Predictive Modeling

PROJECT REPORT

Gowdham S
PGPDSA.O.DEC23.A

Table of Contents

List of Figures

List of Tables

Data Dictionary of Problem 1

Column	Description
Lread	writes (transfers per second) between system memory and user memory
Lwrite	Reads (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 transferred 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.
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

Data Dictionary of Problem 2

Column	Description
Wife_age	Wife's age (numerical)
Wife_ education	Wife's education (categorical) 1=uneducated, 2, 3, 4=tertiary
Husband_education	Husband's education (categorical) 1=uneducated, 2, 3, 4=tertiary
No_of_children_born	Number of children ever born (numerical)
Wife_religion	Wife's religion (binary) Non-Scientology, Scientology
Wife_Working	Wife's now working? (binary) Yes, No
Husband_Occupation	Husband's occupation (categorical) 1, 2, 3, 4(random)
Standard_of_living_index	Standard-of-living index (categorical) 1=very low, 2, 3, 4=high
Media_exposure	Media exposure (binary) Good, Not good

Problem 1

Context

The comp-activ database contains activity measures of computer systems. Data was collected from a Sun Sparcstation 20/712 with 128 Mbytes of memory, functioning in a multi-user university department. Users were involved in various tasks, including internet access, file editing, and CPU-intensive programs.

Objective

The objective is to establish a linear equation for predicting the 'usr' (the percentage of time CPUs operate in user mode). The aim is to analyze different system attributes to discern their impact on the system's 'usr' mode.

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

Solution:

The problem is to develop a predictive linear regression model for estimating the percentage of time CPUs operate in user mode ('usr') within a computer system. Data is sourced from the comp-activ database, gathered from a Sun Sparcstation 20/712 in a multi-user university department setting. The objective is to establish a linear equation using system attributes to predict 'usr' mode percentage, providing insights into system behaviour and performance.

- To begin with, we just see the top 5 rows of the data using the head() method. Since the number of columns is high, we can see the data is separated into 2 parts as shown below,

index	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	ppgout
0	1	0	2147	79	68	0.2	0.2	40671	53995	0	0
1	0	0	170	18	21	0.2	0.2	448	8385	0	0
2	15	3	2162	159	119	2	2.4	125473.5	31950	0	0
3	0	0	160	12	16	0.2	0.2	125473.5	8670	0	0
4	5	1	330	39	38	0.4	0.4	125473.5	12185	0	0

pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
0	0	0	1.6	2.6	16	26.4	CPU_Bound	4670	1730946	95
0	0	0	0	0	15.63	16.83	Not_CPU_Bound	7278	1869002	97
0	0	1.2	6	9.4	150.2	220.2	Not_CPU_Bound	702	1021237	87
0	0	0	0.2	0.2	15.6	16.8	Not_CPU_Bound	7248	1863704	98
0	0	0	1	1.2	37.8	47.6	Not_CPU_Bound	633	1760253	90

Table 1 Sample data

- We used shape attribute of pandas library [df1.shape] to determine the number of rows and columns of the dataframe df1,

Shape	Description
8192, 22	We have 8192 rows and 22 columns in the dataset

Table 2 Shape of comp-activ Dataset

- To check the types of data, we used info() method of pandas library to list the basic information of the data such as rows count, columns count and datatype of the columns.

Column	Non-Null Count	Data Type
lread	8192 non-null	int64
lwrite	8192 non-null	int64
scall	8192 non-null	int64
sread	8192 non-null	object
swrite	8192 non-null	float64
fork	8192 non-null	float64
exec	8192 non-null	float64
rchar	8192 non-null	float64
wchar	8192 non-null	float64
pgout	8192 non-null	float64
ppgout	8192 non-null	float64
pgfree	8192 non-null	float64
pgscan	8192 non-null	float64
atch	8177 non-null	float64
pgin	8088 non-null	float64
ppgin	8192 non-null	float64
pflt	8192 non-null	float64
vflt	8192 non-null	int64
runqsz	8192 non-null	int64

Table 3 Basic Information of comp-activ Dataset

- From the above table, We have **1** object, **8** int and **13** float data types in the dataset. Also we can see that there are **104** null rows in the rchar and **15** null rows in the wchar columns.
- To check if there are any duplicate columns, we used df1.duplicated().sum(). We can see that there are no duplicate rows in the dataset.
- Once the above is done, we found the statistical summary of the dataset using describe method()

Index	Count	Mean	STD	Min	25%	50%	75%	Max
lread	8192	13.42	15.16	0	2	7	20	47
lwrite	8192	6.66	9.29	0	0	1	10	25
scall	8192	2294.48	1593.09	109	1012	2051.5	3317.25	6775.12
sread	8192	199.78	146.76	6	86	166	279	568.5
swrite	8192	137.97	97.14	7	63	117	185	368
fork	8192	1.56	1.59	0	0.4	0.8	2.2	4.9
exec	8192	1.93	2.03	0	0.2	1.2	2.8	6.7
rchar	8192	178884.1	174589.2	278	34860.5	125473.5	265394.8	611196.1
wchar	8192	75645.54	71262.96	1498	22977.75	46619	106037	230625.9
pgout	8192	1.42	2.2	0	0	0	2.4	6
ppgout	8192	2.56	4.04	0	0	0	4.2	10.5
pgfree	8192	3.16	4.98	0	0	0	5	12.5
pgscan	8192	0	0	0	0	0	0	0
atch	8192	0.39	0.56	0	0	0	0.6	1.5
pgin	8192	6.39	7.68	0	0.6	2.8	9.76	23.51
ppgin	8192	9.14	11.16	0	0.6	3.8	13.8	33.6
pflt	8192	105.64	101.55	0	25	63.8	159.6	361.5
vflt	8192	175.62	162.5	0.2	45.4	120.4	251.8	561.4
runqsz	8192	0.53	0.5	0	0	1	1	1

freemem	8192	1387.62	1605.76	55	231	579	2002.25	4659.12
freeswap	8192	1328520	420782.7	10990	1042624	1289290	1730380	2243187
usr	8192	86.25	9.75	61.5	81	89	94	99

Table 4 Statistical Summary of comp-activ Dataset

- Before proceeding with the Univariate and Bivariate analysis, We will impute the missing values in rchar and wchar with median values as the data is numerical and skewed. We can go with mean in case of normally distributed columns.
- After imputing missing values we can see that there are no missing values in the rchar and wchar columns and we verified this using isnull() method.

Index	Null Values
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

Table 5 Missing Value Check

- First, we selected the numerical and categorical columns from the dataframe and assigned them to the variables using select_dtypes(). We then proceeded with the univariate and bivariate analysis.

Variable Name	Columns
Numerical Cols	Lread, lwrite, scall, sread, swrite, fork, exec, rchar, wchar, pgout, ppgout, pgfree, pgscan, atch, pgin, ppgin, pflt, vflt, freemem, freeswap & usr
Categorical Cols	runqsz

Table 6 Numerical & Categorical Columns of comp-activ Dataset

- After separating the numerical and categorical columns, we created boxplots for each of the numerical columns using seaborn library

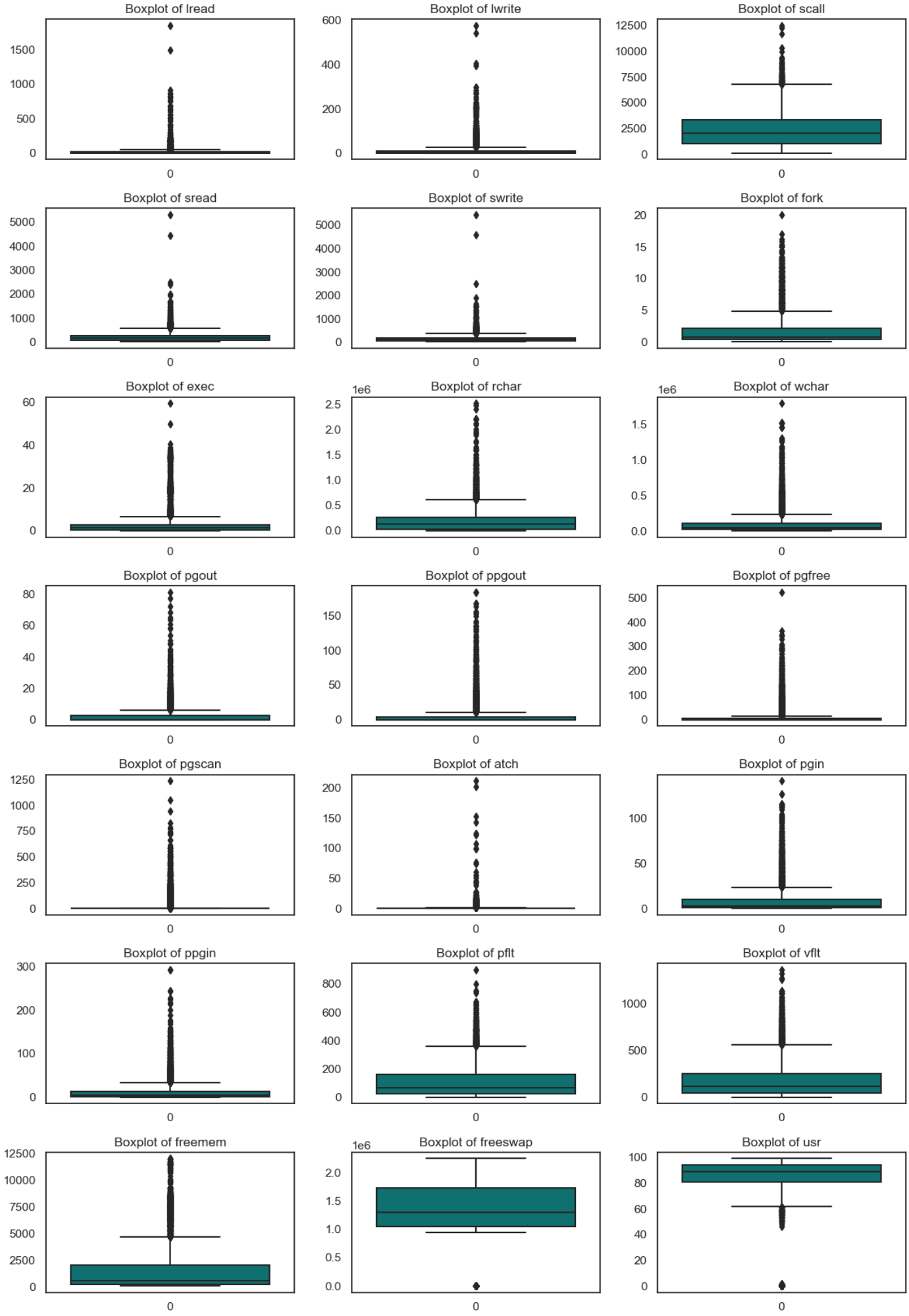


Figure 1 Box Plot of Numerical Columns

- Then we plotted a histogram for the numerical columns followed by the distribution of categorical columns.

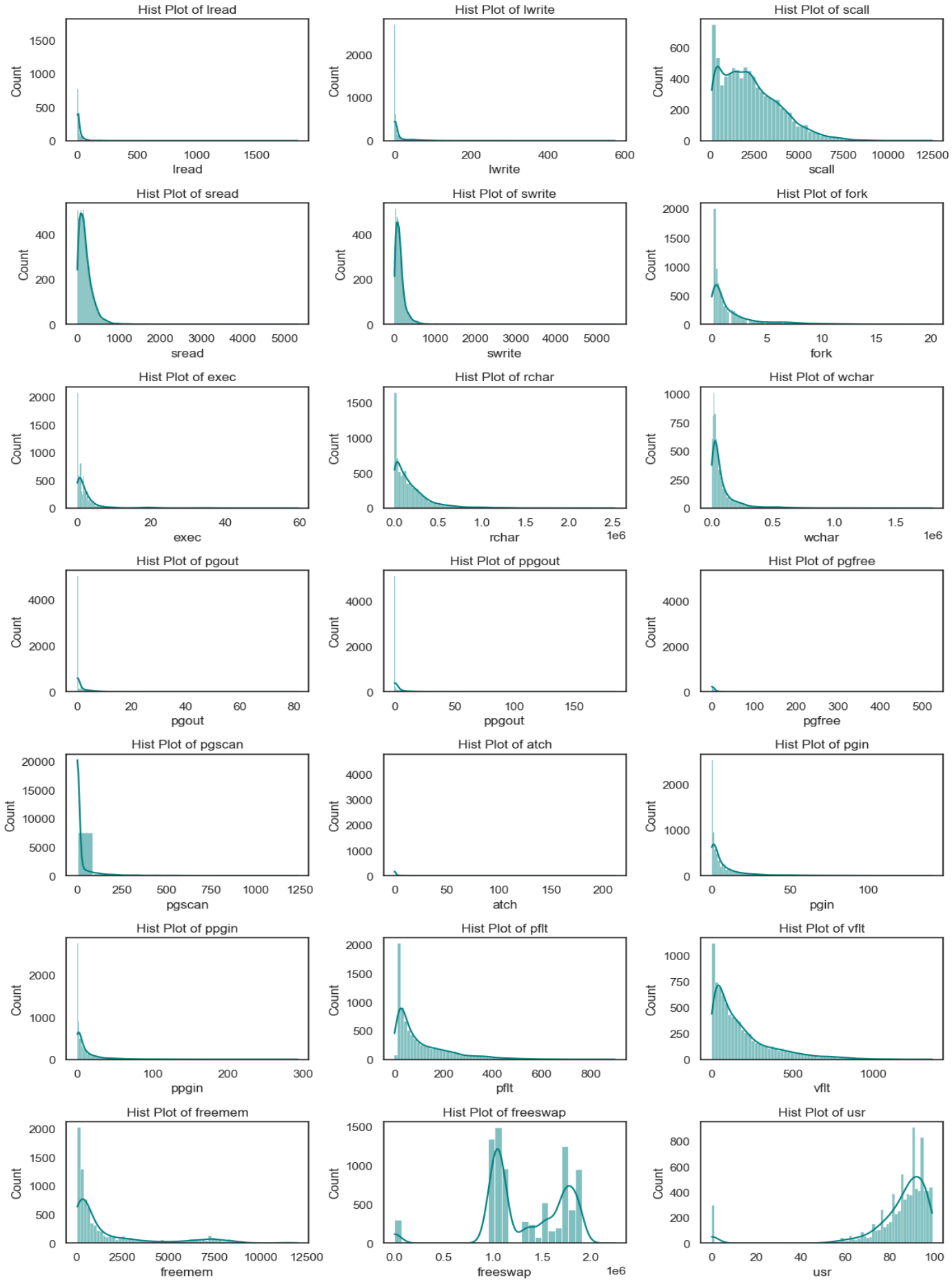


Figure 2 Hist Plot of Numerical Columns

- From the above box plots we can clearly see that there are outliers in all the numerical variables. But these need not to be outliers. We can check this further in the pre-processing stage. Also, we can see that the data is mostly right skewed. usr column is left skewed. The histogram of freeswap shows that the data is somewhat bimodal.

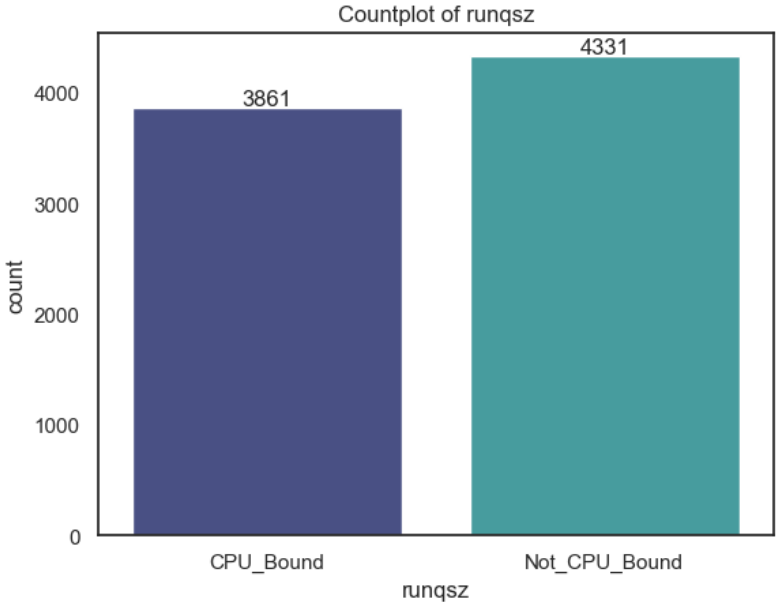
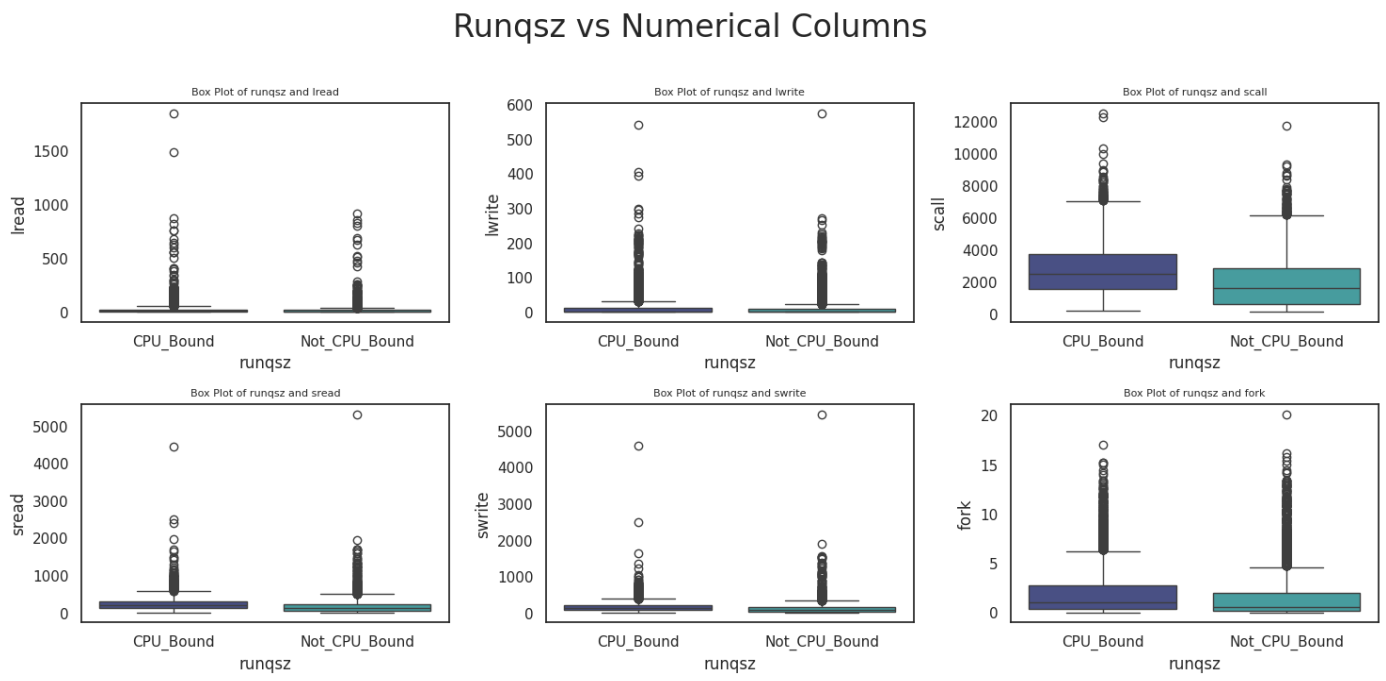


Figure 3 Count Plot of Categorical Column - runqz

- Once the univariate analysis is done, we then proceeded with the bivariate analysis. This step involves examining how pairs of variables interact with each other to uncover relationships within the data.
- First, we plotted box plots of each categorical variable with all the numerical variables.
- This technique allows us to understand how the distribution of numerical variables varies across different categories, providing deeper insights into potential associations within the data.
- Then, we further explored the relationships by plotting a heatmap showing the correlation between the variables.
- The heatmap provides a visual representation of the strength and direction of relationships between variables, aiding in identifying potential patterns or dependencies.



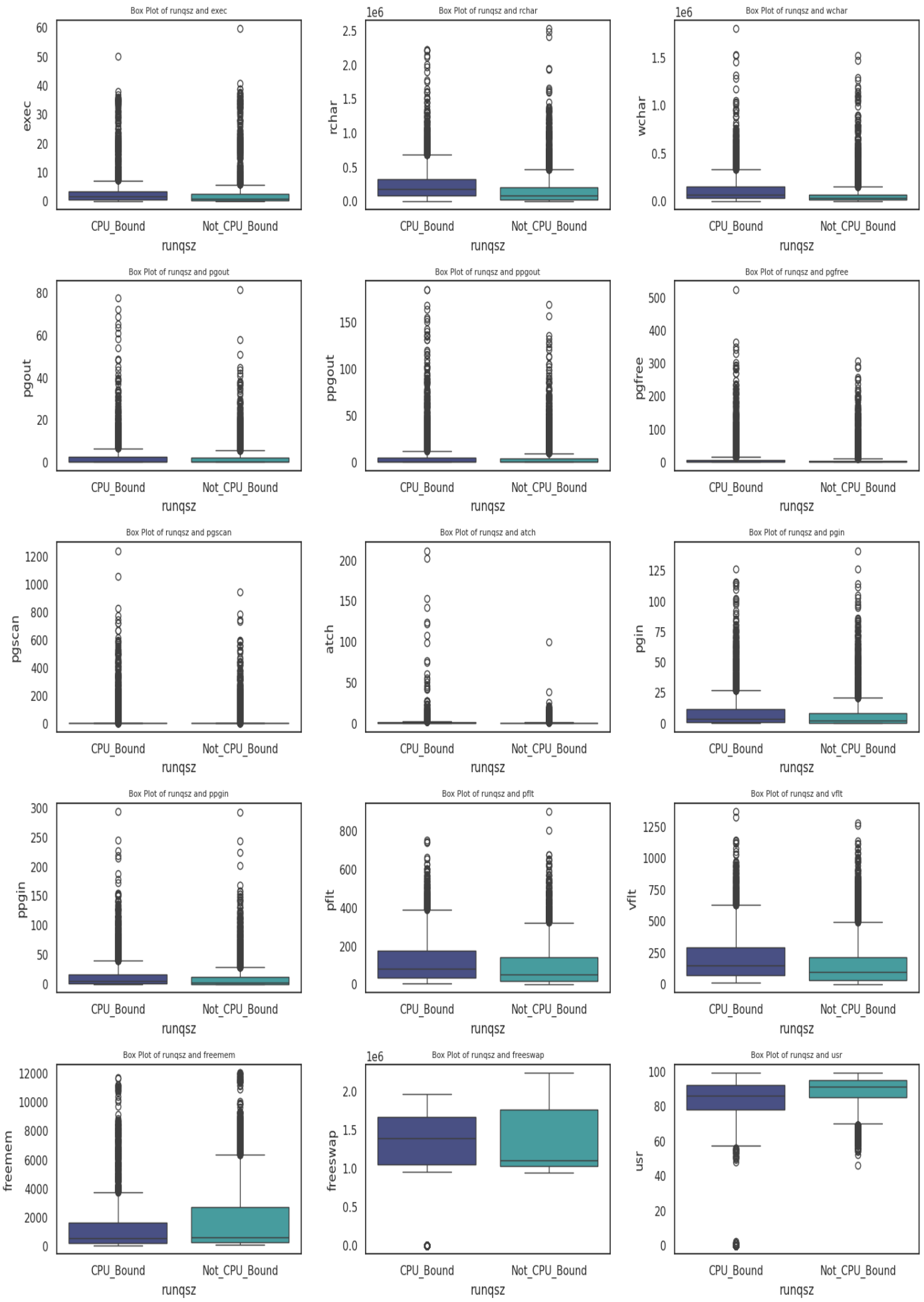


Figure 4 Box Plot of runqsz vs Numerical Columns

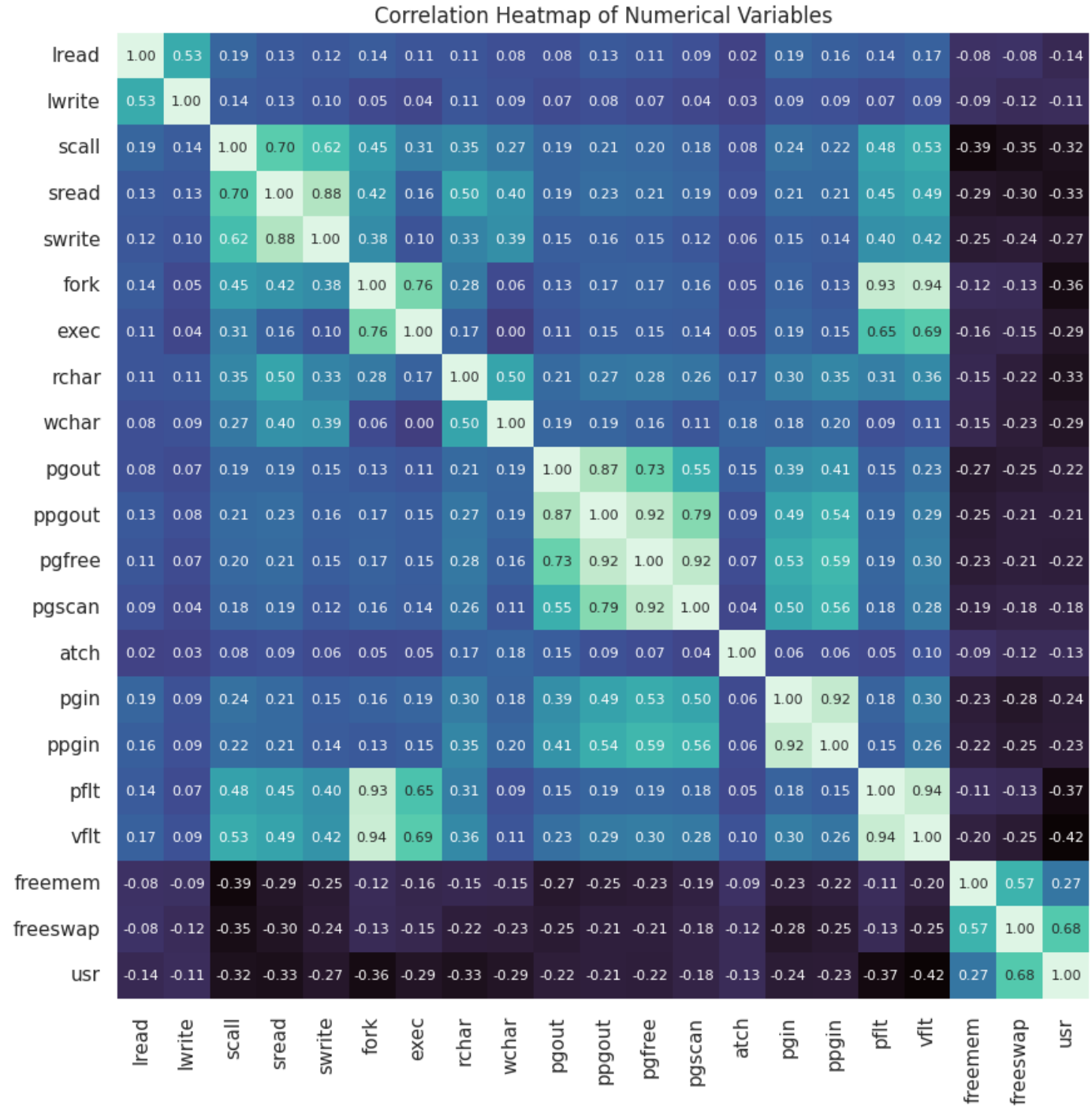


Figure 5 Correlation Heatmap of Numerical Variables

Observations:

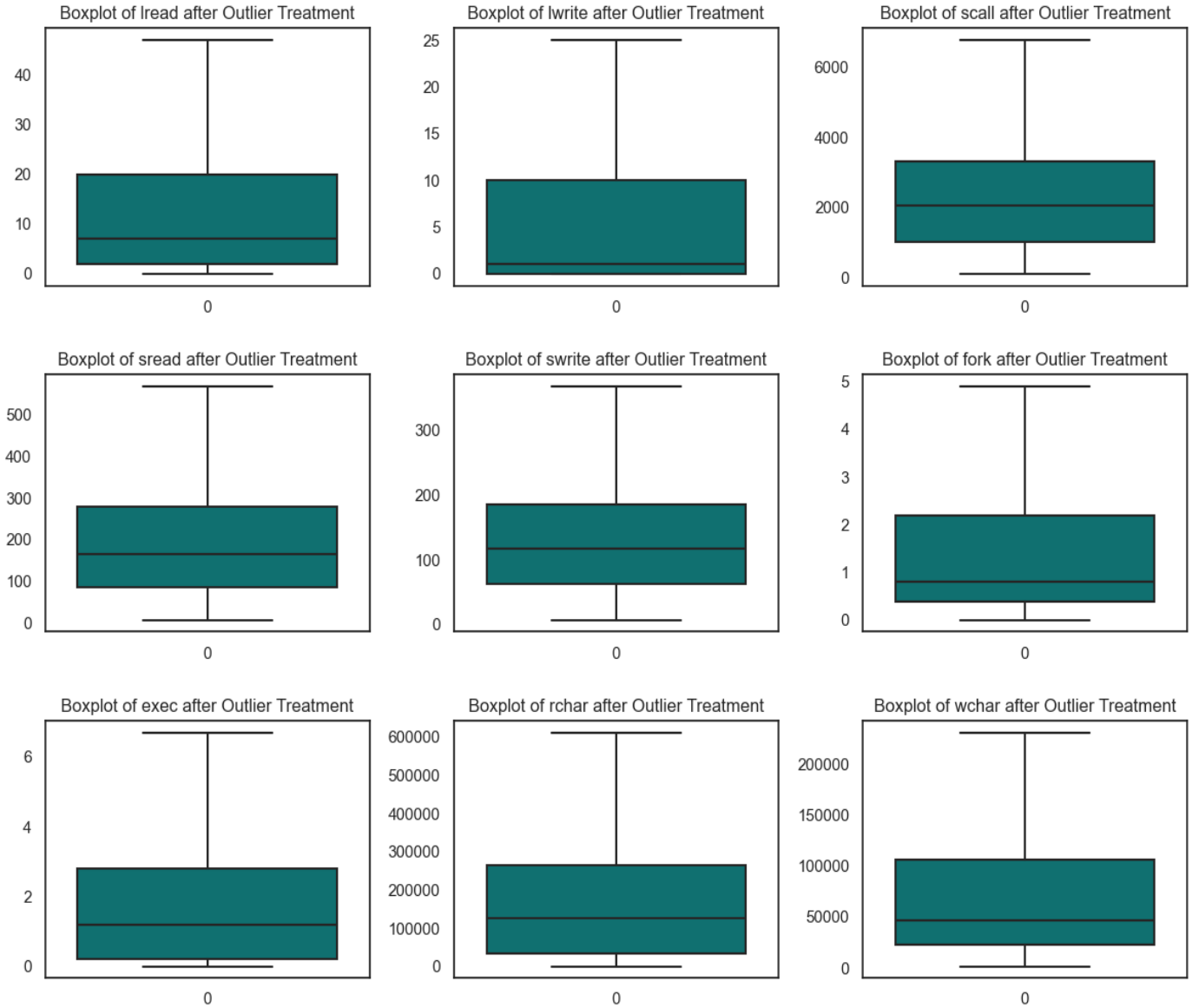
- Majority of the times the process run queue size was Not CPU Bound.
- On an average 83.9% of times the cpus run in user mode.
- There are no duplicate rows present in the data. Available Impressions, Matched Queries, Impressions, Clicks, Spend, Revenue are highly correlated. This means that these variables tend to move together. For example, if there are more Available Impressions, there are also likely to be more Matched Queries, Impressions, Clicks, Spend, and Revenue.
- As observed before, number of page out requests per second is also highly correlated to the number of pages, paged out per second variable.
- Similarly, both the page fault variables – pflt & vflt are highly correlated with the fork variable.

Problem 1 - Data Preprocessing

Prepare the data for modelling: - Missing Value Treatment (if needed) - Outlier Detection (treat, if needed) - Feature Engineering - Encode the data - Train-test split.

Solution:

- After treating missing values using the median value, our dataset is now devoid of any missing values, ensuring the integrity of our data for further analysis. Refer Table 5
- From Figure 1, we can see outliers in every columns & we prioritize outlier treatment before model training. By identifying and addressing outliers, we aim to enhance the robustness and effectiveness of our model.
- To systematically handle outliers, we have devised a function called **handle_outliers**. This function calculates upper and lower bound values using a predetermined equation. These bounds serve as thresholds beyond which data points are considered outliers and are subsequently treated to mitigate their impact on our analysis.
- We can see the below Box plots of numerical variables before and after the treatment of outliers.
- We can clearly see that there are no outliers after the treatment.



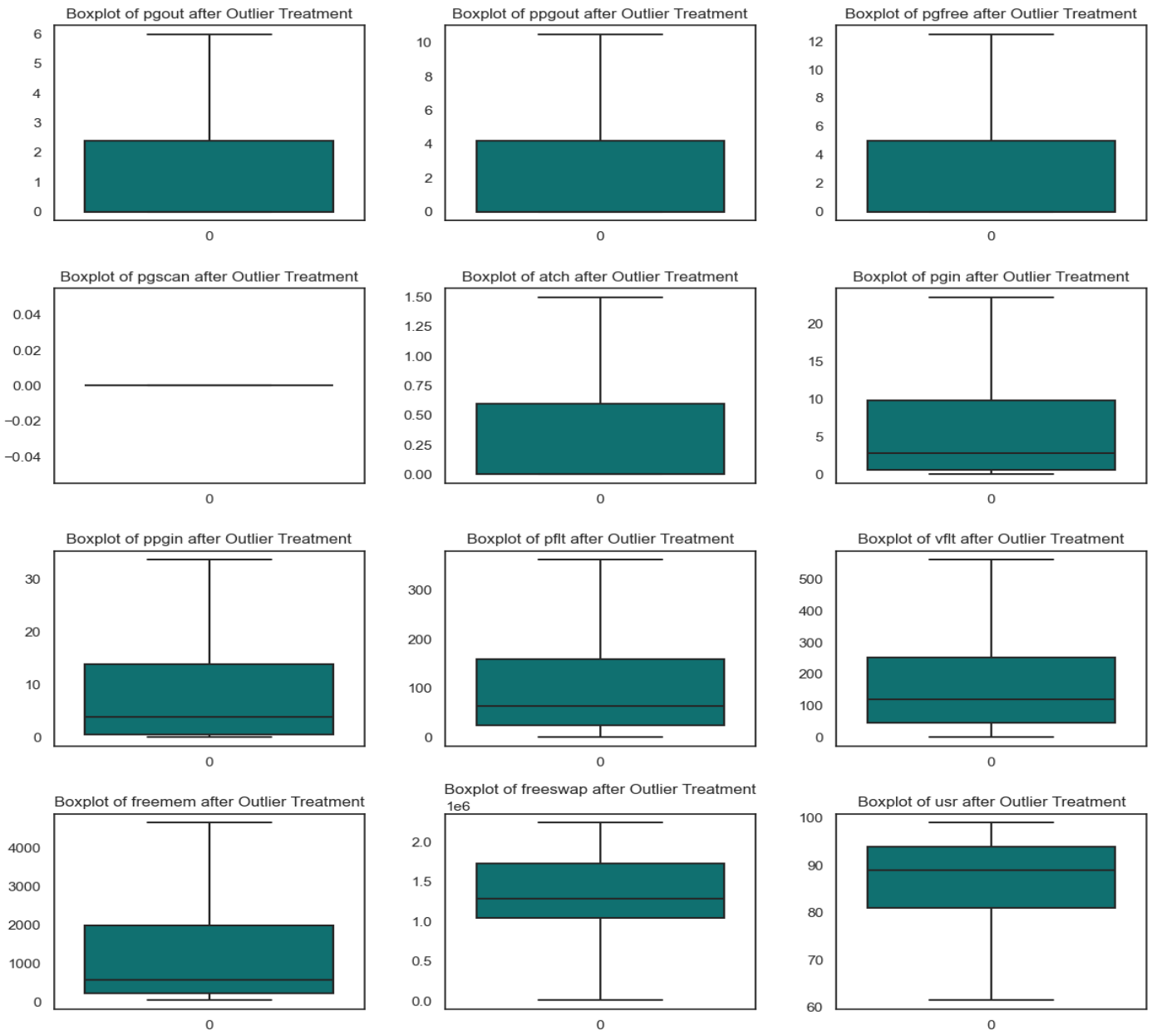


Figure 6 Box Plot of Numerical Variables After Outlier Treatment

- After treating the outliers, we see that there is quite a big difference between the min and max values between the variables. So we proceeded the scaling the data.

index	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout
0	-0.82	-0.72	-0.09	-0.82	-0.72	-0.85	-0.85	-0.79	-0.30	-0.65
1	-0.89	-0.72	-1.33	-1.24	-1.20	-0.85	-0.85	-1.02	-0.94	-0.65
2	0.10	-0.39	-0.08	-0.28	-0.20	0.28	0.23	-0.31	-0.61	-0.65
3	-0.89	-0.72	-1.34	-1.28	-1.26	-0.85	-0.85	-0.31	-0.94	-0.65
4	-0.56	-0.61	-1.23	-1.10	-1.03	-0.73	-0.76	-0.31	-0.89	-0.65

ppgout	pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	usr
-0.63	-0.64	0.00	-0.69	-0.62	-0.59	-0.88	-0.92	2.04	0.96	0.90
-0.63	-0.64	0.00	-0.69	-0.83	-0.82	-0.89	-0.98	2.04	1.28	1.10
-0.63	-0.64	0.00	1.44	-0.05	0.02	0.44	0.27	-0.43	-0.73	0.08
-0.63	-0.64	0.00	-0.69	-0.80	-0.80	-0.89	-0.98	2.04	1.27	1.21
-0.63	-0.64	0.00	-0.69	-0.70	-0.71	-0.67	-0.79	-0.47	1.03	0.39

Table 7 Sample Data after Scaling

- We will now proceed with encoding the object column 'runqsz' using pd.get_dummies() method. We call it as One Hot encoding.

index	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout
0	-0.82	-0.72	-0.09	-0.82	-0.72	-0.85	-0.85	-0.79	-0.30	-0.65
1	-0.89	-0.72	-1.33	-1.24	-1.20	-0.85	-0.85	-1.02	-0.94	-0.65
2	0.10	-0.39	-0.08	-0.28	-0.20	0.28	0.23	-0.31	-0.61	-0.65
3	-0.89	-0.72	-1.34	-1.28	-1.26	-0.85	-0.85	-0.31	-0.94	-0.65
4	-0.56	-0.61	-1.23	-1.10	-1.03	-0.73	-0.76	-0.31	-0.89	-0.65

ppgout	pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	usr	runqsz
-0.63	-0.64	0.00	-0.69	-0.62	-0.59	-0.88	-0.92	2.04	0.96	0.90	0
-0.63	-0.64	0.00	-0.69	-0.83	-0.82	-0.89	-0.98	2.04	1.28	1.10	1
-0.63	-0.64	0.00	1.44	-0.05	0.02	0.44	0.27	-0.43	-0.73	0.08	1
-0.63	-0.64	0.00	-0.69	-0.80	-0.80	-0.89	-0.98	2.04	1.27	1.21	1
-0.63	-0.64	0.00	-0.69	-0.70	-0.71	-0.67	-0.79	-0.47	1.03	0.39	1

Table 8 Sample Data after One Hot Encoding

- We splitted the data into train and test using the train_test_split method from the sklearn library. We splitted the data into training and test in the percentage 70:30.

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

Solution:

Linear regression models the relationship between a dependent variable and one or more independent variables, assuming a linear association. In simple linear regression, there's one independent variable, while multiple linear regression involves several. The goal is to estimate coefficients that minimize the difference between observed and predicted values. It's widely applied across fields for prediction and understanding variable relationships.

Model 1 – Sklearn method

We assess the model's performance using Rsquare values and Root Mean Squared Error (RMSE) for both the training and test datasets.

R-squared (R^2):

- R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables in the model.
- It ranges from 0 to 1, where 1 indicates that the model perfectly explains the variability in the data, and 0 indicates that the model explains none of the variability. Higher R-squared values indicate better model fit.
- The R-squared values on both the training and test datasets are quite high, indicating that a significant proportion of the variance in the target variable is explained by the predictor variables.
- The R-squared value on the training data (0.7884) is slightly higher than that on the test data (0.7877), suggesting that the model generalizes well to unseen data.

Root Mean Squared Error (RMSE):

- RMSE measures the average magnitude of the errors between predicted and observed values. It is the square root of the average of the squared differences between predicted and observed values.
- RMSE is in the same units as the target variable, and lower values indicate better model performance.
- The RMSE values on both the training and test datasets are relatively low, indicating that the model's predictions are close to the actual values on average.
- The RMSE on the training data (0.4600) is slightly higher than that on the test data (0.4610), which is consistent with the observation that the model performs slightly better on the test data.

Mean Absolute Error (MAE):

- MAE measures the average magnitude of the errors between predicted and observed values.
- It is the average of the absolute differences between predicted and observed values.
- MAE is less sensitive to outliers compared to RMSE, and lower values indicate better model performance.
- The MAE values on both the training and test datasets are also relatively low, indicating that, on average, the absolute differences between the predicted and actual values are small.
- Similar to RMSE, the MAE on the training data (0.3428) is slightly higher than that on the test data (0.3407), suggesting good generalization performance of the model.

Overall, the model seems to perform well, with high R-squared values and low error metrics on both the training and test datasets. The slight difference in performance between the training and test datasets suggests that the model is not overfitting, and it generalizes well to unseen data.

These metrics indicate a good fit for both the training and test datasets, with similar performance between them. The model doesn't seem to suffer from overfitting or underfitting, as the Rsquare values are reasonably high and the RMSE values are relatively low.

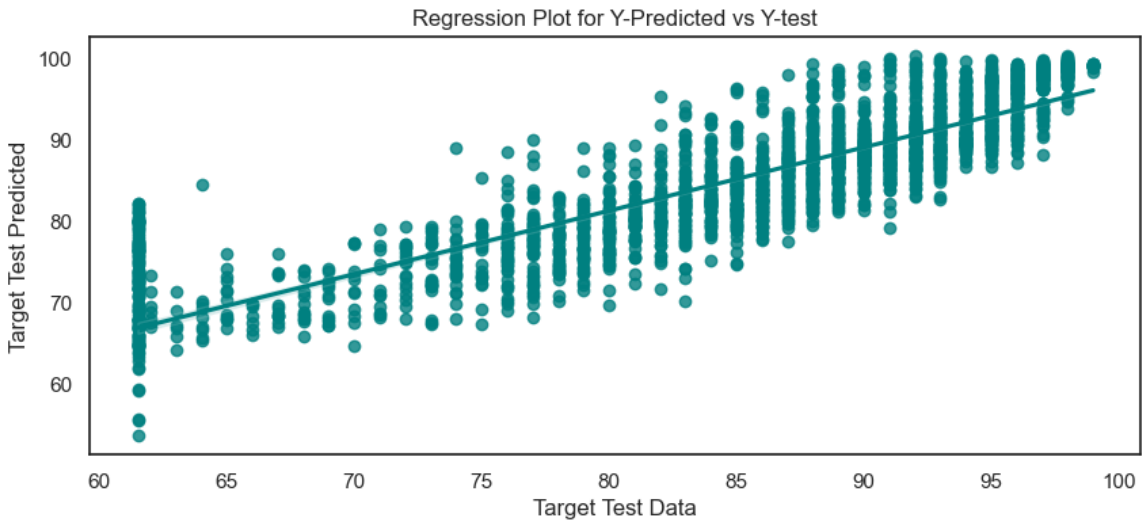


Figure 7 Regression Plot for Y-Predicted vs Y-test

Model 2 - Statsmodel method

We built a model using stats model and OLS method, we see that the Adjusted Rsquare is equal to the Rsquare value which is 0.79.

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.796
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1115.
Date:	Sun, 12 May 2024	Prob (F-statistic):	0.00
Time:	16:31:08	Log-Likelihood:	-3600.4
No. Observations:	5734	AIC:	7243.
Df Residuals:	5713	BIC:	7383.
Df Model:	20		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0905	0.009	-9.935	0.000	-0.108	-0.073
lread	-0.0987	0.014	-7.071	0.000	-0.126	-0.071
lwrite	0.0459	0.013	3.671	0.000	0.021	0.070
scall	-0.1085	0.010	-10.566	0.000	-0.129	-0.088
sread	0.0046	0.015	0.305	0.760	-0.025	0.034
swrite	-0.0540	0.014	-3.777	0.000	-0.082	-0.026
fork	0.0048	0.022	0.222	0.824	-0.037	0.047
exec	-0.0668	0.011	-6.220	0.000	-0.088	-0.046
rchar	-0.0925	0.009	-10.598	0.000	-0.110	-0.075
wchar	-0.0395	0.008	-5.232	0.000	-0.054	-0.025
pgout	-0.0832	0.020	-4.098	0.000	-0.123	-0.043
ppgout	-0.0317	0.033	-0.973	0.330	-0.096	0.032
pgfree	0.0432	0.024	1.769	0.077	-0.005	0.091
pgscan	-1.33e-16	1.77e-17	-7.521	0.000	-1.68e-16	-9.84e-17
atch	0.0362	0.008	4.394	0.000	0.020	0.052
pgin	0.0158	0.022	0.703	0.482	-0.028	0.060
ppgin	-0.0771	0.023	-3.415	0.001	-0.121	-0.033
pflt	-0.3500	0.021	-16.957	0.000	-0.390	-0.310
vflt	-0.0911	0.024	-3.830	0.000	-0.138	-0.044
freemem	-0.0755	0.008	-9.038	0.000	-0.092	-0.059
freeswap	0.3812	0.008	46.472	0.000	0.365	0.397
runqsz	0.1657	0.013	12.819	0.000	0.140	0.191

Omnibus:	1103.645	Durbin-Watson:	2.016
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2372.553
Skew:	-1.119	Prob(JB):	0.00
Kurtosis:	5.219	Cond. No.	1.57e+17

Figure 8 OLS Summary 1

- RMSE on training data is 0.453 & RMSE on test data is 0.477
- We calculated the VIF (Variance Inflation Factor) for identifying important variables and eliminating the ones that are not significant and have high multicollinearity.

VIF values:

const	29.229332
lread	5.350560
lwrite	4.328397
scall	2.960609
sread	6.420172
swrite	5.597135
fork	13.035359
exec	3.241417
rchar	2.133616
wchar	1.584381
pgout	11.360363
ppgout	29.404223
pgfree	16.496748
pgscan	NaN
atch	1.875901
pgin	13.809339
ppgin	13.951855
pflt	12.001460
vflt	15.971049
freemem	1.961304
freeswap	1.841239
runqsz	1.156815

Figure 9 VIF Values

The VIF values indicate that the features fork, pgout, ppgout, pgfree, pgin, ppgin, pflt and vflt are correlated with one or more independent features. Since pgscan doesn't have any non-zero values, we can remove it and check the performance of models

Model 3 - Statsmodel method

We then built a model using stats model and OLS method, we see that the Adjusted Rsquare is equal to the Rsquare value which is 0.79.

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.796
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1115.
Date:	Sun, 12 May 2024	Prob (F-statistic):	0.00
Time:	16:31:09	Log-Likelihood:	-3600.4
No. Observations:	5734	AIC:	7243.
Df Residuals:	5713	BIC:	7383.
Df Model:	20		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0905	0.009	-9.935	0.000	-0.108	-0.073
lread	-0.0987	0.014	-7.071	0.000	-0.126	-0.071
lwrite	0.0459	0.013	3.671	0.000	0.021	0.070
scall	-0.1085	0.010	-10.566	0.000	-0.129	-0.088
sread	0.0046	0.015	0.305	0.760	-0.025	0.034
swrite	-0.0540	0.014	-3.777	0.000	-0.082	-0.026
fork	0.0048	0.022	0.222	0.824	-0.037	0.047
exec	-0.0668	0.011	-6.220	0.000	-0.088	-0.046
rchar	-0.0925	0.009	-10.598	0.000	-0.110	-0.075
wchar	-0.0395	0.008	-5.232	0.000	-0.054	-0.025
pgout	-0.0832	0.020	-4.098	0.000	-0.123	-0.043
ppgout	-0.0317	0.033	-0.973	0.330	-0.096	0.032
pgfree	0.0432	0.024	1.769	0.077	-0.005	0.091
atch	0.0362	0.008	4.394	0.000	0.020	0.052
pgin	0.0158	0.022	0.703	0.482	-0.028	0.060
ppgin	-0.0771	0.023	-3.415	0.001	-0.121	-0.033
pflt	-0.3500	0.021	-16.957	0.000	-0.390	-0.310
vflt	-0.0911	0.024	-3.830	0.000	-0.138	-0.044
freemem	-0.0755	0.008	-9.038	0.000	-0.092	-0.059
freeswap	0.3812	0.008	46.472	0.000	0.365	0.397
runqsz	0.1657	0.013	12.819	0.000	0.140	0.191

Omnibus:	1103.645	Durbin-Watson:	2.016
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2372.553
Skew:	-1.119	Prob(JB):	0.00
Kurtosis:	5.219	Cond. No.	18.6

Figure 10 OLS Summary 2

- On dropping 'pgscan' R-squared & Adj R-squared remains the same.
- RMSE on training data is 0.453 & RMSE on test data is 0.477
- Now we will proceed with dropping the page variables "pgin","ppgin","pgout","ppgout" as they are highly correlated to each other and we can verify this from the heatmap Figure 5.

Model 4 - Statsmodel method

We then built a model using stats model and OLS method, we see that the Adjusted Rsquare is equal to the Rsquare value which is 0.79.

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.792
Model:	OLS	Adj. R-squared:	0.792
Method:	Least Squares	F-statistic:	1364.
Date:	Sun, 12 May 2024	Prob (F-statistic):	0.00
Time:	16:31:09	Log-Likelihood:	-3652.7
No. Observations:	5734	AIC:	7339.
Df Residuals:	5717	BIC:	7452.
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0858	0.009	-9.347	0.000	-0.104	-0.068
lread	-0.1148	0.014	-8.258	0.000	-0.142	-0.088
lwrite	0.0579	0.013	4.631	0.000	0.033	0.082
scall	-0.1107	0.010	-10.694	0.000	-0.131	-0.090
sread	0.0094	0.015	0.616	0.538	-0.021	0.039
swrite	-0.0565	0.014	-3.920	0.000	-0.085	-0.028
fork	0.0363	0.021	1.703	0.089	-0.005	0.078
exec	-0.0710	0.011	-6.579	0.000	-0.092	-0.050
rchar	-0.1026	0.009	-11.842	0.000	-0.120	-0.086
wchar	-0.0461	0.008	-6.078	0.000	-0.061	-0.031
pgfree	-0.0710	0.008	-8.668	0.000	-0.087	-0.055
atch	0.0253	0.008	3.154	0.002	0.010	0.041
pflt	-0.3283	0.021	-16.003	0.000	-0.369	-0.288
vflt	-0.1446	0.023	-6.375	0.000	-0.189	-0.100
freemem	-0.0749	0.008	-8.907	0.000	-0.091	-0.058
freeswap	0.3858	0.008	46.891	0.000	0.370	0.402
runqsz	0.1576	0.013	12.114	0.000	0.132	0.183

Omnibus:	1034.315	Durbin-Watson:	2.012
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2101.299
Skew:	-1.078	Prob(JB):	0.00
Kurtosis:	5.037	Cond. No.	11.2

Figure 11 OLS Summary 3

- On dropping "pgin","ppgin","pgout","ppgout" R-squared & Adj R-squared remains almost same with minor change from the previous model.
- RMSE on training data is 0.457 & RMSE on test data is 0.481
- From the above summary we can see that "sread","fork" columns have high p values greater than 0.05. So we proceeded with dropping them.

Model 5 - Statsmodel method

We then built a model using stats model and OLS method, we see that the Adjusted Rsquare is equal to the Rsquare value which is 0.79.

- On dropping "sread","fork”, R-squared & Adj R-squared remains the same without any major difference.
- RMSE on training data is 0.457 & RMSE on test data is 0.481.

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.792
Model:	OLS	Adj. R-squared:	0.792
Method:	Least Squares	F-statistic:	1558.
Date:	Sun, 12 May 2024	Prob (F-statistic):	0.00
Time:	16:31:09	Log-Likelihood:	-3654.3
No. Observations:	5734	AIC:	7339.
Df Residuals:	5719	BIC:	7438.
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0859	0.009	-9.360	0.000	-0.104	-0.068
lread	-0.1152	0.014	-8.292	0.000	-0.142	-0.088
lwrite	0.0577	0.012	4.623	0.000	0.033	0.082
scall	-0.1113	0.010	-11.348	0.000	-0.130	-0.092
swrite	-0.0472	0.011	-4.402	0.000	-0.068	-0.026
exec	-0.0663	0.010	-6.402	0.000	-0.087	-0.046
rchar	-0.1011	0.008	-12.984	0.000	-0.116	-0.086
wchar	-0.0467	0.008	-6.181	0.000	-0.061	-0.032
pgfree	-0.0719	0.008	-8.808	0.000	-0.088	-0.056
atch	0.0245	0.008	3.056	0.002	0.009	0.040
pflt	-0.3137	0.019	-16.828	0.000	-0.350	-0.277
vflt	-0.1267	0.020	-6.253	0.000	-0.166	-0.087
freemem	-0.0749	0.008	-8.906	0.000	-0.091	-0.058
freeswap	0.3874	0.008	47.611	0.000	0.371	0.403
runqsz	0.1578	0.013	12.132	0.000	0.132	0.183

Omnibus:	1030.895	Durbin-Watson:	2.011
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2077.681
Skew:	-1.078	Prob(JB):	0.00
Kurtosis:	5.012	Cond. No.	9.78

Figure 12 OLS Summary 4

- We calculated the VIF (Variance Inflation Factor) for identifying important variables and eliminating the ones that are not significant and have high multicollinearity.

const	2.299597
lread	5.202090
lwrite	4.236100
scall	2.653110
swrite	3.093484
exec	2.961891
rchar	1.668198
wchar	1.557147
pgfree	1.814947
atch	1.736401
pflt	9.617778
vflt	11.397922
freemem	1.951286
freeswap	1.779452
runqsz	1.151110

Figure 13 ols_model4 VIF

- VIF for pflt and vflt seems to be high. So we proceeded with removing the vflt to check the p value and VIF values again.
- There seems to be no major improvement in the R Square and RMSE values after removing the not so significant variables.

Model 6 - Statsmodel method

We then built a model using stats model and OLS method, we see that the Adjusted Rsquare is equal to the Rsquare value which is 0.79.

- On dropping "vflt", R-squared & Adj R-squared remains almost the same without any major difference.

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.791
Model:	OLS	Adj. R-squared:	0.790
Method:	Least Squares	F-statistic:	1664.
Date:	Sun, 12 May 2024	Prob (F-statistic):	0.00
Time:	16:31:09	Log-Likelihood:	-3673.8
No. Observations:	5734	AIC:	7376.
Df Residuals:	5720	BIC:	7469.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0854	0.009	-9.276	0.000	-0.103	-0.067
lread	-0.1252	0.014	-9.044	0.000	-0.152	-0.098
lwrite	0.0638	0.012	5.109	0.000	0.039	0.088
scall	-0.1125	0.010	-11.435	0.000	-0.132	-0.093
swrite	-0.0581	0.011	-5.468	0.000	-0.079	-0.037
exec	-0.0801	0.010	-7.891	0.000	-0.100	-0.060
rchar	-0.1086	0.008	-14.057	0.000	-0.124	-0.093
wchar	-0.0405	0.008	-5.390	0.000	-0.055	-0.026
pgfree	-0.0797	0.008	-9.839	0.000	-0.096	-0.064
atch	0.0220	0.008	2.748	0.006	0.006	0.038
pflt	-0.4073	0.011	-36.540	0.000	-0.429	-0.385
freemem	-0.0771	0.008	-9.141	0.000	-0.094	-0.061
freeswap	0.3961	0.008	49.265	0.000	0.380	0.412
runqsz	0.1560	0.013	11.958	0.000	0.130	0.182

Omnibus:	941.691	Durbin-Watson:	2.007
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1831.070
Skew:	-1.008	Prob(JB):	0.00
Kurtosis:	4.897	Cond. No.	6.39

Figure 14 OLS Summary 5

- RMSE on training data is 0.459 & RMSE on test data is 0.484.

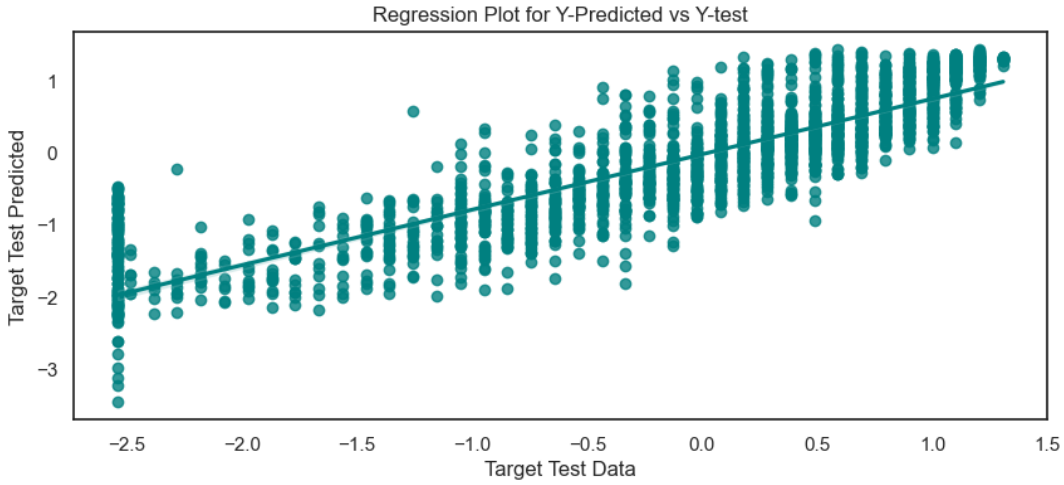


Figure 15 Final Regression Plot for Y-Predicted vs Y-test

- VIF for all the features is less than 5 except lread Figure 16. It means that we have reduced strong collinearity in the model. Also vif is based on business decisions
- There are no features with p-value greater than 0.05

- There seems to be no major improvement in the R Square and RMSE values after removing the not so significant variables.
- It would be better to go with the SKlearn model for prediction and Statsmodel model for interpretation and understand which variables are playing a major role in the model.

const	2.299429
lread	5.132697
lwrite	4.210329
scall	2.652093
swrite	3.012500
exec	2.827150
rchar	1.629358
wchar	1.530348
pgfree	1.773083
atch	1.732355
pflt	3.415993
freemem	1.947978
freeswap	1.726586
runqsz	1.150556

Figure 16 Final VIF Values

Testing the Assumptions of Linear Regression

- For Linear Regression, we need to check if the following assumptions hold:-
  - Linearity
  - Independence
  - Homoscedasticity
  - Normality of error terms
  - No strong Multicollinearity
- First we plotted a Fitted and Residual plot and from the plot we could not find any pattern. So the assumption of Linearity and Independence are satisfied

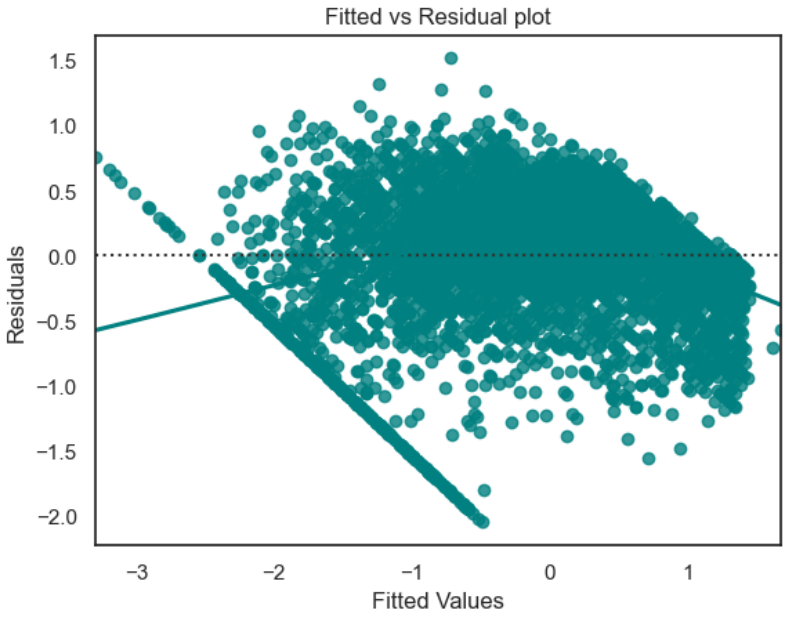


Figure 17 Fitted vs Residual Plot

- For Normality we performed shapiro test and Since p-value < 0.05, the residuals are not normal as per shapiro test.

- For Homoscedasticity, we used het_goldfeldquandt method and Since p-value > 0.05 we can say that the residuals are not homoscedastic.

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.

Solution:

Linear Equation from sklearn:

usr = -0.09592023749931163+ -0.0846083*(lread)+ 0.0475896*(lwrite)+ -0.1041308*(scall)+ 0.0220646*(sread)+ -0.0575855*(swrite)+ -0.0049898*(fork)+ -0.0629167*(exec)+ -0.094862*(rchar)+ -0.0370017*(wchar)+ -0.1077935*(pgout)+ 0.0048442*(ppgout)+ 0.0288333*(pgfree)+ 0.0*(pgscan)+ 0.0374168*(atch)+ -0.0002233*(pgin)+ -0.0640797*(ppgin)+ -0.3421722*(pflt)+ -0.0982258*(vflt)+ -0.0786642*(freemem)+ 0.396291*(freeswap)+ 0.1803818*(runqsz)

Linear Equation from statsmodel:

usr = -0.08540283890881964 + -0.1251862*(lread) + 0.0637883*(lwrite) + -0.1124736*(scall) + -0.0580753*(swrite) + -0.0801422*(exec) + -0.1085651*(rchar) + -0.040479*(wchar) + -0.0796952*(pgfree) + 0.0220477*(atch) + -0.4073339*(pflt) + -0.0770556*(freemem) + 0.3961317*(freeswap) + 0.1560367*(runqsz)

- Monitoring memory (freemem) and swap space (freeswap) helps prevent system crashes.
- Active user presence (usr metric) is crucial for maintaining system responsiveness.
- Linear regression models suggest 'runqsz' and 'fork' as influential factors on 'usr'.
- Monitor system call counts and I/O operations for performance optimization.
- 'lread', 'lwrite', 'scall', 'sread', 'swrite', 'rchar', 'wchar', 'pgout', 'ppgout', 'pgscan', 'freemem', 'freeswap': These coefficients have negligible values close to zero, indicating that they have little impact on the 'usr' metric.
- 'fork', 'exec', 'pgfree', 'pgin', 'ppgin', 'pflt', 'vflt': These coefficients have small values indicating a relatively small impact on the 'usr' metric.
- 'atch': This coefficient has a larger positive value, suggesting that an increase in 'atch' leads to a corresponding increase in the 'usr' metric.
- 'runqsz': This coefficient has a relatively large positive value, indicating that the 'usr' metric is positively influenced by the 'runqsz' (process run queue size) variable.

Recommendations:

- Metrics related to memory (freemem) and swap space (freeswap) indicate available memory and swap space on the system. Monitoring these values can be critical for avoiding system crashes due to resource exhaustion. The presence of non-zero values in these columns suggests that the system has not exhausted its resources in most cases..
- Monitor system call counts, system read/write operations, and I/O operations closely to identify and address performance bottlenecks promptly. Optimize resource allocation based on patterns

of process creation and execution. Keep a close watch on memory and swap space usage to prevent system crashes.

Problem 2

Context

As a statistician at the Republic of Indonesia Ministry of Health, I am analyzing data from a Contraceptive Prevalence Survey encompassing information from 1473 married females. The survey focuses on women who were not pregnant or uncertain of their pregnancy status during the survey, aiming to understand their contraceptive choices based on demographic and socio-economic attributes.

Objective

The objective is to predict whether married women in Indonesia will opt for a contraceptive method based on a comprehensive analysis of their demographic and socio-economic attributes. By leveraging machine learning techniques, the goal is to develop a predictive model that accurately classifies contraceptive usage, providing valuable insights to inform policy decisions and interventions regarding family planning and reproductive health.

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

Solution:

Develop a predictive model to classify contraceptive usage among married women in Indonesia based on demographic and socio-economic attributes. Utilizing data from a Contraceptive Prevalence Survey of 1473 married females, the goal is to provide insights into contraceptive behaviour to inform policy decisions and improve family planning initiatives.

- To begin with, we just see the top 5 rows of the data using the head() method.

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
0	24	Primary	Secondary	3	Scientology	No	2	High	Exposed	No
1	45	Uneducated	Secondary	10	Scientology	No	3	Very High	Exposed	No
2	43	Primary	Secondary	7	Scientology	No	3	Very High	Exposed	No
3	42	Secondary	Primary	9	Scientology	No	3	High	Exposed	No
4	36	Secondary	Secondary	8	Scientology	No	3	Low	Exposed	No

Table 9 Sample Contraceptive data

- We used shape attribute of pandas library [df2.shape] to determine the number of rows and columns of the dataframe df,

Shape	Description
1473, 10	We have 1473 rows and 10 columns in the dataset

Table 10 Shape of Contraceptive Dataset

- To check the types of data, we used info() method of pandas library to list the basic information of the data such as rows count, columns count and datatype of the columns.

Column	Non-Null Count	Data Type
Wife_age	1402 non-null	float64
Wife_ education	1473 non-null	object
Husband_education	1473 non-null	object
No_of_children_born	1452 non-null	float64
Wife_religion	1473 non-null	object
Wife_Working	1473 non-null	object
Husband_Occupation	1473 non-null	int64
Standard_of_living_index	1473 non-null	object
Media_exposure	1473 non-null	object
Contraceptive_method_used	1473 non-null	object

Table 11 Basic Information of Contraceptive Dataset

- From the above table, We have **1** int, **2** float and **7** object data types in the dataset. Also we can see that there are **71** null rows in the Wife_age and **21** null rows in the No_of_children_born columns.
- To check if there are any duplicate columns, we used df2.duplicated().sum(). We can see that there are 80 duplicate rows in the dataset. We will drop the duplicates in the pre-processing step
- Once the above is done, we found the statistical summary of the dataset using describe method()

index	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Wife_age	1402	NaN	NaN	NaN	32.61	8.275	16	26	32	39	49
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	NaN	NaN	NaN	3.254	2.365	0	1	3	4	16
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	NaN	NaN	NaN	2.138	0.865	1	1	2	3	4
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

Table 12 Statistical Summary of Contraceptive Dataset

- Before proceeding with the Univariate and Bivariate analysis, We will impute the missing values in Wife_age and No_of_children_born with median values as the data is numerical and skewed. We can go with mean in case of normally distributed columns.
- After imputing missing values we can see that there are no missing values in the rchar and wchar columns and we verified this using isnull() method.

Index	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

Table 13 Missing Value Check after Imputing

- First, we selected the numerical and categorical columns from the dataframe and assigned them to the variables using select_dtypes(). We then proceeded with the univariate and bivariate analysis.

Variable Name	Columns
Numerical Cols	Wife_age No_of_children_born, Husband_Occupation
Categorical Cols	Wife_ education, Husband_education, Wife_religion, Wife_Working, Standard_of_living_index, Media_exposure, Contraceptive_method_used

Table 14 Numerical & Categorical Columns of Contraceptive Dataset

- After separating the numerical and categorical columns, we created boxplots for each of the numerical columns using seaborn library

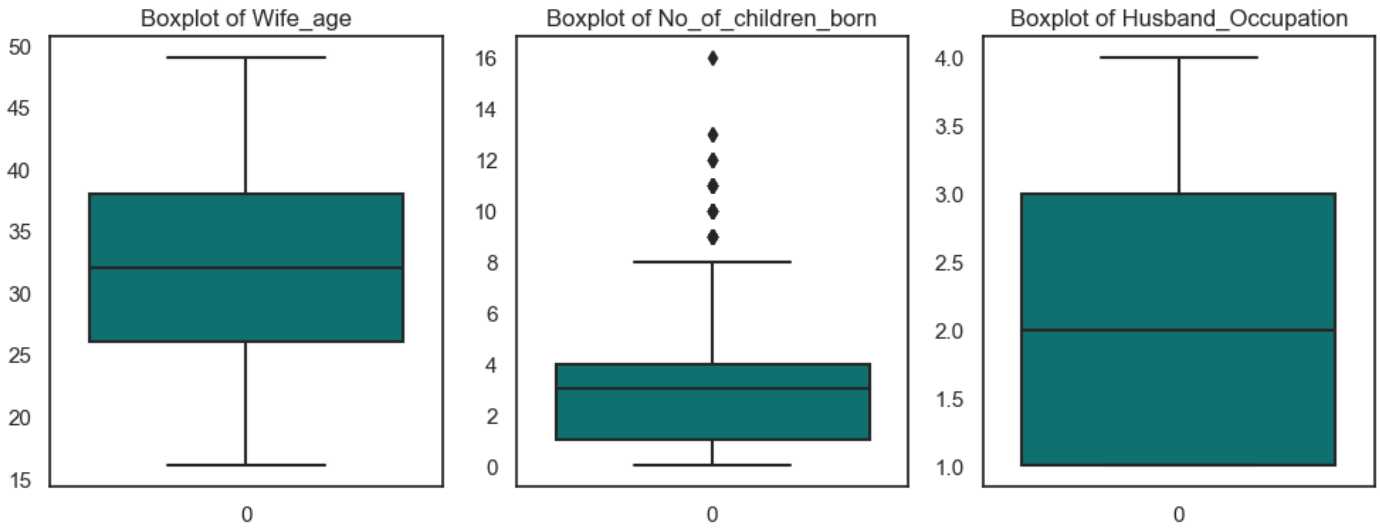


Figure 18 Box Plot of Numerical Columns

- Then we plotted a histogram for the numerical columns followed by the distribution of categorical columns.

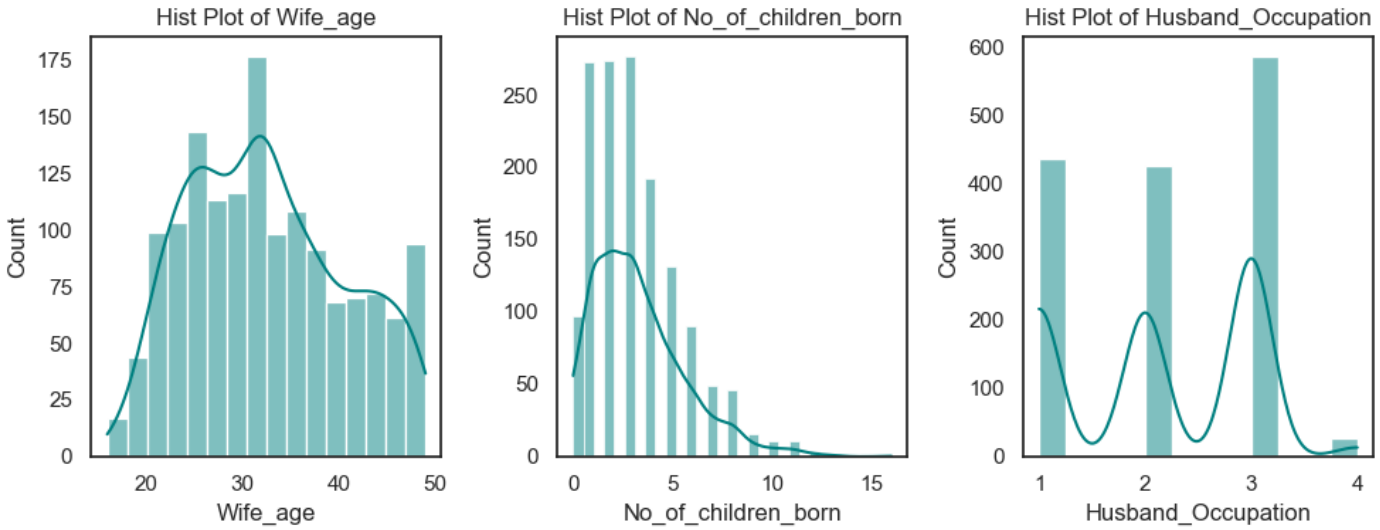


Figure 19 Hist Plot of Numerical Columns

- From the above box plots we can clearly see that there are outliers in No_of_children_born variable. But these need not to be outliers. We can check this further in the pre-processing

stage. Also, we can see that the data is mostly right skewed. Usr column is left skewed. The histogram of freeswap shows that the data is somewhat bimodal.

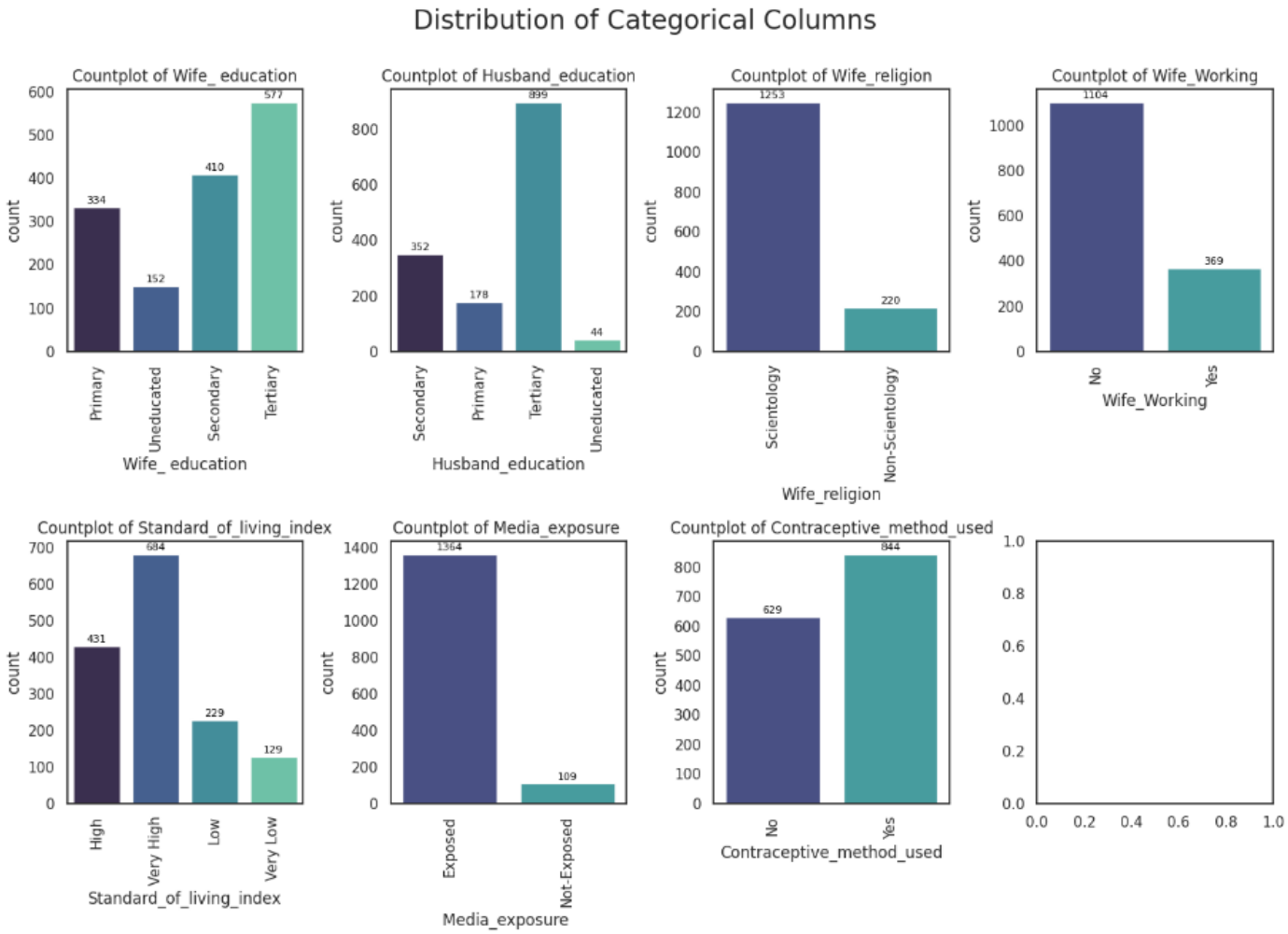


Figure 20 Count Plot of Categorical Columns

- Once the univariate analysis is done, we then proceeded with the bivariate analysis. This step involves examining how pairs of variables interact with each other to uncover relationships within the data.
- First, we plotted a pairgrid which shows us the relationship between numerical variables facilitating the identification of patterns and potential correlations.
- Then we plotted box plots of each categorical variable with all the numerical variables.
- By generating box plots for each categorical variable against all numerical variables, we were able to discern how the distribution of numerical data varies across different categories. This analysis shed light on the nuanced associations present within the dataset, offering a comprehensive understanding of the underlying patterns.
- Then, we further explored the relationships by plotting a heatmap showing the correlation between the variables.
- The heatmap provides a visual representation of the strength and direction of relationships between variables, aiding in identifying potential patterns or dependencies.

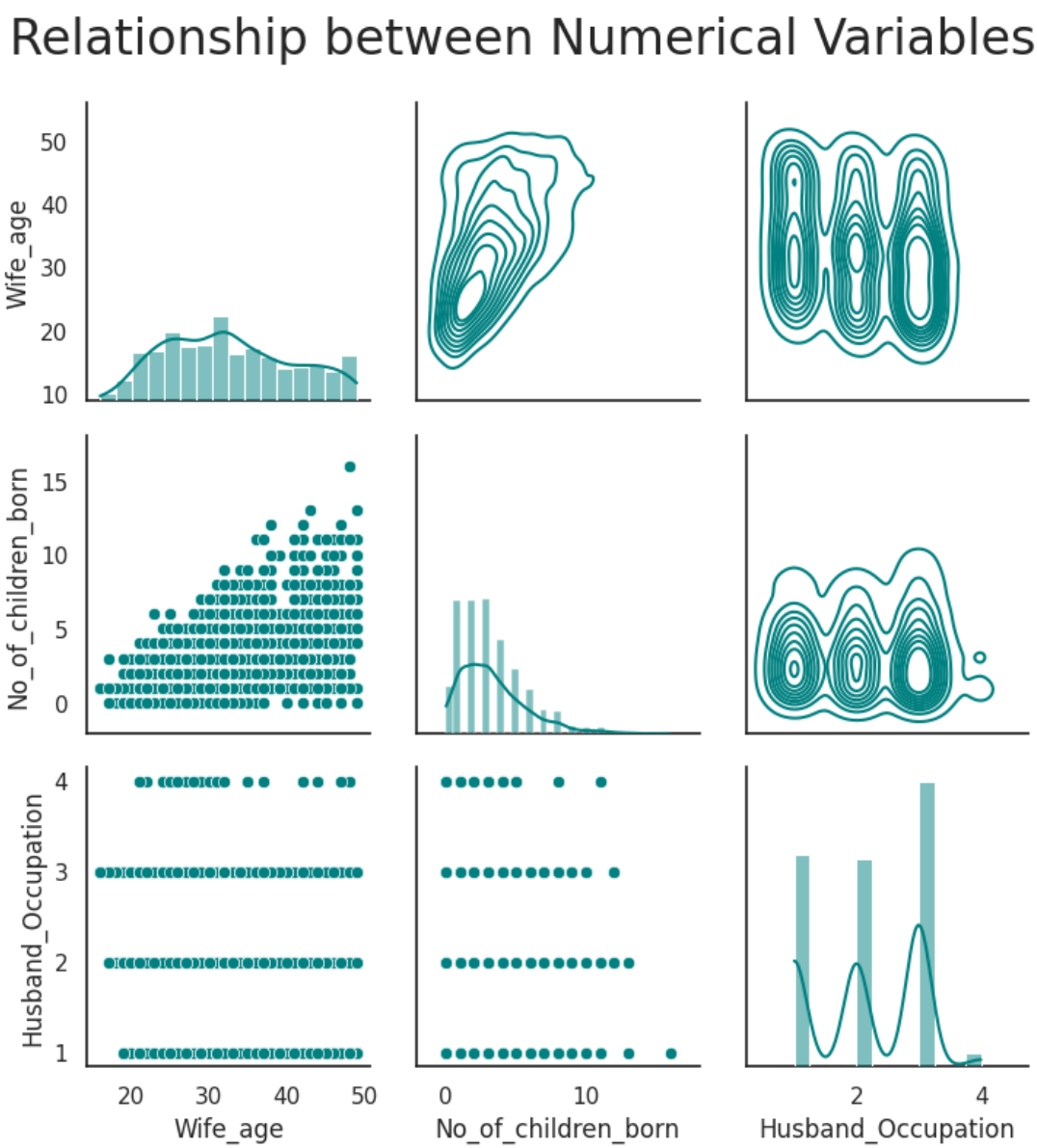


Figure 21 Pairgrid showing relationship between numerical variables

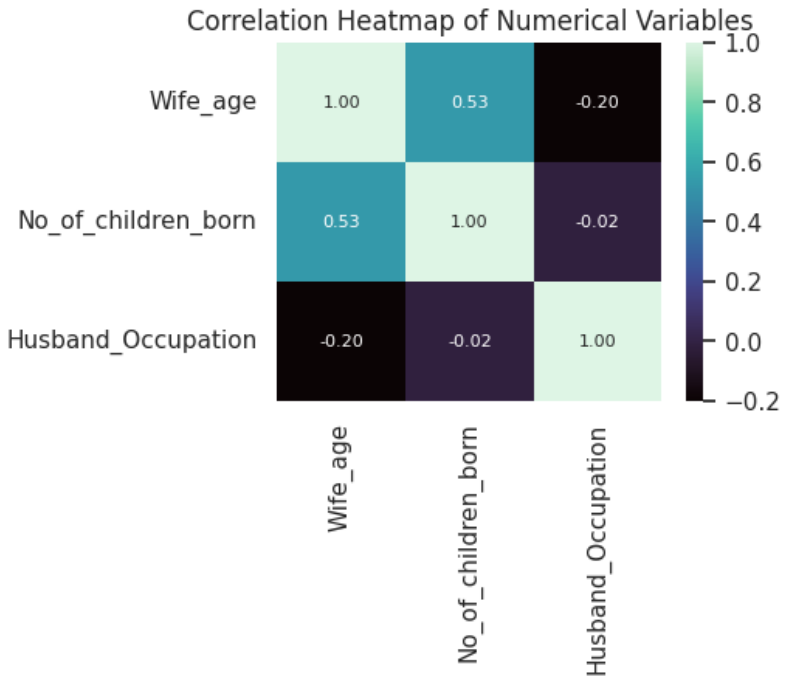


Figure 22 Correlation Heatmap of Numerical Variables

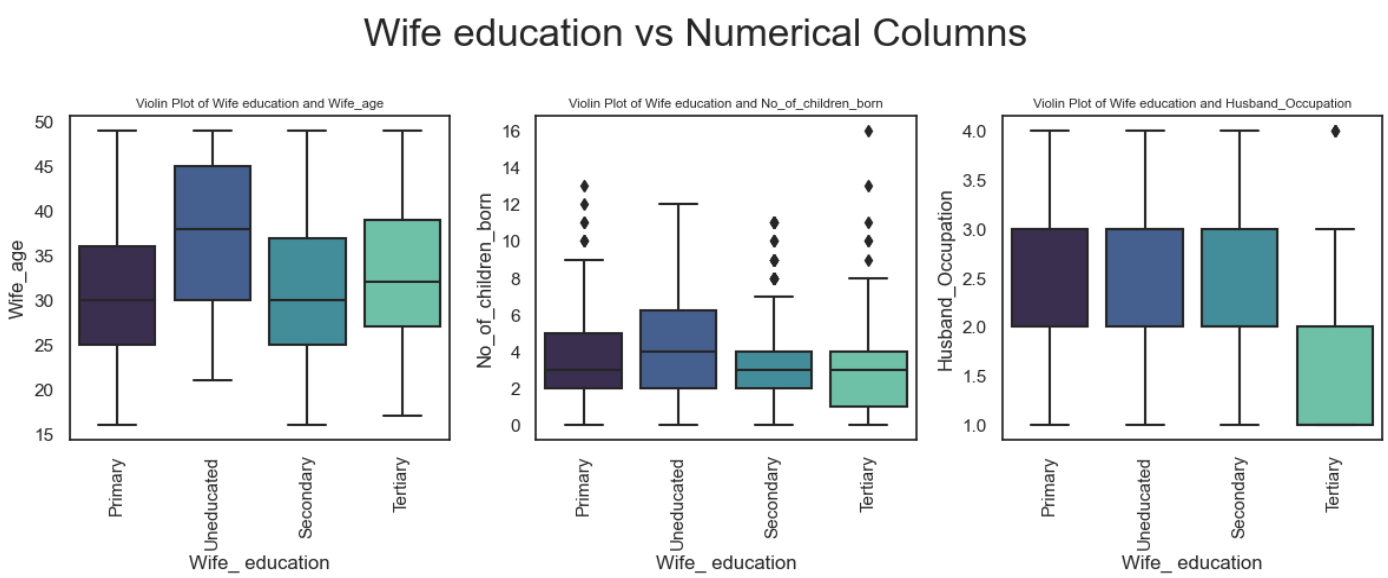


Figure 23 Wife Education vs Numerical Columns

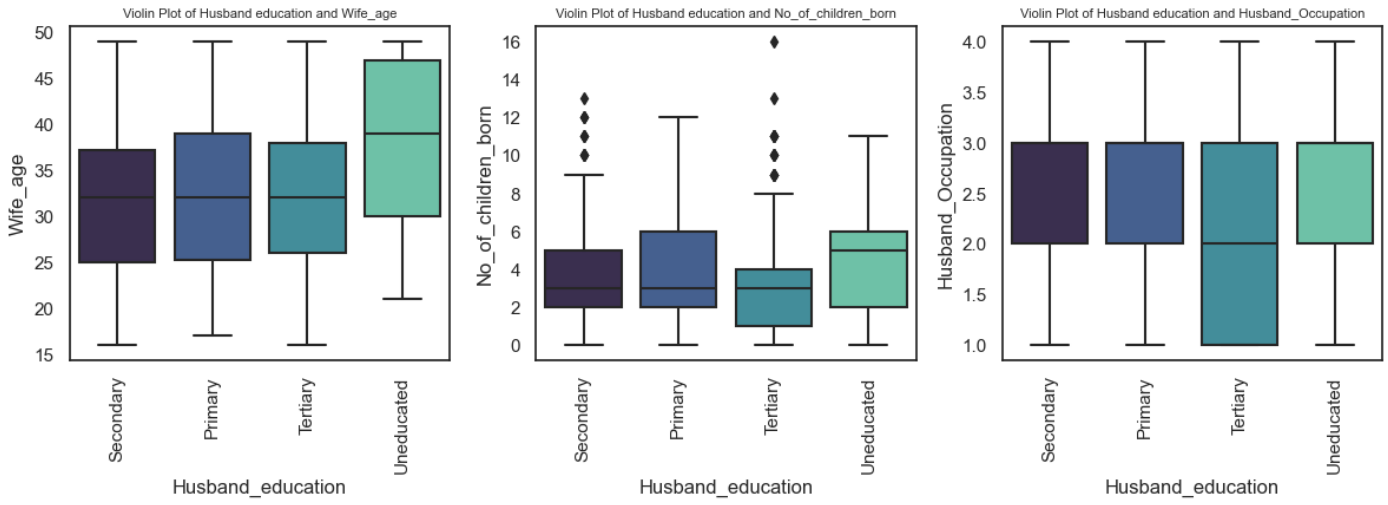


Figure 24 Husband Education vs Numerical Columns

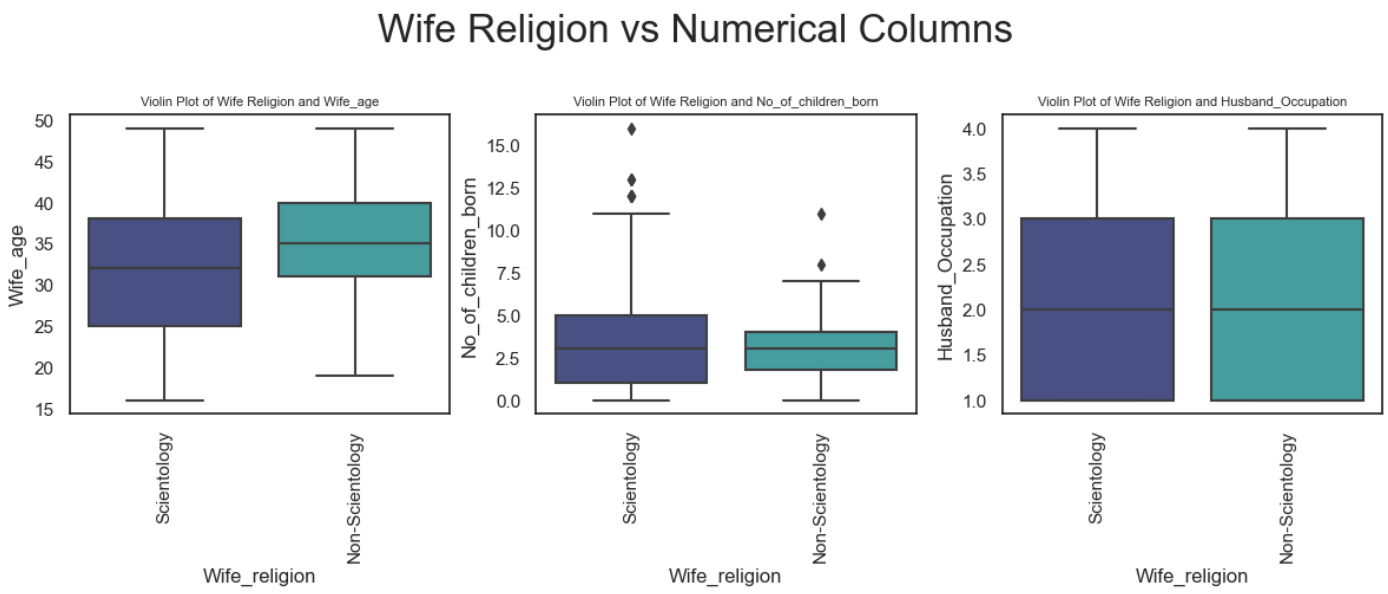


Figure 25 Wife Religion vs Numerical Columns

Wife Working vs Numerical Columns

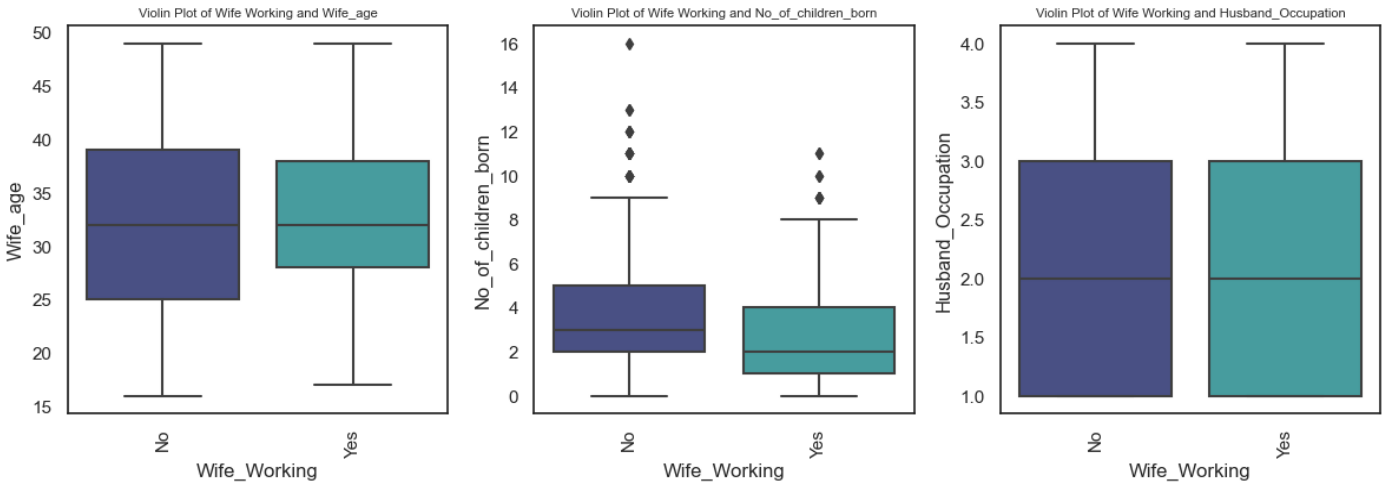


Figure 26 Wife Working vs Numerical Columns

Standard of living index vs Numerical Columns

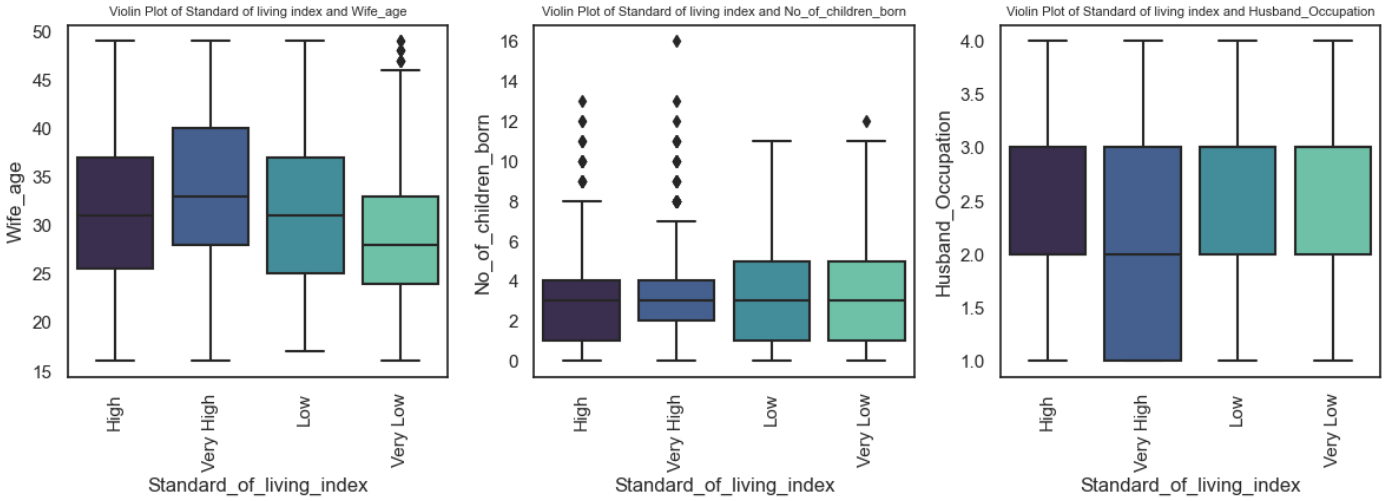


Figure 27 Standard of Living Index vs Numerical Columns

Media Exposure vs Numerical Columns

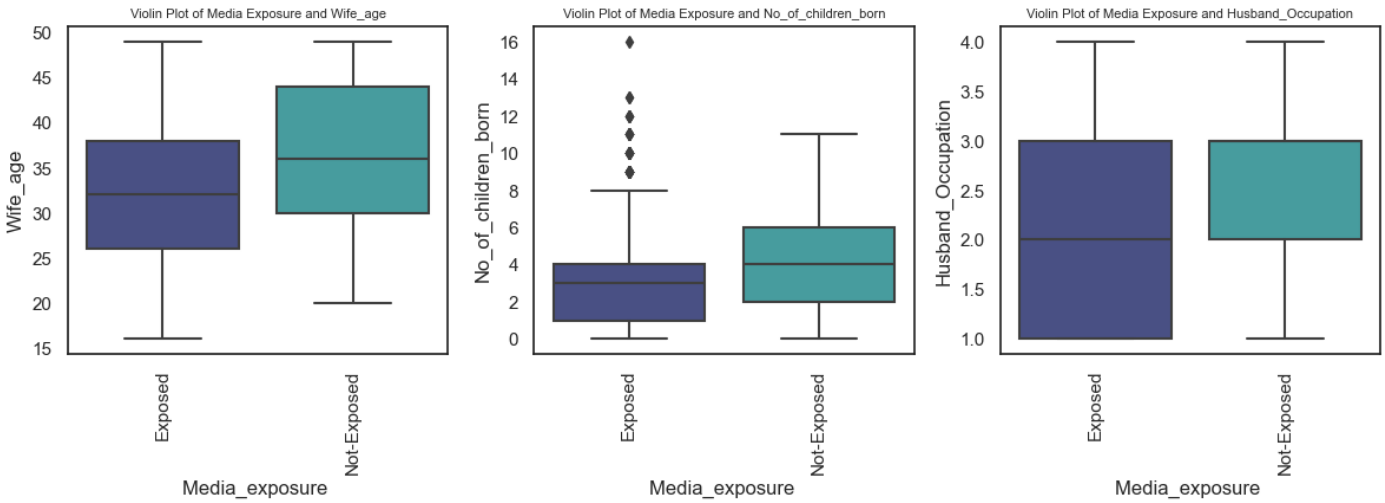


Figure 28 Media Exposure vs Numerical Columns

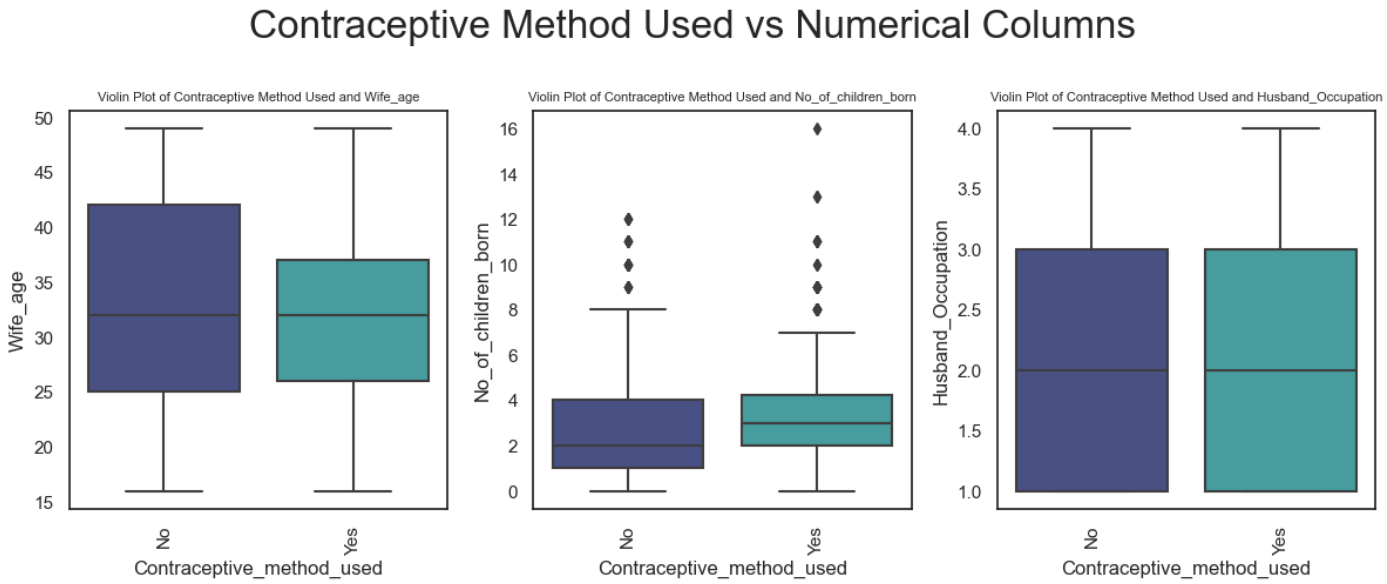


Figure 29 Contraceptive Method vs Numerical Columns

Observations:

- From the 5-point summary of the object type variables, we can see that Tertiary is the most frequent education level of both Husband and Wife.
- Scientology is the most frequent religion that is followed by the women and majority of them are not working.
- The Standard of living index is very high amongst the people and majority of them are exposed to media. This means that the people might be from a city or an urban area.
- Majority of the women have used a contraceptive method.
- Majority of the people had 1 or 2 children but a few have more than 10 children as well.
- The pairgrid does not indicate any major trend/correlation between the variables.

Problem 2 - Data Preprocessing

Prepare the data for modelling: - Missing Value Treatment (if needed) - Outlier Detection (treat, if needed) - Feature Engineering - Encode the data - Train-test split.

Solution:

- After treating missing values using the median value, our dataset is now devoid of any missing values, ensuring the integrity of our data for further analysis. Refer Table 13
- From Figure 18, we can see outliers in No_of_children_born columns. These values need not to be outliers. Rather these can be true value and here there is no huge difference between the min and max values. So, we decided not to treat the outliers.
- We found that there are 80 duplicate rows in our data. Now we will remove the duplicate rows from our data.

```
Index([  79,  167,  224,  270,  299,  394,  414,  462,  492,  528,  576,  585,
        586,  622,  627,  646,  655,  682,  688,  694,  717,  720,  721,  738,
        755,  769,  790,  799,  806,  833,  852,  872,  877,  916,  919,  923,
        961,  968,  970,  971,  974,  981,  982,  984,  986,  994, 1081, 1106,
       1134, 1160, 1187, 1193, 1199, 1206, 1215, 1217, 1220, 1221, 1224, 1228,
       1238, 1239, 1246, 1260, 1263, 1264, 1268, 1270, 1278, 1280, 1285, 1318,
       1330, 1349, 1355, 1367, 1384, 1387, 1389, 1423, 1431, 1440, 1447, 1467,
       1469],
```

Figure 30 Index of Duplicate Rows

- Once the duplicates were removed, we can see there 1393 rows in our data.
- We then proceeded with label encoding of the object columns. Since, our target variable is binary, we coded Contraceptive used No - 0 and Yes - 1

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
0	24	0	1	3	1	0	2	0	0	0
1	45	3	1	10	1	0	3	2	0	0
2	43	0	1	7	1	0	3	2	0	0
3	42	1	0	9	1	0	3	0	0	0
4	36	1	1	8	1	0	3	1	0	0

Table 15 Contraceptive Data after Label Encoding

- We splitted the data into train and test using the train_test_split method from the sklearn library. For all the models, we splitted the data into training and test in the percentage 70:30.

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 hyper parameters 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

Solution:

Model 1 – Logistic Regression Model

Logistic regression is a statistical technique used for binary classification tasks, where the goal is to predict the probability of a binary outcome based on input features. It models the relationship between the independent variables and the log-odds of the dependent variable using a logistic (sigmoid) function, which transforms the output into a probability between 0 and 1. Logistic regression is valued for its simplicity and interpretability.

- We trained the logistic regression model using the train data.
- Once the model is trained, we can assess the model's performance using Classification report and Confusion Matrix for both the training and test datasets.
- Before that we need to understand what is a confusion matrix, classification report and what can be interpreted from AUC-ROC curve

Confusion Matrix:

- A confusion matrix is a table that summarizes the performance of a classification model on a set of test data for which the true values are known.
- From the confusion matrix, we can interpret the following:
  - True Positives (TP): The number of correctly predicted positive instances.
  - True Negatives (TN): The number of correctly predicted negative instances.
  - False Positives (FP): The number of negative instances incorrectly predicted as positive.
  - False Negatives (FN): The number of positive instances incorrectly predicted as negative.
- The confusion matrix provides a clear understanding of the model's performance in terms of correct and incorrect predictions for each class.

Classification Report:

- A classification report provides a comprehensive summary of various evaluation metrics for a classification model, including precision, recall, F1-score, and support, for each class.
- From the classification report, we can interpret the following:
    - Precision: The proportion of true positive predictions out of all positive predictions made by the model.
    - Recall: The proportion of true positive predictions out of all actual positive instances.
    - F1-score: The harmonic mean of precision and recall, providing a balance between the two metrics.
    - Support: The number of actual occurrences of each class in the test dataset.
- The classification report provides insights into the model's performance for each class, helping identify potential areas of improvement and understanding the model's behavior across different classes.

AUC-ROC Curve:

- The AUC-ROC (Area Under the Receiver Operating Characteristic Curve) plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) at different threshold settings
- The AUC-ROC curve provides a comprehensive measure of a binary classification model's discriminative power, where a higher AUC value indicates better performance in distinguishing between positive and negative instances.

- Below is the classification report and confusion matrix for the train and test data

Train Data

	precision	recall	f1-score	support
0	0.64	0.45	0.53	423
1	0.66	0.81	0.72	548
accuracy			0.65	971
macro avg	0.65	0.63	0.63	971
weighted avg	0.65	0.65	0.64	971

Test Data

	precision	recall	f1-score	support
0	0.62	0.41	0.50	191
1	0.61	0.79	0.69	226
accuracy			0.62	417
macro avg	0.62	0.60	0.59	417
weighted avg	0.62	0.62	0.60	417

Figure 31 Classification Report - Logistic Regression

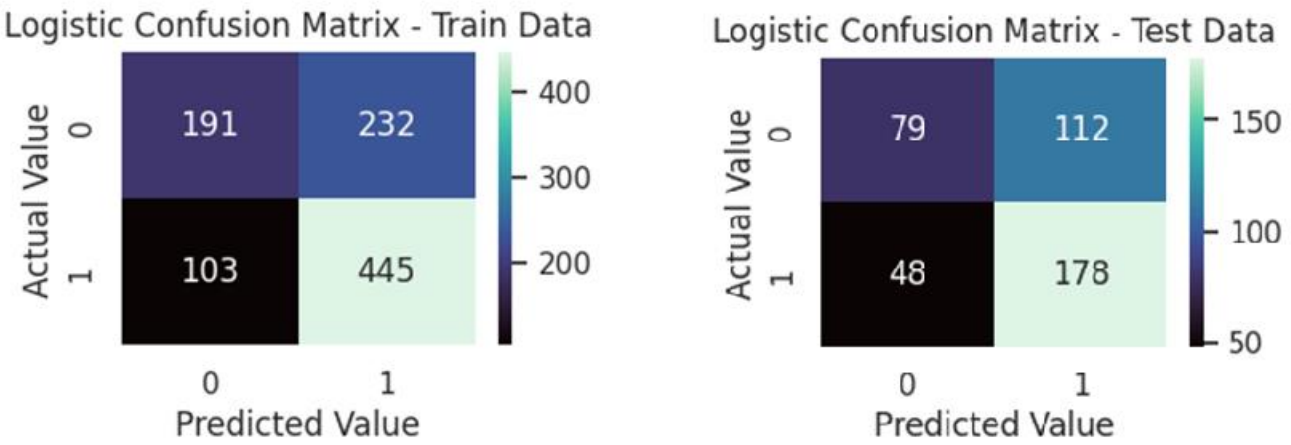


Figure 32 Logistic Regression Confusion Matrix

- Once we plotted the classification report and confusion matrix, we then proceeded with plotting AUC and ROC curve.

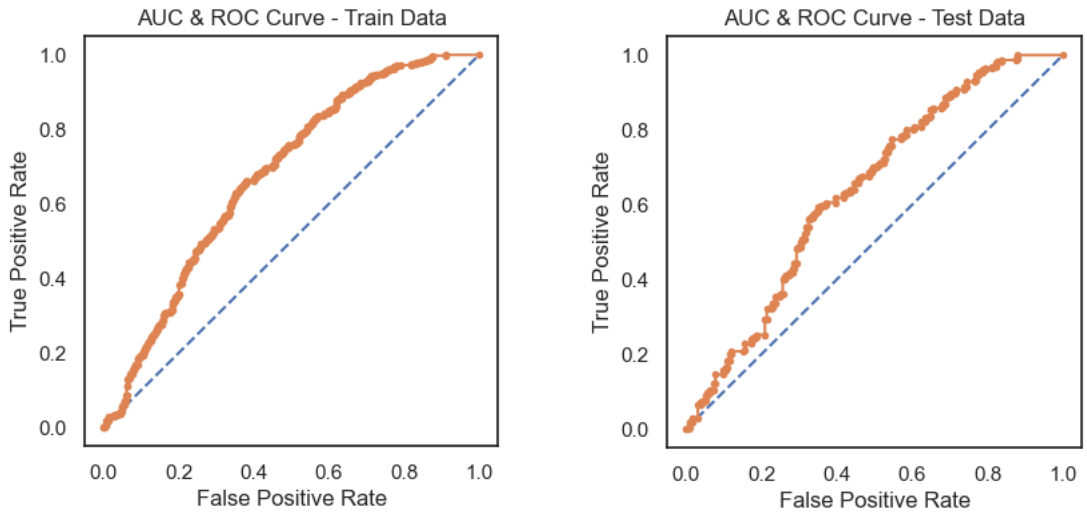


Figure 33 Logistic Regression AUC & ROC Curve

Observations:

- Eventhough the curve is not perfect but the curve is OK ,the AUC(Area under the curve) of the train data model is 67.9%.
- Test curve is similar to the train data curve AUC but slightly vary in the initial points.
- The Area under curve for test data as 64%.
- From the confusion matrix of train data:
  - 191 instances of class 0 were correctly classified as class 0.
  - 232 instances of class 0 were incorrectly classified as class 1.
  - 103 instances of class 1 were incorrectly classified as class 0.
  - 445 instances of class 1 were correctly classified as class 1.
- From the confusion matrix of test data:
  - 79 instances of class 0 were correctly classified as class 0.
  - 112 instances of class 0 were incorrectly classified as class 1.
  - 48 instances of class 1 were incorrectly classified as class 0.
  - 178 instances of class 1 were correctly classified as class 1.

Model 2 – LDA Model

Linear Discriminant Analysis (LDA) is a statistical method for classification and dimensionality reduction, finding linear combinations of features to best separate classes. It maximizes between-class variance while minimizing within-class variance to create discriminant functions for classification. LDA assumes normality of features and equal covariance matrices among classes, making it useful in pattern recognition, machine learning, and statistics.

- We trained the LDA model using the train data.
- Once the model is trained, we assessed the model's performance using Classification report and Confusion Matrix for both the training and test datasets.
- Below is the classification report, confusion matrix, AUC-ROC curve for the train and test data

Train Data

	precision	recall	f1-score	support
0	0.65	0.45	0.53	423
1	0.65	0.81	0.72	548
accuracy			0.65	971
macro avg	0.65	0.63	0.63	971
weighted avg	0.65	0.65	0.64	971

Test Data

	precision	recall	f1-score	support
0	0.63	0.41	0.50	191
1	0.61	0.80	0.69	226
accuracy			0.62	417
macro avg	0.62	0.60	0.59	417
weighted avg	0.62	0.62	0.60	417

Figure 34 Classification Report - LDA

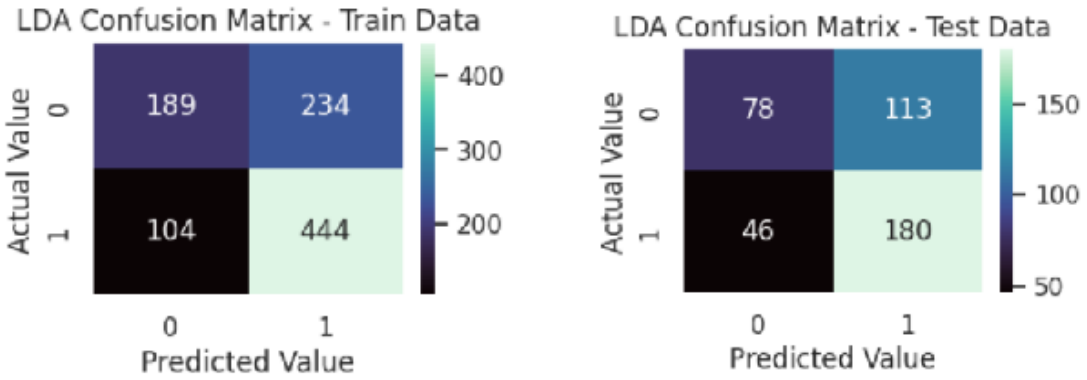


Figure 35 LDA Confusion Matrix

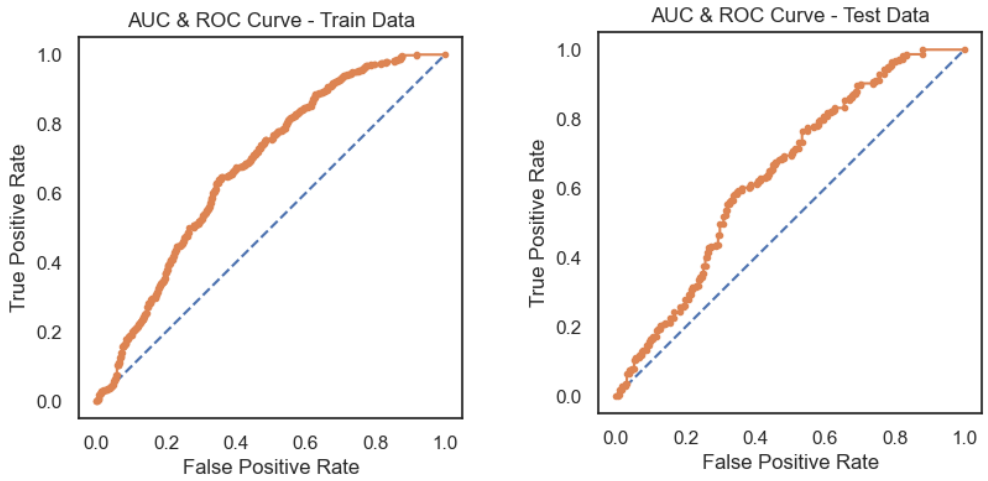


Figure 36 LDA AUC & ROC Curve

Observations:

- The LDA model looks more or less similar to the Logistic Regression models in terms of Accuracy, Recall value for train and test data. The AUC and ROC curves also do not show a significant difference compared to the other models built.
- The Area under curve for train data is 67.8% & test data as 64.2%.
- LDF for the above model will be 2.168+ X1(-0.075)+ X2(0.143) + X3(0.190) + X4(0.303) + X5(-0.771) + X6(-0.031) + X7(-0.050) + X8(-0.038) + X9*(-0.947)
- So from the above equation the following can be summarized
- The coeff of X4 predictor is largest & X9 predictor is smallest in magnitude thus it helps in discriminating the target the best
- From the confusion matrix of train data:
    - 189 instances of class 0 were correctly classified as class 0.
    - 234 instances of class 0 were incorrectly classified as class 1.
    - 104 instances of class 1 were incorrectly classified as class 0.

    - 444 instances of class 1 were correctly classified as class 1.
- From the confusion matrix of test data:
    - 78 instances of class 0 were correctly classified as class 0.
    - 113 instances of class 0 were incorrectly classified as class 1.
    - 46 instances of class 1 were incorrectly classified as class 0.
    - 180 instances of class 1 were correctly classified as class 1.
-

Model 3 – CART Model

The CART (Classification and Regression Trees) method is a versatile algorithm for both classification and regression tasks. It recursively divides the data based on feature values to create a binary tree. Using criteria like Gini impurity for classification or variance reduction for regression, it selects splits to best separate the data. CART produces interpretable trees that predict the target variable from input features, making it popular in machine learning for its simplicity and effectiveness.

- We trained the CART model using the train data.
- Once the model is trained, we can assess the model's performance using Classification report and Confusion Matrix for both the training and test datasets.
- We can see the important features using feature_importances_ property and from that we can see that Wife_age has highest score among the other variables.

	Imp
Wife_age	0.312269
Wife_ education	0.102887
Husband_education	0.064766
No_of_children_born	0.242403
Wife_religion	0.031966
Wife_Working	0.048991
Husband_Occupation	0.064559
Standard_of_living_index	0.111301
Media_exposure	0.020859

Figure 37 Important Features Before Pruning

- We then proceeded with pruning our CART model using GridSearchCV.
- The grid search method helps in finding the optimal combination of hyperparameters for a machine learning model.
- Hyperparameters are settings that govern the learning process of the model and can significantly impact its performance.
- The hyperparameters being tuned include the maximum number of features to consider (max_features), the complexity parameter (ccp_alpha), the maximum depth of the tree (max_depth), the minimum number of samples required to be at a leaf node (min_samples_leaf), and the criterion for splitting (criterion).
- The grid search systematically explores different combinations of these hyperparameters, evaluating each combination using cross-validation.
- The goal is to find the combination that results in the best performance
- Once the optimal parameters were found, we trained a new model with the training data.

	Imp
Wife_age	0.291877
No_of_children_born	0.255335
Standard_of_living_index	0.104884
Wife_ education	0.094903
Husband_Occupation	0.090474
Wife_Working	0.056188
Husband_education	0.054442
Wife_religion	0.030597
Media_exposure	0.021301

Figure 38 Important Features After Pruning

Train

	precision	recall	f1-score	support
0	0.91	0.91	0.91	423
1	0.93	0.93	0.93	548
accuracy			0.92	971
macro avg	0.92	0.92	0.92	971
weighted avg	0.92	0.92	0.92	971

Test

	precision	recall	f1-score	support
0	0.59	0.52	0.55	191
1	0.63	0.70	0.66	226
accuracy			0.62	417
macro avg	0.61	0.61	0.61	417
weighted avg	0.61	0.62	0.61	417

Figure 39 Classification Report - CART

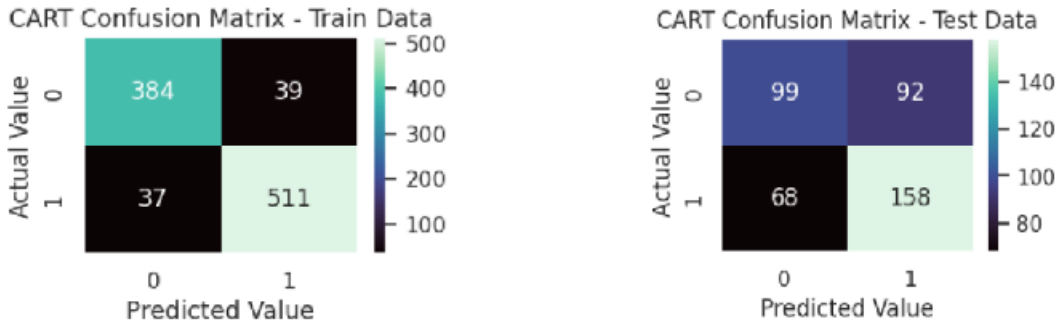


Figure 40 CART Confusion Matrix

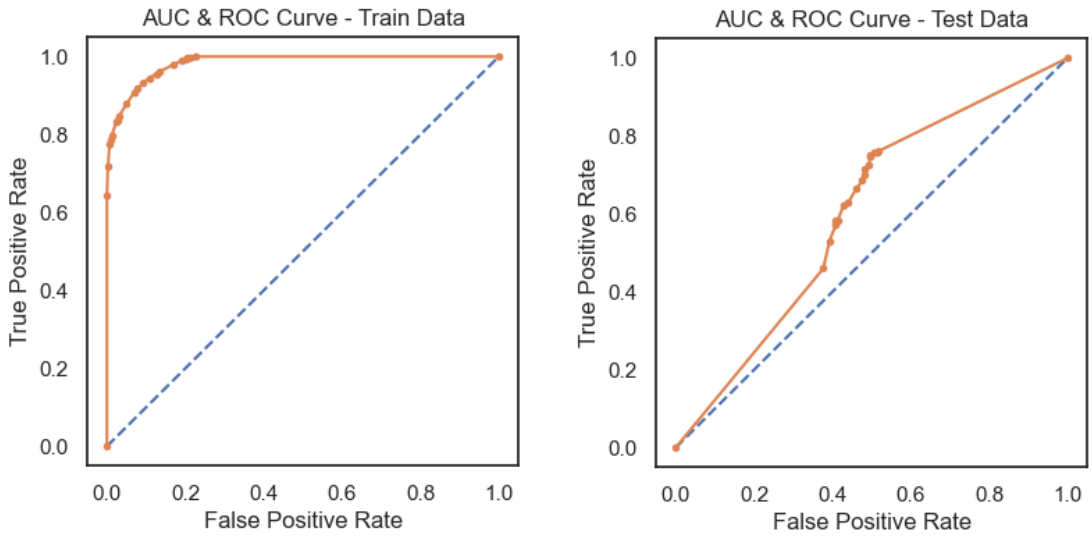


Figure 41 CART AUC & ROC Curve

- The Area under curve for train data is 98.3% & test data as 60.1%. We can see that our model performs well on train data and fails to perform on test data. We call it as overfitting.

- From the confusion matrix of train data:
  - 384 instances of class 0 were correctly classified as class 0.
  - 39 instances of class 0 were incorrectly classified as class 1.
  - 37 instances of class 1 were incorrectly classified as class 0.
  - 511 instances of class 1 were correctly classified as class 1.
- From the confusion matrix of test data:
  - 99 instances of class 0 were correctly classified as class 0.
  - 92 instances of class 0 were incorrectly classified as class 1.
  - 68 instances of class 1 were incorrectly classified as class 0.
  - 158 instances of class 1 were correctly classified as class 1.

Model	Accuracy		Recall		Precision		AUC	
	Train	Test	Train	Test	Train	Test	Train	Test
Logistic Regression	65	62	81	79	66	61	68	64
LDA	65	62	81	80	65	61	68	64
CART	92	62	93	70	93	63	98	60

Table 16 Model Metrics

Based on the tabulated data Table 16:

- The CART model achieves the highest accuracy on the training set (92%), but its accuracy on the test set is relatively lower (62%).
- LDA and Logistic Regression models have similar accuracy scores on both the training and test sets, with 65% accuracy.
- CART has the highest recall on the training set (93%), but its recall on the test set drops to 62%, indicating potential overfitting.
- LDA and Logistic Regression models have consistent recall scores on both the training and test sets, around 62-62%.
- CART has the highest precision on both the training and test sets, with 93% and 70%, respectively.
- LDA and Logistic Regression models have similar precision scores on both the training and test sets, with precision around 81-68%.
- CART has the highest AUC score on the training set (98%), but its AUC score on the test set drops to 60%.
- LDA and Logistic Regression models have similar AUC scores on both the training and test sets, with AUC around 68-64%.

Observations:

- The Logistic Regression and LDA models seem to generalize better to unseen data compared to the CART model, as indicated by their consistent performance across training and test sets.
- The CART model appears to overfit the training data, as evidenced by the large discrepancy between its performance on the training and test sets.
- While the CART model has the highest accuracy and precision on the training set, its performance on the test set is significantly lower, suggesting that it may not generalize well to new data.

- Overall, the Logistic Regression and LDA models exhibit more stable performance across different evaluation metrics and datasets, making them potentially better choices for this classification task.

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

Solution:

- As per the Logistic Regression model, the wife's education, no. of children born is very important in deciding whether the women will use contraceptive methods or not.
- The CART model also indicates that the wife's education, no. of children born are important. Therefore, these features are highly important.
- Both the models also indicated that the Husband's education is also important, and in real life that makes sense. This feature can influence the wife's decision to use contraceptive methods.

Recommendations:

- It's clear that factors like age, education, socio-economic status, and media exposure greatly influence whether married women use contraceptives.
- Women living in areas with higher living standards tend to use contraceptives more.
- Women aged 25 to 35, especially those with better education, are more likely to use contraceptives.
- The husband's education level is also important in determining whether his wife will use contraceptives.
- It's important to understand why some women without children are still using contraceptives.
- Media exposure is another factor affecting contraceptive use.
- The Ministry of Health in Indonesia can educate women who don't use contraceptives about their benefits and usage.