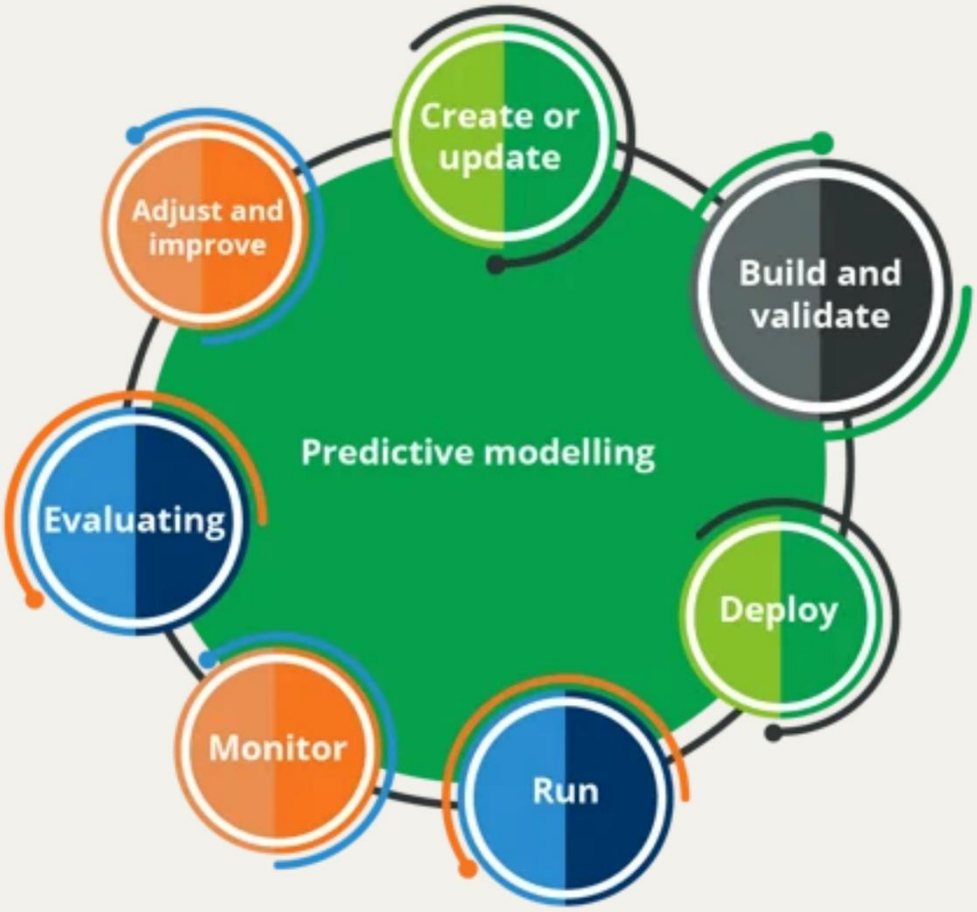
PREDICTIVE MODELLING CODED PROJECT



BY
R.SUKANYA

CONTENT

LIST OF FIGURES

LIST OF TABLES

Scoring guide (Rubric) - PM Project Rubric

Criteria	Points
Problem 1 - Define the problem and perform exploratory Data Analysis- Problem definition - Check shape, Data types, statistical summary - Univariate analysis - Multivariate analysis - Use appropriate visualizations to identify the patterns and insights - Key meaningful observations on individual variables and the relationship between variables	**8**
Problem 1 - Data Pre-processing Prepare the data for modelling: - Missing Value Treatment (if needed) - Outlier Detection (treat, if needed) - Feature Engineering - Encode the data - Train-test split	**5**
Problem 1- Model Building - Linear regression - Apply linear Regression using Sklearn - Using Statsmodels Perform checks for significant variables using the appropriate method - Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare.	**8**
Problem 1 - Business Insights & Recommendations - Comment on the Linear Regression equation from the final model and impact of relevant variables (atleast 2) as per the equation - Conclude with the key takeaways (actionable insights and recommendations) for the business	**5**
Problem 2 - Define the problem and perform exploratory Data Analysis - Problem definition - Check shape, Data types, statistical summary - Univariate analysis - Multivariate analysis - Use appropriate visualizations to identify the patterns and insights - Key meaningful observations on individual variables and the relationship between variables	**8**
Problem 2 - Data Pre-processing Prepare the data for modelling: - Missing value Treatment (if needed) - Outlier Detection(treat, if needed) - Feature Engineering (if needed) - Encode the data - Train-test split	**3**
Problem 2 - Model Building and Compare the Performance of the Models - Build a Logistic Regression model - Build a Linear Discriminant Analysis model - Build a CART model - Prune the CART model by finding the best hyperparameters using GridSearch - Check the performance of the models across train and test set using different metrics - Compare the performance of all the models built and choose the best one with proper rationale	**11**
Problem 2 - Business Insights & Recommendations - Comment on the importance of features based on the best model - Conclude with the key takeaways (actionable insights and recommendations) for the business	**6**
Business Report Quality - Adhere to the business report checklist	**6**
TOTAL	**60**

Data Description

Problem 1

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

Problem 2

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

Problem - 1

The comp-activ database comprises activity measures of computer systems. Data was gathered from a Sun Sparcstation 20/712 with 128 Mbytes of memory, operating in a multi-user university department. Users engaged in diverse tasks, such as internet access, file editing, and CPU-intensive programs.

Being an aspiring data scientist, you aim to establish a linear equation for predicting 'usr' (the percentage of time CPUs operate in user mode). Your goal is to analyze various system attributes to understand their influence on the system's 'usr' mode.

1.1 Define the problem and perform EDA (Exploratory Data Analysis)

1.1.1 Importing the libraries

Import the necessary libraries for data processing, data visualization, modelling, validation, data splitting, building the linear regression model and to check the model performance.

1.1.2 Reading and loading the dataset compactiv.xls:

Load and read the dataset using the read_excel() function of pandas. The first five lines of the dataset is as follows:

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

5 rows × 22 columns

Table 1.1: Reading the first 5 rows of the dataset

1.1.3 Checking the shape of the dataset:

Find out the number of rows and columns in the dataset using the shape command.
The dataset has 8192 rows and 22 columns.

1.1.4 Checking the datatypes of all the columns:

Use the info() command to find the datatypes of all the columns in the dataset. It also tells you if the dataset has missing values.

```
 #   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
```

Search

Table 1.2: Checking the datatype of all the columns

Insights:

- There are 21 numerical datatype column and 1 object datatype column.
- The 'runqsz' column is a categorical column.
- All columns have non-null values. The columns rchar and wchar have only 8088 and 8177 records, instead of 8192 records. This needs to be investigated further.

1.1.5 Checking the statistical summary:

Use the describe() function to get the statistical summary of the object type columns.

	count	unique	top	freq
runqsz	8192	2	Not_CPU_Bound	4331

Table 1.3: Statistical summary of object type column

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

Table 1.4: Statistical summary of numerical type columns

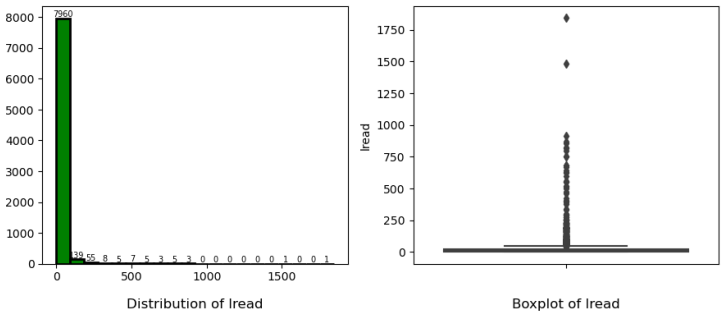
Insights:

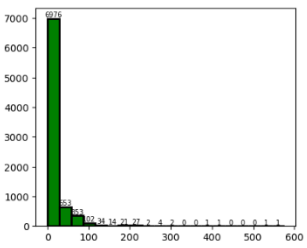
- The skewness is maximum for atch variable followed by lread column.
- Freeswap column has the maximum mean of 1328126 followed by rchar which has a mean of 197385.
- The median value is maximum for freeswap column which is 1289289 followed by rchar.
- The range of rchar is the maximum ranging from a minimum of 278 to a maximum of 2526649.
- The variability is the most in freeswap column with standard deviation of 422019.
- The rchar column has the maximum value in the dataset.

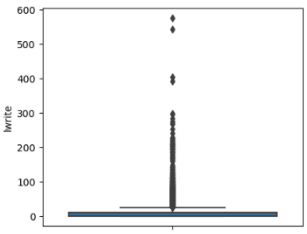
1.1.6 Univariate Analysis

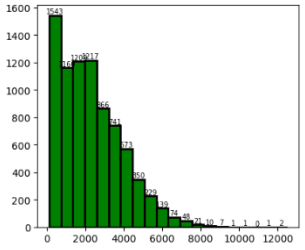
Univariate analysis of all the fields is done using a histogram and a boxplot. First, we divide the dataset into a dataset with numerical fields and another dataset with categorical fields.

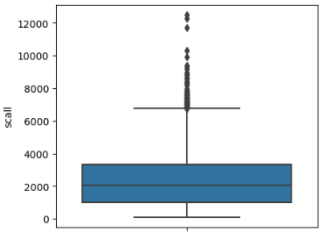
1.1.6.1 Numerical columns

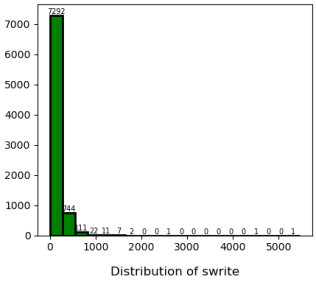


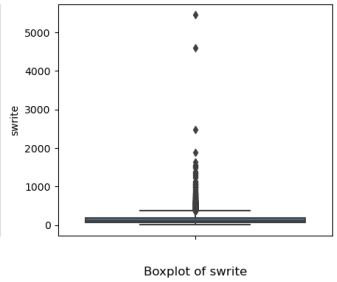

Distribution of lwrite

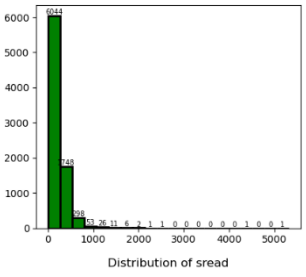

lwrite
Boxplot of lwrite

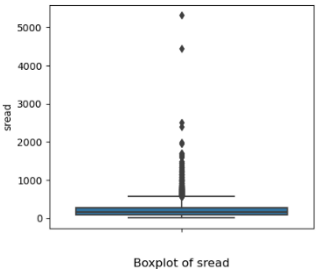

Distribution of scall

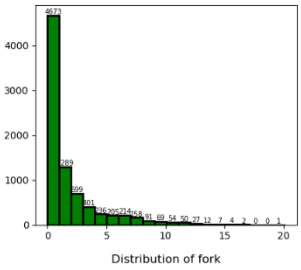

scall
Boxplot of scall

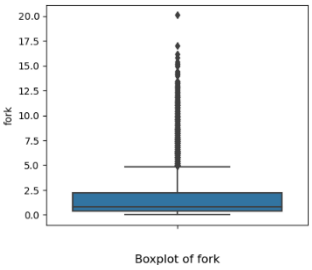

Distribution of swrite

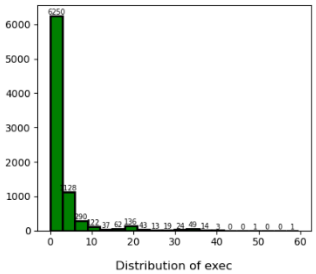

swrite
Boxplot of swrite

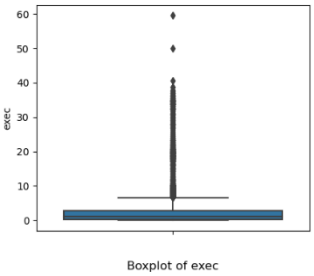

Distribution of sread

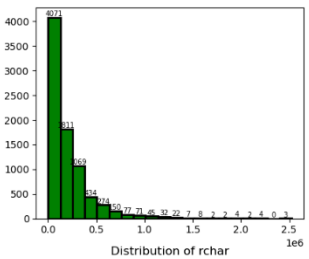

sread
Boxplot of sread

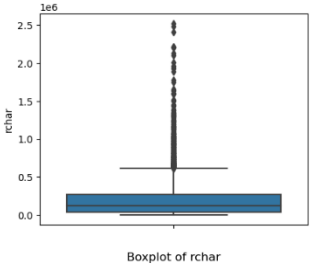

Distribution of fork

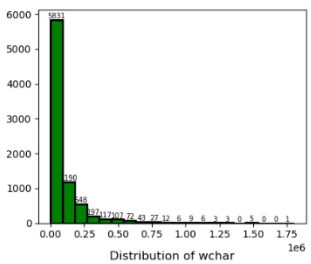

fork
Boxplot of fork

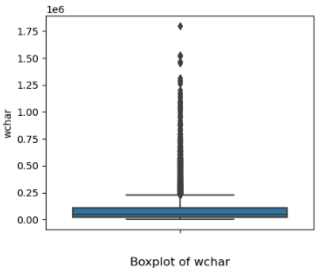

Distribution of exec

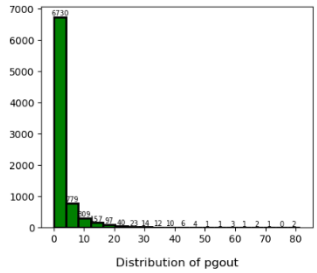

exec
Boxplot of exec

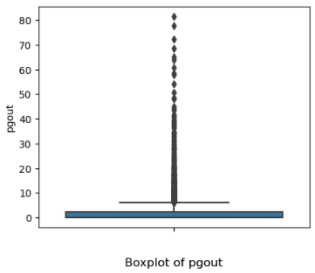

Distribution of rchar
1e6

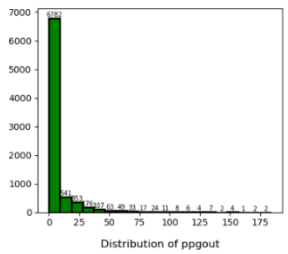

1e6
rchar
Boxplot of rchar

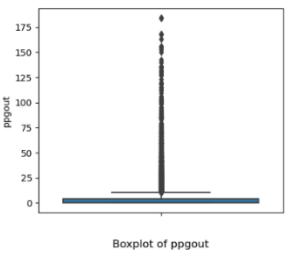

Distribution of wchar
1e6


1e6
wchar
Boxplot of wchar


Distribution of pgout


pgout
Boxplot of pgout


Distribution of ppgout


ppgout
Boxplot of ppgout

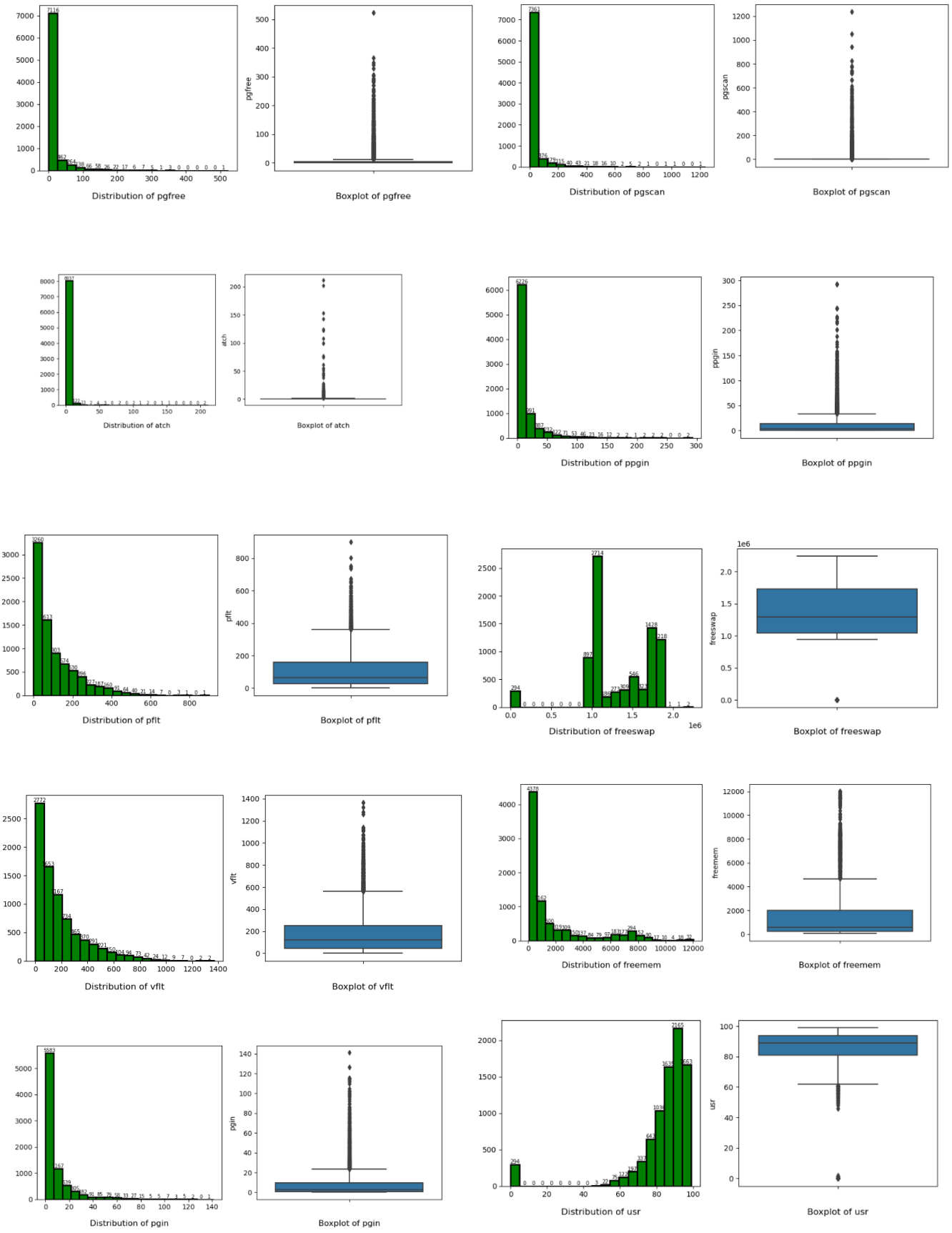


Fig 1.1: Univariate analysis of Numerical columns

1.1.6.2 Categorical columns

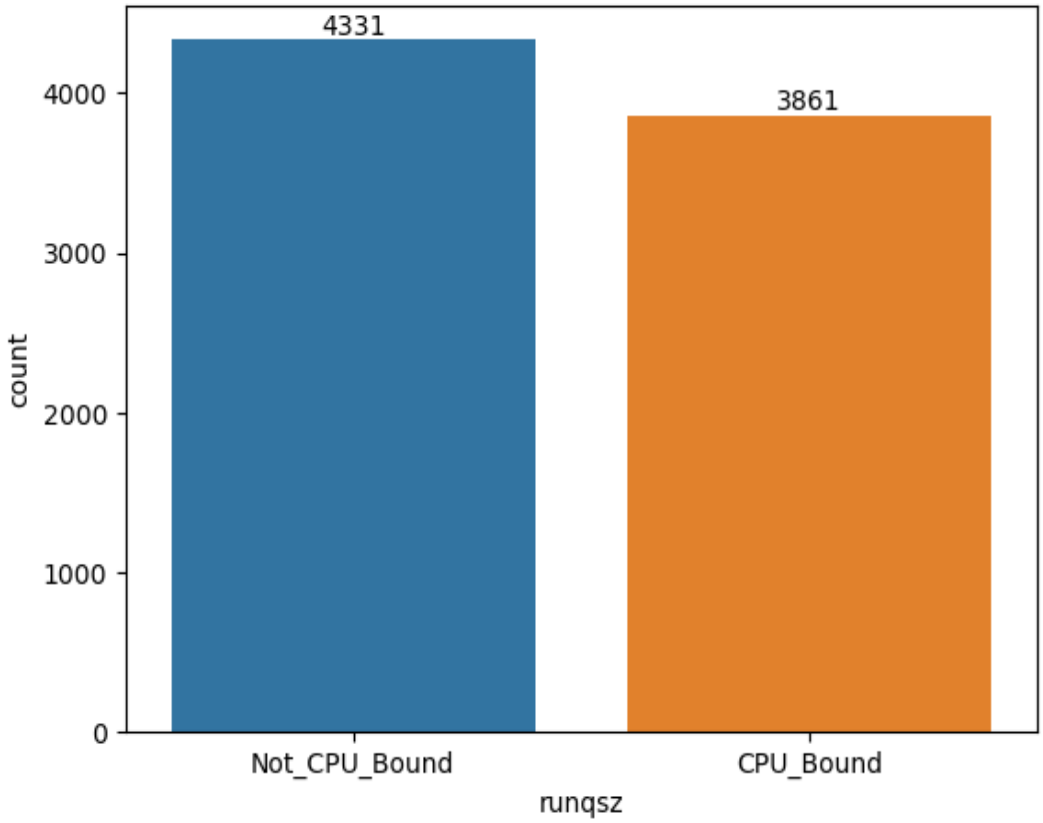


Fig 1.2: Univariate Analysis of Object type column - runqsz

Insights:

- There are 21 numerical fields in the dataset.
- All numerical columns are right-skewed, except freeswap and usr fields. The right-skewed fields have the mean values greater than the medians.
- The freeswap and usr columns are left-skewed.
- All fields have outliers.
- The median of the usr field is at 89 while the mean is 83.
- The free memory, Freemem field has a median of 579 with mean of 1763.
- The outliers need to be treated.
- We observe that the count of Not_CPU_Bound process run queue size is more than that of CPU_Bound queue size. Not_CPU_Bound is 4331 and CPU_Bound is 3861.

1.1.7 Multivariate Analysis - Visualizations to identify the pattern and insights

1.1.7.1 Numeric vs Numeric

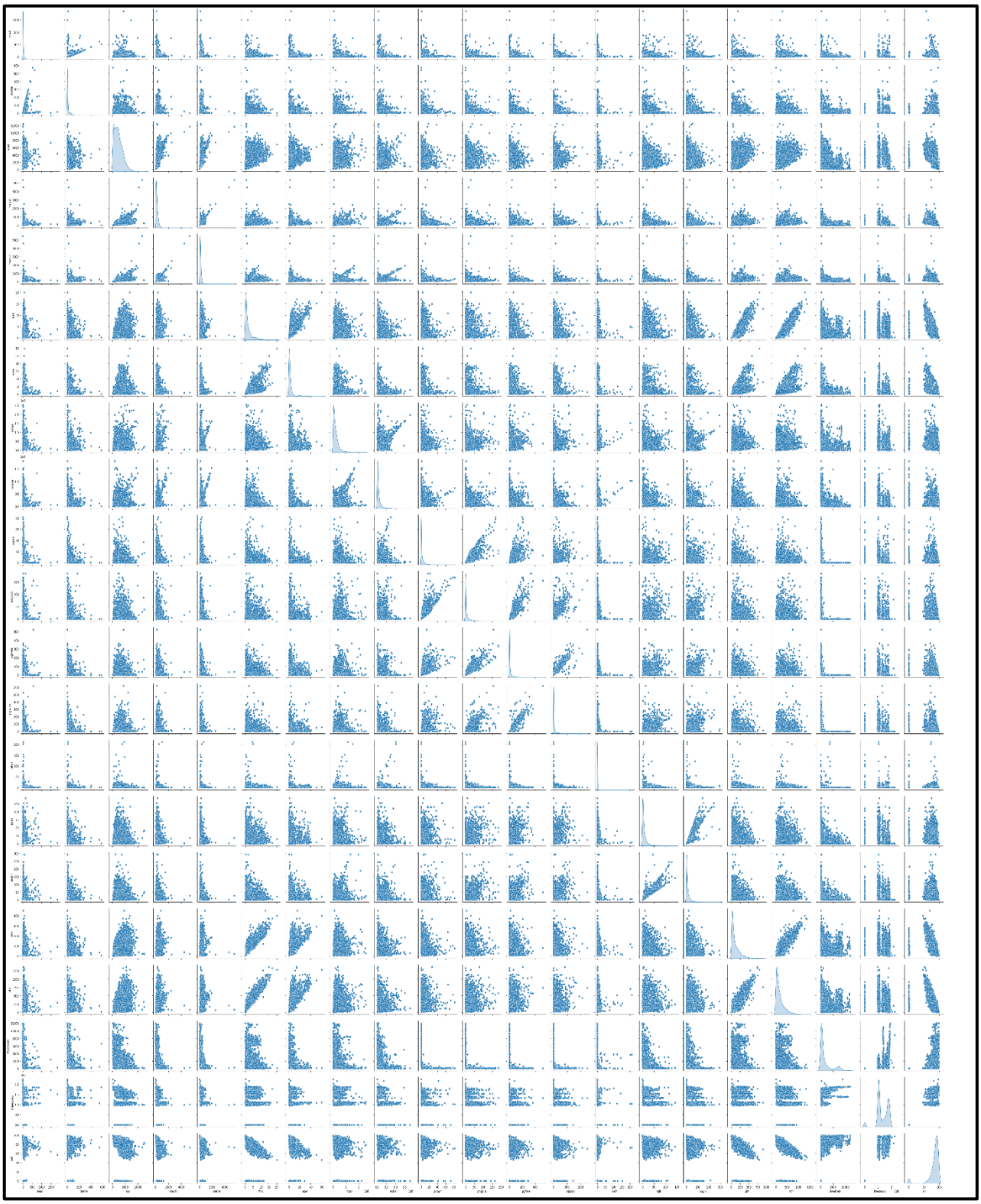
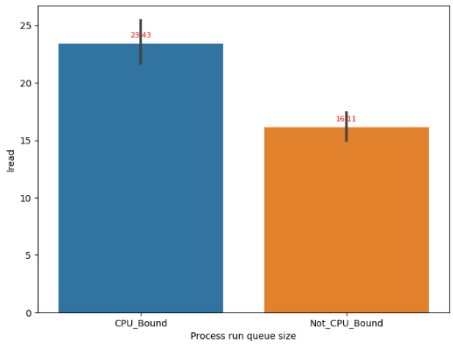
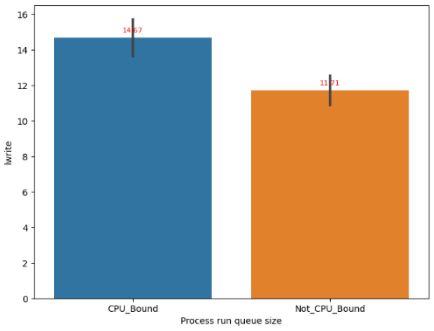


Fig 1.3: Pairplot of all the Numeric vs Numeric variables

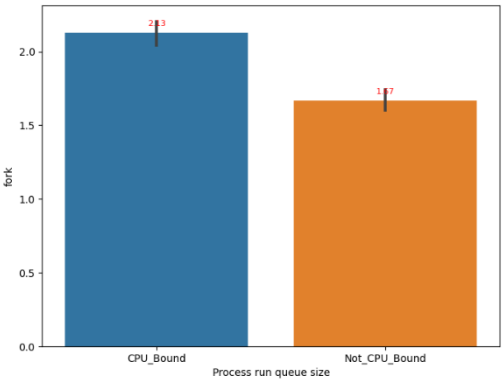
1.1.7.2 Categorical vs Numerical



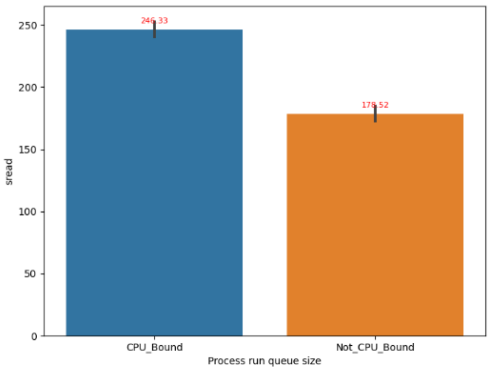
Bar plot of lread against Process run queue size



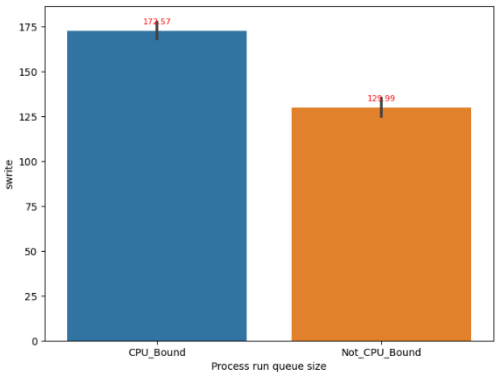
Bar plot of lwrite against Process run queue size



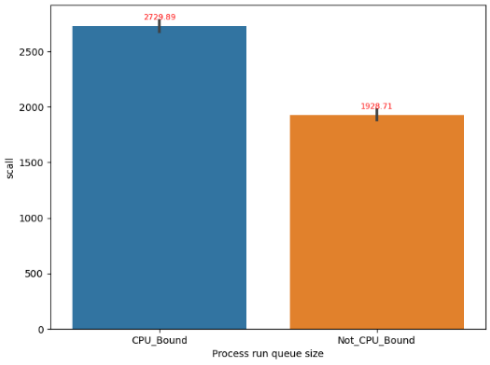
Bar plot of fork against Process run queue size



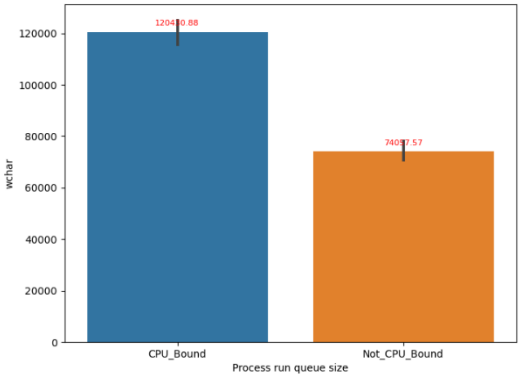
Bar plot of sread against Process run queue size



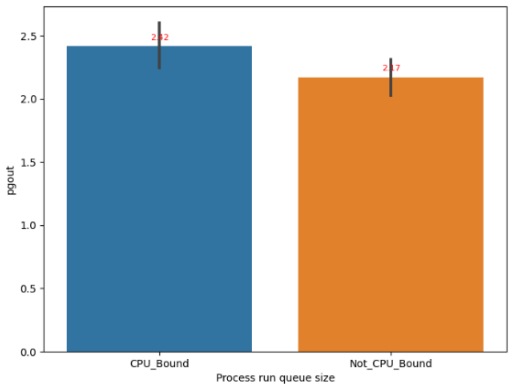
Bar plot of swrite against Process run queue size



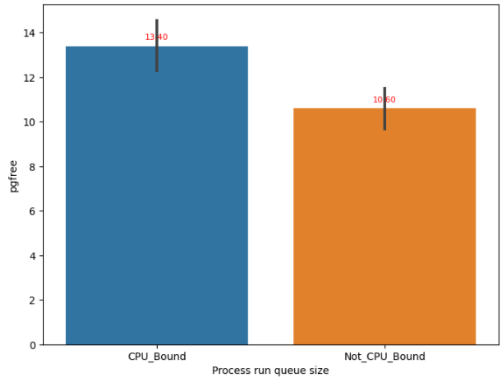
Bar plot of scall against Process run queue size



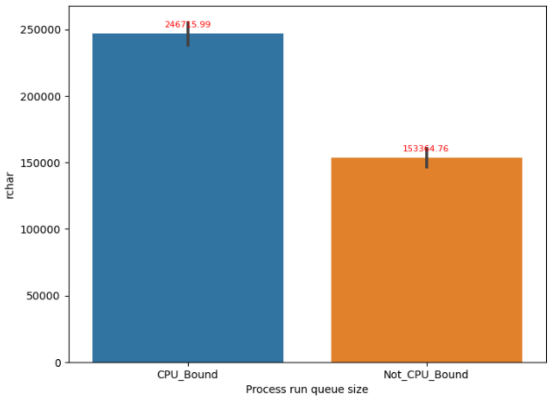
Bar plot of wchar against Process run queue size



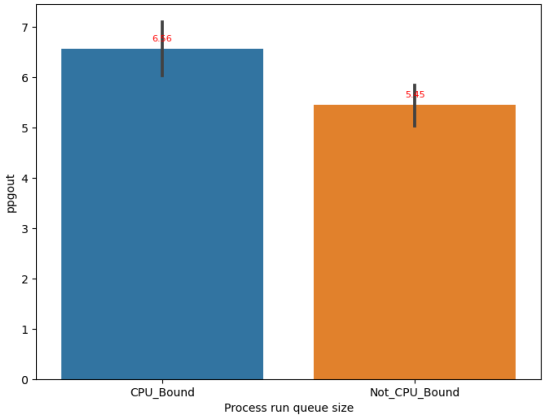
Bar plot of pgout against Process run queue size



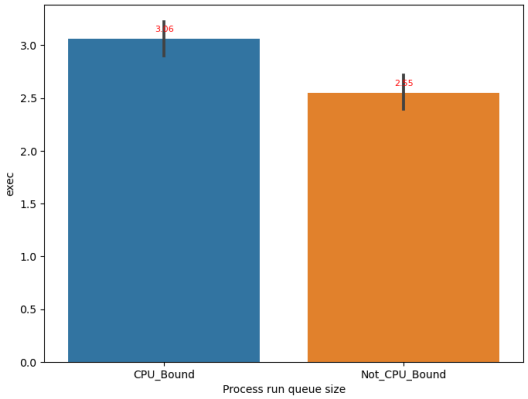
Bar plot of pgfree against Process run queue size



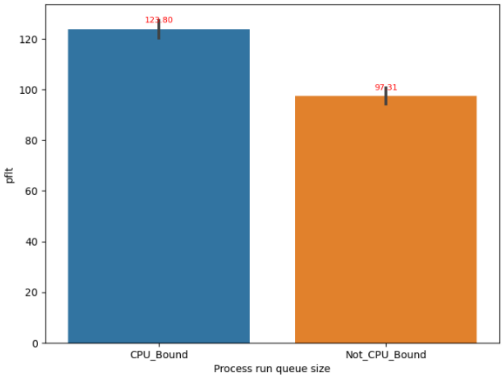
Bar plot of rchar against Process run queue size



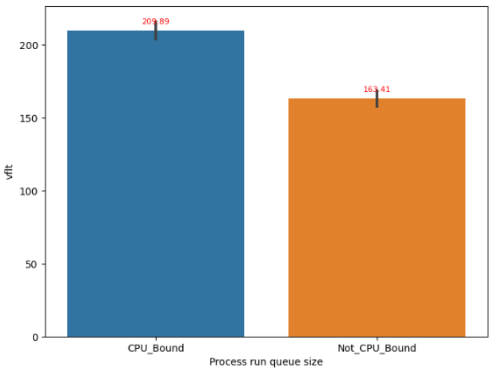
Bar plot of ppgout against Process run queue size



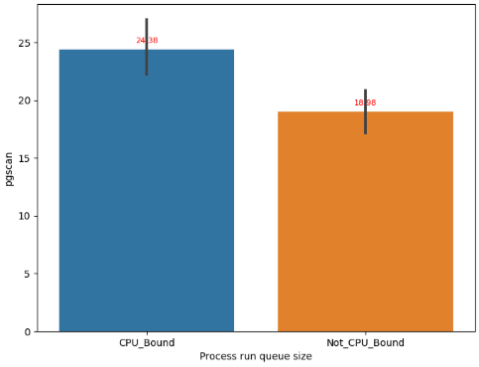
Bar plot of exec against Process run queue size



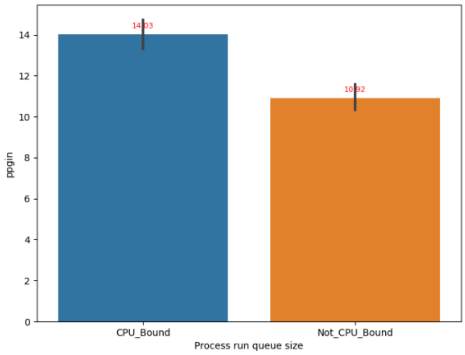
Bar plot of pflt against Process run queue size



Bar plot of vflt against Process run queue size



Bar plot of pgscan against Process run queue size



Bar plot of ppgin against Process run queue size

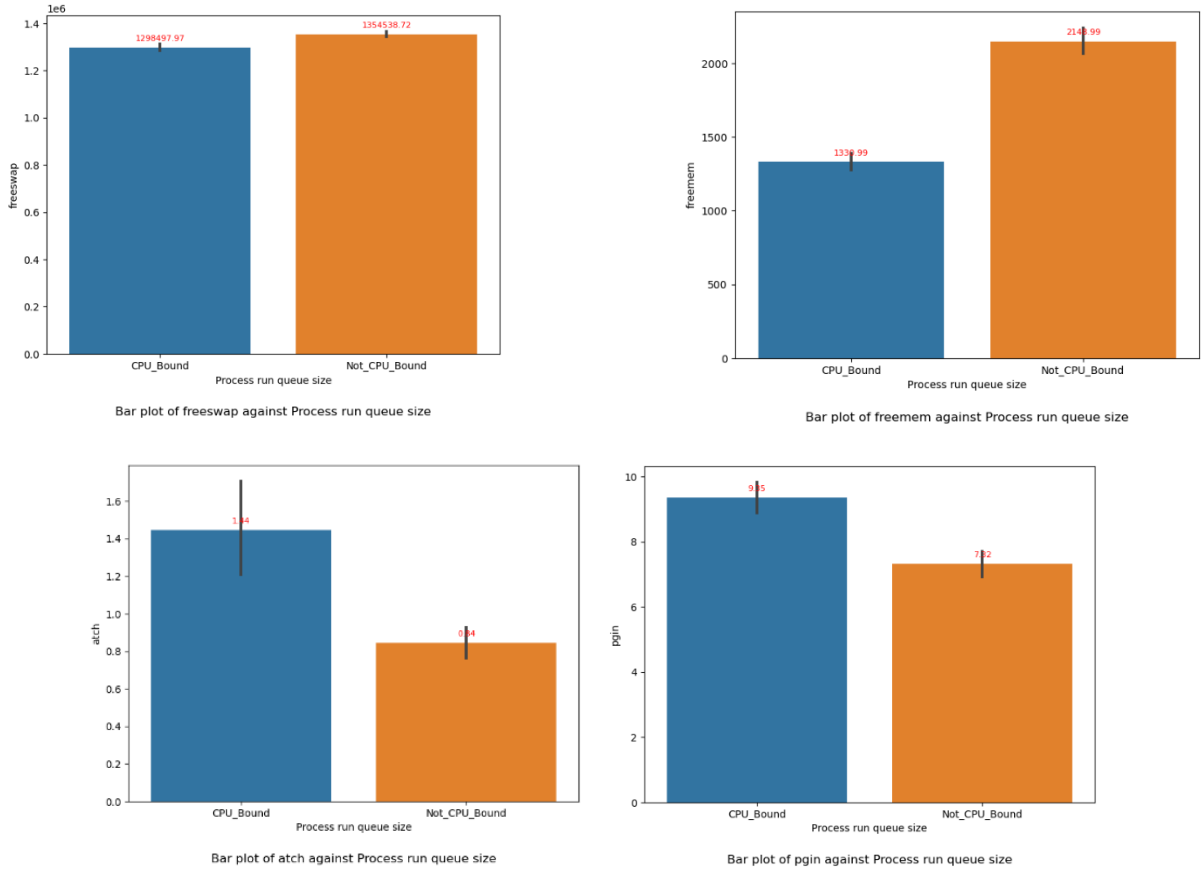


Fig 1.4: Barplot of runqsz (Run queue size) vs Numerical variables

1.1.7.3 Multivariate Analysis

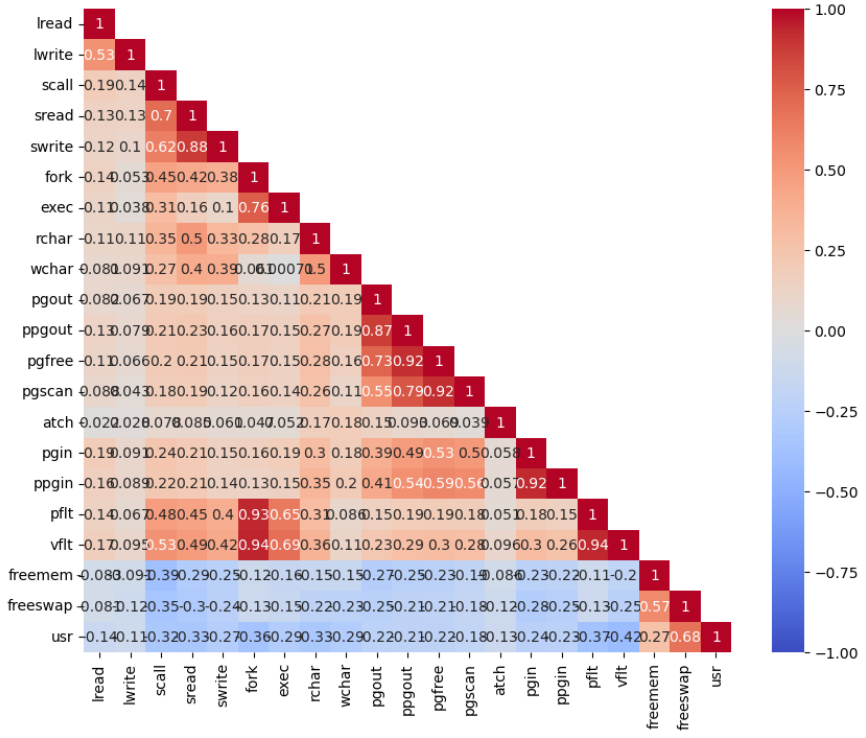


Fig 1.5: Correlation Plot of all the numerical variables

1.1.8 Key meaningful observations on individual variables and the relationship between variables:

The process run queue size vs lread, lwrite, system call scall, system read sread, system write swrite, fork, exec, rchar, wchar, pgout, ppgout, pgfree, atch, pgin, ppgin, pflt, vflt is more for CPU_Bound as compared to Not_CPU_Bound.

The process run queue size vs free memory freemem, freeswp and usr is more for Not_CPU_Bound as compared to CPU_Bound.

The correlation is maximum between:

- fork and vflt
- fork and pflt
- pflt and vflt
- pgin and ppgin
- pgfree and pgscan
- ppgout and pgfree

We can see some correlation between the various fields.

1.2 Data Preprocessing – Prepare the data for modelling

1.2.1 Missing Value Treatment

Check the dataset for any missing values using the isnull() function.

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

Table 1.5: Checking for missing values

We see that there are 104 missing values in the column ‘rchar’ and 15 missing values in ‘wchar’ column. We need to treat them using the median value in both the columns. After filling the null values with the median value, we see that there are no missing values in the dataset.

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

Table 1.6: Treating the missing values

1.2.2 Outlier Detection

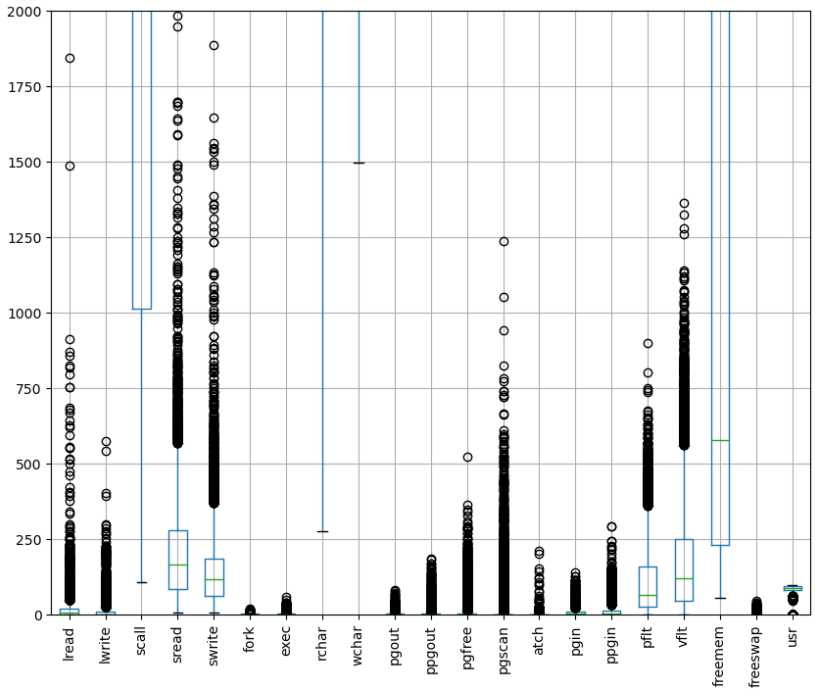


Fig 1.6: Outlier detection in the numerical fields of the dataset

Define a function which returns the Upper and Lower limit to detect outliers for each feature. Call the function with the column names. Cap & floor the values beyond the outlier boundaries.

Observations:

We see that all outliers have been removed and treated. We can proceed with the scaling of data.

After outlier treatment,

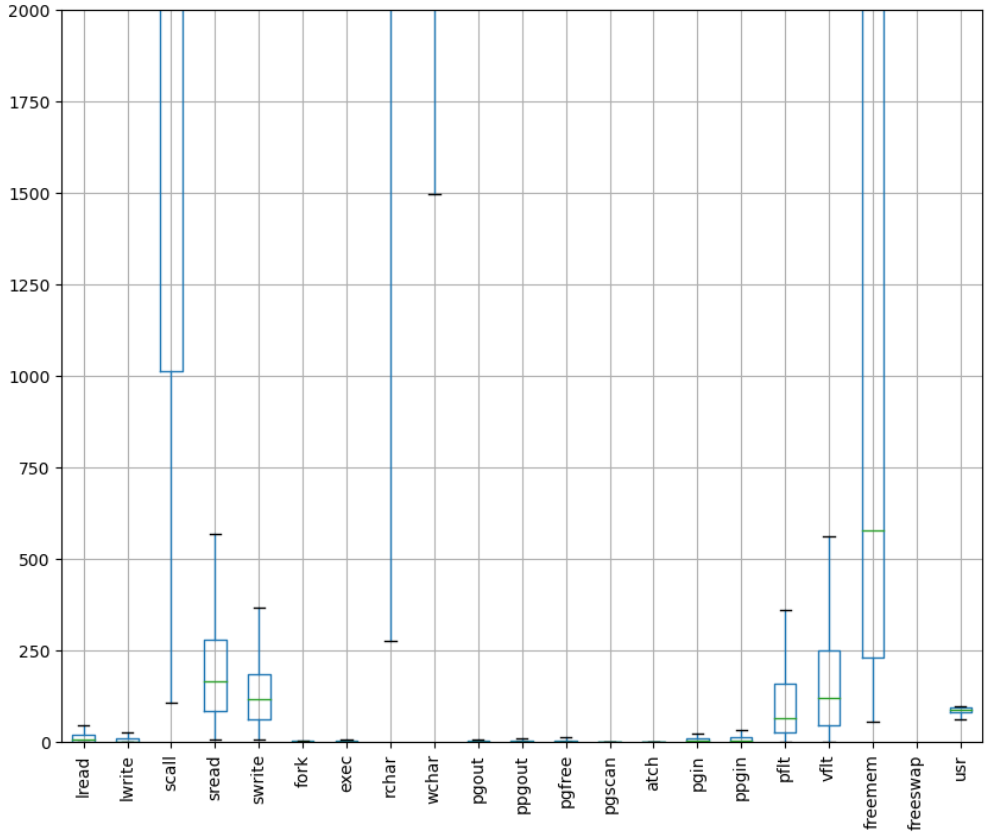


Fig 1.7: Outlier treatment in the numerical fields of the dataset

1.2.3 Feature Engineering

Feature engineering is a crucial step in the machine learning pipeline. It involves creating new features or modifying existing features which might enable the machine learning models to predict more accurately.

We can create new features that capture the interaction between different system attributes. For example, a feature can be created that represents the ratio of 'lread' to 'lwrite' or 'pgin' to 'pgout'. These new features might help in capturing the relationships between different system attributes more effectively.
For this dataset, no new feature has been created.

1.2.4 Encoding the data

One hot encoding is a technique that we use to represent categorical variables as numerical values in a machine learning model.

The get_dummy() function is used to convert the categorical columns into binary output. Here, runqsz is the only categorical column. On running the get_dummy function on 'runqsz' column, Python creates two columns, namely, runqsz_CPU_Bound and runqsz_Not_CPU_Bound. The column runqsz_CPU_Bound has been dropped so that there is no linear dependancy between the 2 columns.

scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	usr	runqsz_Not_CPU_Bound
2147.0	79.0	68.0	0.2	0.2	40671.0	53995.0	0.0	...	0.0	0.0	1.6	2.6	16.00	26.40	4659.125	1730946.0	95.0	0
170.0	18.0	21.0	0.2	0.2	448.0	8385.0	0.0	...	0.0	0.0	0.0	0.0	15.63	16.83	4659.125	1869002.0	97.0	1
2162.0	159.0	119.0	2.0	2.4	125473.5	31950.0	0.0	...	0.0	1.2	6.0	9.4	150.20	220.20	702.000	1021237.0	87.0	1
160.0	12.0	16.0	0.2	0.2	125473.5	8670.0	0.0	...	0.0	0.0	0.2	0.2	15.60	16.80	4659.125	1863704.0	98.0	1
330.0	39.0	38.0	0.4	0.4	125473.5	12185.0	0.0	...	0.0	0.0	1.0	1.2	37.80	47.60	633.000	1760253.0	90.0	1

Table 1.7 Encoding the data- creating dummy column

1.2.5 Train-Test split

- First, split the dataset into dependent variables and independent variable. X dataset contains all the variables except the dependent variable 'usr' which needs to be dropped from the dataset. Y dataset contains only the dependent variable 'usr' in it.
- Split X and Y into training and test dataset in 70:30 ratio using the train_test_split() function. This function needs to be imported from sklearn.model_selection library.

- X_train has 5734 rows and 21 columns.
  X_test 2458 rows and 21 columns.

1.3 Model Building – Linear Regression

1.3.1 Apply Linear Regression using sklearn

Step 1: Invoke the Linear Regression function and find the best fit model on the training data.

Step 2: Do the predictions using the training data as well as the test data.

Train Predictions:

```
array([91.5078012 , 91.77883105, 74.85526321, ..., 84.49418847,
       84.15725271, 92.95606428])
```

Test Predictions:

```
array([96.91549254, 90.34324284, 77.86534314, ..., 97.58162341,
       90.9160509 , 79.45653718])
```

1.3.2 Using Statsmodels Perform checks for significant variables using the appropriate method:

Step 1: Add the constant term to the X_train and X_test datasets.

Step 2: Fit the linear regression model using the Ordinary Least Squares method. Build the statsmodel using OLS() function and then fit it.

Step 3: All the summary statistics of the linear regression model are returned by the model.summary() method. The p-value and many other values/statistics are known by this method. Predictions about the data are found by the model.summary() method.

Dep. Variable:	usr	R-squared:	0.796
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1115.
Date:	Tue, 02 Jan 2024	Prob (F-statistic):	0.00
Time:	21:28:49	Log-Likelihood:	-16657.
No. Observations:	5734	AIC:	3.336e+04
Df Residuals:	5713	BIC:	3.350e+04
Df Model:	20		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.1217	0.316	266.106	0.000	83.502	84.741
lread	-0.0635	0.009	-7.071	0.000	-0.081	-0.046
lwrite	0.0482	0.013	3.671	0.000	0.022	0.074
scall	-0.0007	6.28e-05	-10.566	0.000	-0.001	-0.001
sread	0.0003	0.001	0.305	0.760	-0.002	0.002
swrite	-0.0054	0.001	-3.777	0.000	-0.008	-0.003
fork	0.0293	0.132	0.222	0.824	-0.229	0.288
exec	-0.3212	0.052	-6.220	0.000	-0.422	-0.220
rchar	-5.167e-06	4.88e-07	-10.598	0.000	-6.12e-06	-4.21e-06
wchar	-5.403e-06	1.03e-06	-5.232	0.000	-7.43e-06	-3.38e-06
pgout	-0.3688	0.090	-4.098	0.000	-0.545	-0.192
ppgout	-0.0766	0.079	-0.973	0.330	-0.231	0.078
pgfree	0.0845	0.048	1.769	0.077	-0.009	0.178
pgscan	4.002e-14	1.62e-16	247.538	0.000	3.97e-14	4.03e-14
atch	0.6276	0.143	4.394	0.000	0.348	0.908
pgin	0.0200	0.028	0.703	0.482	-0.036	0.076
ppgin	-0.0673	0.020	-3.415	0.001	-0.106	-0.029
pflt	-0.0336	0.002	-16.957	0.000	-0.037	-0.030
vflt	-0.0055	0.001	-3.830	0.000	-0.008	-0.003
freemem	-0.0005	5.07e-05	-9.038	0.000	-0.001	-0.000
freeswap	8.832e-06	1.9e-07	46.472	0.000	8.46e-06	9.2e-06

Table 1.8 StatsModel.summary()

Step 4: We observe that the R-squared value is 0.796 and Adjusted R-squared value is 0.795. These are good scores and the model is trained well.

Null Hypothesis: There is no correlation between the variables in the dataset.

Alternate Hypothesis: Atleast one column is correlated with the other variables in the dataset.

Observe the variables with high p_values, higher than 0.05.

Insights:

- If the p_value < significance level(0.05), there is enough evidence to reject the null hypothesis that there is no correlation in the dataset. Hence alternate hypothesis is true, that is, there exists correlation in the population. So, any change in the

independent variable also influences the dependent variable. So, these fields are statistically significant.

- If p_value > 0.05, then, there is enough evidence not to reject the null hypothesis. Hence, there is no correlation between these variables in the dataset. Hence, such variable can be removed from the dataset as they are not statistically significant.

- The columns whose p_value is more than 0.05 are: fork, sread, pgin, ppgout and pgfree.

- We drop these columns one by one and rebuild the model after each drop and check the p_value.

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.795
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1588.
Date:	Tue, 02 Jan 2024	Prob (F-statistic):	0.00
Time:	21:49:22	Log-Likelihood:	-16668.
No. Observations:	5734	AIC:	3.337e+04
Df Residuals:	5719	BIC:	3.346e+04
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.0589	0.311	270.482	0.000	83.450	84.668
lread	-0.0661	0.009	-7.386	0.000	-0.084	-0.049
lwrite	0.0501	0.013	3.826	0.000	0.024	0.076
scall	-0.0007	5.95e-05	-11.186	0.000	-0.001	-0.001
swrite	-0.0058	0.001	-5.502	0.000	-0.008	-0.004
exec	-0.3583	0.048	-7.401	0.000	-0.453	-0.263
rchar	-5.32e-06	4.35e-07	-12.226	0.000	-6.17e-06	-4.47e-06
wchar	-4.868e-06	1.02e-06	-4.780	0.000	-6.86e-06	-2.87e-06
pgout	-0.3429	0.038	-8.967	0.000	-0.418	-0.268
pgscan	1.884e-15	8.47e-16	2.224	0.026	2.23e-16	3.54e-15
atch	0.5982	0.142	4.200	0.000	0.319	0.877
ppgin	-0.0608	0.007	-9.337	0.000	-0.074	-0.048
pflt	-0.0398	0.001	-37.432	0.000	-0.042	-0.038
freemem	-0.0005	5.06e-05	-9.324	0.000	-0.001	-0.000
freeswap	8.935e-06	1.86e-07	48.063	0.000	8.57e-06	9.3e-06
runqsz_Not_CPU_Bound	1.6017	0.126	12.702	0.000	1.354	1.849

Omnibus:	1051.708	Durbin-Watson:	2.013
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2222.302
Skew:	-1.077	Prob(JB):	0.00

Table 1.9 StatsModel summary after dropping insignificant columns

Step 5: Check the Variance Inflation Factor (VIF) after dropping the insignificant columns.

	Features	VIF Score
0	const	28.175532
1	lread	5.302801
2	lwrite	4.294132
12	pflt	3.441335
4	swrite	3.012188
5	exec	2.842866
3	scall	2.652892
8	pgout	2.045289
13	freemem	1.945844
10	atch	1.860459
14	freeswap	1.757212
6	rchar	1.695078
7	wchar	1.536906
11	ppgin	1.518497
15	runqsz_Not_CPU_Bound	1.155447
9	pgscan	NaN

Table 1.10 VIF Score (1)

Step 6: We see that the feature ‘vflt’ has a very high VIF score of around 12.1. We drop this column and remodel the train dataset. The feature ‘pgscan’ shows a NaN value, hence this feature has to be dropped from the dataset and the model has to be rebuilt. Recheck the statsmodel summary and the VIF score after dropping both the columns.

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.795
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1588.
Date:	Tue, 02 Jan 2024	Prob (F-statistic):	0.00
Time:	21:50:21	Log-Likelihood:	-16668.
No. Observations:	5734	AIC:	3.337e+04
Df Residuals:	5719	BIC:	3.346e+04
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.0589	0.311	270.482	0.000	83.450	84.668
lread	-0.0661	0.009	-7.386	0.000	-0.084	-0.049
lwrite	0.0501	0.013	3.826	0.000	0.024	0.076
scall	-0.0007	5.95e-05	-11.186	0.000	-0.001	-0.001
swrite	-0.0058	0.001	-5.502	0.000	-0.008	-0.004
exec	-0.3583	0.048	-7.401	0.000	-0.453	-0.263
rchar	-5.32e-06	4.35e-07	-12.226	0.000	-6.17e-06	-4.47e-06
wchar	-4.868e-06	1.02e-06	-4.780	0.000	-6.86e-06	-2.87e-06
pgout	-0.3429	0.038	-8.967	0.000	-0.418	-0.268
atch	0.5982	0.142	4.200	0.000	0.319	0.877
ppgin	-0.0608	0.007	-9.337	0.000	-0.074	-0.048
pflt	-0.0398	0.001	-37.432	0.000	-0.042	-0.038
freemem	-0.0005	5.06e-05	-9.324	0.000	-0.001	-0.000
freeswap	8.935e-06	1.86e-07	48.063	0.000	8.57e-06	9.3e-06
runqsz_Not_CPU_Bound	1.6017	0.126	12.702	0.000	1.354	1.849

Omnibus:	1051.708	Durbin-Watson:	2.013
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2222.302
Skew:	-1.077	Prob(JB):	0.00
Kurtosis:	5.159	Cond. No.	7.59e+06

Table 1.11 StatsModel summary after dropping all the insignificant features

	Features	VIF Score
0	const	28.175532
1	lread	5.302801
2	lwrite	4.294132
11	pflt	3.441335
4	swrite	3.012188
5	exec	2.842866
3	scall	2.652892
8	pgout	2.045289
12	freemem	1.945844
9	atch	1.860459
13	freeswap	1.757212
6	rchar	1.695078
7	wchar	1.536906
10	ppgin	1.518497
14	runqsz_Not_CPU_Bound	1.155447

Table 1.12 Final VIF Scores

Insights:

After dropping the features causing strong multicollinearity and the statistically insignificant ones, our model performance hasn't dropped sharply . This shows that these variables did not have much predictive power.

Step 7. The OLS regression results is as follows:

OLS Regression Results

Dep. Variable:	usr	**R-squared:**	0.795
Model:	OLS	**Adj. R-squared:**	0.795
Method:	Least Squares	**F-statistic:**	1588.
Date:	Tue, 02 Jan 2024	**Prob (F-statistic):**	0.00
Time:	21:55:37	**Log-Likelihood:**	-16668.
No. Observations:	5734	**AIC:**	3.337e+04
Df Residuals:	5719	**BIC:**	3.346e+04
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.0589	0.311	270.482	0.000	83.450	84.668
lread	-0.0661	0.009	-7.386	0.000	-0.084	-0.049
lwrite	0.0501	0.013	3.826	0.000	0.024	0.076
scall	-0.0007	5.95e-05	-11.186	0.000	-0.001	-0.001
swrite	-0.0058	0.001	-5.502	0.000	-0.008	-0.004
exec	-0.3583	0.048	-7.401	0.000	-0.453	-0.263
rchar	-5.32e-06	4.35e-07	-12.226	0.000	-6.17e-06	-4.47e-06
wchar	-4.868e-06	1.02e-06	-4.780	0.000	-6.86e-06	-2.87e-06
pgout	-0.3429	0.038	-8.967	0.000	-0.418	-0.268
atch	0.5982	0.142	4.200	0.000	0.319	0.877
ppgin	-0.0608	0.007	-9.337	0.000	-0.074	-0.048
pflt	-0.0398	0.001	-37.432	0.000	-0.042	-0.038
freemem	-0.0005	5.06e-05	-9.324	0.000	-0.001	-0.000
freeswap	8.935e-06	1.86e-07	48.063	0.000	8.57e-06	9.3e-06
runqsz_Not_CPU_Bound	1.6017	0.126	12.702	0.000	1.354	1.849

Omnibus:	1051.708	**Durbin-Watson:**	2.013
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2222.302
Skew:	-1.077	**Prob(JB):**	0.00
Kurtosis:	5.159	**Cond. No.**	7.59e+06

Table 1.13 OLS regression summary

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

Make predictions on the test set.

y_pred_train

```
694     90.990787
5535    91.779952
4244    74.579718
2472    80.857658
7052    98.255460
          ...
7935    81.186779
5192    94.623197
3980    84.611937
235     84.587346
5157    92.963889
Length: 5734, dtype: float64
```

y_pred_test

```
3894    96.880734
4276    90.324494
3414    78.201606
4165    78.183283
7385    78.153402
          ...
4744    98.364441
6918    81.153065
1556    97.635037
1577    90.883277
453     79.722028
Length: 2458, dtype: float64
```

Root Mean Square Error (RMSE):

```
The Root Mean Square Error (RMSE) of the model for the training set is
4.427

The Root Mean Square Error (RMSE) of the model for the test set is 4.66
9
```

Mean Squared Error (MSE)

```
The mean squared error is 4.43.
```

R-squared Value – Coefficient of determination

```
R-squared value is  79.53
```

1.3.4 Testing for assumptions of Linear Regression

a. Test for Linearity

If the scatter plot or pairplot of residuals do not follow any pattern, then the model is said to be linear.

	Actual Values	Predicted Values	Residuals
0	91.0	90.990787	0.009213
1	94.0	91.779952	2.220048
2	61.5	74.579718	-13.079718
3	83.0	80.857658	2.142342
4	94.0	98.255460	-4.255460

Table 1.14 Actual Values, Predicted Values, Residuals

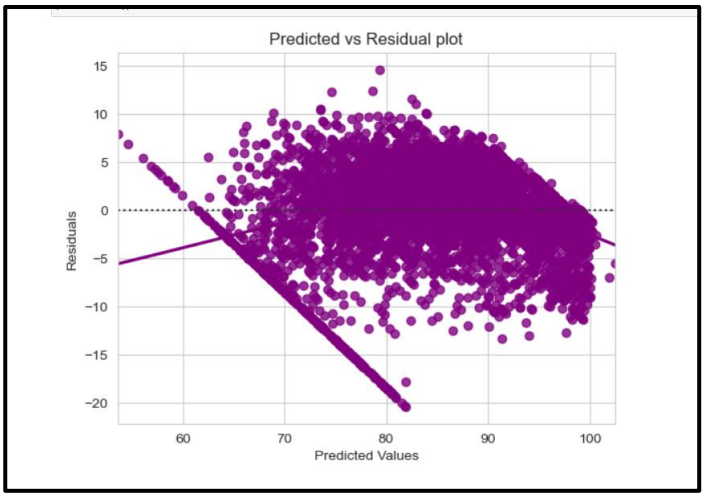


Fig 1.8: Plot for Test of Linearity

No pattern is seen in the plot of residuals vs predicted values. Hence, the assumption of linearity and independence of predictors is satisfied.

b. Test for Homoscedasticity

Goldfeld Quandt test will be performed to test for homoscedasticity.

Null Hypothesis: H0- Homoscedasticity - Variance does not change.

Alternate Hypothesis: H1 - Heteroscedasticity - Variance changes.

```
The f-value and p-value are found to be:

(1.1185205131843772, 0.0013959368769108644, 'increasing')
```

Insights:

Since p-value < 0.05 we can say that the residuals are heteroscedastic. This test fails but we go ahead with the modeling.

c. Durbin-Watson Test

The Durbin-Watson test is a statistical test that measures autocorrelation in the residuals of a regression analysis. The Durbin-Watson statistic close to 2 shows no autocorrelation.

Output of Durbin Watson test:

```
The Durbin-Watson statistic is 2.013264968917853
```

The value is approximately equal to 2 states there is no auto correlation between errors. Or residuals.

d. Test for Normality

The skewness is `-1.077224138843297`

On performing the Shapiro test for normality, the p_value is found to be `1.26256991635666 02e-42.`

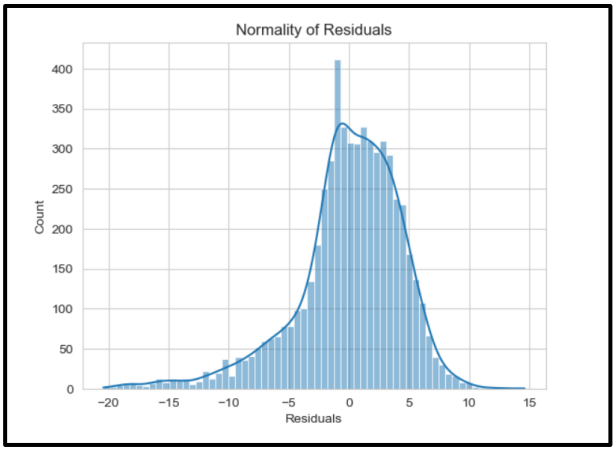


Fig 1.9: Plot for Test for Normality

Since p-value < 0.05, the residuals are not normal as per shapiro test. But visually the data looks normal so we ignore the result of shapiro test.

The QQ curve plot: A Q-Q plot, or quantile-quantile plot, is a scatter plot that shows the relationship between the ordered values of a sample and the corresponding percentiles of a normal distribution. If the data is normally distributed, the points in a Q-Q plot should fall along a straight line with a slope of 1 and an intercept of 0. If the points deviate from the diagonal line, the variable is not normally distributed.

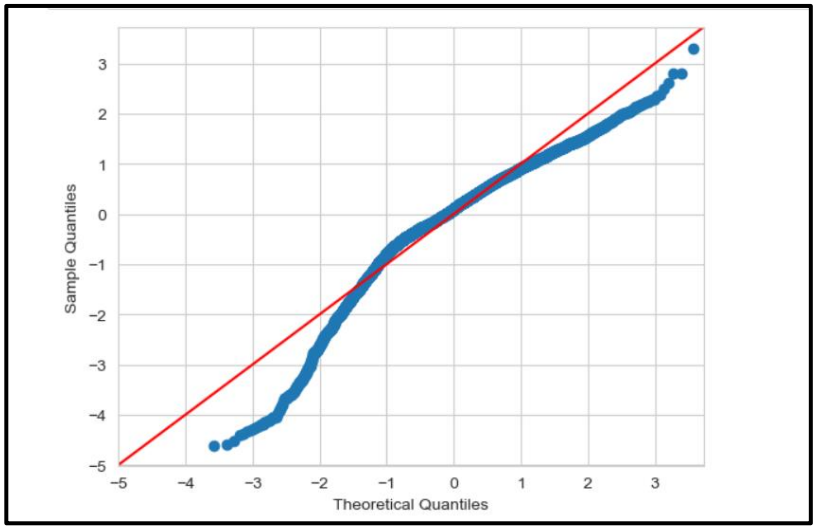


Fig 1.10: QQ Plot for Quantile-Quantile Plot

A pairplot of actual values vs predicted values is shown:

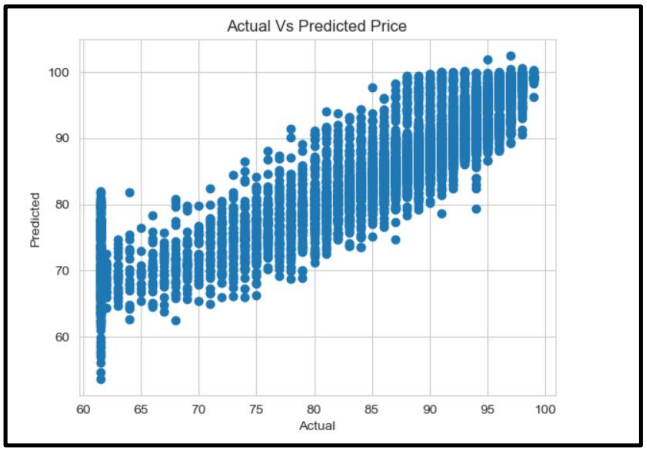


Fig 1.11: Pairplot of Actual price vs predicted price

1.4 Business Insights and Recommendations

1.4.1 Comment on the Linear Regression equation from the final model and impact of relevant variables (atleast 2) as per the equation.

const	84.058858
lread	-0.066101
lwrite	0.050061
scall	-0.000666
swrite	-0.005801
exec	-0.358337
rchar	-0.000005
wchar	-0.000005
pgout	-0.342872
atch	0.598161
ppgin	-0.060811
pflt	-0.039771
freemem	-0.000472
freeswap	0.000009
runqsz_Not_CPU_Bound	1.601689

dtype: float64

Table 1.15: Coefficients and constants of the significant features

Linear Regression Equation:

```
(84.058858) * const + (-0.066101) * lread + (0.050061) * lwrite + (-0.0
00666) * scall + (-0.005801) * swrite + (-0.358337) * exec + (-5e-06) *
rchar + (-5e-06) * wchar + (-0.342872) * pgout + (0.598161) * atch + (-
0.060811) * ppgin + (-0.039771) * pflt + (-0.000472) * freemem + (9e-06
) * freeswap + (1.601689) * runqsz_Not_CPU_Bound +
```

Insights:

- 1 unit increase in the 'lread' leads to a 0.066 times decrease in the usr keeping all other predictors constant.
- 1 unit increase in the 'lwrite' leads to a 0.00066 times decrease in the usr keeping all other predictors constant

- 1 unit increase in the number of system calls 'scall' leads to a 0.0058 times decrease in the usr keeping all other predictors constant.
- 1 unit increase in the 'atch' leads to a 0.59 times increase in the usr keeping all other predictors constant.
- If 'runqsz_Not_CPU_Bound' is present, it increases the usr by a factor of 1.6 keeping all other predictors constant.
- **The 2 most important features are 'runqsz_Not_CPU_Bound' and 'atch', followed by 'exec' and 'pgout'.**

1.4.2 Conclude with the key takeaways (actionable insights and recommendations) for the business.

- The R-squared value and the adjusted R-squared value are almost the same, that is, 79.5%.
- This tells us that the linear regression model has an accuracy of 79.5%.
- R-squared is a summary statistic that quantifies the proportion of variation in the outcome variable explained by the explanatory variable.
- Also, the other measures which are RMSE(Root Mean Squared Error) and MSE(Mean Squared Error) are very low. Lower these values, better the model is.
- Two most significant features are 'runqsz' and 'atch', which have a positive correlation with the dependent variable 'usr'.
- One unit increase in 'runqsz' leads to an increase of 1.6 times the 'usr' variable. One unit increase in 'atch' variable leads to an increase in 0.59 times of the 'usr' variable.
- Two other variables which are negatively correlated to 'usr' variable are 'exec' and 'pgout'.
- One unit increase in 'exec' leads to a decrease of 0.35 times of the 'usr' variable. One unit increase in 'pgout' leads to a decrease of 0.34 times of the 'usr' variable.
- **Thus, the most significant system attributes which influence the 'usr' variable are the process run queue size (runqsz), number of page attaches (atch), number of system execute calls per second(exec) and number of page out requests (pgout) per second.**
- For the business to do better, the user should focus on the four significant variables and try to tune the performance of the dataset by increasing or decreasing the four variables.

Problem 2:

In your role as a statistician at the Republic of Indonesia Ministry of Health, you have been entrusted with a dataset containing information from a Contraceptive Prevalence Survey. This dataset encompasses data from 1473 married females who were either not pregnant or were uncertain of their pregnancy status during the survey.

Your task involves predicting whether these women opt for a contraceptive method of choice. This prediction will be based on a comprehensive analysis of their demographic and socio-economic attributes.

Data Description

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 Performing Exploratory Data Analysis (EDA)

2.1.1 Importing Libraries

Import the necessary libraries for data processing, data visualization, modelling, validation, data splitting, building the linear regression model and to check the model performance.

2.1.2 Reading and loading the dataset compactiv.xls:

Load and read the dataset using the read_excel() function of pandas. The first five lines of the dataset is as follows:

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

Table 2.1: Reading the first 5 rows of the Contraceptive_method_dataset

2.1.3 Checking the shape of the dataset:

Find out the number of rows and columns in the dataset using the shape command.

The dataset has 1473 rows and 10 columns.

2.1.4 Checking the datatypes of all the columns:

Use the info() command to find the datatypes of all the columns in the dataset. It also tells you if the dataset has missing values.

```
<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
```

Table 2.2: Checking the datatype of all the columns

Insights:

- Out of 10 columns, 3 are numeric type and 7 are object type columns.
- The categorical columns are not in encoded format.
- There are missing values in 2 columns, namely, Wife_age and No_of_children_born.

2.1.5 Checking the statistical summary:

Use the describe() function to get the statistical summary of the object type columns.

	count	mean	std	min	25%	50%	75%	max
Wife_age	1402.0	32.606277	8.274927	16.0	26.0	32.0	39.0	49.0
No_of_children_born	1452.0	3.254132	2.365212	0.0	1.0	3.0	4.0	16.0
Husband_Occupation	1473.0	2.137814	0.864857	1.0	1.0	2.0	3.0	4.0

Table 2.3: Statistical summary of numerical type columns

	count	unique	top	freq
Wife_ education	1473	4	Tertiary	577
Husband_education	1473	4	Tertiary	899
Wife_religion	1473	2	Scientology	1253
Wife_Working	1473	2	No	1104
Standard_of_living_index	1473	4	Very High	684
Media_exposure	1473	2	Exposed	1364
Contraceptive_method_used	1473	2	Yes	844

Table 2.4: Statistical summary of object type columns

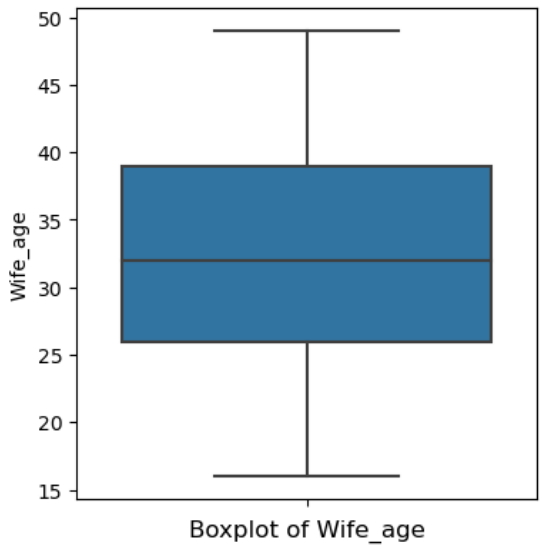
Insights:

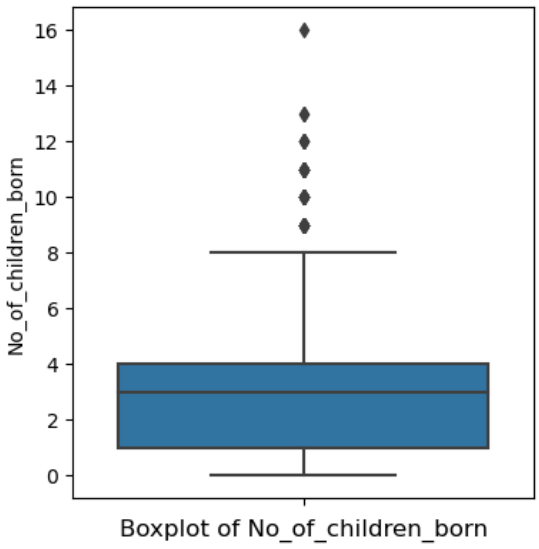
- The columns Wife_age and No_of_children_born have missing values as the count is less than the total number of rows in the dataset.
- The minimum age of a wife opting or not opting for contraceptive is 16 years and the maximum age is 49 years with the mean age being 32 years.
- The maximum number of children born are 16. The average number of children is 3. The median number of children born is also 3.
- Educated wives count upto a number of 577 while educated husbands count up to 899.
- Most of the wives practice Scientology religion.
- Most of the wives do not work. Out of a total of 1473, 1253 wives do not work.
- A very high standard of living is enjoyed by 684 people in Indonesia.
- 844 people have used contraceptives while the remaining have not used.
- Most of the women in Indonesia are media exposed.

2.1.6 Univariate Analysis

2.1.6.1 Numerical Columns

Univariate analysis of all the fields is done using a histogram and a boxplot. First, we divide the dataset into a dataset with numerical fields and another dataset with categorical fields.





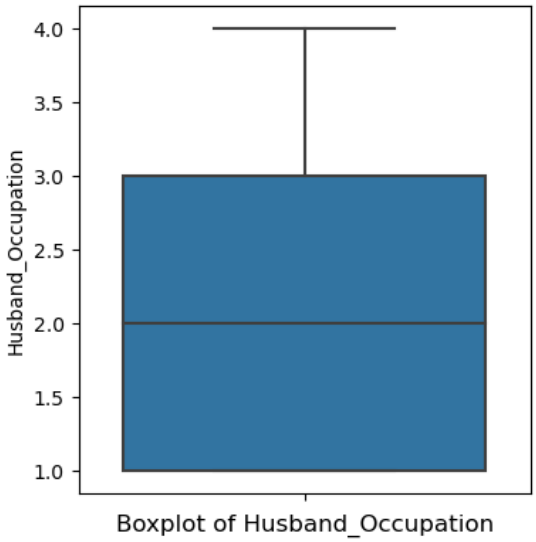
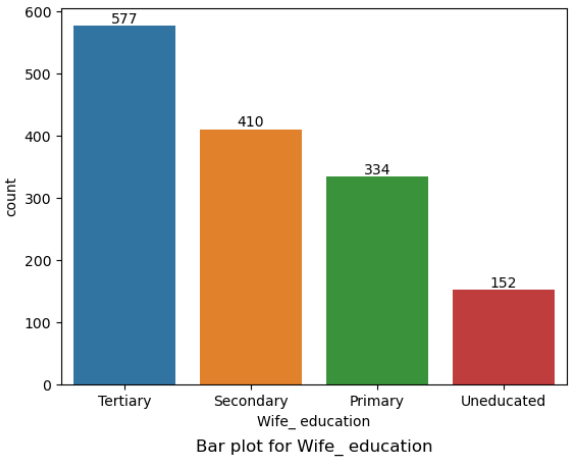
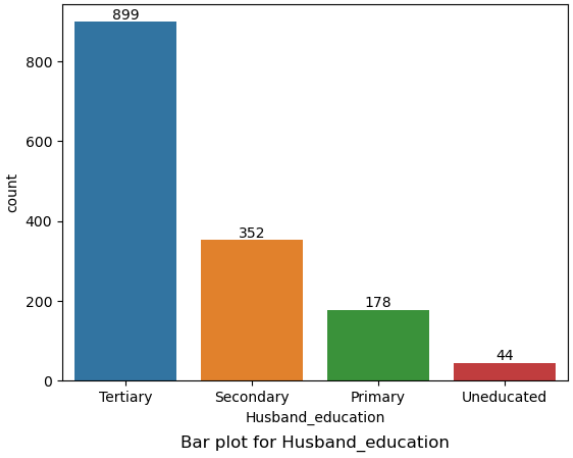
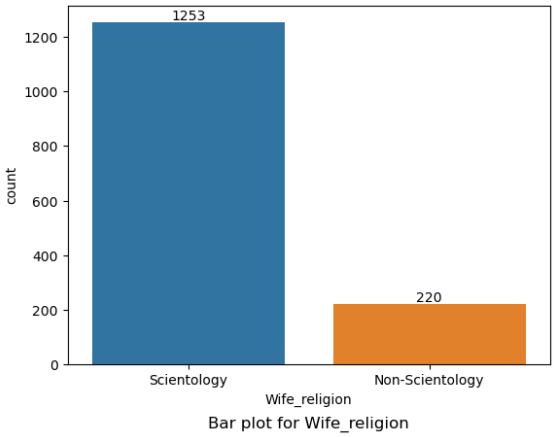


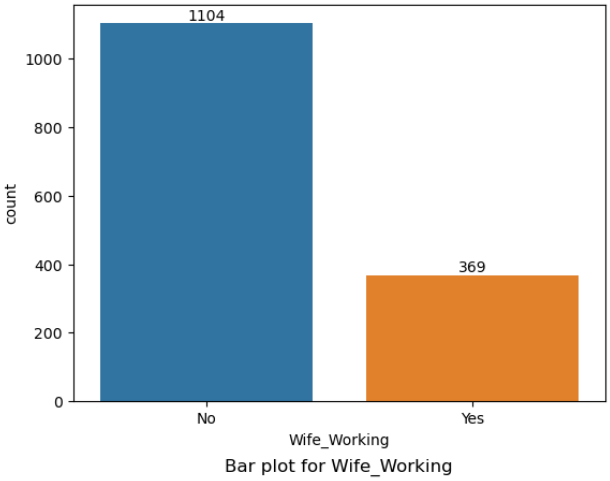
Fig 2.1 Univariate analysis of Numerical columns

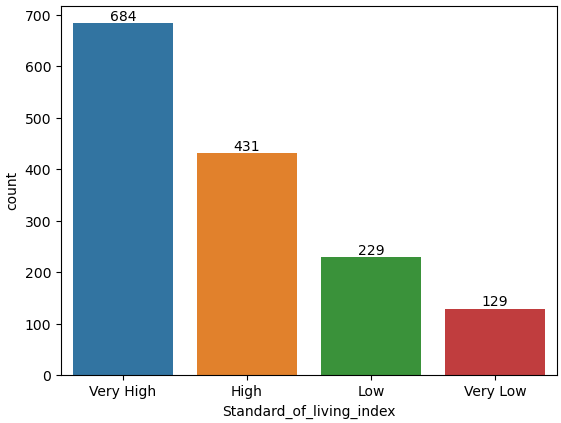
2.1.6.2 Categorical columns



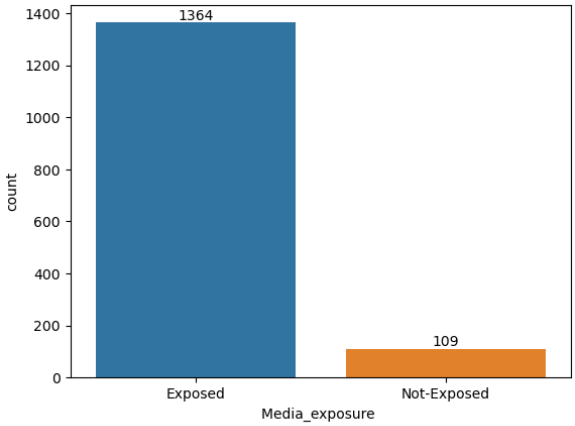




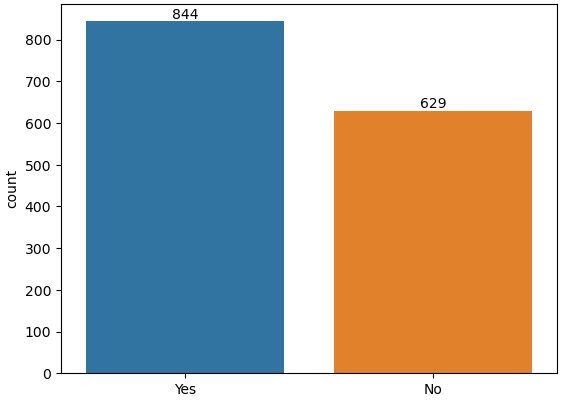




Bar plot for Standard_of_living_index



Bar plot for Media_exposure



Bar plot for Contraceptive_method_used

Fig 2.2 Univariate analysis of categorical columns

2.1.7 Multivariate Analysis - Visualizations to identify the pattern and insights

2.1.7.1 Numeric vs Numeric

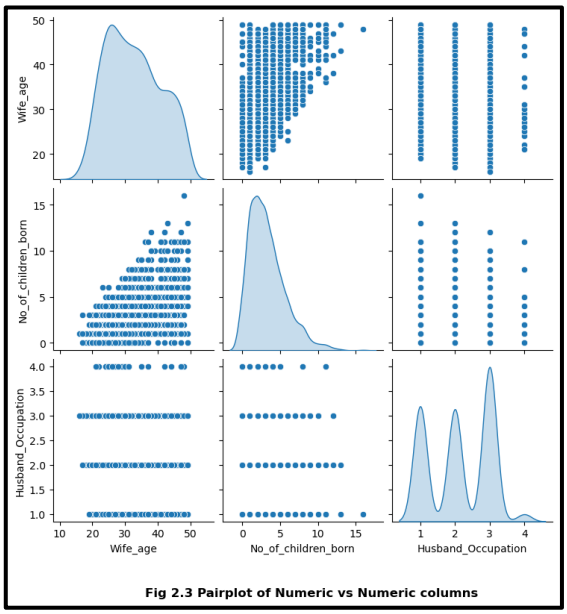


Fig 2.3 Pairplot of Numeric vs Numeric columns

2.1.7.2 Numerical vs Categorical

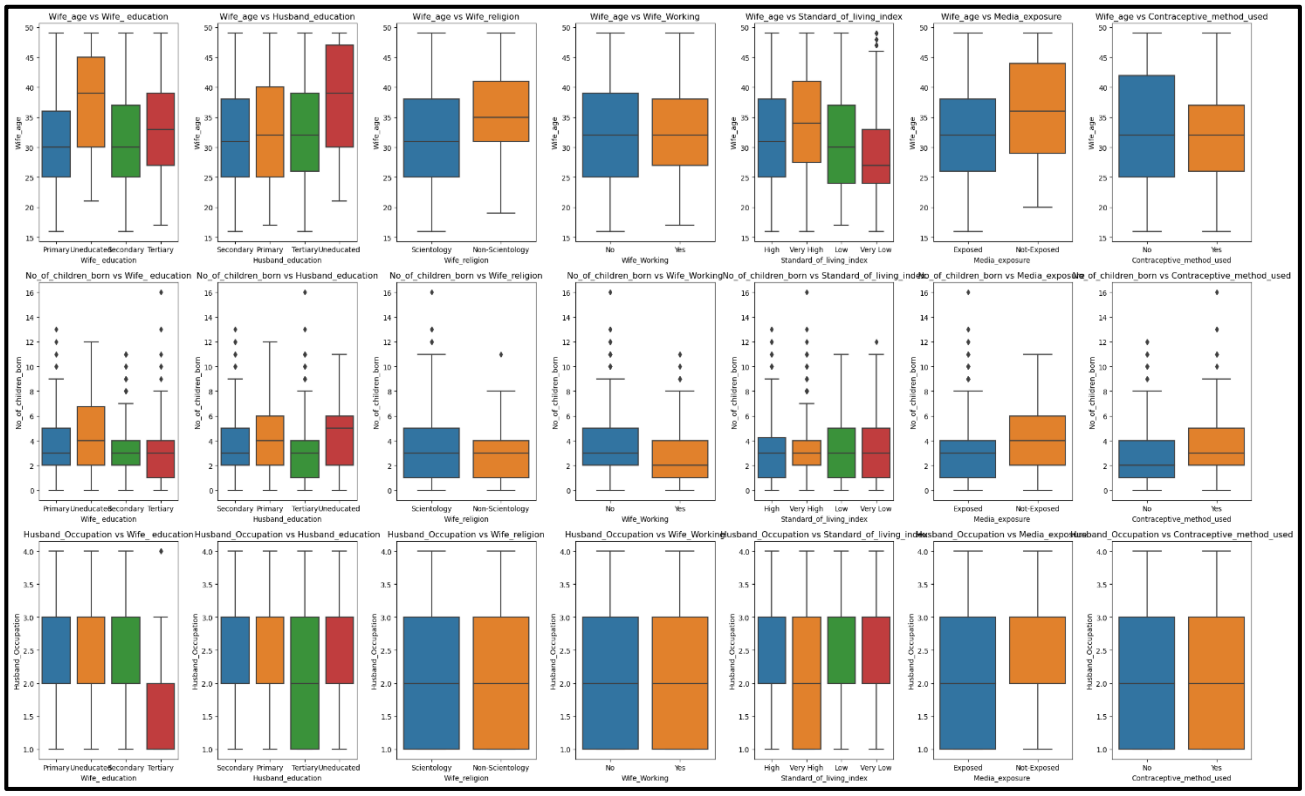


Fig 2.4 Boxplots of Numerical vs Categorical columns

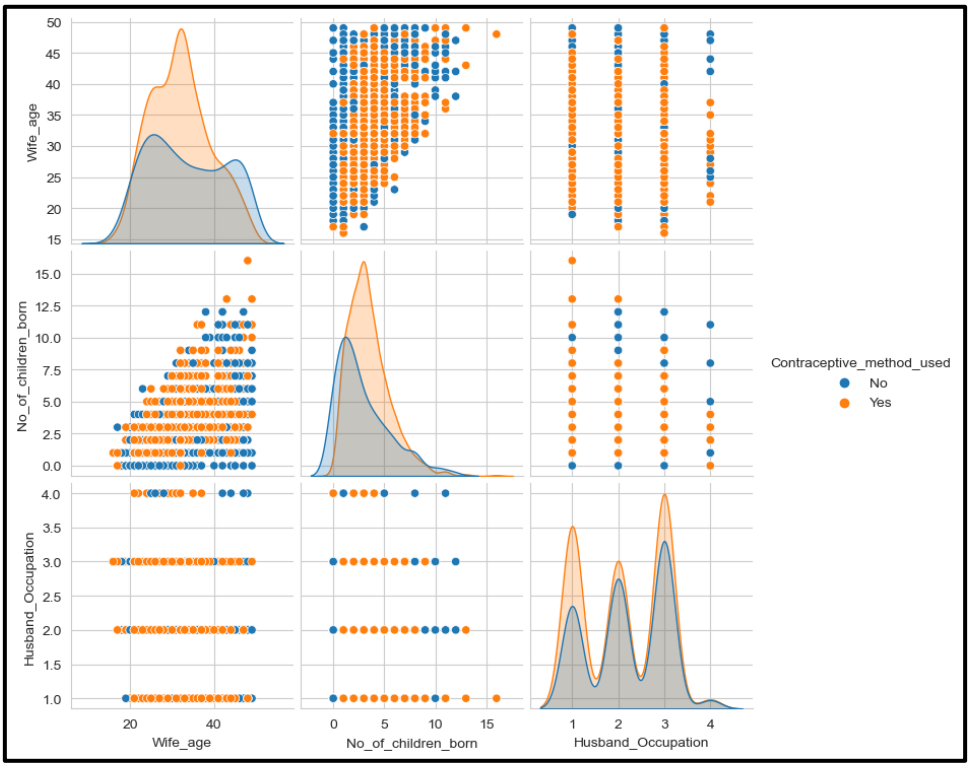


Fig 2.5: Pairplot - Contraceptive_method_used vs Numerical Fields

2.1.7.3 Categorical vs Categorical

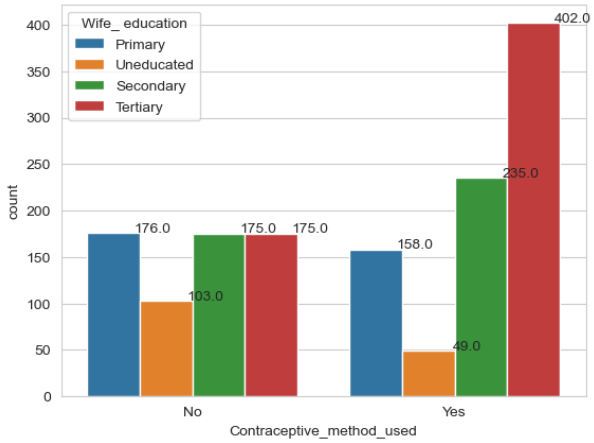


Fig 2.6 a) Contraceptive Method vs Wife_education

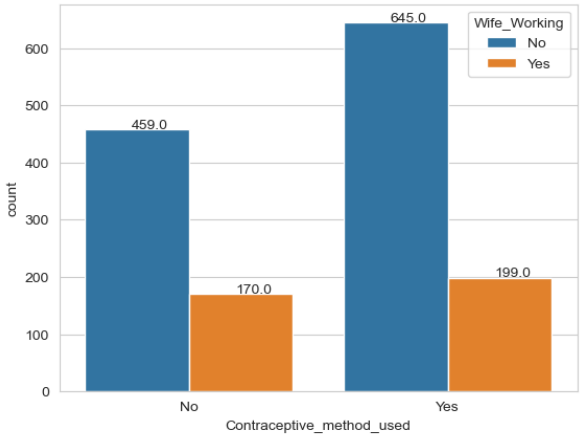


Fig 2.6 b) Contraceptive Method vs Wife_Working

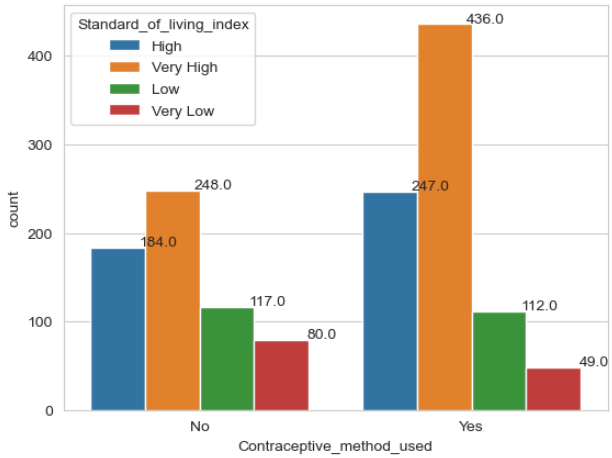


Fig 2.6 c) Contraceptive Method vs Standard_of_living_index

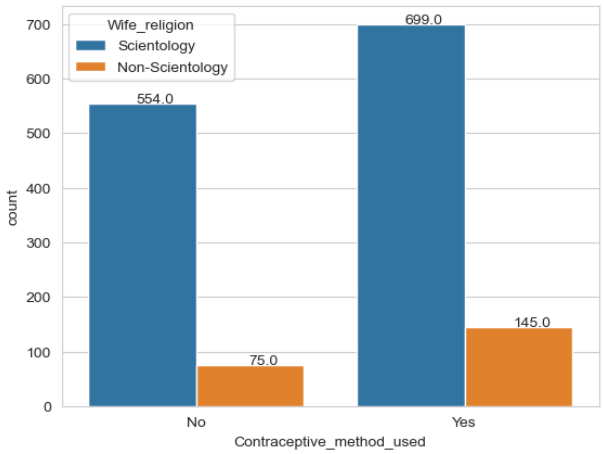


Fig 2.6 d) Contraceptive Method vs Wife_religion

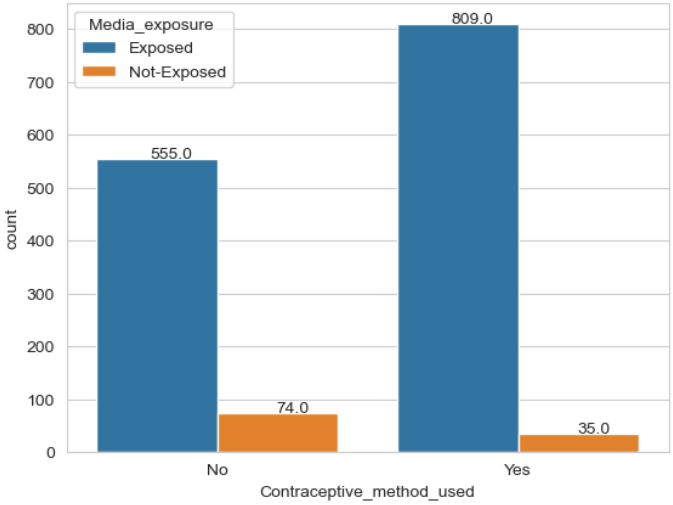


Fig 2.6 e) Contraceptive Method vs Media_exposure

Figure 2.6 Contraceptive Method vs Categorical columns

2.1.7.4 Correlation Plot

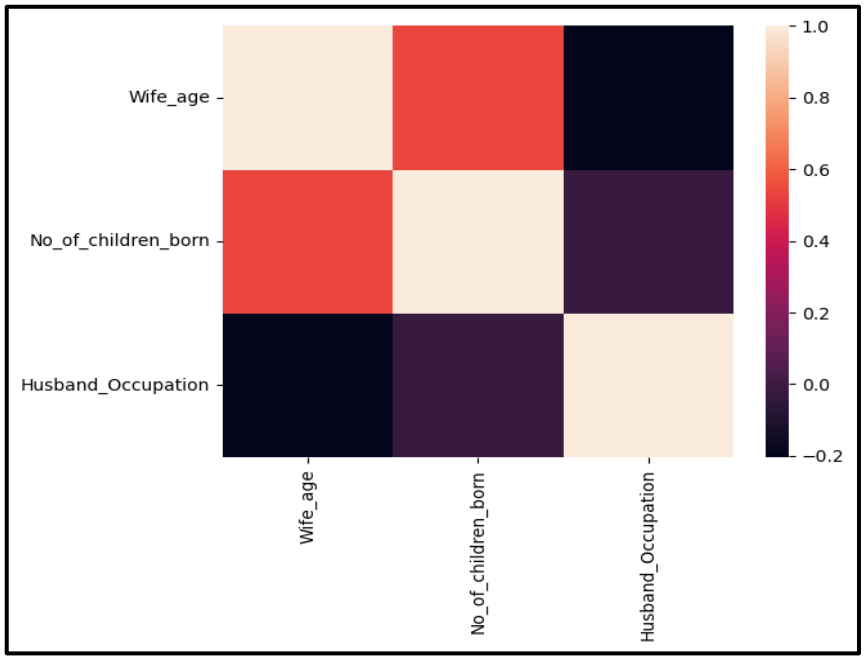


Figure 2.7 Heatmap of Numerical Variables

2.1.7.5 Key meaningful observations on individual variables and the relationship between variables:

1. Number of children born has outliers.

2. The median age of women using contraceptives is 32 years.

3. The count of highly educated women is 577 while that of uneducated women is 152. The number of women uneducated is far less as compared to women having at least basic education.

4. Very few, almost negligible number of husbands are uneducated. The count is just 44. Most of the husbands are highly educated with a count of almost 899. Most of them have primary or secondary education.

5. 1253 women practice Scientology religion while only 220 are non-scientologists.

6. Working wives outnumber the non-working wives.

7. 129 people have very low standard of living while 684 enjoy very high standard of living.

8. 1364 people are exposed to media as compared to 109 who have no media exposure.

9. 844 women have used contraceptives while 629 have not used any contraceptives.

10. Uneducated women fall in the median age of 40 years.

11. The median age of working women using contraceptives is slightly more than the median age of women not using contraceptives.

12. The median age of uneducated women is more for the number of children born than that of educated women.

13. Highly educated women have the maximum usage of contraceptives.

14. 436 people having very high standard of living use contraceptives. 248 people who have high standard of living but do not use conraceptives.

15. 809 people who are exposed to media use contraceptives.

2.2 Data Preprocessing – Prepare the data for modelling

2.2.1 Missing Value Treatment

Check the dataset for any missing values using the isnull() function.

```
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
```

Table 2.5 Missing Values

We see that the columns 'Wife_age' and 'No_of_children_born' have missing values in them. One column 'Wife_age' has 71 missing values and 'No_of_children_born' has 21 missing values. We treat the missing values with the median values of the median of the respective columns.

```
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
```

Table 2.6: Treating the missing values

2.2.2. Outlier Detection

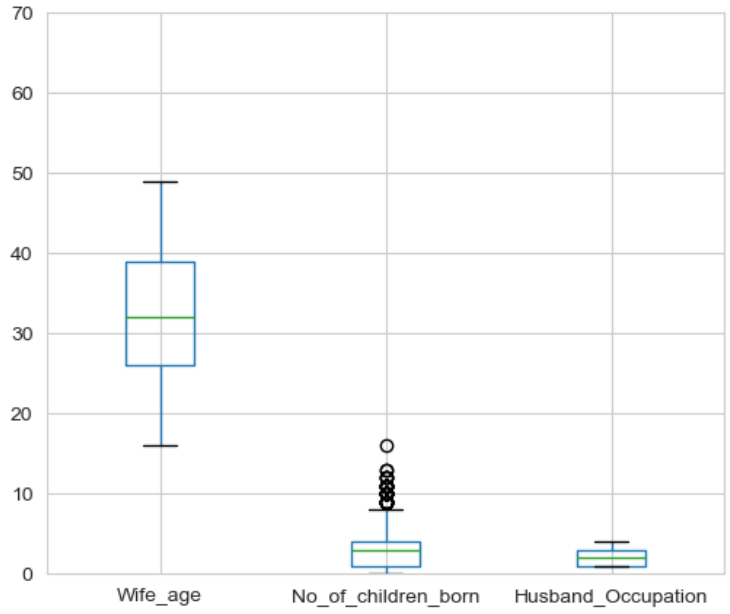


Fig 2.8: Outlier detection in the numerical fields of the dataset

Define a function which returns the Upper and Lower limit to detect outliers for each feature. Call the function with the column names. Cap & floor the values beyond the outlier boundaries. After outlier treatment, we see that the outliers disappear.

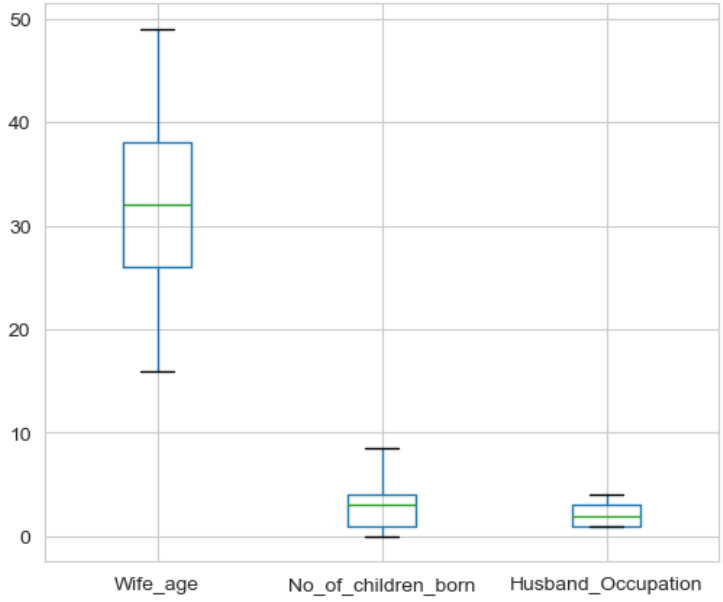


Fig 2.9: After Outlier Removal

Observations:

We see that all outliers have been removed and treated. We can proceed with the scaling of data.

2.2.3 Feature Engineering

Feature engineering is a crucial step in the machine learning pipeline. It involves creating new features or modifying existing features which might enable the machine learning models to predict more accurately.

We can create new features that capture the interaction between different system attributes. For example, a feature can be created that represents the ratio of 'Wife age' to 'No of children born'. These new features might help in capturing the relationships between different system attributes more effectively.

For this dataset, no new feature has been created.

2.2.4 Encoding the data

One hot encoding as well as ordinal encoding are used to convert the categorical columns into numerical values for the machine learning modelling.

One hot encoding is used for the columns for 'Wife_religion', 'Wife_Working', 'Contraceptive_method_used' and 'Media_exposure'. The get_dummy() function is used to convert the categorical columns into binary output.

Ordinal encoding is used for columns 'Wife_education', 'Husband_education' and 'Standard_of_living_index' to convert them into numerical values according to the rankings. For example: Very low standard of living gets a value of 1, low gets a value of 2, high standard of living gets a value of 3 and very high standard of living gets a value of 4.

	Wife_age	Wife_ education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure
0	24.0	2	3	3.0	1	0	2.0	3	0
1	45.0	1	3	8.5	1	0	3.0	4	0
2	43.0	2	3	7.0	1	0	3.0	4	0
3	42.0	3	2	8.5	1	0	3.0	3	0
4	36.0	3	3	8.0	1	0	3.0	2	0

Table 2.7 Encoding the categorical column values into numerical values

As all the columns are now having numerical values, our data is ready for machine learning modelling.

2.2.5 Train-Test split

- First, split the dataset into dependent variables and independent variable. X dataset contains all the variables except the dependent variable 'Contraceptive_method_used' which needs to be dropped from the dataset. Y dataset contains only the dependent variable 'Contraceptive_method_used' in it.

- Split X and Y into training and test dataset in 70:30 ratio using the train_test_split() function. This function needs to be imported from sklearn.model_selection library.

- X_train has 1031 rows and 9 columns.
  X_test 442 rows and 9 columns.

2.3 Building Different Models

2.3.1 Building the Logistic Model

Step 1: Build the logistic regression model using the logistic regression function (LogisticRegression()) and fit the model on the training dataset.

Step 2: Predict the model for the training as well as the testing dataset.

Step 3: Check the accuracy of the model.

Step 4: Generate the confusion matrix and the classification report.

Insights:

- We have a 68.7% accuracy on the testing dataset of the Logistic Regression model and 67.3% accuracy on the training dataset.
- Confusion Matrix:

```
[[ 96  97]
 [ 41 208]]
```

	precision	recall	f1-score	support
0	0.70	0.50	0.58	193
1	0.68	0.84	0.75	249
accuracy			0.69	442
macro avg	0.69	0.67	0.67	442
weighted avg	0.69	0.69	0.68	442

Table 2.8 Classification Report – Logistic Regression Model

- Intercept of the Logistic Regression model is -0.35620442
- On fitting the coefficients to the features into a dataframe, we get:

	0
Wife_age	-0.077199
Wife_ education	0.567930
Husband_education	-0.022191
No_of_children_born	0.362477
Wife_religion	-0.484623
Wife_Working	-0.014965
Husband_Occupation	0.079342
Standard_of_living_index	0.229071
Media_exposure	-0.478825

Table 2.9 Coefficients with the features – Logistic Regression

- We observe that Wife_education and No_of_children_born are the two features which positively affect the dependent variable, 'Contraceptive_method_used'.
- 1 unit increase in Wife_education increases the contraceptive_method_used by 0.567 units. A 1 unit increase in No_of_children_born increases the contraceptive_method_used by 0.36 units.
- Also, Wife_religion and Media_exposure negatively impact the dependent variable.
- 1 unit increase in Wife_religion and Media_exposure decreases the contraceptive_method_used by almost 0.48 units.
- Therefore, the most significant features are Wife_education, Wife-religion, Media_exposure and No_of_children_born.

2.3.2 Building the LDA Model

Step 1: Import the necessary libraries like LinearDiscriminantAnalysis from sklearn.metrics.

Step 2: Build the LDA model by applying LinearDiscriminantAnalysis() function.

Step 3: Fit the model on the training dataset.

Step 4: Predict the model based on the testing dataset.

Step 5: Print the classification report.

	precision	recall	f1-score	support
0	0.71	0.47	0.57	193
1	0.67	0.85	0.75	249
accuracy			0.68	442
macro avg	0.69	0.66	0.66	442
weighted avg	0.69	0.68	0.67	442

Table 2.10 Classification Report – LDA Model

Accuracy of Training and Testing dataset:

```
The accuracy of the training set is 0.6547041707080504
The accuracy of the testing set is 0.665158371040724
```

Insights:

- We have a 68% accuracy in the LDA model.
- The coefficient for the LDA model is -0.34837862
- The weights of each of the features in the LDA model is as follows:

	0
Wife_age	-0.075698
Wife_ education	0.573923
Husband_education	-0.026745
No_of_children_born	0.353946
Wife_religion	-0.506511
Wife_Working	-0.005046
Husband_Occupation	0.074468
Standard_of_living_index	0.237545
Media_exposure	-0.498047

Table 2.11 Weights of coefficients – LDA

Linear Discriminant Function:

LDF = **-0.34837862** + **(-0.075698)**Wife_age + **(0.573923)**Wife_education + **(-0.026745)***Husband_education + **(0.353946)** * No_of_children_born + **(-0.506511)** * Wife_religion + **(-0.005046)** * Wife_Working + **(0.074468)** * Husband_Occupation + **(0.237545)** * Standard_of_living_index + **(-0.498047)** * Media_exposure

The most important features are:

1. Wife_education has a positive correlation with the contraceptive method used. 1 unit increase in Wife_education leads to 0.57 units increase in contraceptive_method_used.

2. Wife_religion has a negative correlation with the contraceptive method used. 1 unit increase in Wife_religion leads to a decrease of 0.5 units of contraceptive_method_used.

3. 1 unit increase in Media_exposure leads to a 0.49 units decrease of contraceptive method used.

4. 1 unit increase in No_of_children born leads to an increase of 0.35 units of contraceptive method used.

5. Hence, the most important features are Wife_education, Wife_religion, Media_exposure and No_of_children_born.

2.3.3. Building the CART Model

Step 1: Import some important libraries like the DecisionTreeClassifier.

Step 2: CART model expects all the columns to be of numerical datatype. Hence all object type columns are converted to numerical type columns. The object type features Wife_education, Husband_education, Wife_religion, Wife_working, Standard_of_living_index.

```
 #   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
```

Table 2.12: Datatype of columns in the dataset

Step 3: Convert the object datatype to numerical datatype using the pandas.Categorical() function.

```
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Wife_age                  1473 non-null   float64
 1   Wife_ education           1473 non-null   int8
 2   Husband_education         1473 non-null   int8
 3   No_of_children_born       1473 non-null   float64
 4   Wife_religion             1473 non-null   int8
 5   Wife_Working              1473 non-null   int8
 6   Husband_Occupation        1473 non-null   int64
 7   Standard_of_living_index  1473 non-null   int8
 8   Media_exposure            1473 non-null   int8
 9   Contraceptive_method_used 1473 non-null   int8
dtypes: float64(2), int64(1), int8(7)
memory usage: 44.7 KB
```

Table 2.13: Datatype of columns in the dataset

Step 4: Build the CART model using DecisionTreeClassifier() function with 'gini' as a criterion. The decision tree is huge and overfitting issues may occur.

Step 5: Fit the CART model on a training dataset.

2.3.4 Prune the CART model by finding the best hyperparameters using GridSearch

Rebuild the Cart model after pruning using the max_depth = 7, min_samples_leaf=10, min_samples_split = 30 in the DecisionTreeClassifier() function. Pruning reduces the size of decision trees by removing parts of the tree that do not provide power to classify instances.

Insights:

- Check for the feature importance or the weightage of each feature.

	Imp
Wife_age	0.281857
Wife_ education	0.186002
Husband_education	0.000000
No_of_children_born	0.450607
Wife_religion	0.001121
Wife_Working	0.004275
Husband_Occupation	0.015198
Standard_of_living_index	0.034321
Media_exposure	0.026620

Table 2.14 Weightage of each feature – CART Model

- As we see that the feature 'Husband_education' has a weight of 0, it has never been used in the decision tree node splitting, hence this column can be safely removed. Also, the columns Wife_religion and Wife_Working have very low weighted values, hence can be safely removed.

- The columns with the most significance are No_of_children_born and Wife_age followed by Wife_education and Standard_of_living_index.

Predictions: Make the predictions on the test and training dataset.

Model Performance: Generate the AUC score for the test as well as training dataset.

AUC score for training dataset is 0.80525

AUC score for testing dataset is 0.75548

ROC curve for the training data:

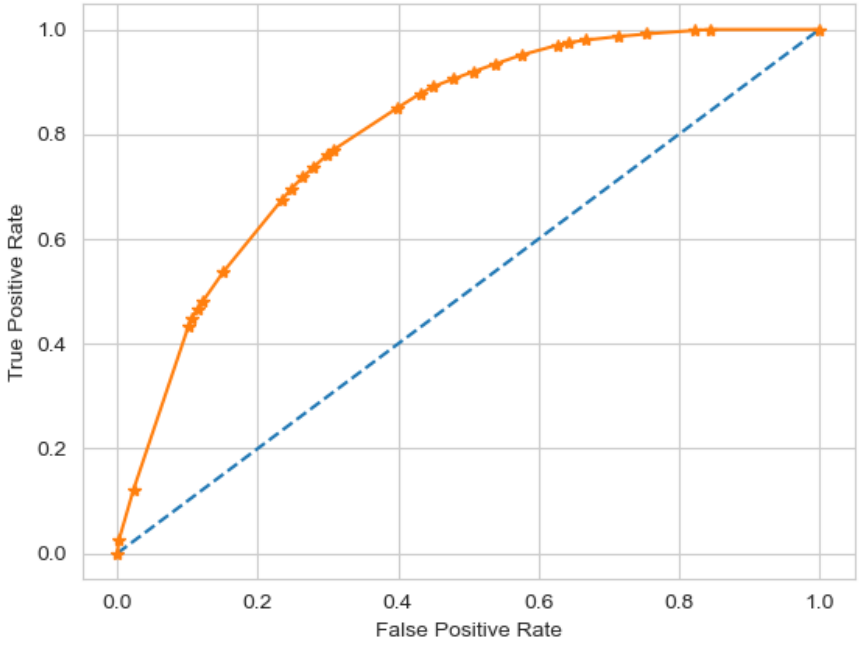


Fig 2.10: ROC Curve for Training data

ROC Curve for testing data:

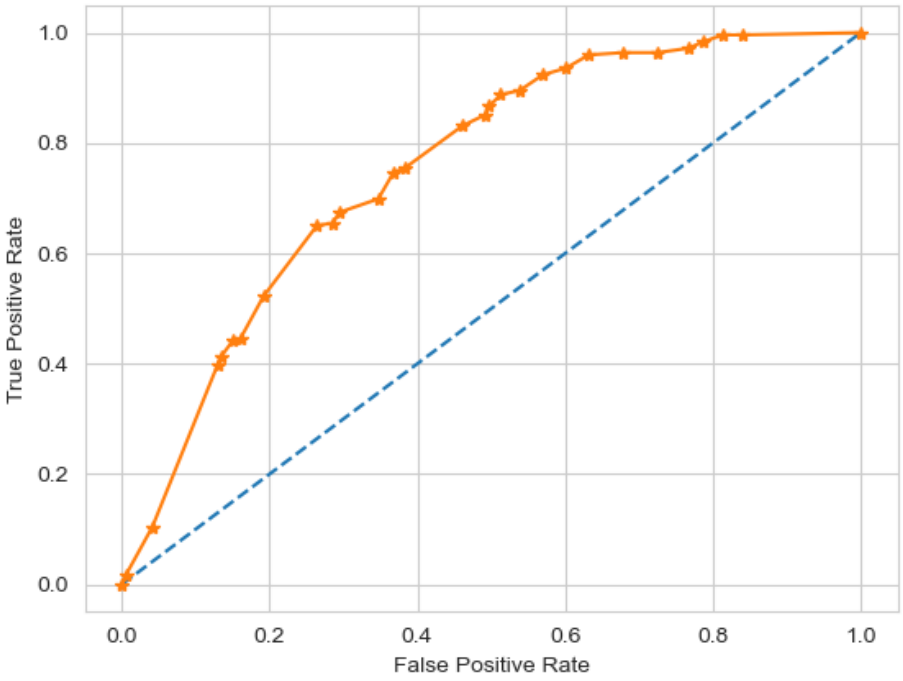


Fig 2.11: ROC Curve for Testing data

We see that the AUC is slightly better for the training data as compared to the testing data. The AUC score is 0.80 for training data while it is 0.75 for testing data. Hence, training data is better than the testing data.

Classification Report for the Training and Testing data:

	precision	recall	f1-score	support
0	0.77	0.57	0.66	436
1	0.74	0.88	0.80	595
accuracy			0.75	1031
macro avg	0.75	0.72	0.73	1031
weighted avg	0.75	0.75	0.74	1031

Table 2.15 Classification Report for Training data – CART

	precision	recall	f1-score	support
0	0.73	0.51	0.60	193
1	0.69	0.85	0.76	249
accuracy			0.70	442
macro avg	0.71	0.68	0.68	442
weighted avg	0.71	0.70	0.69	442

Table 2.16 Classification Report for Testing data – CART

The ROC AUC scores tell us how efficient the model is. AUC score of 0.8 and above is considered to be excellent. We see that the AUC score for the training dataset is 0.80525 and that of the testing set is 0.75548. The model is highly efficient.

Accuracy of the model is given by:

```
The accuracy of the training data is  0.7468477206595538
The accuracy of the testing data is  0.7013574660633484
```

Thus, we see that the accuracy of the training set in the CART model is 74.6% and the training of the test set is 70.1%.

2.3.5 Check the performance of the models across train and test set using different metrics.

As already calculated, the accuracy of the three models, namely, logistic Regression, LDA and CART models is shown below:

Model	Accuracy on Training Dataset	Accuracy on Testing Dataset	Overall Accuracy
Logistic Regression	67.3%	68.7%	68%
LDA	65.47%	66.51%	68%
CART	74.68%	70.13%	70%

Table 2.17 Comparison of accuracies of the three models

2.3.6 Compare the performance of all the models built and choose the best one with proper rationale.

We see from the above table that the accuracy of the training and testing dataset for all the three models is almost the same, so no issues of overfitting or underfitting.

The accuracy of the training dataset is the highest for CART model, followed by Logistic Regression and then LDA model.

The accuracy of the testing dataset is the highest for CART model, followed by Logistic Regression and then LDA model.

The overall accuracy of the CART model is the highest at 70%.

Hence, we choose the **CART model** over the other models to get the most significant features which will help us in the business.

2.4 Business Insights and Recommendations

2.4.1 Comment on the importance of features based on the best model.

	Imp
Wife_age	0.281857
Wife_ education	0.186002
Husband_education	0.000000
No_of_children_born	0.450607
Wife_religion	0.001121
Wife_Working	0.004275
Husband_Occupation	0.015198
Standard_of_living_index	0.034321
Media_exposure	0.026620

Table 2.18 Importance of features - CART

Best Model is the CART model.

We observe that Number of children born is the most important feature for the use of contraceptives followed by Wife age and Wife education. Both these features are the most important features in the list of features which determine the contraceptive method used.

1 unit increase in Number of children born leads to an increase of 0.45 units of contraceptive method used.

1 unit increase in Wife age leads to an increase of 0.28 units of contraceptive method used.

1 unit increase in Wife education leads to an increase of 0.186 units of contraceptive method used.

2.4.2 Conclude with the key takeaways (actionable insights and recommendations) for the business

We conclude that CART is the best model to classify the features of the given dataset.

Number of children born is the most important feature which determines the dependent variable, which is contraceptive method used. This is followed by Wife age and Wife education.

After a requisite number of children are born, maybe the wives use contraceptives. From a business perspective, wives with median age of 32 and above can be targeted for the sale of contraceptives.

Also, wife's age is an important determinant of contraceptive method used. Maybe wives after a particular age group of 35 and above do not prefer to have children due to medical reasons. Hence, women of 35 age group and above is a good target group for the sale of contraceptives.

Wife education plays a significant role in determining the use of contraceptives. It is seen that women who are educated tend to use contraceptives after a certain age group in order to not have children, as compared to women who are not very educated. Hence, this group of women can be a target group for the sale of contraceptives.

Thus, as a business initiative, we can immediately spring into action with the three target groups to increase our sale of contraceptives.

****