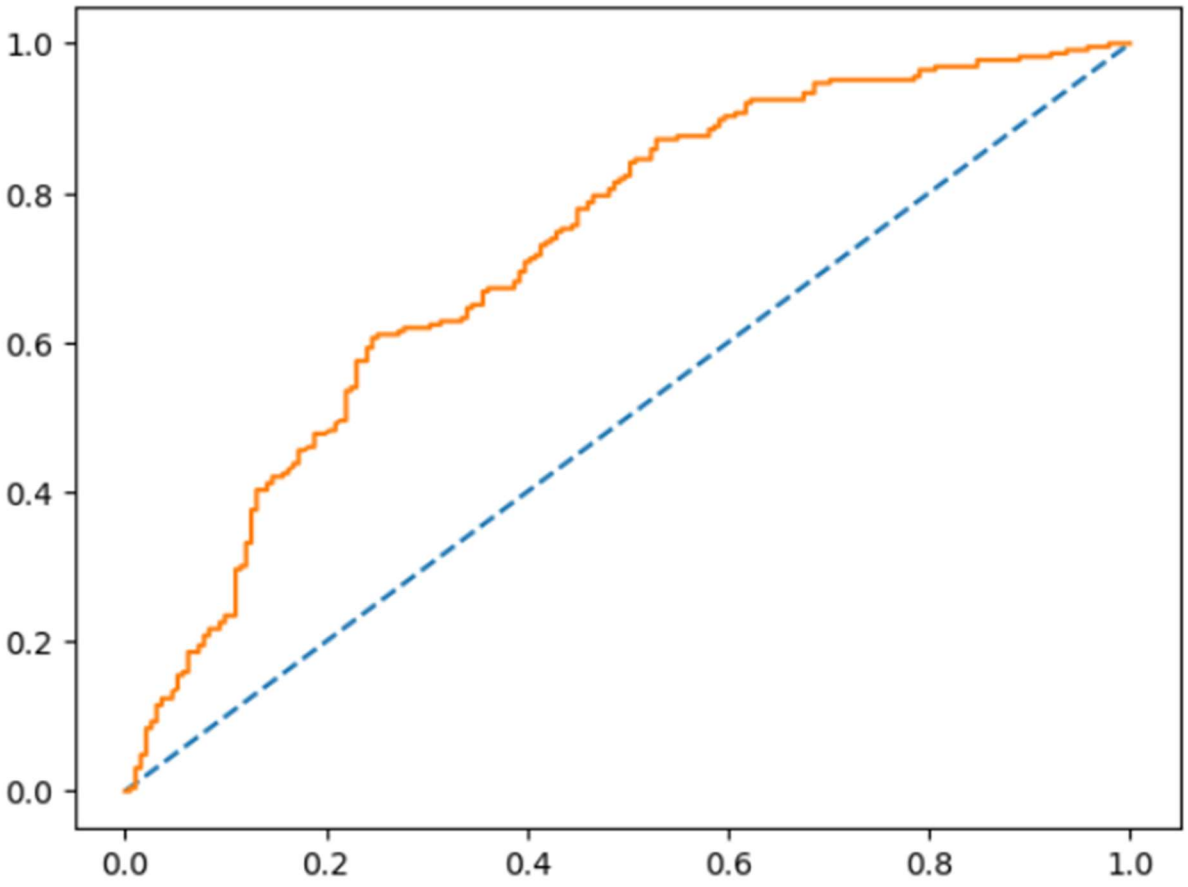
The curve is typically obtained by plotting 1 – specificity (False Positive Rate, FPR) on the x-axis and sensitivity (True Positive Rate, TPR) on the y-axis

ROC Curve for the training data:



Area under the ROC curve: AUC ROC score on training data: 0.77

ROC curve on test data:



AUC ROC score on test data: 0.73

Higher the AUC ROC score tends to 1 better is the model.

3. Performance metrics for the model using CART:

Important variables from CART model are determined to be:

	Imp
Wife_age	0.328969
No_of_children_born	0.171066
Wife_education	0.128195
Standard_of_living_index	0.102563
Husband_Occupation	0.100894
Husband_education	0.068860
Wife_Working	0.050528
Wife_religion	0.029139
Media_exposure	0.019785

Predicted probabilities:

Predicted probabilities by the model on training and test data:

```
     0    1
0  0.0  1.0
1  0.0  1.0
2  1.0  0.0
3  0.5  0.5
4  1.0  0.0
     0    1
0  1.0  0.0
1  1.0  0.0
2  0.0  1.0
3  0.0  1.0
4  1.0  0.0
```

Insights: For row 0 in test dataset, the probability of predictor variable belonging to class 0 is 1 whereas the probability of predictor variable belonging to class 1 is 0

AUC ROC score for the training data is 0.76 whereas for testing data is 0.58. This indicates that the model has overfitted.

Performance metrics on the regularized decision tree model:

Criteria used: DecisionTreeClassifier(max_depth=30, min_samples_leaf=10, min_samples_split=10,
random_state=1)

Importances of variables in regularized decision tree:

	Imp
Wife_age	0.307578
No_of_children_born	0.290978
Wife_education	0.231368
Husband_Occupation	0.056006
Standard_of_living_index	0.050708
Wife_Working	0.023300
Husband_education	0.022593
Media_exposure	0.012237
Wife_religion	0.005231

Predicted probabilities of training and test data respectively:

	0	1
0	0.083333	0.916667
1	0.000000	1.000000
2	0.600000	0.400000
3	0.066667	0.933333
4	1.000000	0.000000

	0	1
0	1.0	0.0
1	1.0	0.0
2	0.0	1.0
3	0.0	1.0
4	1.0	0.0

Accuracy:

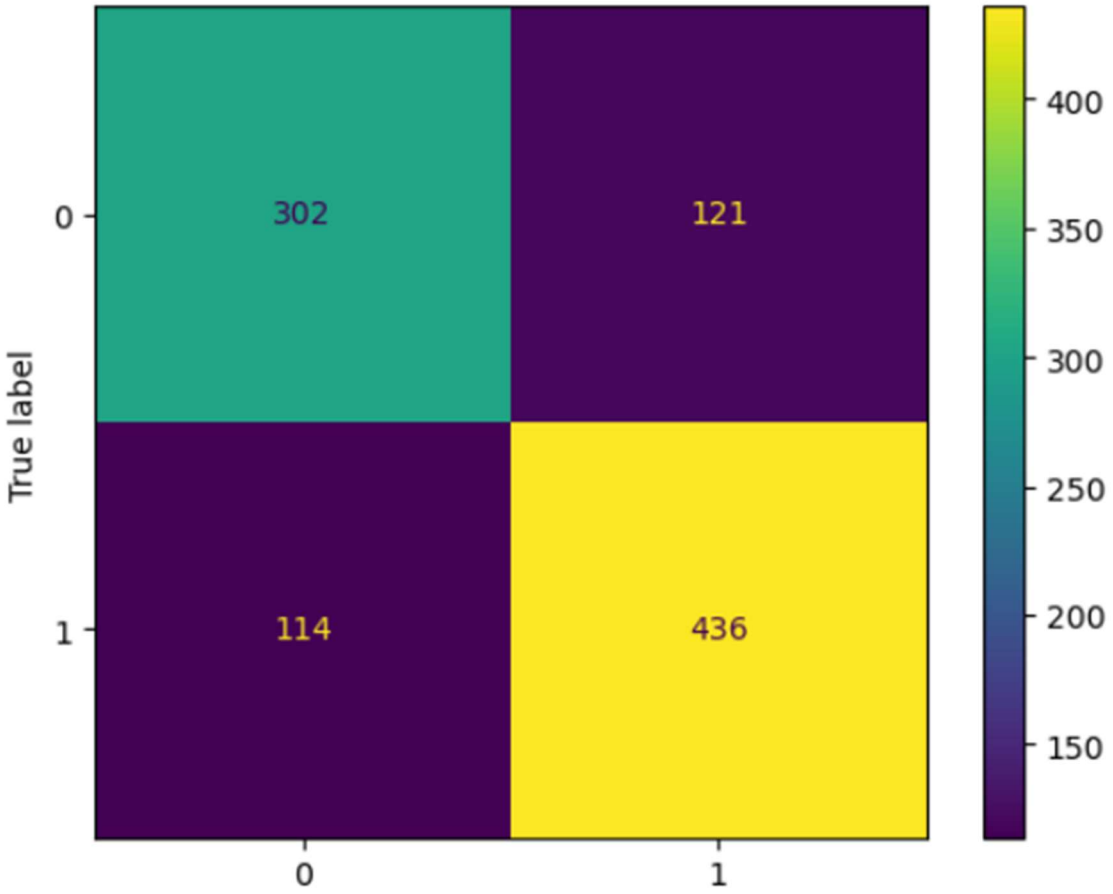
Accuracy of regularized decision tree model on training data: 0.7584789311408017

Accuracy of regularized decision tree on test data: 0.6546762589928058

Insights: Accuracy for the training data is 75% whereas that of testing data is 65%. Accuracies of training and test data are close to each other which indicates that the model has decently generalized.

Confusion matrix:

Confusion matrix of training data:



Insights:

There are a total of 973 observations in training data.

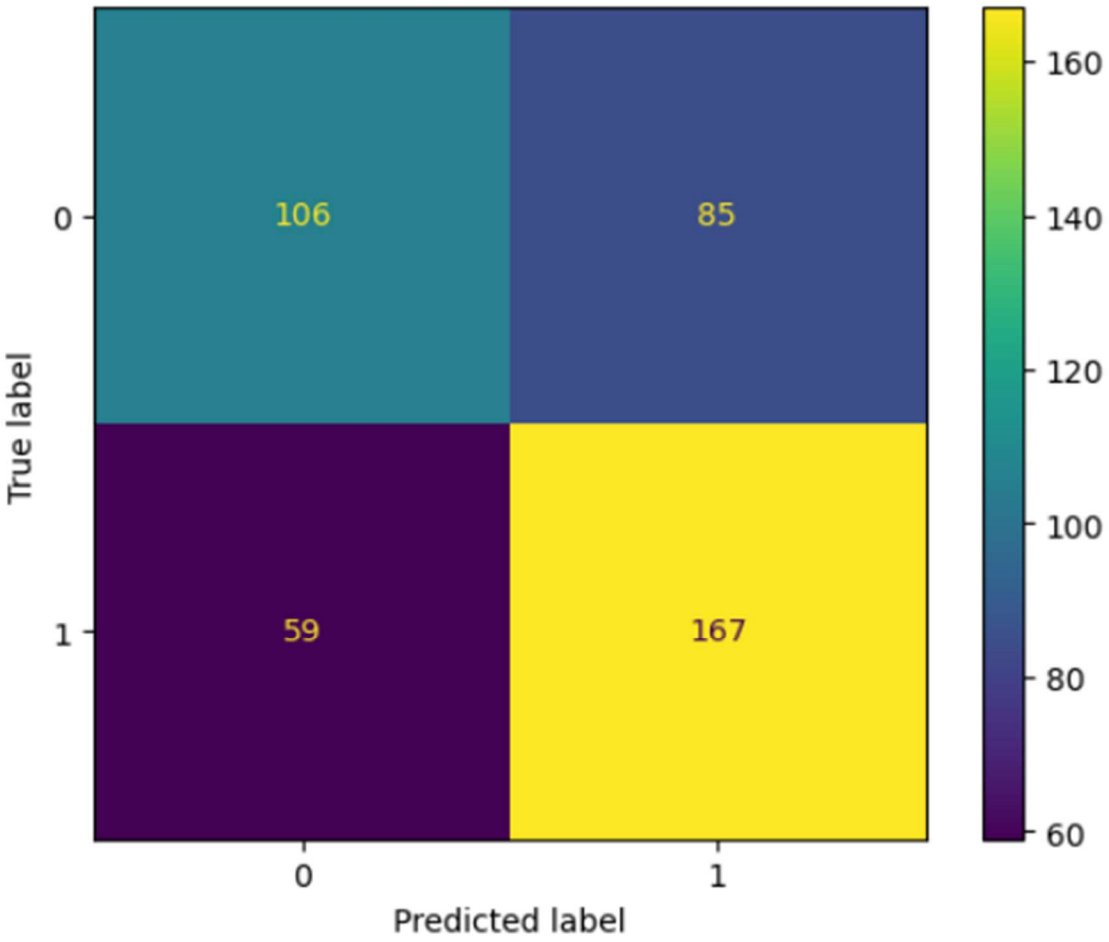
Out of which 302 negatives have been correctly classified as negatives

121 observations are actually negative but have been classified as positive

114 observations are positive but have been classified as negative

436 observations are positive and predicted as positive.

Confusion matrix on test data:



Insights:

There are a total of 417 observations in training data.

Out of which 106 negatives have been correctly classified as negatives

85 observations are actually negative but have been classified as positive

59 observations are positive but have been classified as negative

167 observations are positive and predicted as positive.

Classification report for training data and test data:

Classification report for training data:

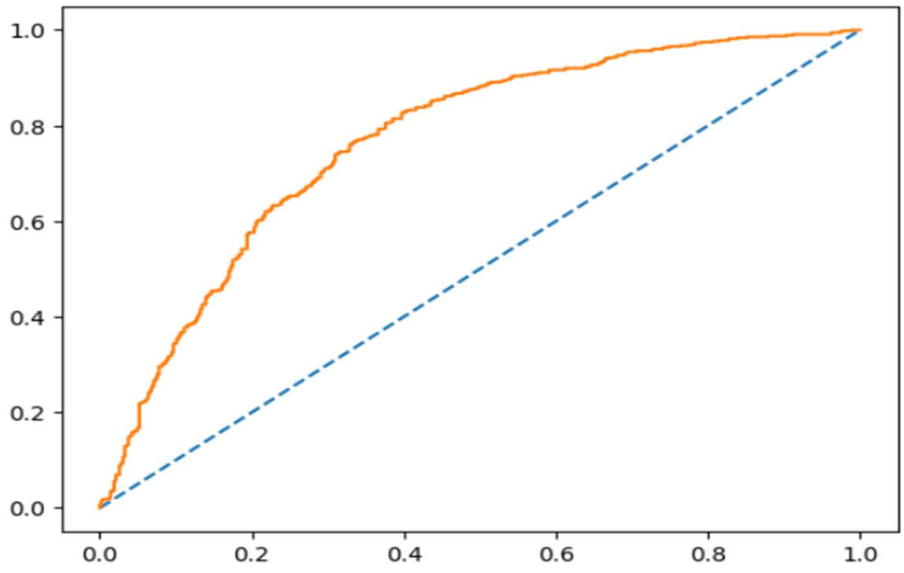
	precision	recall	f1-score	support
0	0.73	0.57	0.64	423
1	0.72	0.84	0.77	550
accuracy			0.72	973
macro avg	0.72	0.71	0.71	973
weighted avg	0.72	0.72	0.72	973

Classification report for test data:

	precision	recall	f1-score	support
0	0.70	0.51	0.59	191
1	0.66	0.81	0.73	226
accuracy			0.68	417
macro avg	0.68	0.66	0.66	417
weighted avg	0.68	0.68	0.67	417

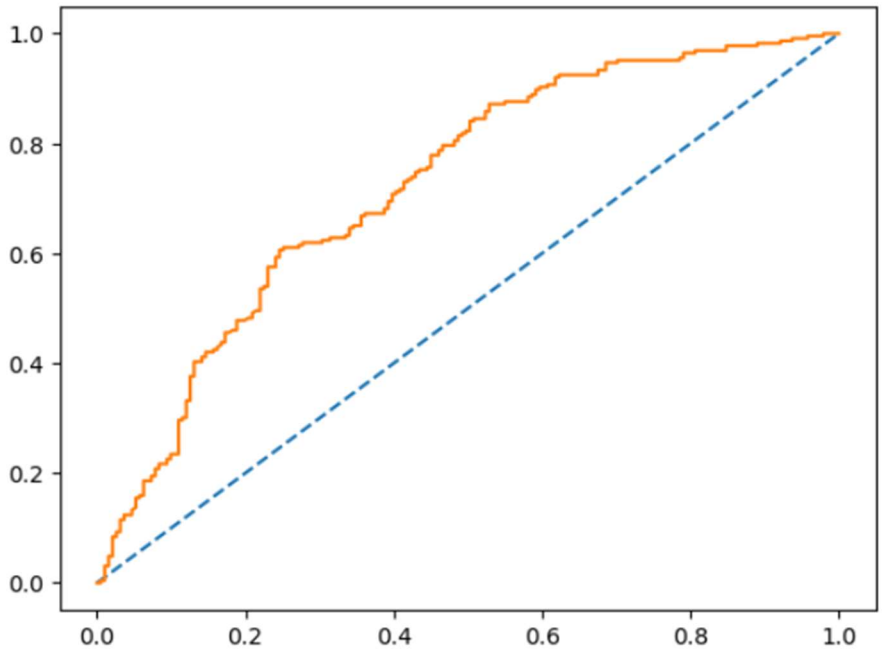
The above chart shows the precision and recall values. Row with index 0 indicates the precision recall and f-score values if 0 is considered to be positive.
Row with index 1 indicates the precision, recall and f-score values if 1 is considered to be positive.

ROC Curve:
The curve is typically obtained by plotting 1 – specificity (False Positive Rate, FPR) on the x-axis and sensitivity (True Positive Rate, TPR) on the y-axis
ROC Curve for the training data:



Area under the ROC curve: AUC ROC score on training data: 0.76

ROC curve on test data:



AUC ROC score on test data: 0.72

Higher the AUC ROC score tends to 1 better is the model.

Comparing all the models:

Accuracy of the model is highest for the model built using regularized decision tree.

According to business understanding, contraceptive method used is related to health care. We are more interested in the reduction of false negatives in this case as it is not desired to misinterpret women who have used contraceptive method as not used.

Let's consider the recall values for all the 3 models considering 1 as positive. (contraceptive method used)

Recall for all the 3 models is : 0.81

Since it is the same, we will consider accuracy to find the best fit model.

Accuracy for training and test data for logistic regression or LDA model are 72% and 67% respectively.

Accuracy for training and test data of initial model for CART is 98% and 58%

This indicates that the decision tree has overfitted.

Accuracy of the regularized decision tree model is 75% for training data and 65% for testing data.

Considering the accuracies and ROC scores of all the models built, either LDA or logistic regression model can be chosen to predict the values as the scores of trained and test data are comparable and the test data has high accuracy in these models compared to others. We choose LDA model as it gives good accuracy and we will be able to interpret influence of each independent variable on dependent variable.

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

Steps performed in the problem for predicting whether a women used contraceptive method or not:

1. The dataset has been read.
2. Initial analysis has been performed on the data

    a. First and last rows of the data have been viewed.
    b. Basic information about the dataset has been analysed. We had some null values and integer data types for categorical columns
    c. Overall summary of the data has been analysed.
    d. Null values for Wife_age have been treated with mean and number of children born with 0
    e. Inconsistent datatypes for columns have been realigned
    f. The dataset has been found to contain duplicates. Removed them.
3. The numeric columns have been checked for outliers. No outliers detected. So we did not have to treat outliers.
4. Univariate analysis has been done on the data. Key takeaways are
    a. There are slightly more women in the dataset who have used contraceptive methods.
    b. Most of the wives are exposed to media
5. Bivariate analysis has been done on the data. Key takeaways
    a. Wives having education level 'Tertiary' have used contraceptive methods the most.
    b. Wives having 2-4 children have used contraceptive methods more than the wives having less or more children
    c. Non working and media exposed wives have used contraceptive methods the most
6. Multivariate analysis has been done on the data. Key takeaways
    a. in 41% of cases, a wife and husband having tertiary education have resorted to using contraceptive methods.
    b. Non-working wives tend to use contraceptive methods at a higher age compared to working women.
7. The given data has been encoded for categorical variables to be able to pass it to regression model using one hot encoding
8. Logistic regression has been applied to the data with below parameters obtained using GridSearchCV
9. GridSearchCV(cv=3, estimator=LogisticRegression(n_jobs=-1), n_jobs=-1, param_grid={'penalty': ['l2', 'none'], 'solver': ['sag', 'newton-cg']}, scoring='f1')
10. Performance metrics of the model have been evaluated

11. Linear Discriminant Analysis model from scikit learn has been applied on the dataset
12. Performance metrics for the model have been evaluated.
13. Basic transformation has been done on the dataset to be able to fit it into the decision tree classifier.
14. The data has also been fed into Regularized decision tree with different parameters considering generalization.
15. Performance metrics on the model have been evaluated.

Business insights and actionable references:

1. From the accuracy and AUC ROC scores of the models, either logistic regression or LDA model can be used for prediction for the given problem. Since LDA exhibits slights high accuracy than Logistic regression we will interpret the results and actionable insights from linear discriminant analysis model.
2. Interpreting the selected regression models:
    a. The coefficients of the linear discriminant function obtained for the data is:

0.83 + (-0.09 * Wife_age) + (0.44 * Wife_education_Secondary) + (1.07 * Wife_education_Tertiary) + (-0.24 * Wife_education_Uneducated) + (0 * Husband_education_Secondary) + (-0.09 * Husband_education_Tertiary) + (-0.05* Husband_education_Uneducated) + (1.13 * No_of_children_born_1) + (1.86* No_of_children_born_2) + (2.77* No_of_children_born_3) + (3.15 *No_of_children_born_4) + (3.02 * No_of_children_born_5) + (3.25 * No_of_children_born_6) + (3.39 * No_of_children_born_7) + (2.26 * No_of_children_born_8) + (4.4 * No_of_children_born_9) + (2.75 * No_of_children_born_10) + (-0.32 * Wife_religion_Scientology) + (-0.09 * Wife_Working_Yes) + (-0.14 * Husband_Occupation_2) + (0.1 * Husband_Occupation_3) + (-0.96 * Husband_Occupation_4) + (-0.25 * Standard_of_living_index_Low) + (0.21 * Standard_of_living_index_High) + (-0.91 * Standard_of_living_index_Very Low) + (-0.31 * Media_exposure_Not-Exposed)

From the above equation following actionable insights can be deduced:

1. For a unit increase in Wife age, the probability that the women will use contraceptive methods decreases by 9% keeping all other independent factors constant.
2. Independent variables having higher positive coefficients move the probability value of the equation to 1. This means that to predict if the woman uses a contraceptive method or not we can tweak the variables having higher positive coefficients. Women with such characteristics tend to use contraceptive methods compared to women with other characteristics.
3. Among the different independent variables, the equation suggests that if a woman is having tertiary education or if the woman has more children, the probability that she uses contraceptive methods tends to 1.
4. Media exposure not exposed has negative coefficient it means that, if the woman is not media exposed, probability that she resides to contraceptive methods reduces.

---------- END OF REPORT----------