Image courtesy: https://blog.reffascode.de/tag/machine-learning/

Predictive Modelling Project

Business Report

June 09, 2024

Authored by: Kartik Trivedi

List of Contents

List of Figures

List of Tables

List of Equations

Data Dictionary

Problem 1

Column Name	Column Description	Data Type
lread	Reads (transfers per second) between system memory and user memory	int64
lwrite	writes (transfers per second) between system memory and user memory	int64
scall	Number of systems calls of all types per second	int64
sread	Number of systems read calls per second.	int64
swrite	Number of systems write calls per second.	int64
fork	Number of system fork calls per second.	float64
exec	Number of system exec calls per second.	float64
rchar	Number of characters transferred per second by system read calls	float64
wchar	Number of characters transferred per second by system write calls	float64
pgout	Number of pages out requests per second	float64
ppgout	Number of pages, paged out per second	float64
pgfree	Number of pages per second placed on the free list.	float64
pgscan	Number of pages checked if they can be freed per second	float64
atch	Number of page attaches (satisfying a page fault by reclaiming a page in memory) per second	float64
pgin	Number of pages	float64
ppgin	Number of pages paged in per second	float64
pflt	Number of page faults caused by protection errors (copy	float64
vflt	Number of page faults caused by address translation.	float64
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	object
freemem	Number of memory pages available to user processes.	int64
freeswap	Number of disk blocks available for page swapping.	int64
usr	Portion of time (%) that CPUs run in user mode	int64

Problem 2

Name	Description	Data Type
Wife_age	Wife's age (numerical)	float64
Wife_ education	Wife's education (categorical) 1=uneducated, 2, 3, 4=tertiary	object
Husband_education	Husband's education (categorical) 1=uneducated, 2, 3, 4=tertiary	object
No_of_children_born	Number of children ever born (numerical)	float64
Wife_religion	Wife's religion (binary) Non	object
Wife_working	Wife's now working? (binary) Yes, No	object
Husband_Occupation	Husband's occupation (categorical) 1, 2, 3, 4(random)	int64
Standard_of_living_index	Standard	object
Media_exposure	Media exposure (binary) Good, not good	object
Contraceptive_method_used	Contraceptive method used (class attribute) No, Yes	object

Executive Summary

Problem 1

Background Information

A data scientist has been assigned a project to analyse system activity data and construct a linear equation using various system attributes. This equation aims to predict the system's 'usr' mode based on data collected from a Sun SPARCstation 20/712 with 128 Mbytes of memory, which is stored in the comp-activ database.

Business Objective

Study different system attributes to understand their impact on the system's 'usr' mode and build a linear regression model that could help predict the portion of time (%) that CPU's run-in user mode.

Problem Statement

The objective of this project is to establish a linear equation that could predict portion of time (%) that CPU's run-in user mode represented by column 'usr'. For building this model different system attributes have to be analysed in order to understand their influence on the system's 'usr' mode.

Linear Regression Equation

Ridge Regression Equation:

usr = 89.5666 + (-0.0707 * lread) + (0.0373 * lwrite) + (-0.0003 * scall) + (-0.0020 * sread) + (-0.0106 * swrite) + (-0.2016 * fork) + (-0.4311 * exec) + (-0.0000 * rchar) + (-0.0000 * wchar) + (-0.2632 * pgout) + (0.0675 * ppgout) + (-0.0053 * pgfree) + (-0.0087 * pgscan) + (0.0928 * atch) + (-0.0488 * pgin) + (-0.0137 * ppgin) + (-0.0162 * pflt) + (-0.0095 * vflt) + (0.0001 * freemem) + (0.0000 * freeswap) + (1.7379 * runqsz_Not_CPU_Bound)

Equation 1: Ridge regression equation

Conclusion

1. Final regression equation is: usr = 89.5666 + (-0.0707 * lread) + (0.0373 * lwrite) + (-0.0003 * scall) + (-0.0020 * sread) + (-0.0106 * swrite) + (-0.2016 * fork) + (-0.4311 * exec) + (-0.0000 * rchar) + (-0.0000 * wchar) + (-0.2632 * pgout) + (0.0675 * ppgout) + (-0.0053 * pgfree) + (-0.0087 * pgscan) + (0.0928 * atch) + (-0.0488 * pgin) + (-0.0137 * ppgin) + (-0.0162 * pflt) + (-0.0095 * vflt) + (0.0001 * freemem) + (0.0000 * freeswap) + (1.7379 * runqsz_Not_CPU_Bound)

   Equation 2: Ridge regression equation

2. As per this regression equation we can conclude:
   a. In terms of coefficient value impact of runqsz_Not_CPU_Bound is highest that is if 'runsqz' is 'Not_CPU_Bound' then 'usr' value increases by 1.738 meaning CPUs run 1.738% Portion of time more in user mode provided all other attributes remain constant.
   b. While 'runsqz_Not_CPU_Bound' is a categorical attribute meaning its value can either be 0 or 1, for continuous variables whose values can differ significantly we find that in absolute terms

'exec' has highest impact followed by 'pgout' and their impact is negative meaning provided that other attributes remain constant every unit increase in 'exec' reduces cpu run time in user mode by 0.431% and for 'pgout' this decrease is 0.263%.

c. For continuous attribute which increase cpu run time in user mode, 'atch' has the highest coefficient value of 0.093, that is for every one unit increase in 'atch' if others remain constant then cpu run time in user mode increases by 0.093% For improvement purposes the following recommendation are made

Key Takeaways

While we have discussed the impact of different attributes in predicting 'usr' in terms of the regression equation and coefficient values, it is essential to consider the business context. The scale of values for some variables varies significantly, as highlighted by the statistical summary. For many attributes, the median value is 0, while for a few, it is in the thousands. This range difference indicates that some attributes have higher variance. Despite having lower coefficient values, these attributes can have a significant impact on the equation when multiplied by their respective attribute values as they are highly sensitive. Consequently, they will significantly affect the 'usr' value and play a larger role in determining the percentage of CPU run time in user mode.

In a real-world environment, multiple attributes change together, and the magnitude of change for these sensitive attributes will be significantly higher, thus affecting 'usr' considerably more.

We will examine this effect with the help of a table that outlines the impact each variable has according to the five-point summary.

	Coefficient	min	25%	50%	75%	max
freeswap	0.000004	3.766998	4.161733	4.529277	6.912509	8.964651
runqsz_Not_CPU_Bound	1.737918	0.000000	0.000000	1.737918	1.737918	1.737918
freemem	0.000085	0.004658	0.019139	0.048101	0.166404	0.640804
lwrite	0.037346	0.000000	0.000000	0.037346	0.373462	2.464848
pgout	-0.263236	-0.000000	-0.000000	-0.000000	-0.526471	-2.895591
atch	0.092785	0.000000	0.000000	0.000000	0.000000	0.463926
pgscan	-0.008694	-0.000000	-0.000000	-0.000000	-0.000000	-1.173705
pgfree	-0.005323	-0.000000	-0.000000	-0.000000	-0.026617	-0.372633
ppgout	0.067453	0.000000	0.000000	0.000000	0.269811	2.158484
fork	-0.201552	-0.000000	-0.000000	-0.000000	-0.403105	-1.410867
ppgin	-0.013735	-0.000000	-0.000000	-0.041204	-0.192287	-0.727944
pgin	-0.048839	-0.000000	-0.000000	-0.097678	-0.439551	-1.709366
wchar	-0.000005	-0.007628	-0.117599	-0.237414	-0.539690	-1.842918
sread	-0.001977	-0.011859	-0.171957	-0.330078	-0.555401	-1.120686
exec	-0.431057	-0.000000	-0.000000	-0.431057	-0.862114	-4.741625
lread	-0.070672	-0.000000	-0.141344	-0.494706	-1.413445	-4.805712
scall	-0.000294	-0.032063	-0.300330	-0.606248	-0.985411	-1.990823
rchar	-0.000006	-0.001552	-0.195720	-0.701476	-1.513416	-3.599326
pflt	-0.016207	-0.000000	-0.388962	-1.037231	-2.576872	-5.850633
vflt	-0.009460	-0.000000	-0.425711	-1.135229	-2.383982	-5.619386
swrite	-0.010640	-0.074480	-0.670321	-1.255522	-1.979043	-4.170887
Intercept	89.566590	NaN	NaN	NaN	NaN	NaN

Table 1: Attributes magnitude of change

As expected, 'freeswap', which has the lowest coefficient value in absolute terms, will have the highest values in the regression equation due to its sensitive nature. According to our data, 'freeswap' has a minimum value of 3.766, a maximum of 8.96, and a median of 4.53, meaning it accounts for 3.766% to 8.96% of CPU run time in user mode.

As per the above table, the top three attributes that are pushing up the CPU run time in user mode in median terms are:

1. freeswap
2. runqsz_Not_CPU_Bound
3. freemem

and bringing cpu run time in user mode down in median terms are:

1. swrite

2. vflt
3. pflt

Problem 2

Background Information

The Ministry of Health of the Republic of Indonesia has launched an initiative aimed at gaining a better understanding of contraceptive use among married women and identifying the key factors that influence their decisions. This study will provide valuable insights towards informed policy decisions aimed at improving public health.

Business Context

Study different demographic and socio-economic factor and create a predictive model that could help identify whether married women (who are either not pregnant or are uncertain of their pregnancy status) choose to use a contraceptive method. This insight would help the Ministry of Health of the Republic of Indonesia in making informed policy decisions aimed at improving public health.

Problem Statement

The goal is to identify key demographic and socio-economic factor that could help classify whether married women (who are either not pregnant or are uncertain of their pregnancy status) opt for a contraceptive method of choice. Using this understanding build classification models using different techniques that could help predict whether these women opt for a contraceptive method of choice and identify the best model based on model evaluation metrics.

Important Features

On comparing different classification models on multiple metrics, we identified that logistic regression model is better generalised model, we consider it as the best model.

In logistic regression we would use coefficient values to determine the most important features.

	imp
Wife_education	0.441862
No_of_children_born	0.272439
Standard_of_living_index	0.190277
Husband_education	0.134508
Husband_occupation	0.123461
Wife_age	-0.078586
Wife_working_Yes	-0.168278
Wife_religion_Scientology	-0.397523
Media_exposure_Not-Exposed	-0.508125

Table 2: Feature Importance

Based on the sign of coefficient values we have we can divide them into 2 categories:

1. Positive Coefficients: The value of these features increase the log odds of positive outcome that is being labelled as class 1.

a) For positive coefficients 'Wife_education' has the highest value which means that higher the level of education is amongst wife more likely they are going to use the contraceptive methods.

2. Negative Coefficients: The value of these features increase the log odds of positive outcome decreasing that is being labelled as class 1.

a) For negative coefficients 'Media_exposure_Not-Exposed' has the highest impact meaning women who are not exposed to media are more likely of not using contraceptive methods.

Conclusion

1. Based on the evaluation of various classification techniques for predictive modeling, the logistic regression model demonstrated the best generalization performance on the test data, achieving an accuracy score of 67.7%. This indicates that the model correctly predicts the class labels of the target variable 67.7% of the time.

2. Based on coefficient values of the features, the regression equation is:
log odds (Contraceptive method used) = Intercept + 0.441862Wife_education + 0.272439No_of_children_born + 0.190277Standard_of_living_index + 0.134508Husband_education + 0.123461Husband_occupation - 0.078586Wife_age - 0.168278Wife_working_Yes - 0.397523Wife_religion_Scientology - 0.508125*Media_exposure_Not-Exposed

Equation 3: Logistic regression equation

Where intercept value of model is: [-0.24755177]

Key Takeaways

Based on the model coefficient values most important demographic and socio-economic features for classification are:

1. 'Wife_education' has the highest positive coefficient value of 0.4418 followed by 'No_of_children_born' with value of 0.2724 meaning provided other features remain constant with every increase in level of education amongst wife log-odds value will increase by 0.4418 and with every additional child born log-odds value will increase by 0.2724.
2. 'Media_exposure_Not-Exposed' has the highest negative coefficient value of -0.5081 followed by 'Wife_religion_Scientology' with value of -0.3975 meaning provided other features remain constant if there is no media exposure the log-odds value will decrease by 0.5081 and if 'WIfe_religion' is Scientology the log-odds value will decrease by 0.3975.

In simple terms, if there is no media exposure it is highly unlikely that married women will use contraceptive method and with increase in education levels the probability that a married woman will use contraceptive method also increases.

Problem 1

1.1 Background Information

A data scientist has been assigned a project to analyse system activity data and construct a linear equation using various system attributes. This equation aims to predict the system's 'usr' mode based on data collected from a Sun SPARCstation 20/712 with 128 Mbytes of memory, which is stored in the comp-activ database.

1.2 Business Objective

Study different system attributes to understand their impact on the system's 'usr' mode and build a linear regression model that could help predict the portion of time (%) that CPU's run-in user mode.

1.3 Problem Statement

The objective of this project is to establish a linear equation that could predict portion of time (%) that CPU's run-in user mode represented by column 'usr'. For building this model different system attributes have to be analysed in order to understand their influence on the system's 'usr' mode.

1.4 METHODOLOGY

Import the libraries - Load the data - Check the structure of the data - Check the types of the data – Check for and treat (if needed) missing values - Check the statistical summary - Check for and treat (if needed) data irregularities – Univariate Analysis – Bivariate Analysis – Outlier Treatment – Apply linear regression models – Predict values – Evaluate model – Compare model – Get regression equation – Conclusion

Key Points

1. **Data Collection**: Data was made available using comp-activ database which measures activity of computer systems.
2. **Data Cleaning and Pre-processing:** Dataset was checked for duplicates, missing values, bad data and outliers. Missing values and outliers were found in the dataset, missing values and outliers were treated as per the procedure and data was scaled before applying PCA to treat multi-collinearity in data.
3. **Univariate Analysis:** Individual variables were analyzed using boxplot and histogram to understand distribution, central tendency and variability of variables.
4. **Bivariate Analysis:** All the variables were examined with the aim of gaining deeper insights about the various measures of the activiy.

5. **Visualization Techniques:** In the report we have used histograms, boxplot and count plot for univariate analysis, in bivariate analysis, to understand correlation between numeric variables heatmap and pair plot are used, strip plot is used to understand relationship between categorical and numeric variables.
6. **Tools and Software:** We have carried out the analysis using programming language python on Jupyter notebook. For this analysis Python libraries Numpy, Pandas, Matplotlib, Seaborn, Scipy, Statistics, Scikit-learn, Statsmodel and Math were used.

1.5 Data Overview

1. **Data Description:** Dataset has 8192 rows and 22 columns.

```
shape of the dataset
------------------------------------------------------------------------------

(8192, 22)
```

Table 3: Dataset Shape

2. **Dataset Information:** Of the twenty-two columns in the dataset 1 is object type, 8 are int 64 type and 13 are float type.

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

Table 4: Dataset Information

3. **Missing Value Check:** There are missing values for attributes 'rchar' and 'wchar' which were imputed using median values.

```
missing values
--------------------------------------------------------------------------------

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 5: Missing values information

4. **Duplicate Values:** Data was checked for duplicate values and no duplicates were found

```
checking for duplicates
--------------------------------------------------------------------------------
number of dupliacte rows: 0
```

Table 6: Data Duplicates

5. **Statistical Summary:**

```
statistical summary
--------------------------------------------------------------------------------
```

	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 7: Statistical Summary

6. **Frequency Distribution of Categorical Columns:**

```
value counts for runqsz
--------------------------------------------------------------------------------
runqsz
Not_CPU_Bound    4331
CPU_Bound        3861
Name: count, dtype: int64
```

Table 8: Frequency Distribution of categorical columns

Key observations

1. Based on the first glimpse of data, there are clearly some missing values in the data.

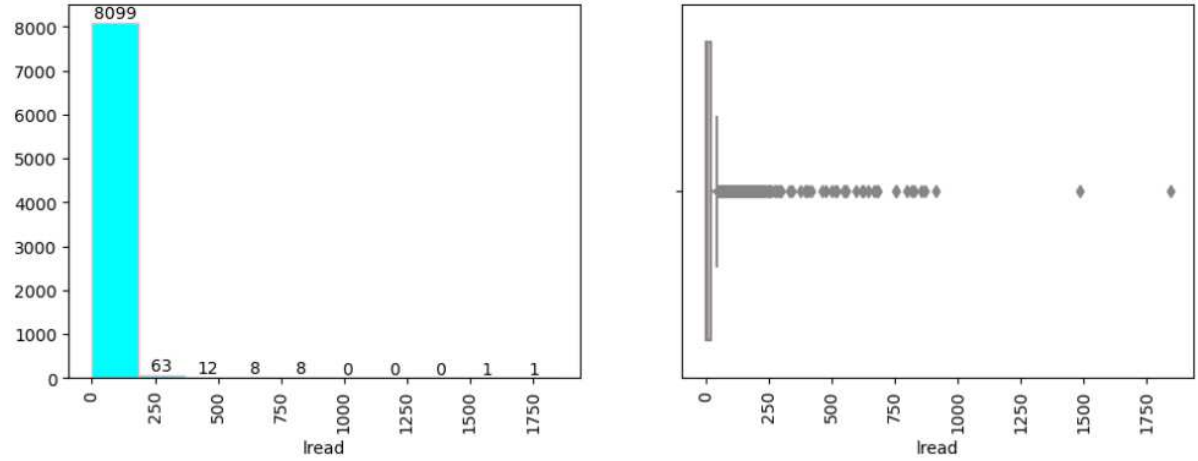
2. Dataset has 8192 rows and 22 columns of which 21 are numeric and 1 is categorical.
3. From the statistical summary we can conclude that there is difference in the scales of numeric data. For some variables there is significant difference in mean and 50% (median) values meaning presence of outliers about which we will understand more during EDA.
4. There are missing values for attributes 'rchar' and 'wchar' which we will impute during pre-processing stage.
5. For categorical attribute 'runqsz' there are only 2 unique values, 4331 records are 'Not_CPU_Bound' and 3861 are 'CPU_Bound'.

1.6 Exploratory Data Analysis

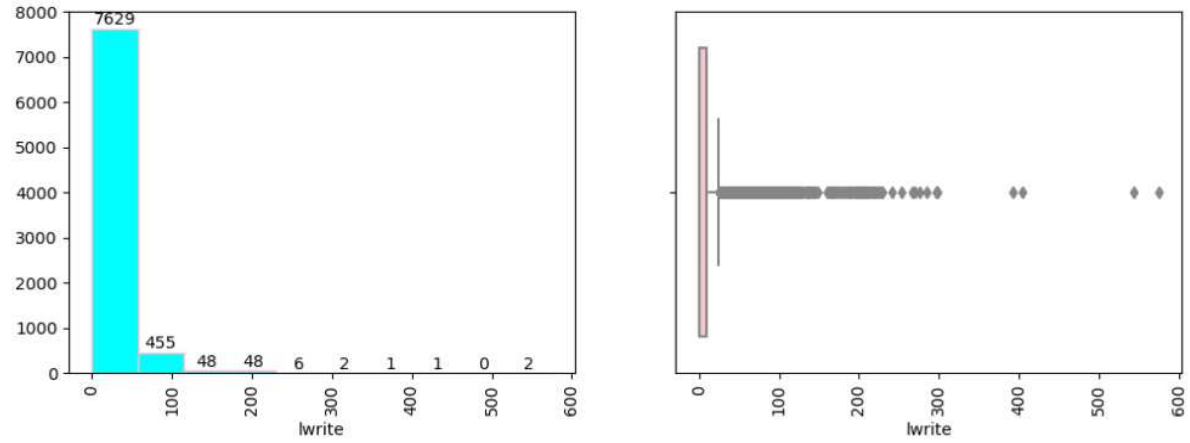
1.6.1 Univariate Analysis

For numeric columns

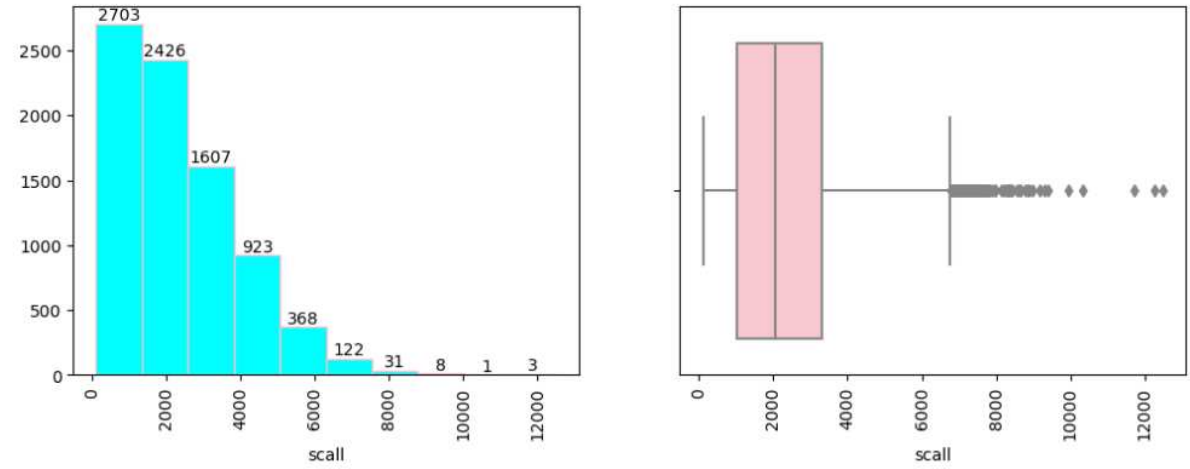
```
Skewness of lread:  13.897852242774922
Distribution of lread
--------------------------------------------------------------------------------
```



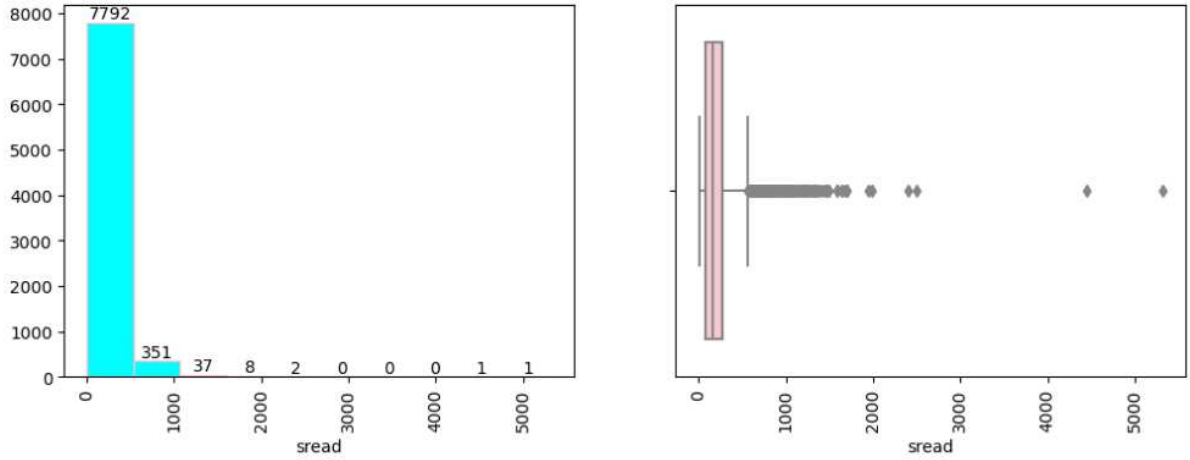

```
Skewness of lwrite:  5.27764452621306
Distribution of lwrite
--------------------------------------------------------------------------------
```



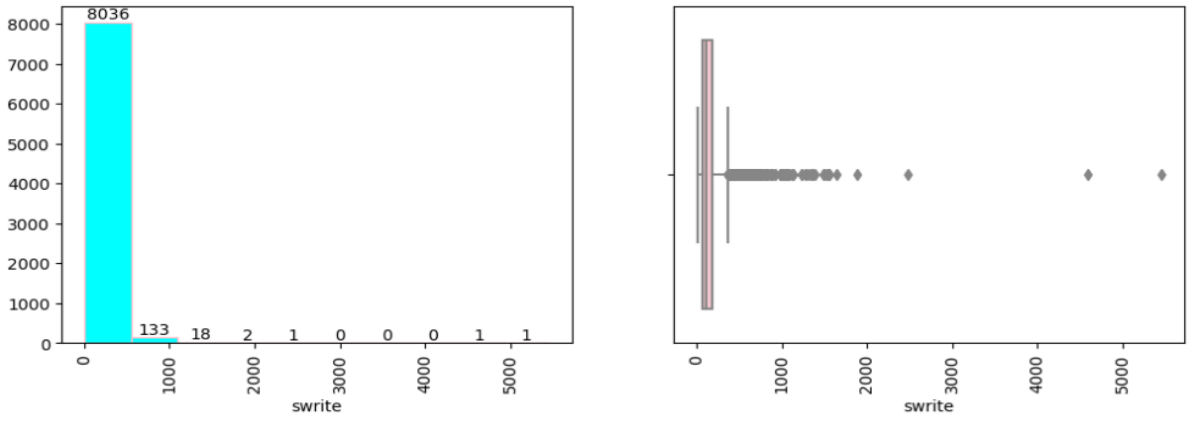
```
Skewness of scall:  0.9025312213201333
Distribution of scall
--------------------------------------------------------------------------------
```



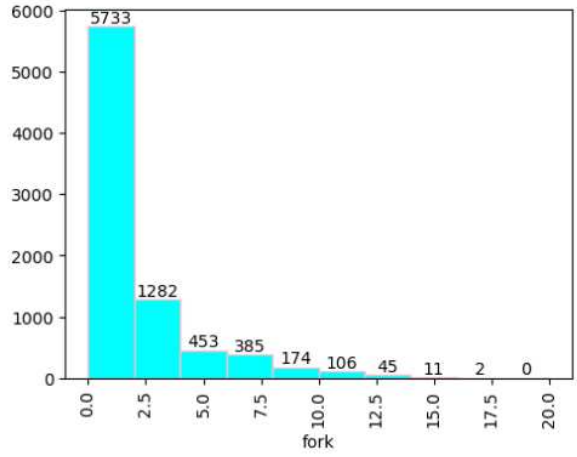

```
Skewness of sread:  5.459465962452425
Distribution of sread
--------------------------------------------------------------------------------
```

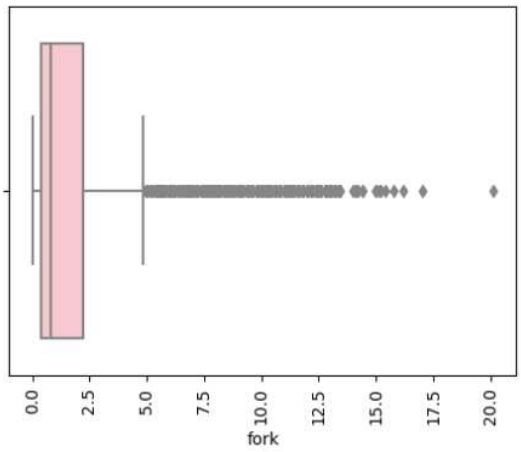



```
Skewness of swrite:  9.605843698195871
Distribution of swrite
--------------------------------------------------------------------------------
```

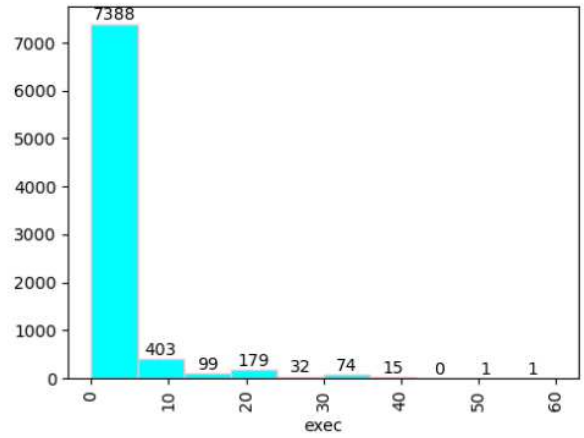


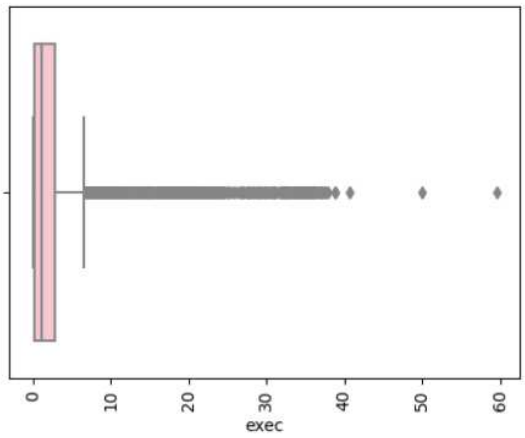
Skewness of fork: 2.2496891391571325
Distribution of fork



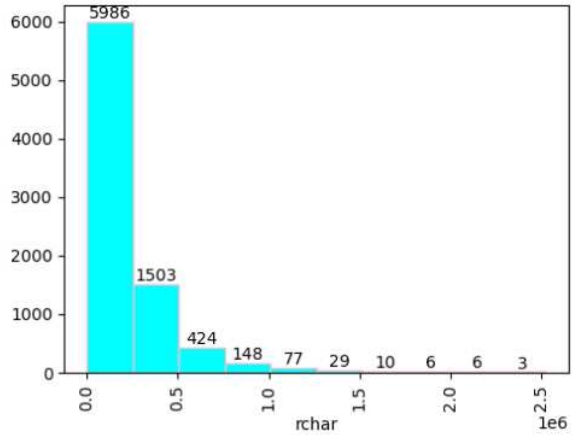


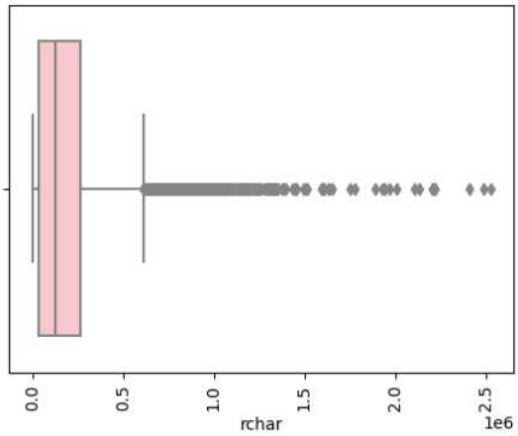
Skewness of exec: 4.069237707552533
Distribution of exec



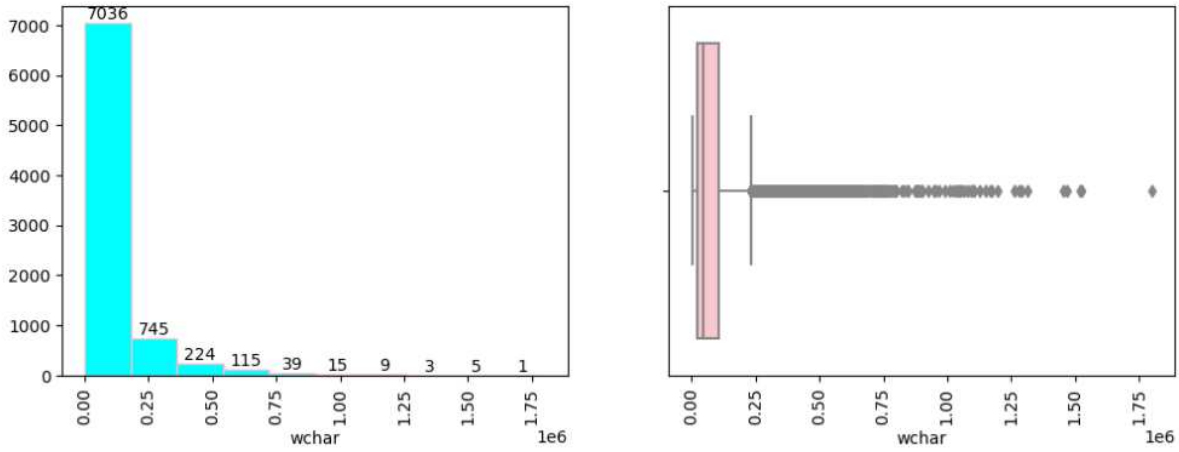


Skewness of rchar: 2.8785581933662114
Distribution of rchar

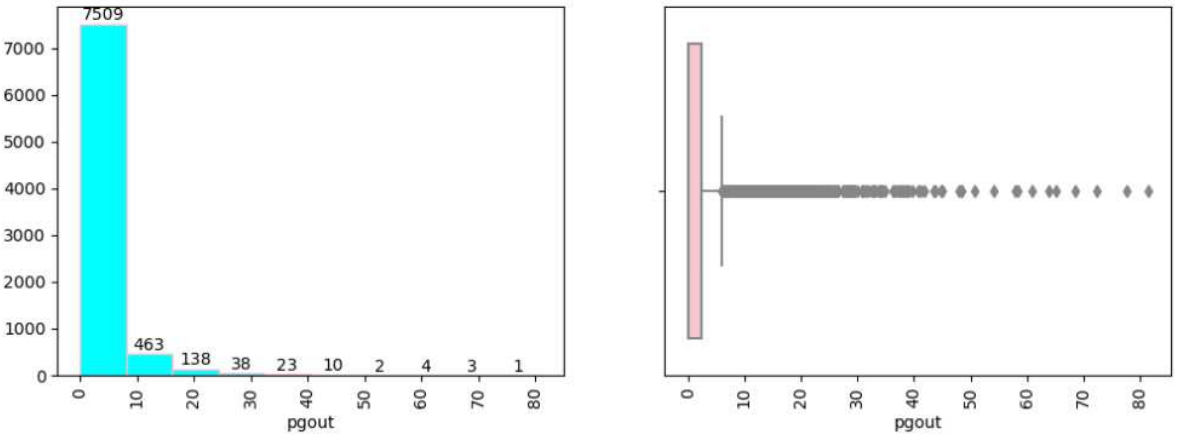




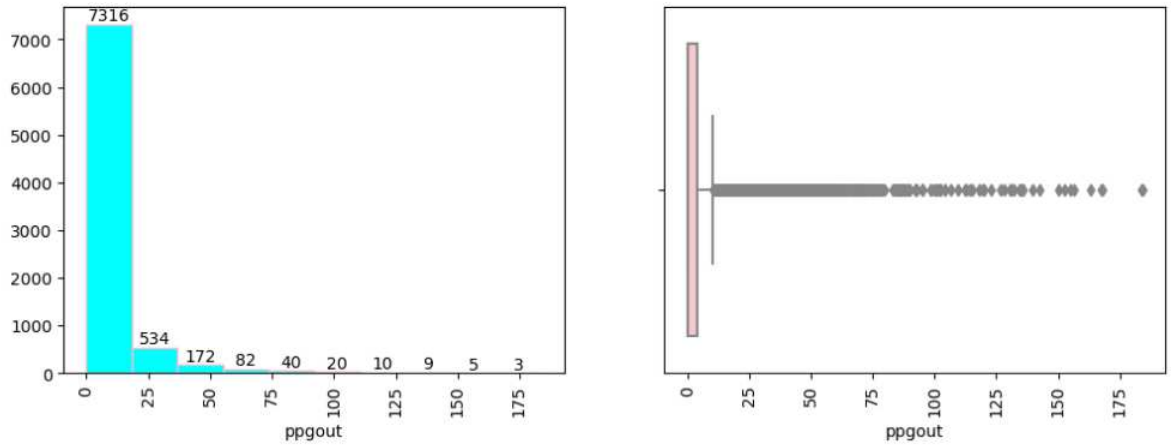
Skewness of wchar: 3.851730992818844
Distribution of wchar



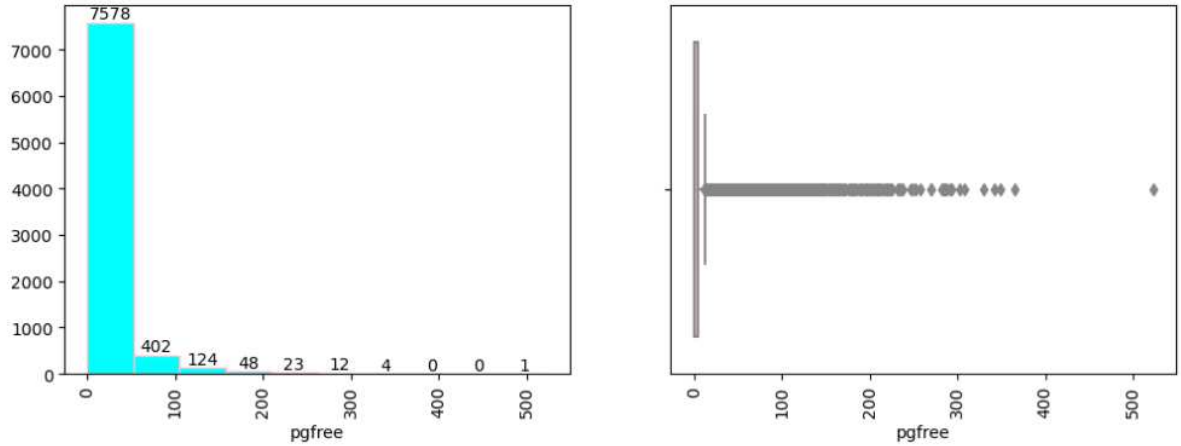
Skewness of pgout: 5.0669841185950535
Distribution of pgout



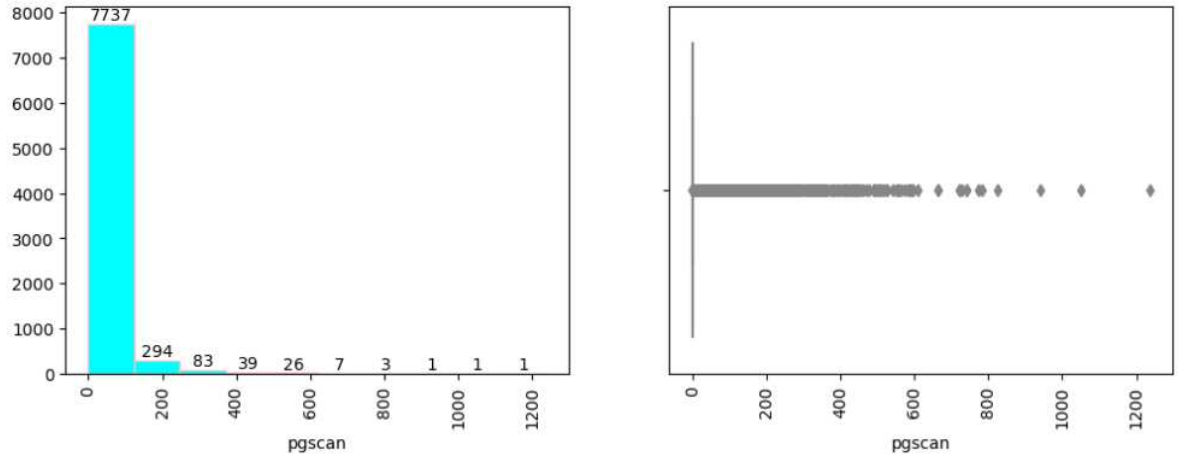
Skewness of ppgout: 4.680441654574661
Distribution of ppgout



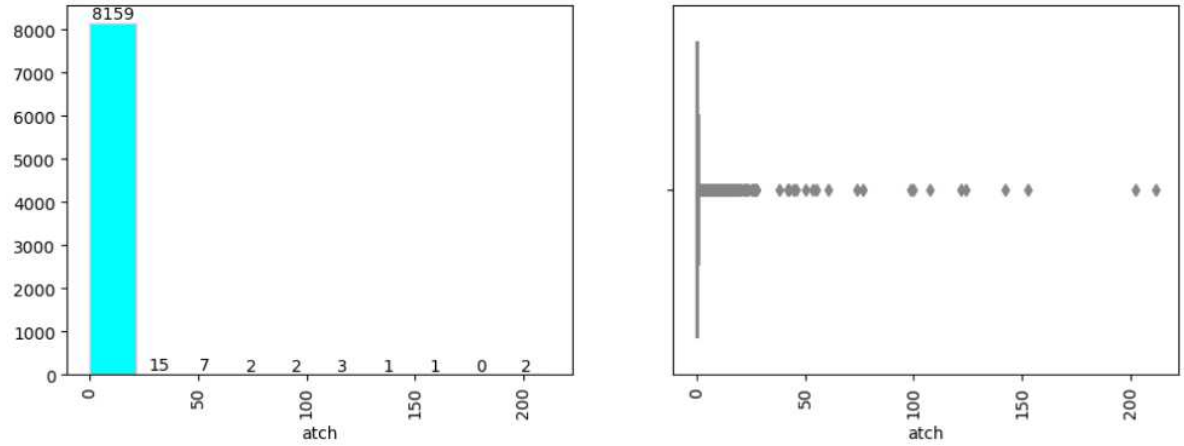
Skewness of pgfree: 4.768191252103855
Distribution of pgfree



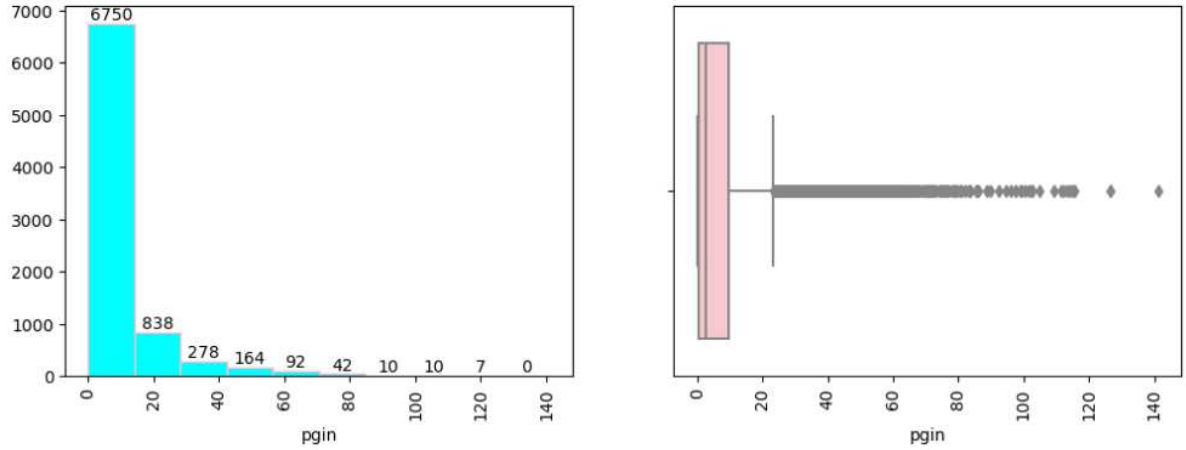
Skewness of pgscan: 5.813415144064877
Distribution of pgscan



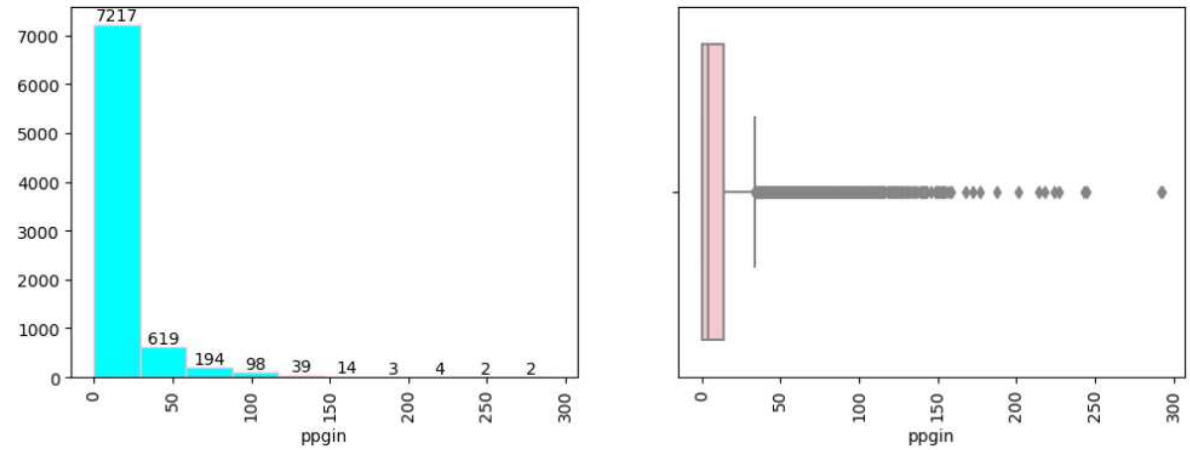
Skewness of atch: 21.542019683247847
Distribution of atch



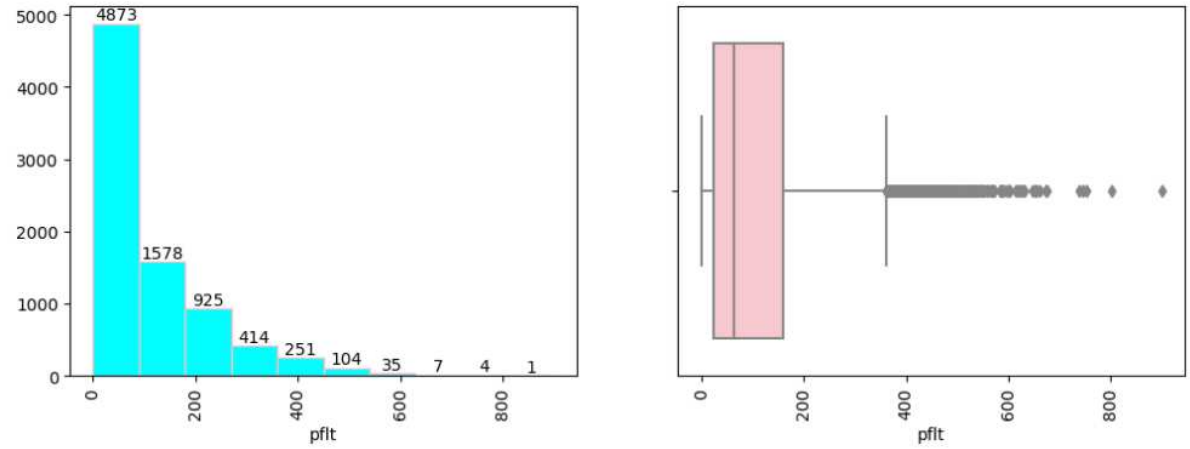
Skewness of pgin: 3.2424124762557356
Distribution of pgin



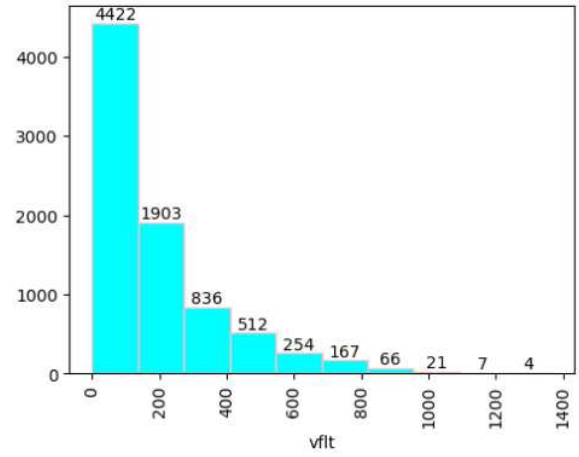
Skewness of ppgin: 3.902764914157577
Distribution of ppgin

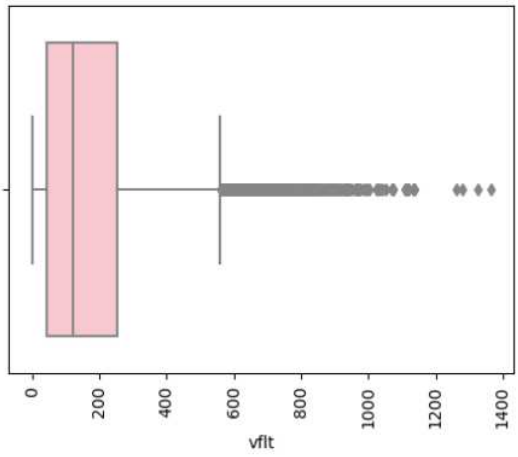


Skewness of pflt: 1.7202841192012033
Distribution of pflt

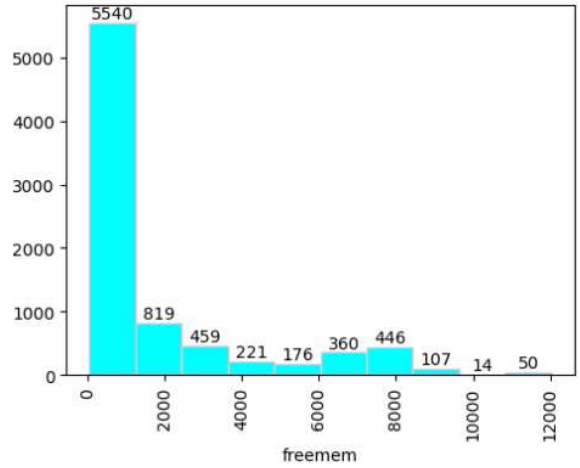


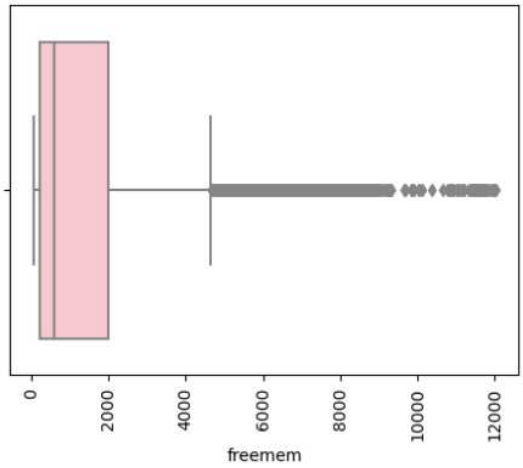
Skewness of vflt: 1.7373265929727528
Distribution of vflt



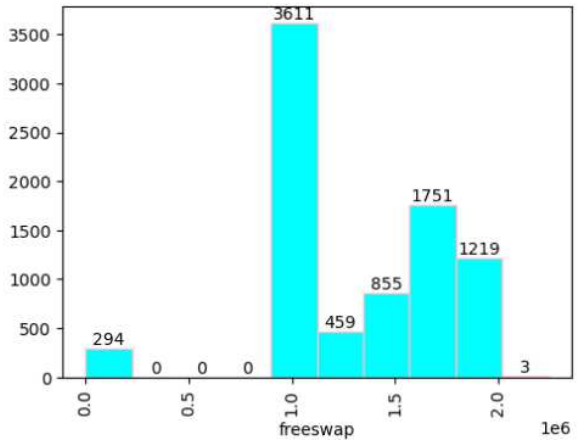


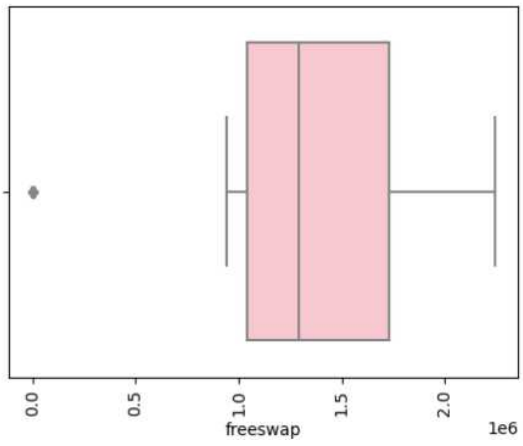
Skewness of freemem: 1.8075546533224125
Distribution of freemem





Skewness of freeswap: -0.7916644438525977
Distribution of freeswap





```
Skewness of usr:  -3.4167496030437094
Distribution of usr
--------------------------------------------------------------------------------
```

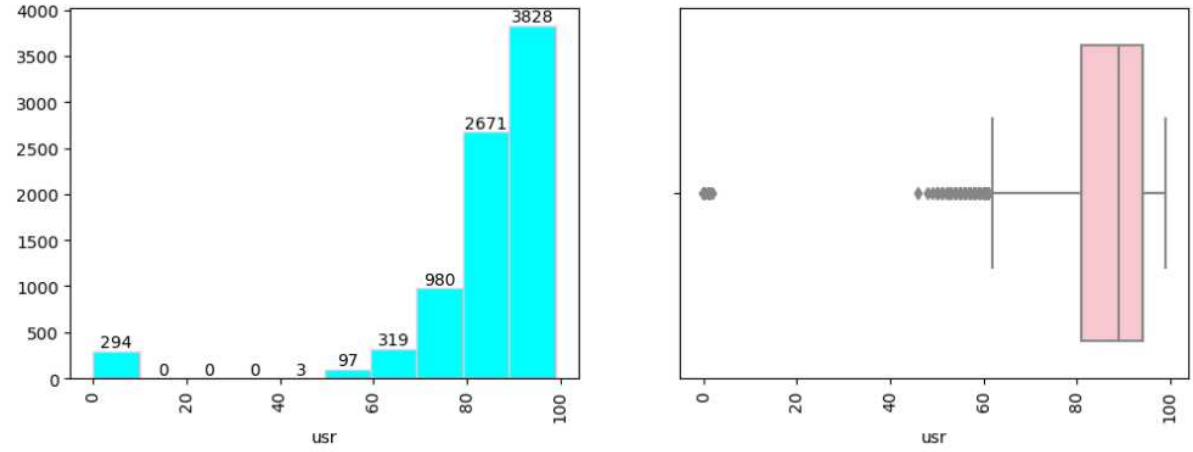


Figure 1: Univariate Analysis numeric columns

Key Observations

1. As we had expected earlier, there are significant number of outliers in the data which we will have to treat.
2. Data is skewed for all attributes. For independent variables only 'freeswap' is left skewed, rest all are right skewed.

For categorical columns

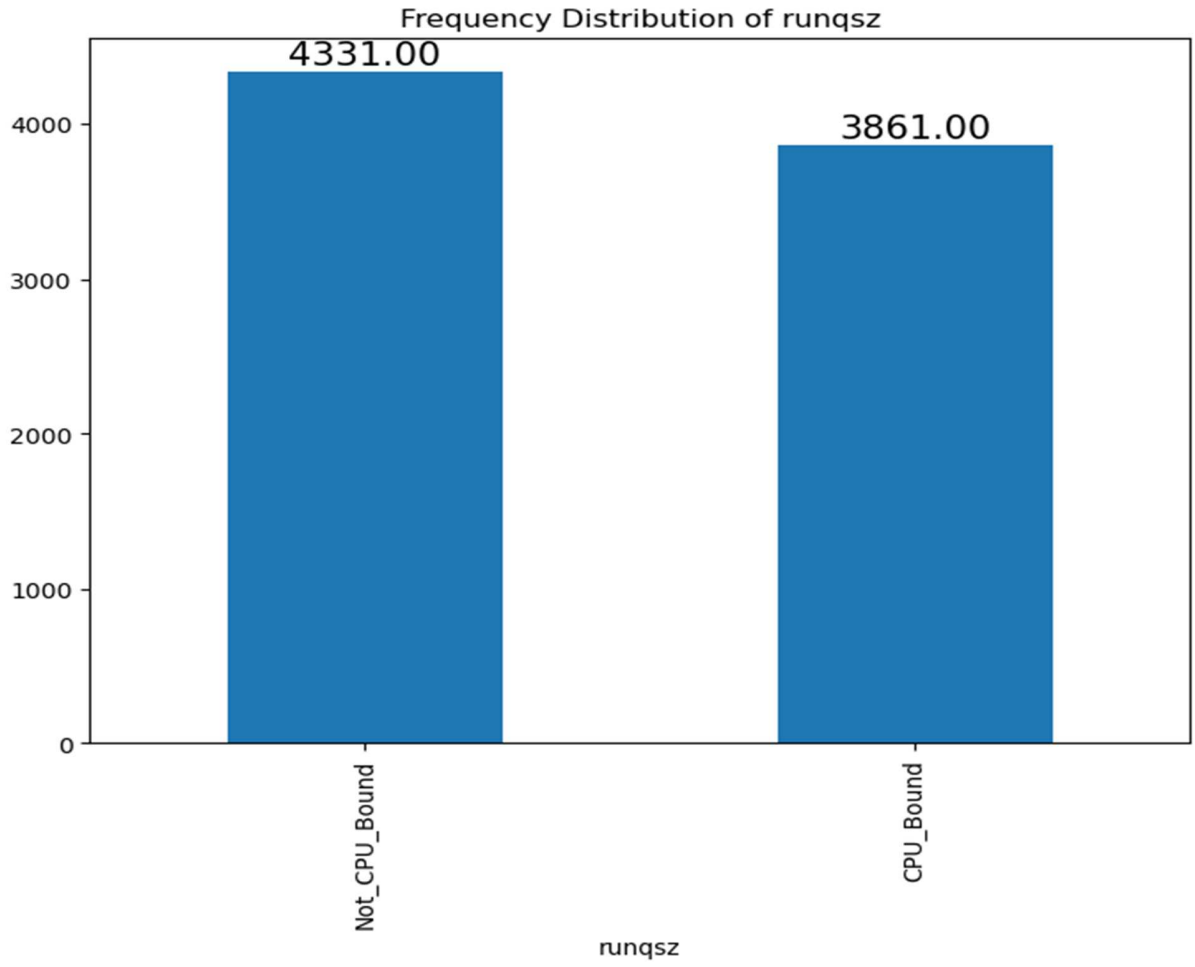


Figure 2: Univariate Analysis categorical columns

1.6.2 Bivariate Analysis

Relation between numeric columns

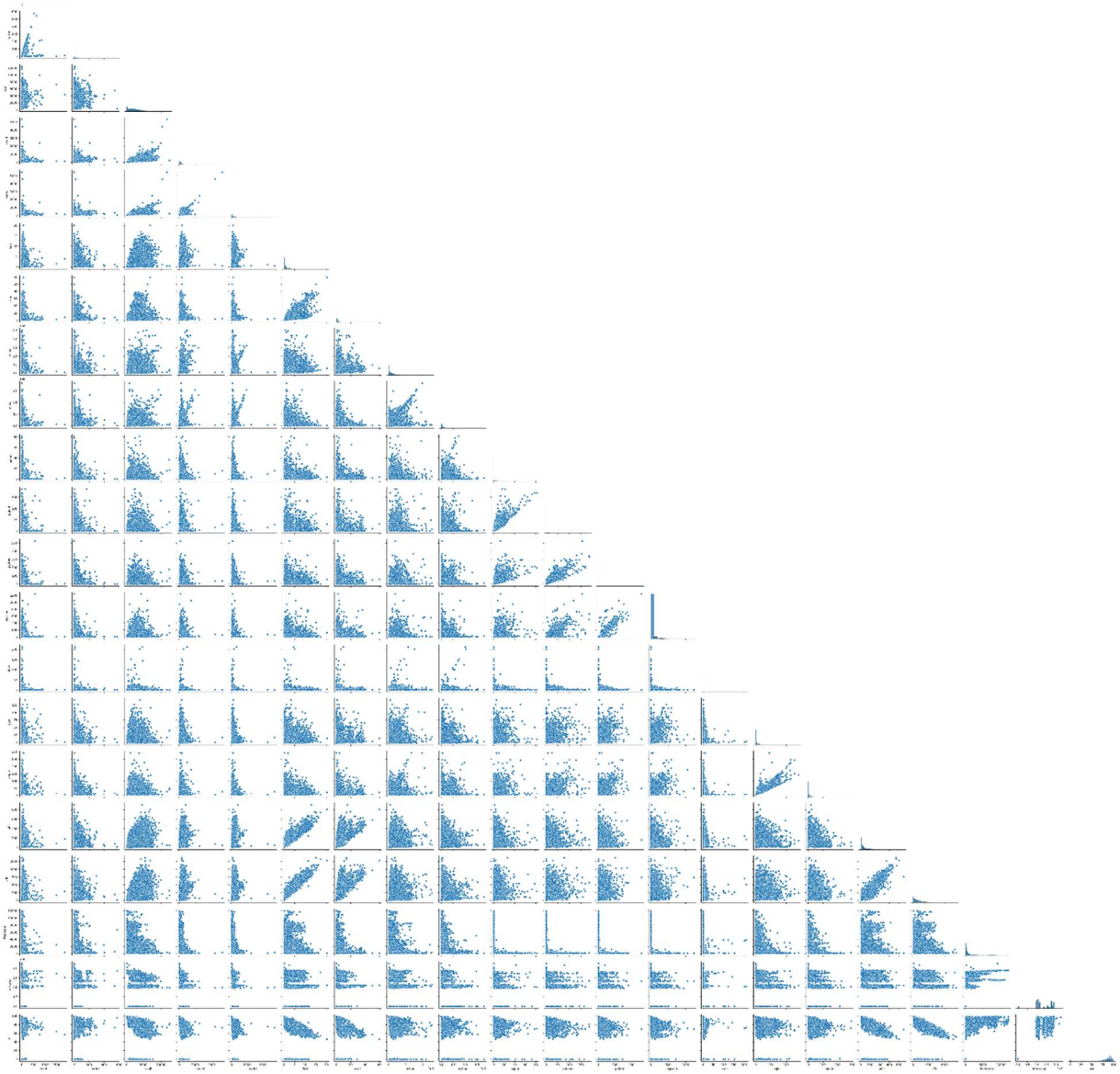


Figure 3: Pair plot

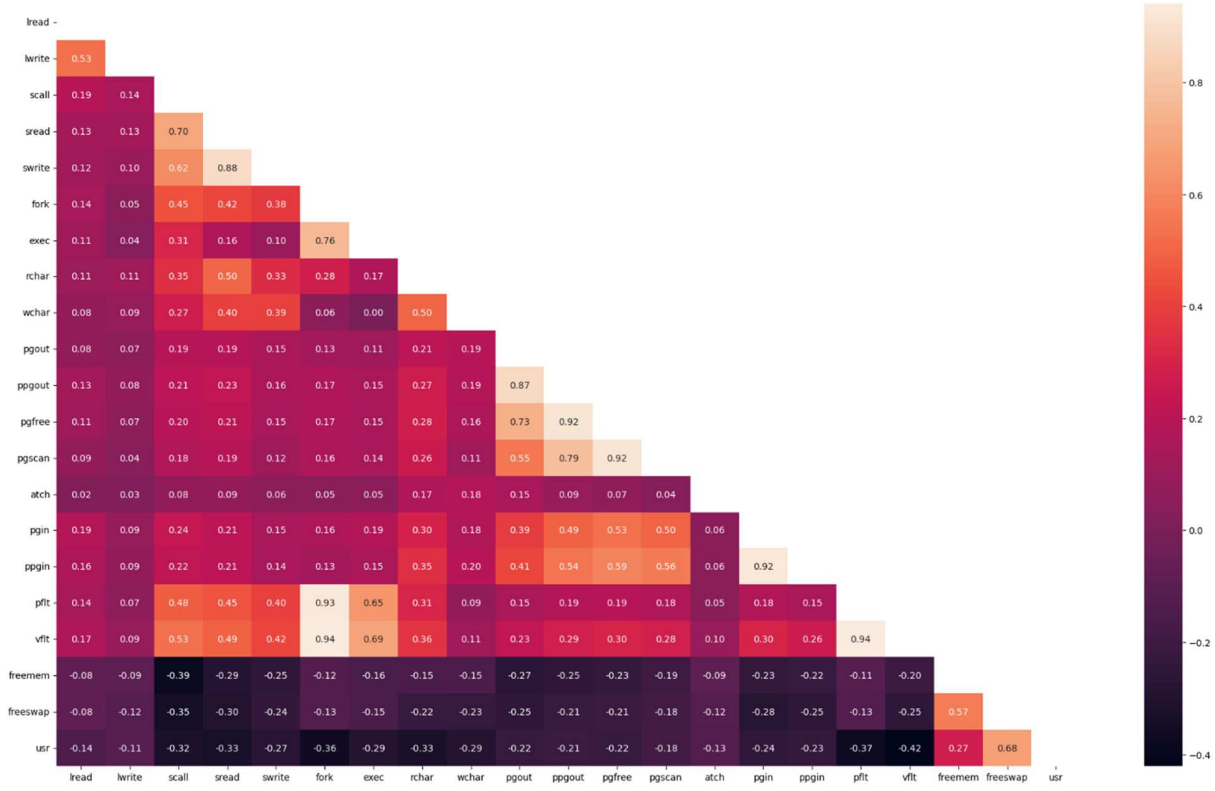


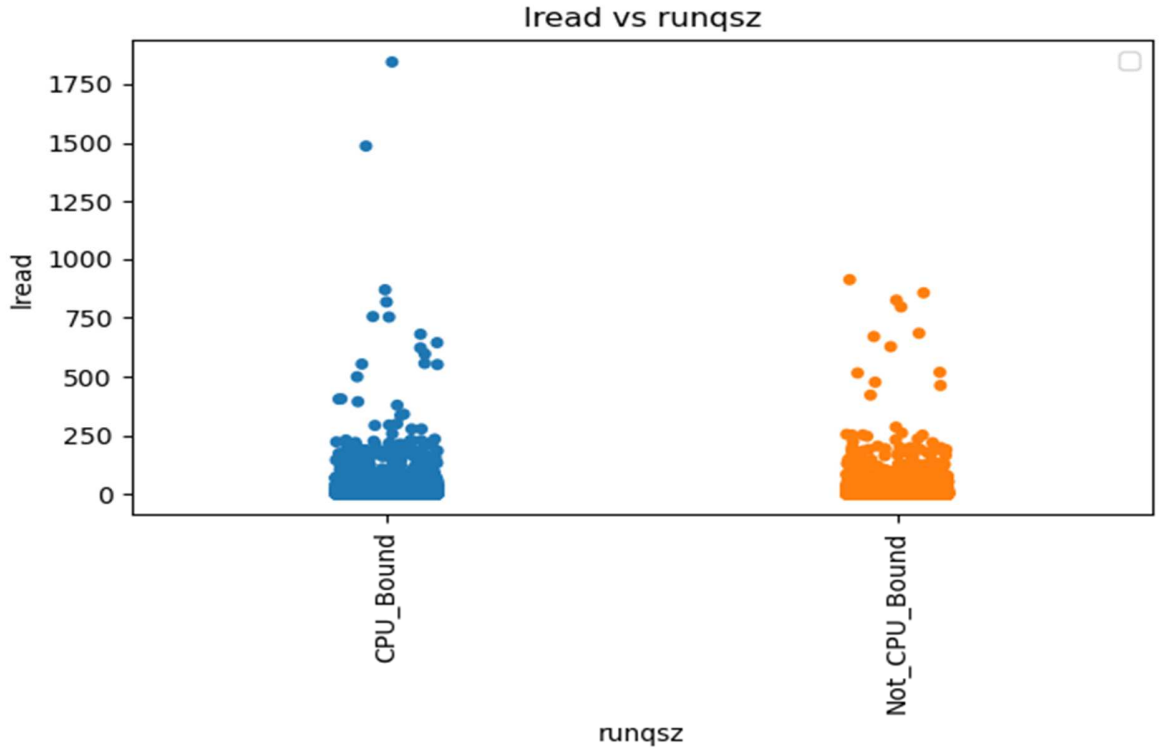
Figure 4: Heatmap

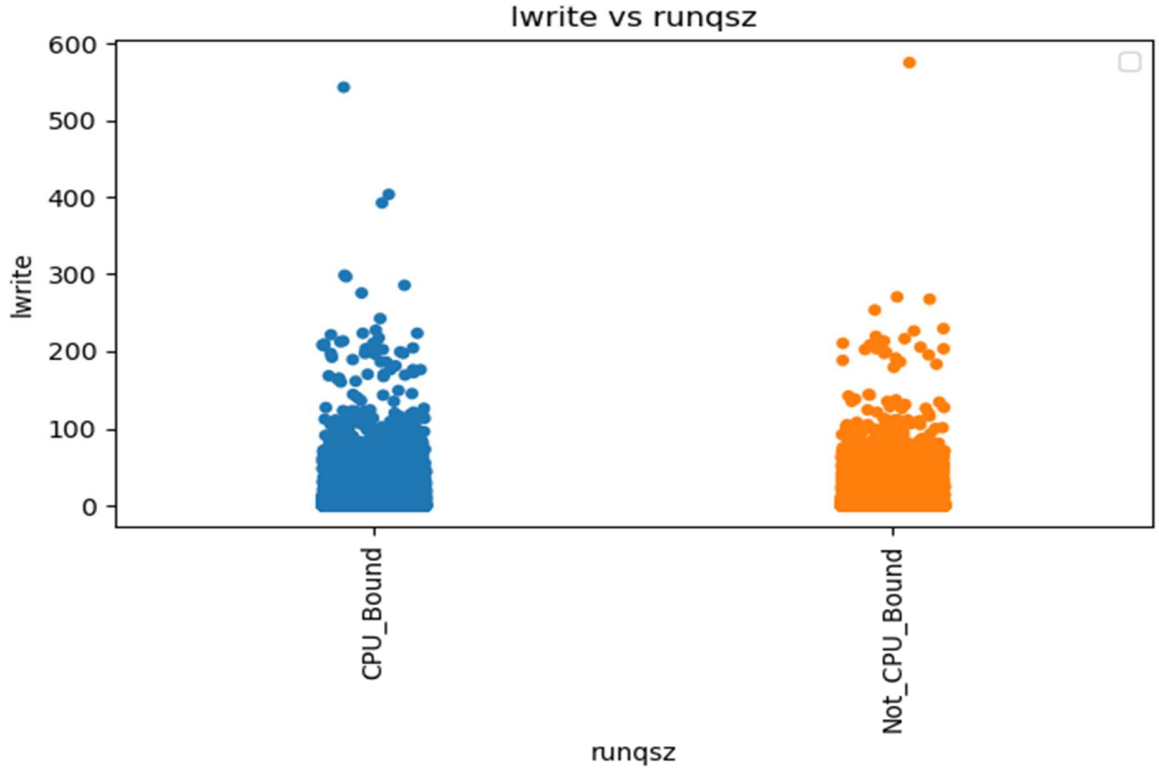
Key Observations

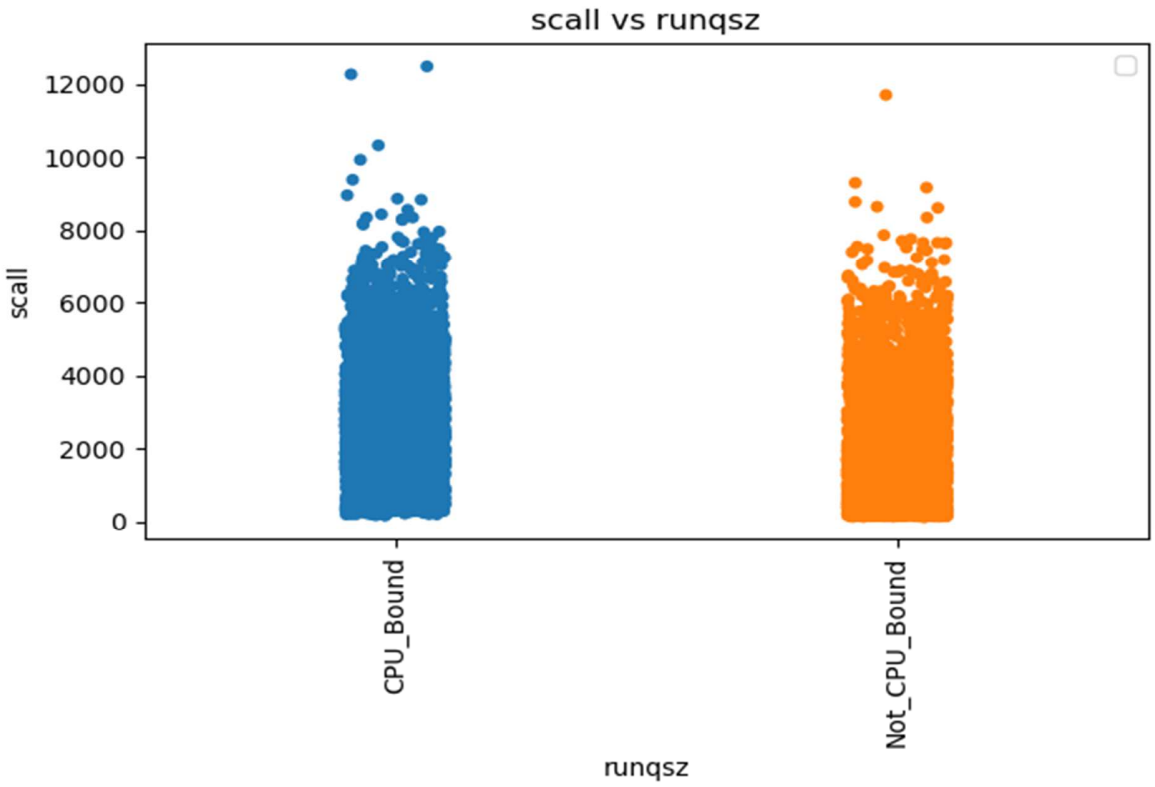
1. 1.From the pair plot we can conclude that there is correlation between more than 1 pair of attributes, to understand exact magnitude we will have to look at heatmap.
2. Based we can say that with 'usr' which is the attribute of interest for us there is some negative correlation with 'fork', 'pflt' and 'vflt'.
3. From the heatmap we can conclude that 'freeswap' has the highest correlation with 'usr' which is moderately strong at 0.68.
4. For multiple pairs correlation is above 0.9 meaning that these attributes are highly correlated to each other, also there are pairs like 'pgscan' and 'ppgout' or 'swrite' and 'iwrite' that have correlation around or over 0.8. While applying regression model we will have drop some of these variables depending upon their effect on the model.

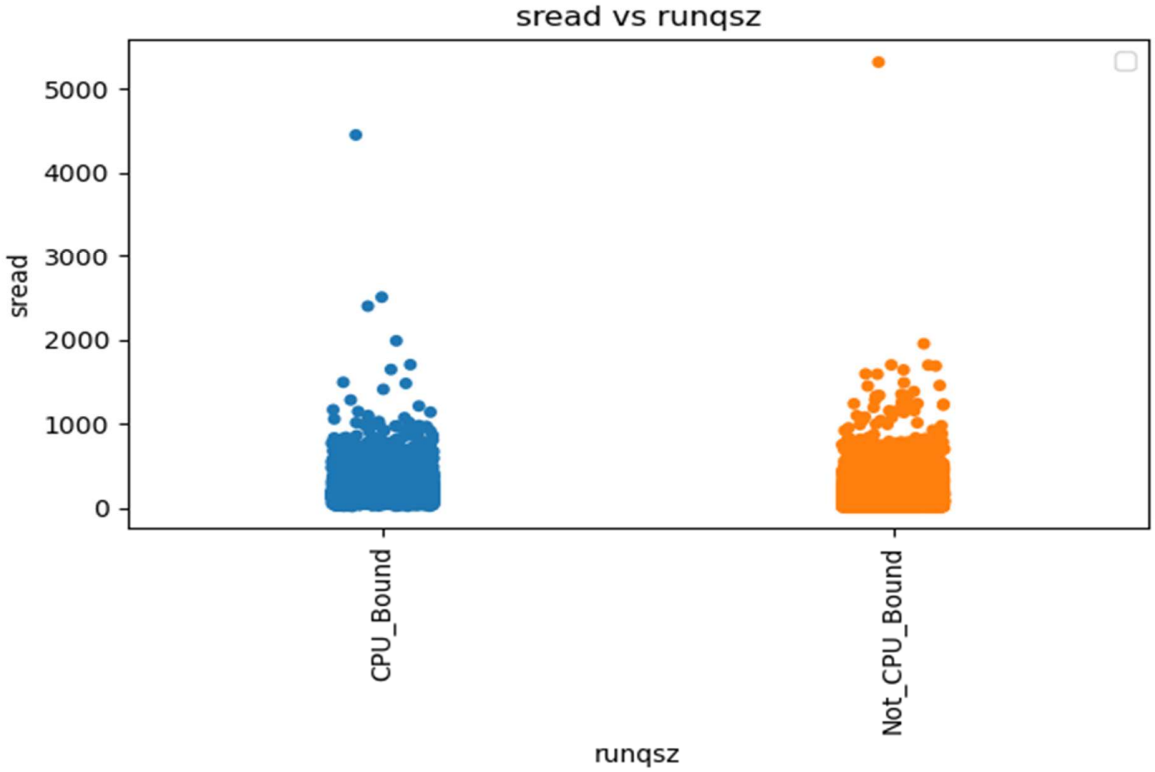
Relation between numeric and categorical columns

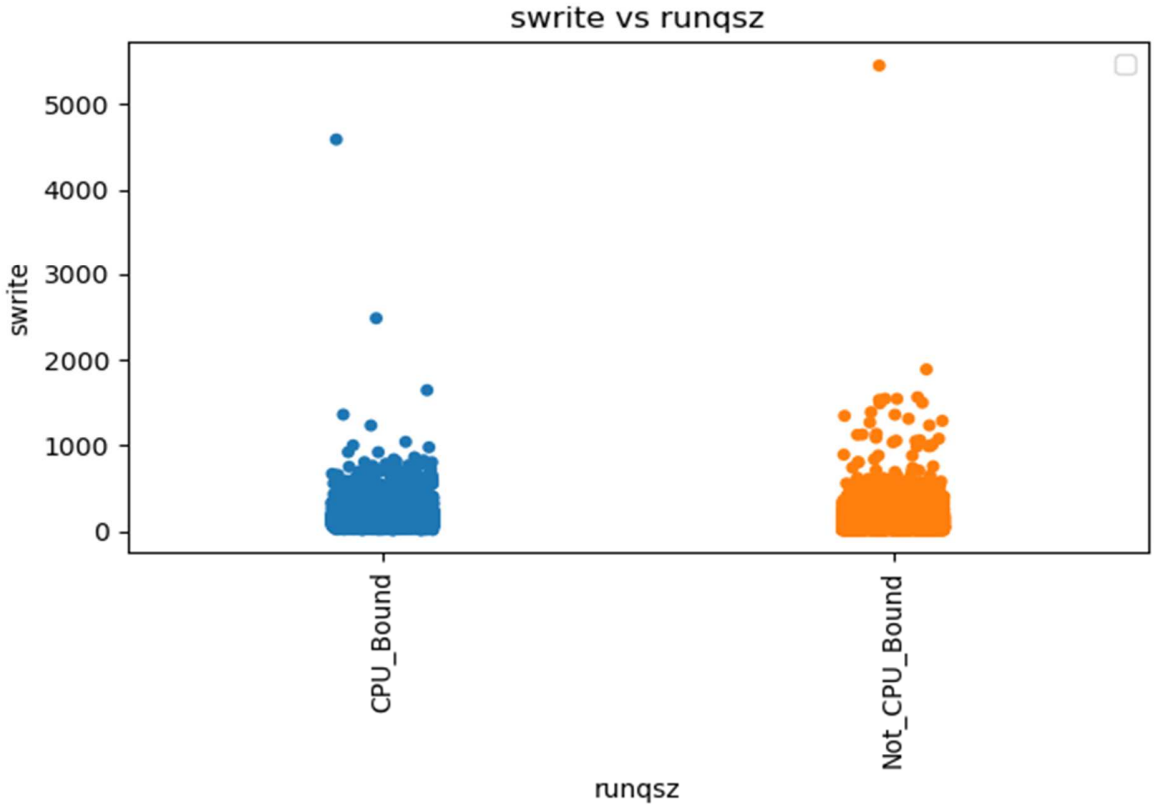
Bivariate analysis for runsqz

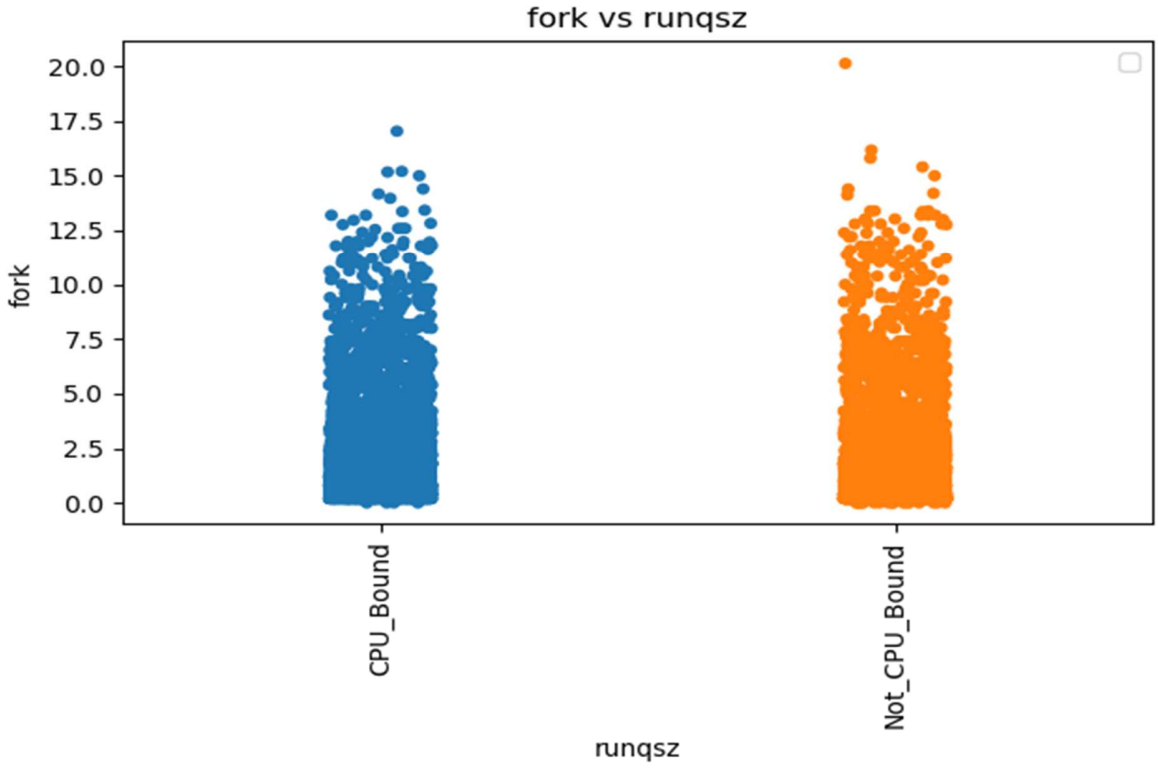


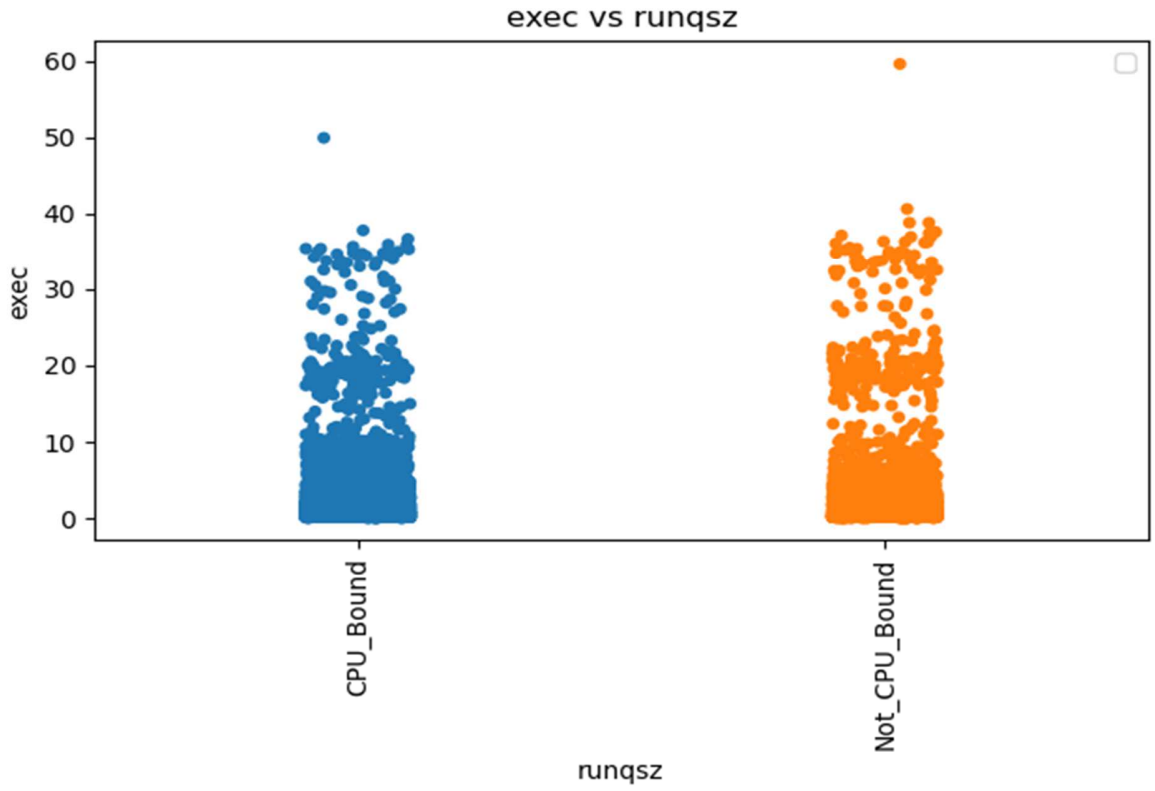


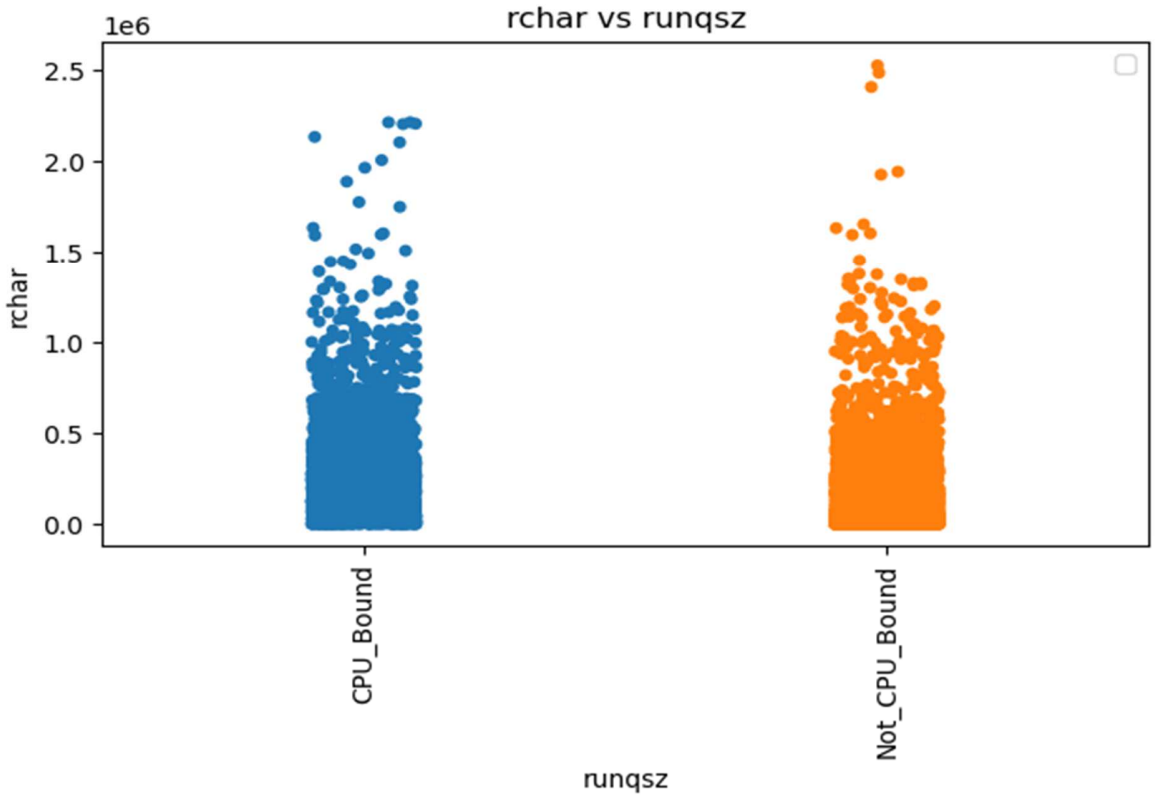

scall vs runqsz
scall
12000
10000
8000
6000
4000
2000
0
CPU_Bound
Not_CPU_Bound
runqsz

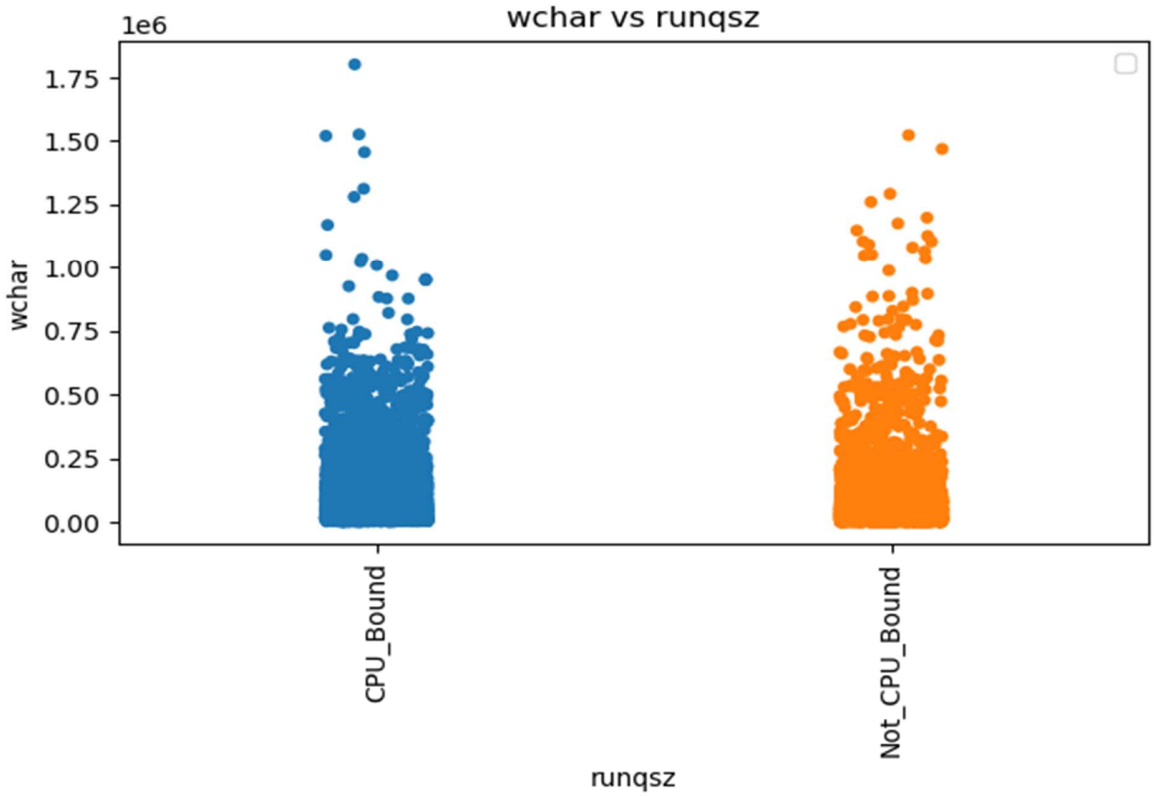

sread vs runqsz
sread
5000
4000
3000
2000
1000
0
CPU_Bound
Not_CPU_Bound
runqsz

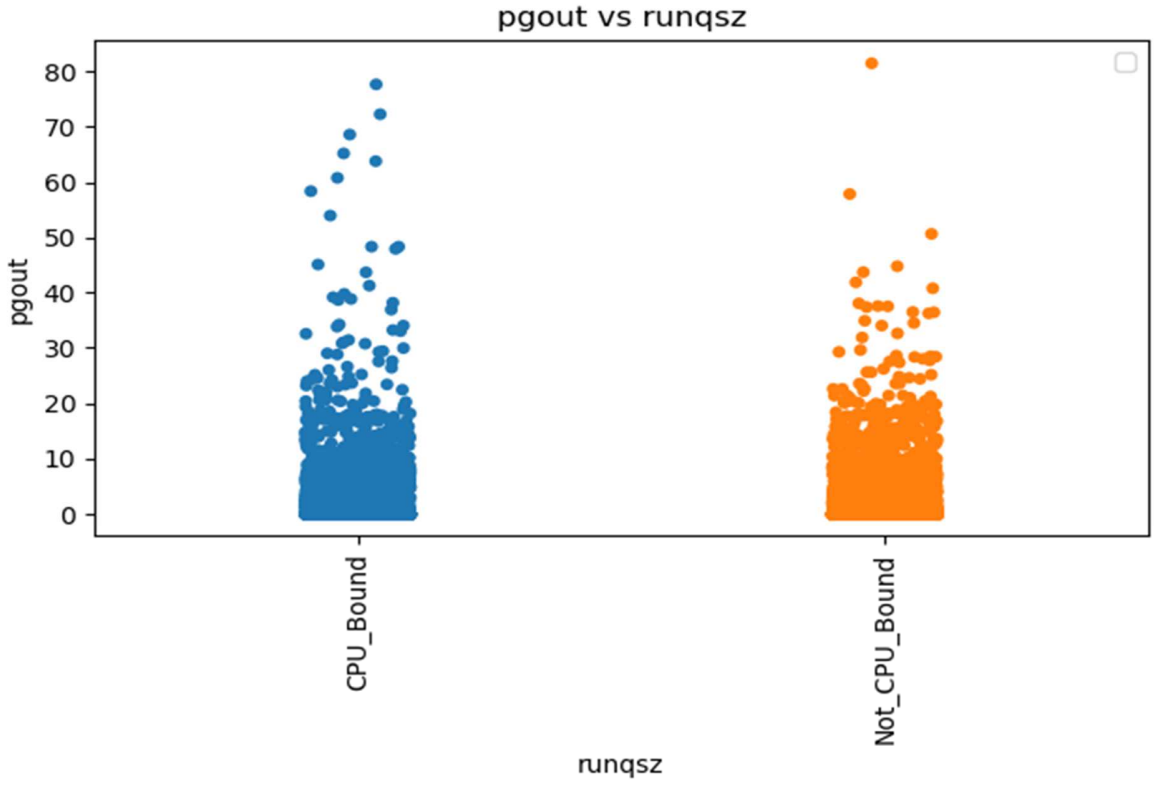

swrite vs runqsz
swrite
5000
4000
3000
2000
1000
0
CPU_Bound
Not_CPU_Bound
runqsz

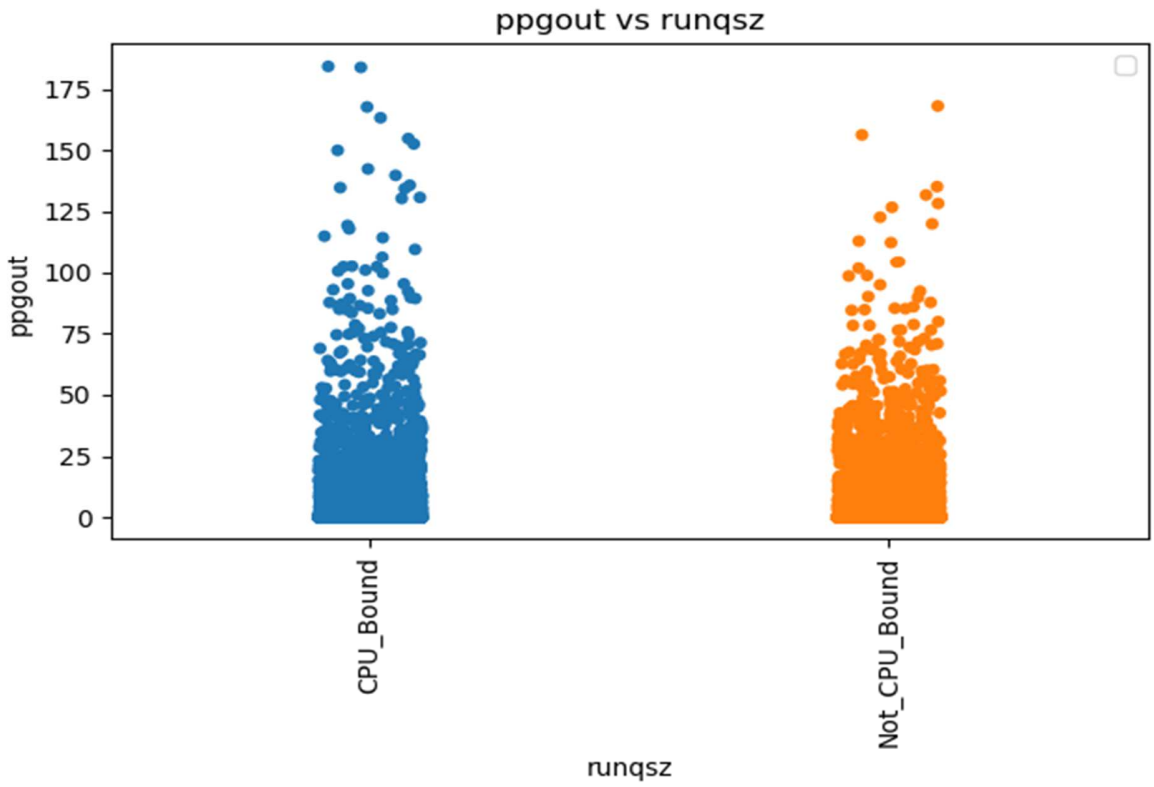

fork vs runqsz
fork
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
CPU_Bound
Not_CPU_Bound
runqsz

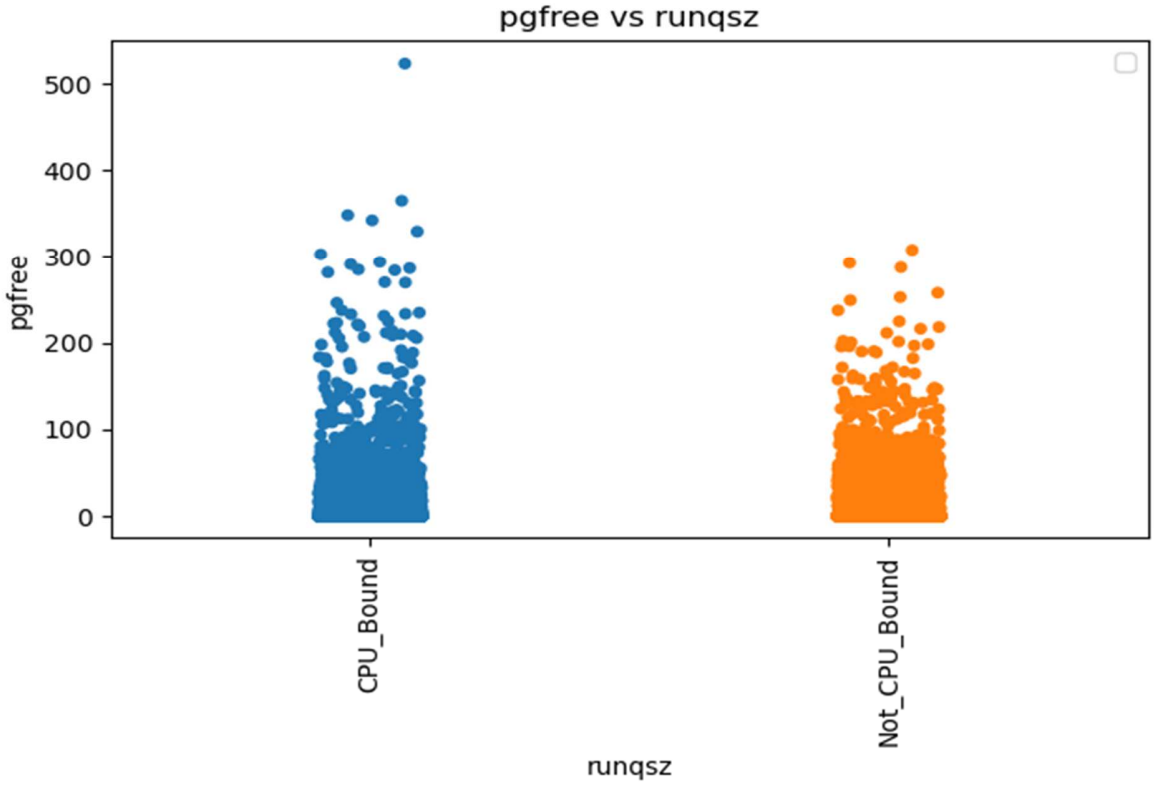

exec vs runqsz
60
50
40
30
20
10
0
exec
CPU_Bound
Not_CPU_Bound
runqsz

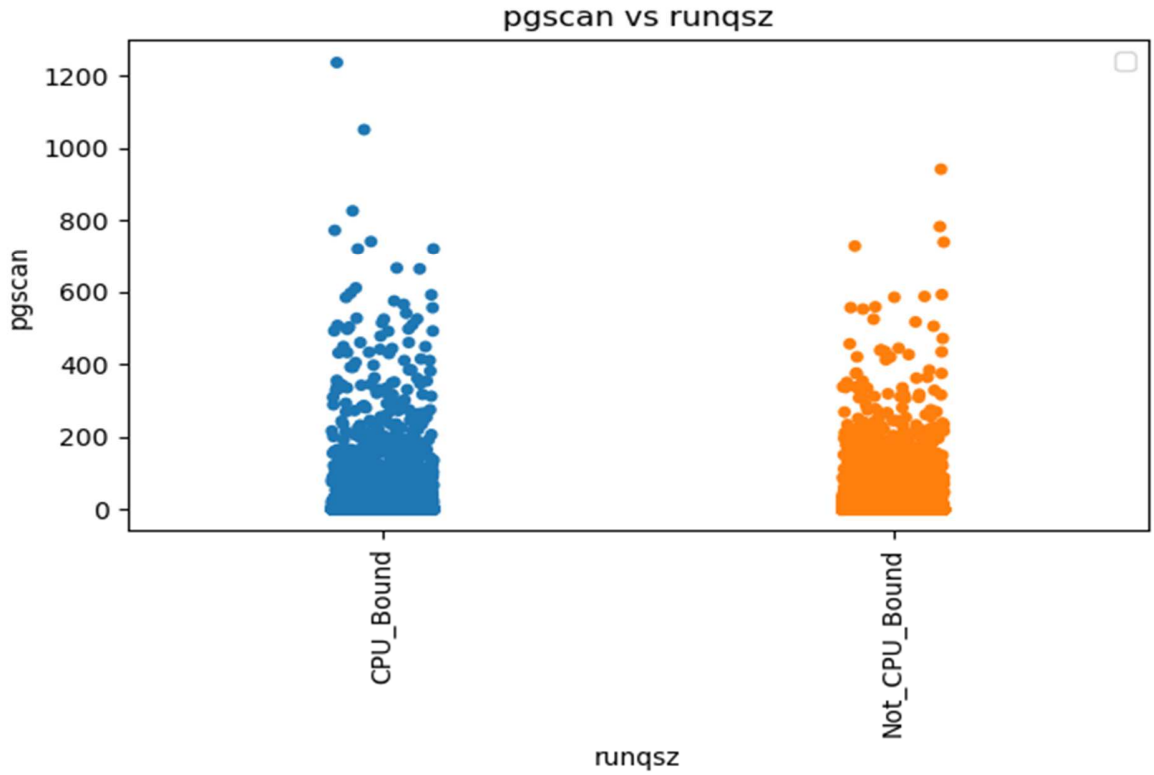

rchar vs runqsz
1e6
2.5
2.0
1.5
1.0
0.5
0.0
rchar
CPU_Bound
Not_CPU_Bound
runqsz

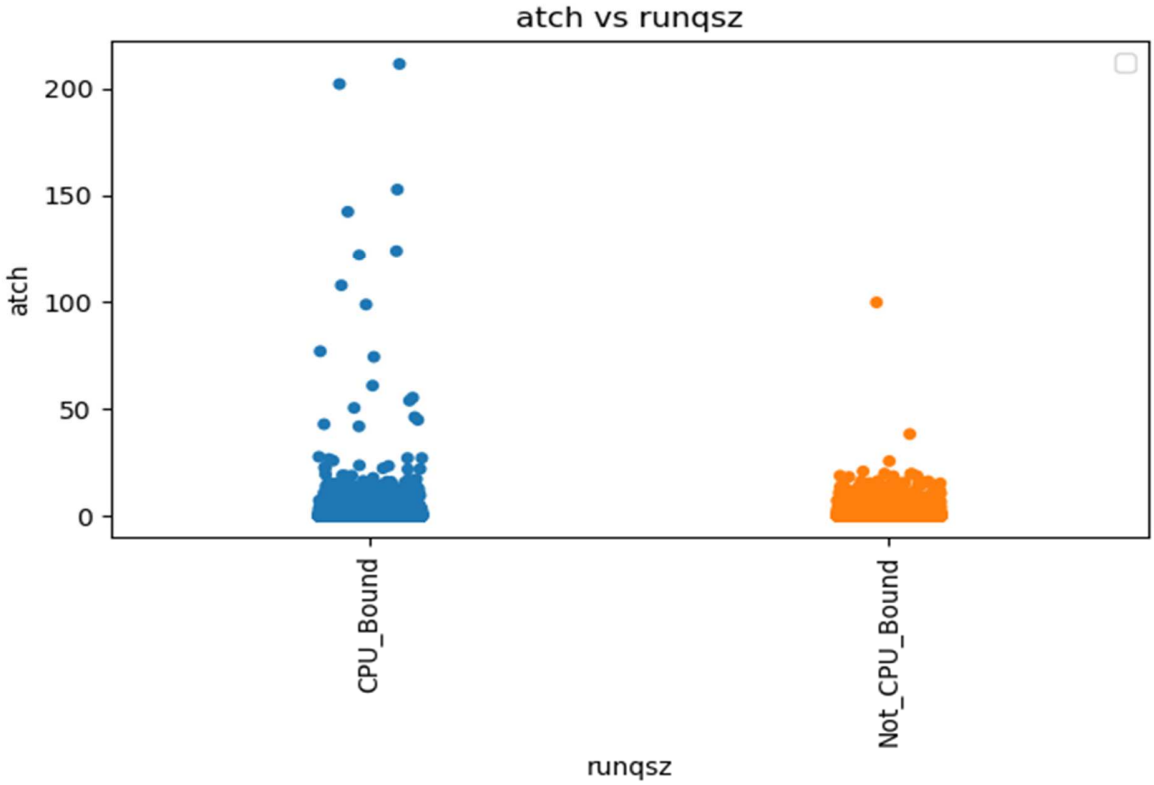

wchar vs runqsz
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
wchar
CPU_Bound
Not_CPU_Bound
runqsz

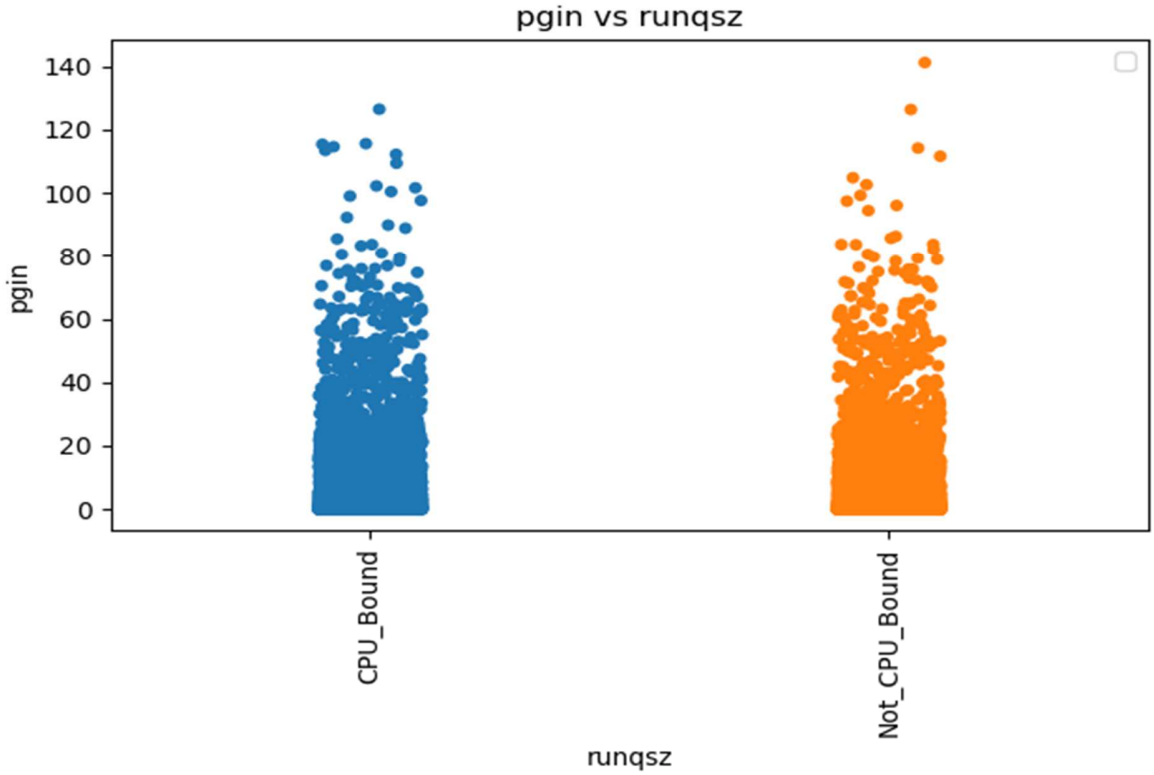

pgout vs runqsz
80
70
60
50
40
30
20
10
0
pgout
CPU_Bound
Not_CPU_Bound
runqsz

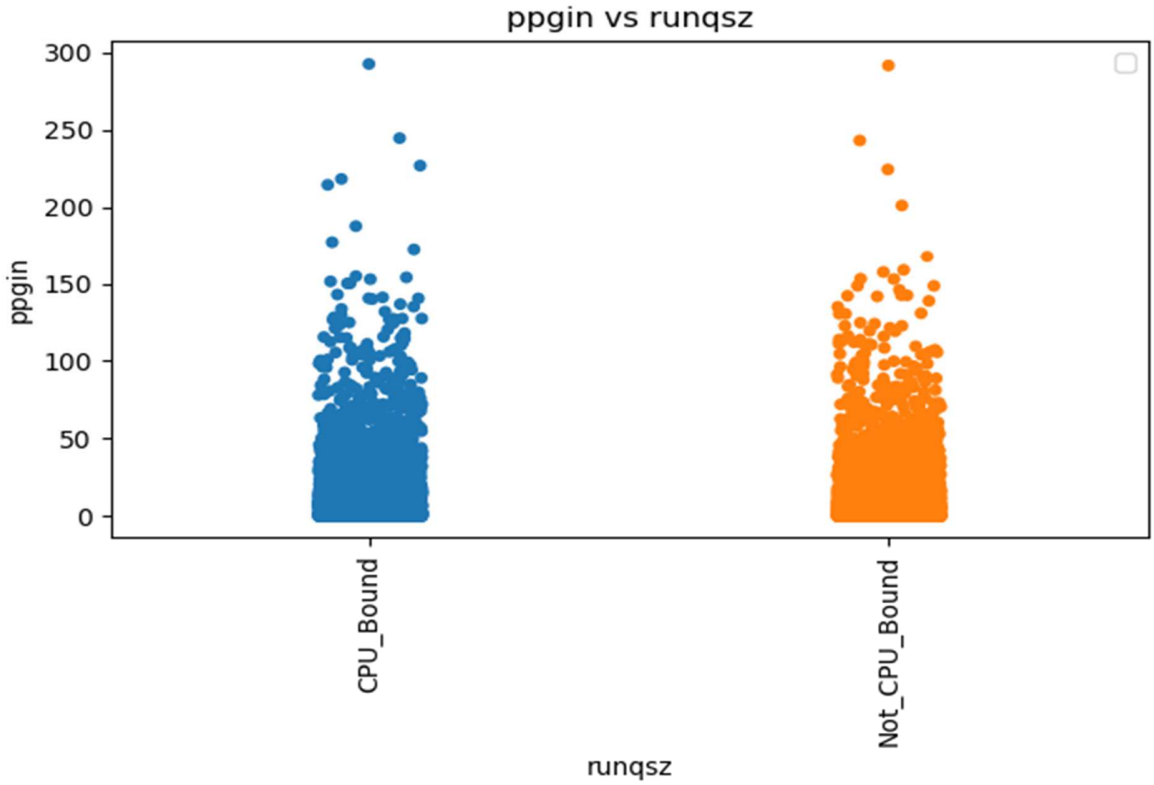

ppgout vs runqsz
ppgout
175
150
125
100
75
50
25
0
CPU_Bound
Not_CPU_Bound
runqsz

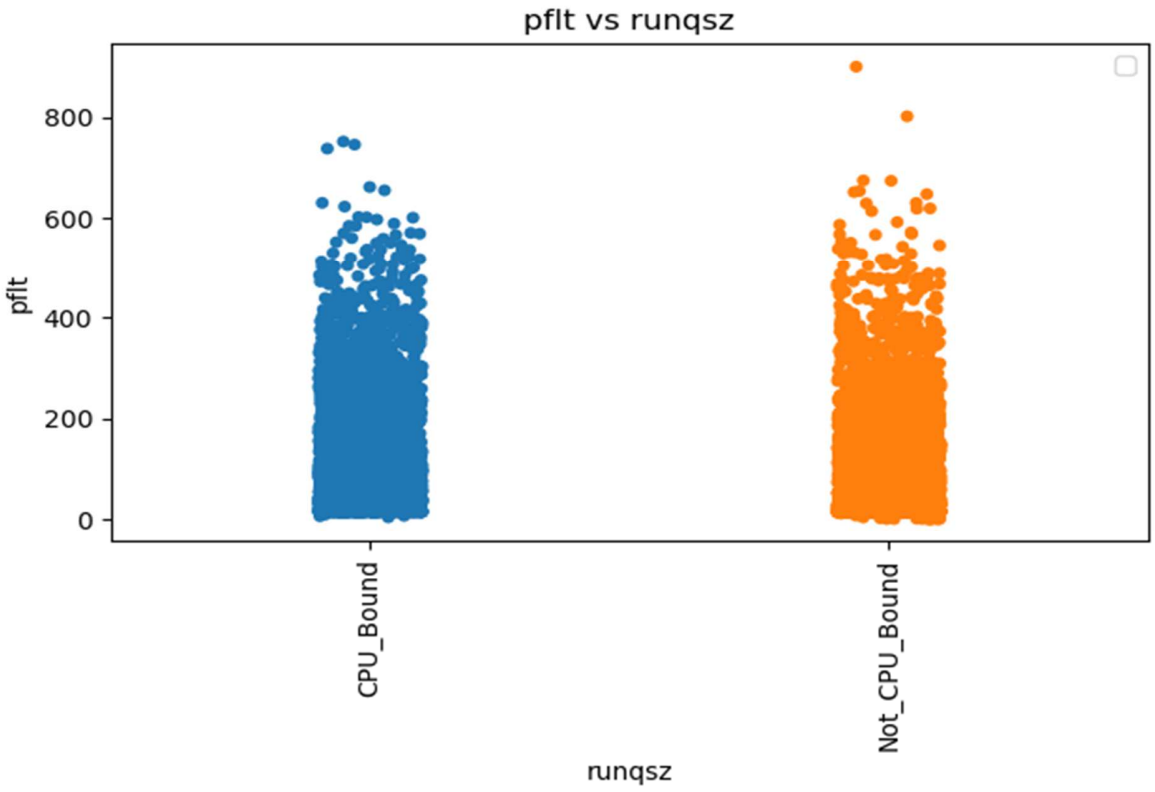

pgfree vs runqsz
pgfree
500
400
300
200
100
0
CPU_Bound
Not_CPU_Bound
runqsz

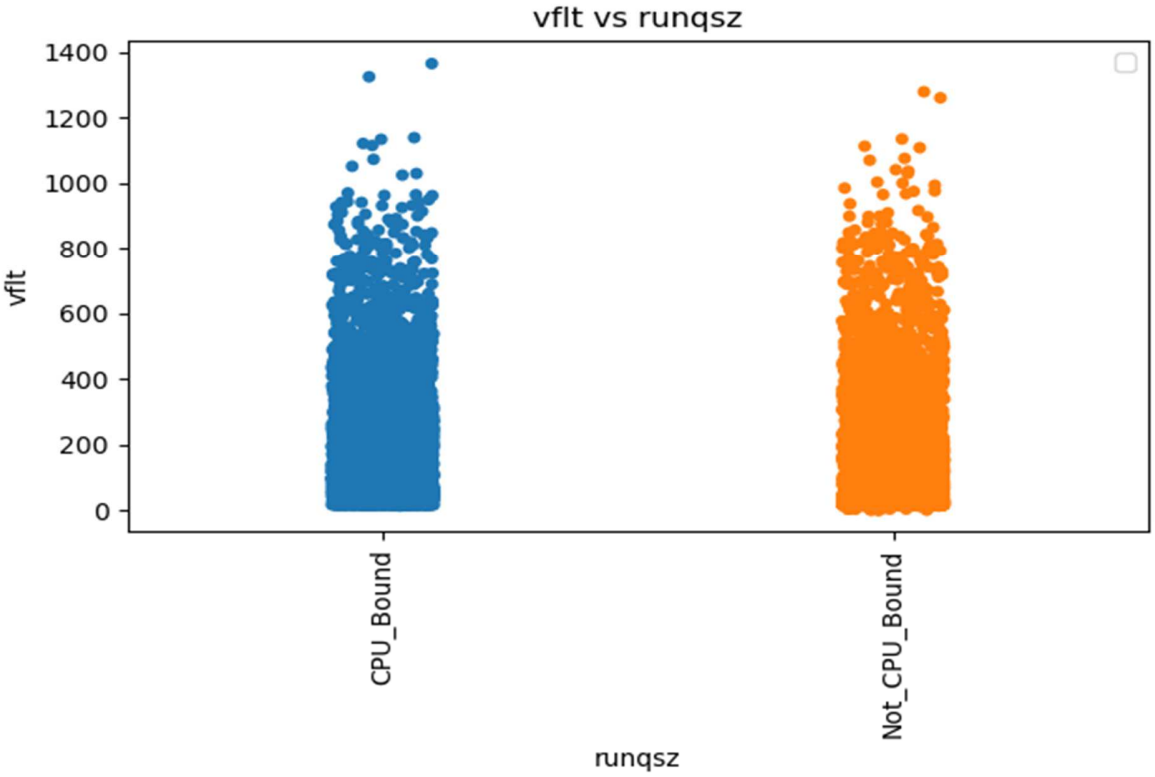

pgscan vs runqsz
pgscan
1200
1000
800
600
400
200
0
CPU_Bound
Not_CPU_Bound
runqsz

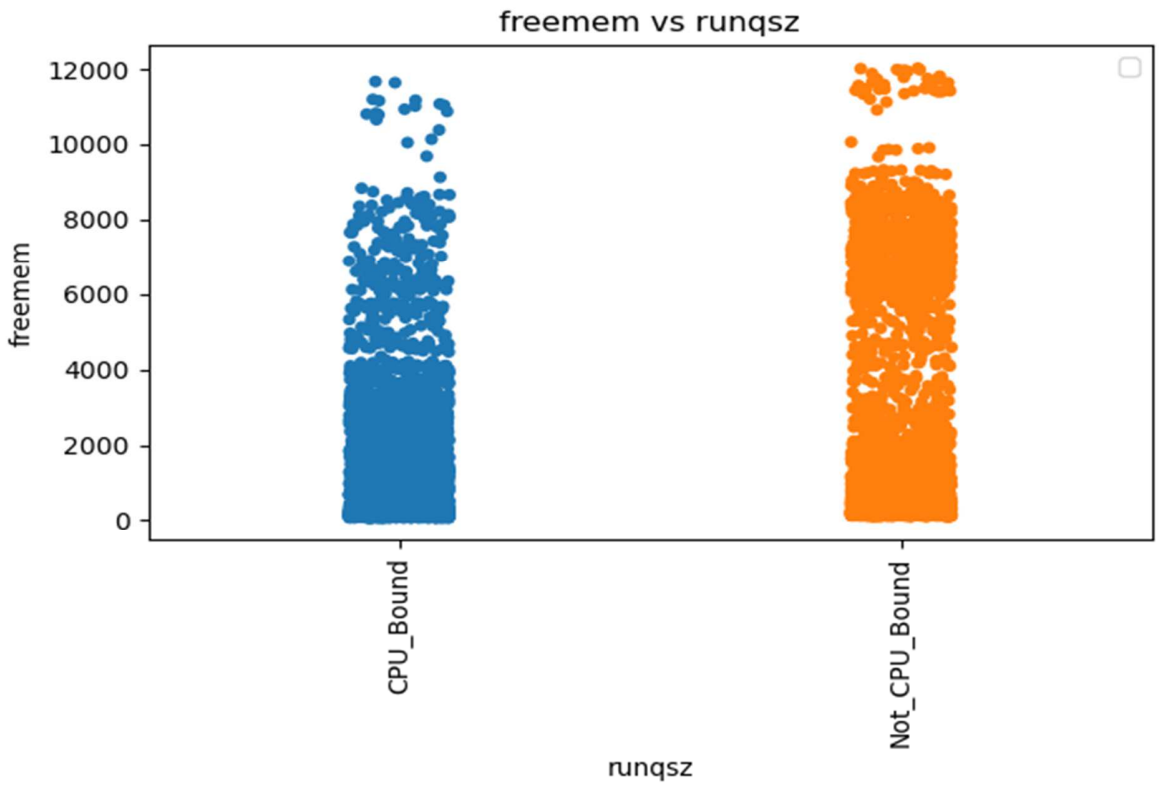

atch vs runqsz
atch
200
150
100
50
0
CPU_Bound
Not_CPU_Bound
runqsz

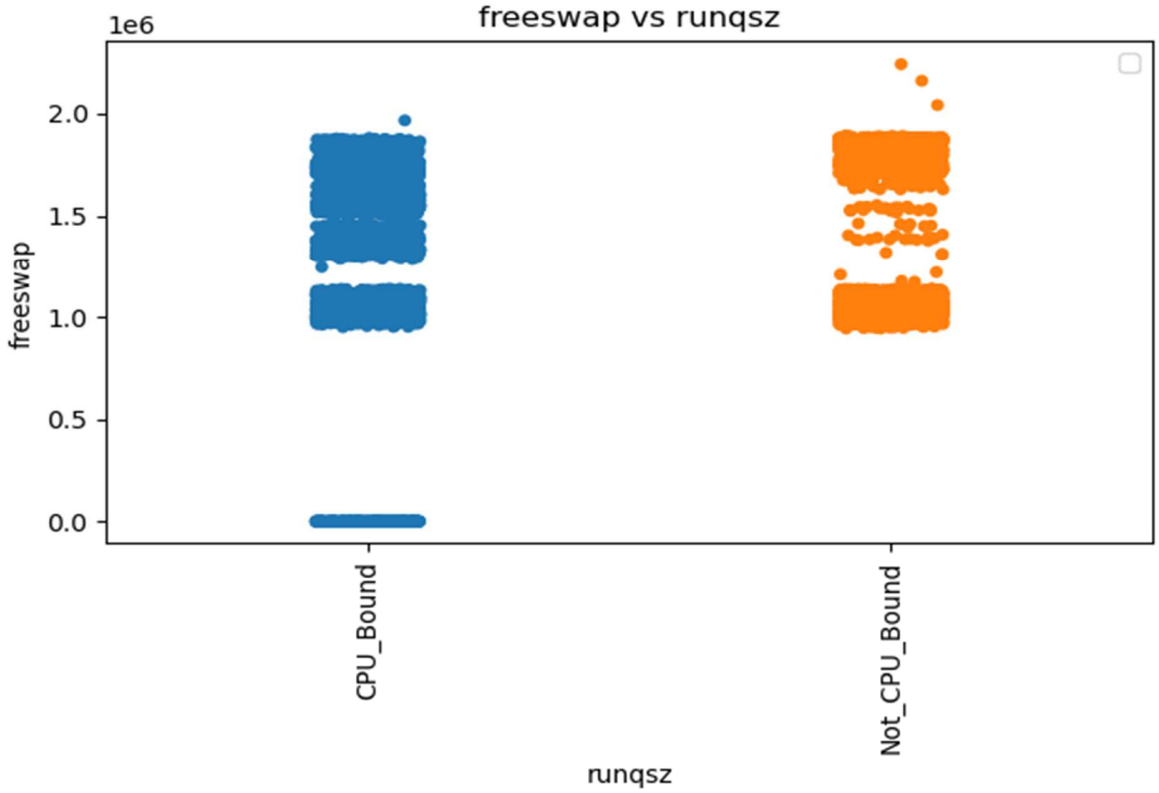

pgin vs runqsz
pgin
0
20
40
60
80
100
120
140
CPU_Bound
Not_CPU_Bound
runqsz


ppgin vs runqsz
ppgin
0
50
100
150
200
250
300
CPU_Bound
Not_CPU_Bound
runqsz


pflt vs runqsz
pflt
800
600
400
200
0
CPU_Bound
Not_CPU_Bound
runqsz


vflt vs runqsz
vflt
1400
1200
1000
800
600
400
200
0
CPU_Bound
Not_CPU_Bound
runqsz


freemem vs runqsz
12000
10000
8000
6000
4000
2000
0
freemem
CPU_Bound
Not_CPU_Bound
runqsz


freeswap vs runqsz
1e6
2.0
1.5
1.0
0.5
0.0
freeswap
CPU_Bound
Not_CPU_Bound
runqsz

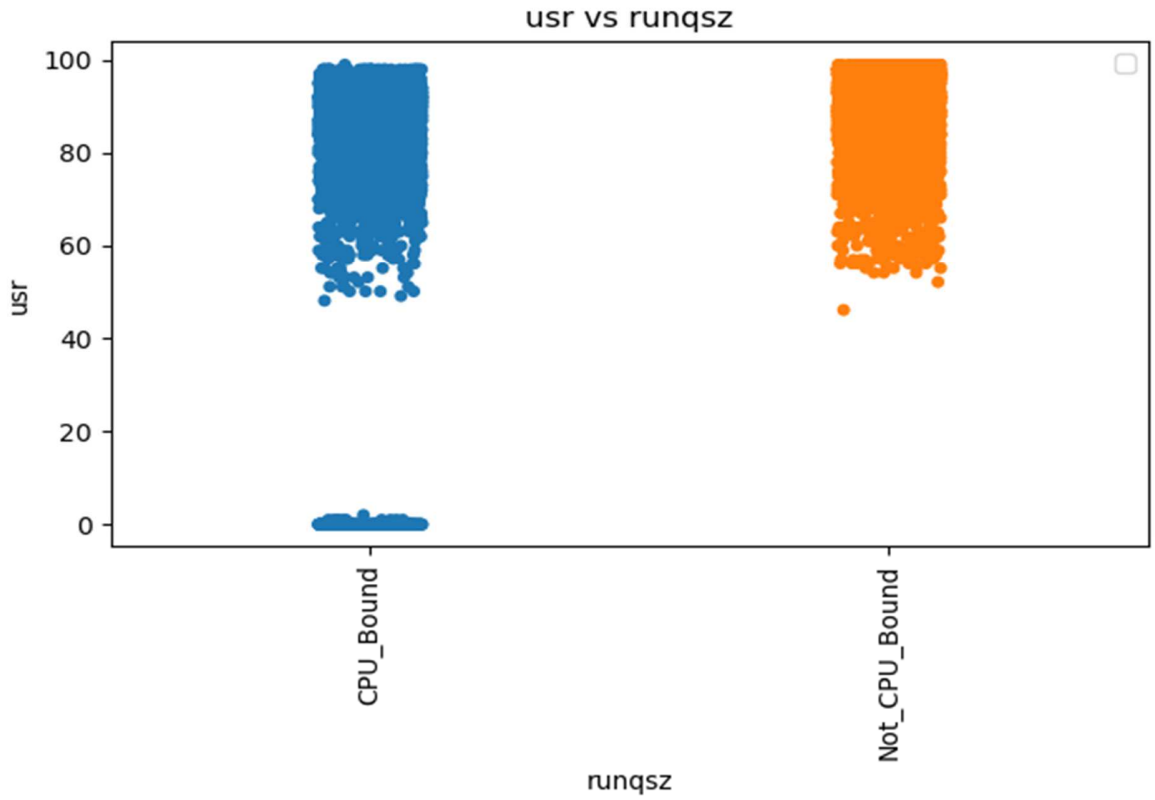


Figure 5: Bivariate analysis runsqz

Key Observations

1. For most dependent variables, the data spread with respect to both types of 'runqsz' is very similar. However, for variables like 'freemem' and 'freeswap,' there are ranges where one type of 'runqsz' shows a higher probability of occurrence.
2. Plot for 'runsqz' with 'usr' which is the variable of interest, for higher 'usr' values it is difficult to diffrentiate for the type of 'runsqz', however, for CPU_Bound type 'runsqz' only there are cases when 'usr' value is 0.

1.7 Outlier Treatment

In the dataset all the numeric attributes are continuous in nature, we will apply outlier treatment to all of them. For outlier treatment we will be imputing 95 percentile and 5 percentile values where ever the values are above or below these values.

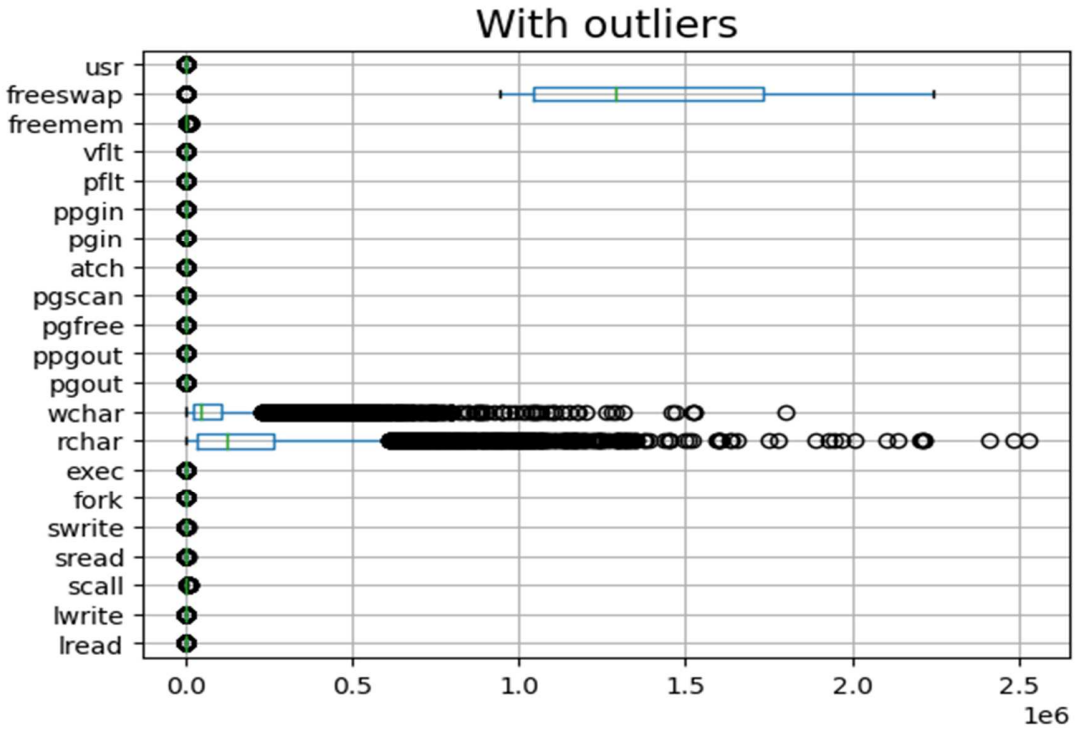


Figure 6: Boxplot numeric columns

As per the original data the range for feature 'rchar' go well over 2500000.

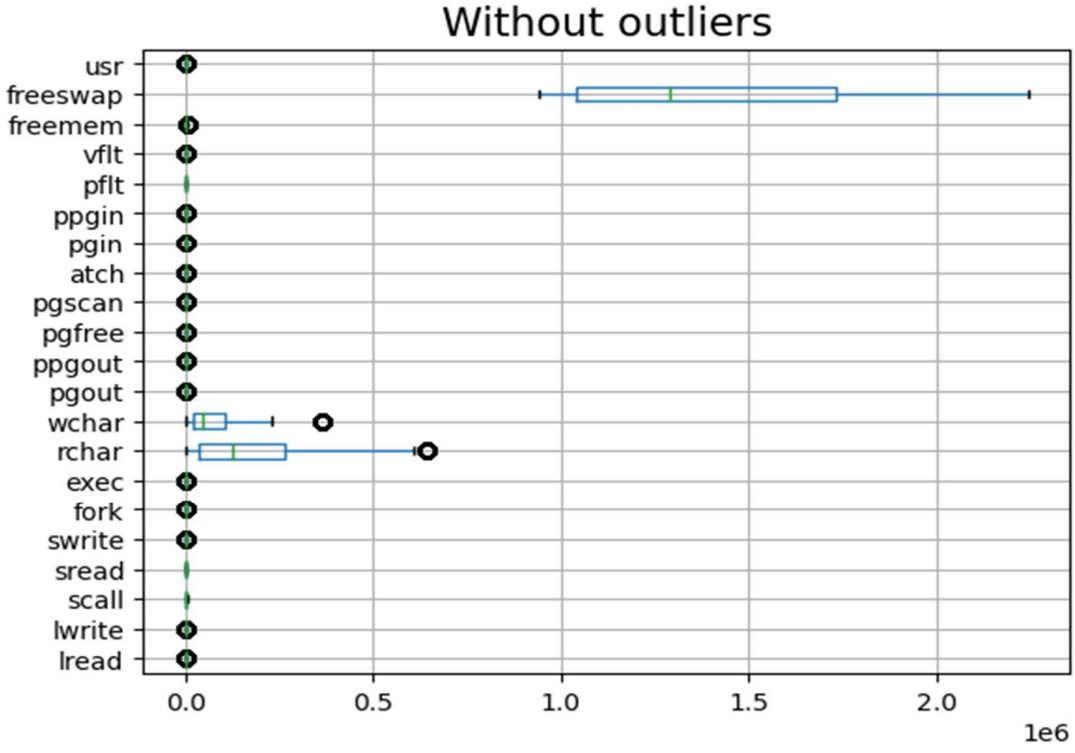


Figure 7: Boxplot outlier treated numeric columns

The outlier treatment has helped bring down the range for features like for 'rchar', it has come down to below 1000000, similarly for 'wchar' from just under 2000000 to below 500000.

1.8 Data Encoding

For data encoding we have created dummy variable by dropping using 'drop first'.

Glimpse of encoded data

lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	usr	runqsz_Not_CPU_Bound
1	0	2147	79	68	0	0	40671	53995	0	...	0	0	1	2	16	26	7567	1730946	95	0
0	0	170	18	21	0	0	448	8385	0	...	0	0	0	0	15	16	7567	1869002	97	1
15	3	2162	159	119	2	2	125473	31950	0	...	0	1	6	9	150	220	702	1021237	87	1
0	0	160	12	16	0	0	125473	8670	0	...	0	0	0	0	15	16	7567	1863704	98	1
5	1	330	39	38	0	0	125473	12185	0	...	0	0	1	1	37	47	633	1760253	90	1

Table 9: Data Overview

Based on first 5 rows of the data we can clearly see that 'runsqz' column is changed to runqsz_Not_CPU_Bound which has binary values 0 and 1 where 0 mean CPU_Bound and 1 means Not_ CPU_Bound.

1.9 Splitting Data

Here data is divided into X and Y where X contains all the independent attributes and Y has response variable. This X and Y are further split into train and test data where for this problem we have taken train to test split ratio of 80:20.

Train data

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	runqsz_Not_
1305	9	2	2128	113	84	1	1	16656	25126	0	...	70	135	0	2	3	68	116	128	1105893	
6407	0	0	230	35	14	0	0	169196	5037	0	...	0	0	0	0	0	15	16	7567	1869339	
961	1	1	166	10	13	0	0	7601	43378	0	...	0	0	0	0	0	0	3	7567	1854920	
4988	1	0	1930	113	128	0	0	134171	361894	1	...	70	135	0	5	10	55	27	136	1539870	
3017	68	66	4083	309	198	7	11	644761	139522	11	...	70	135	1	35	53	203	503	172	1095856	

Table 10: Data Overview

Test data

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	runqsz_Not_(
3566	7	0	4049	261	244	0	0	270608	186598	3	...	70	135	5	1	1	64	183	135	1528261	
899	12	12	2479	297	168	0	1	644761	361894	4	...	5	0	0	22	25	65	165	306	1014872	
2406	7	0	2144	344	296	7	1	203939	12761	0	...	0	0	0	0	0	311	448	7567	1846826	
5447	2	0	1861	223	120	0	0	232693	29510	0	...	0	0	0	4	4	56	136	434	1104610	
6215	2	1	837	42	50	0	0	3692	32235	0	...	0	0	0	1	1	32	179	771	1032677	

Table 11: Data Overview

1.10 Linear Regression

We have built multiple regression models using different techniques, optimize those models and compared them with each other on key metrics to identify the best performing model. For this we first built a basic model using Scikit-learn to identify intercept value understand the model performance.

Model using Sklearn

We implemented that provided us with the intercept value of 89.058. On checking the model performance on test and train data the key metrics values were:

For train data

- 69.76% of the variance in ‘usr’ is explained by predictors in the model.
- Root Mean Square Error (RMSE) score is: 5.472
- Mean Absolute Error (MAE) score is: 3.434

For test data

- 70.33% of the variance in usr is explained by predictors in the model.
- RMSE score is: 5.387
- MAE score is: 3.35

69.76% of the variance in 'usr' is explained by predictors in the model for the train set and for test set it is 70. 33%. Since, there is not a major variation between test set and train set R squared values we can say that our model is not suffering from under-fitting or over-fitting and is stable.

RMSE and MAE values for both test and train data are also similar meaning we have a stable model. But we will now evaluate and optimize the model by checking it for significant variables using Stats model to make it more generalized and simpler.

Model Evaluation

For model evaluation we used statsmodel library to get the model summary for which we added the intercept value represented by const to X_train data as in case of statsmodel, algorithm does not take the intercept value by default.

Model Summary Table

OLS Regression Results

Dep. Variable:	usr	**R-squared:**	0.698
Model:	OLS	**Adj. R-squared:**	0.697
Method:	Least Squares	**F-statistic:**	717.5
Date:	Sun, 09 Jun 2024	**Prob (F-statistic):**	0.00
Time:	09:12:03	**Log-Likelihood:**	-20436.
No. Observations:	6553	**AIC:**	4.092e+04
Df Residuals:	6531	**BIC:**	4.106e+04
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	89.0584	0.486	183.077	0.000	88.105	90.012
lread	-0.0536	0.007	-7.773	0.000	-0.067	-0.040
lwrite	0.0191	0.005	3.606	0.000	0.009	0.029
scall	-0.0005	7.65e-05	-6.441	0.000	-0.001	-0.000
sread	-0.0016	0.001	-1.294	0.196	-0.004	0.001
swrite	-0.0058	0.002	-3.657	0.000	-0.009	-0.003

fork	-0.1750	0.091	-1.930	0.054	-0.353	0.003
exec	-0.3180	0.035	-8.983	0.000	-0.387	-0.249
rchar	-5.023e-06	5.41e-07	-9.281	0.000	-6.08e-06	-3.96e-06
wchar	-6.421e-06	8.13e-07	-7.895	0.000	-8.02e-06	-4.83e-06
pgout	-0.2674	0.040	-6.637	0.000	-0.346	-0.188
ppgout	0.0065	0.018	0.369	0.712	-0.028	0.041
pgfree	0.0162	0.007	2.193	0.028	0.002	0.031
pgscan	-0.0018	0.002	-0.748	0.454	-0.006	0.003
atch	0.0302	0.045	0.665	0.506	-0.059	0.119
pgin	-0.0587	0.020	-3.010	0.003	-0.097	-0.020
ppgin	-0.0107	0.013	-0.828	0.408	-0.036	0.015
pflt	-0.0185	0.002	-8.397	0.000	-0.023	-0.014
vflt	-0.0110	0.002	-6.897	0.000	-0.014	-0.008
freemem	0.0001	3.75e-05	3.190	0.001	4.62e-05	0.000
freeswap	4.183e-06	2.96e-07	14.141	0.000	3.6e-06	4.76e-06
runqsz_Not_CPU_Bound	2.0102	0.149	13.477	0.000	1.718	2.303

Omnibus:	3755.895	**Durbin-Watson:**	2.005
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	34109.326
Skew:	-2.646	**Prob(JB):**	0.00
Kurtosis:	12.844	**Cond. No.**	1.03e+07

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.03e+07. This might indicate that there are
strong multicollinearity or other numerical problems.

Table 12: Model Summary Table

The model created using Statsmodel has R squared value of 0.698, meaning model could explain 69.8% variance in the train set. However, there is multicollinearity in the data which we already knew from the heatmap, we will check the multicollinearity using VIF score.

const	51.621156
lread	4.088893
lwrite	3.391719
scall	3.071692
sread	6.566072
swrite	5.538006
fork	8.986569
exec	2.616221
rchar	2.119306
wchar	1.556212
pgout	4.633536
ppgout	9.068261
pgfree	8.937199
pgscan	3.818786
atch	1.399699
pgin	9.108703
ppgin	9.349767
pflt	10.751295
vflt	15.463960
freemem	1.932632
freeswap	2.256698
runqsz_Not_CPU_Bound	1.209473
dtype: float64	

Table 13: VIF score

VIF score is a measure of multicollinearity where closer the VIF value is to 1, the lower is the multicollinearity. For a regression model a VIF value of up to 5 is acceptable as it means a moderate multicollinearity, a score above between 5 and 10 means high multicollinearity and above 10 is strong multicollinearity, for our data VIF scores for many variables are above 5 and for some even above 10.

We examined the impact on adjusted R-squared values when we dropped variables one at a time starting with variables from the X_train data that have a VIF score above 5. We permanently dropped the variable that causes the least change in the adjusted R-squared value one at a time rechecked the VIF score. We repeated this process until the VIF scores for all remaining variables in the model were below 5.

const	46.701886
lread	4.046324
lwrite	3.370343
scall	2.750611
swrite	2.919208
exec	2.335108
rchar	1.687407
wchar	1.512534
pgout	2.129013
pgscan	1.883239
atch	1.389191
pgin	1.488959
pflt	2.929465
freemem	1.924455
freeswap	2.093056
runqsz_Not_CPU_Bound	1.203833
dtype: float64	

Table 14: VIF score

Finally, we have bought down the VIF values for all the attributes belonging to the dataset below 5, now we will again display the regression summary table.

OLS Regression Results

Dep. Variable:	usr	**R-squared:**	0.693
Model:	OLS	**Adj. R-squared:**	0.693
Method:	Least Squares	**F-statistic:**	985.3
Date:	Sun, 09 Jun 2024	**Prob (F-statistic):**	0.00
Time:	16:42:54	**Log-Likelihood:**	-20482.
No. Observations:	6553	**AIC:**	4.100e+04
Df Residuals:	6537	**BIC:**	4.110e+04
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	88.5056	0.466	190.031	0.000	87.593	89.419
lread	-0.0580	0.007	-8.397	0.000	-0.072	-0.044
lwrite	0.0213	0.005	4.013	0.000	0.011	0.032
scall	-0.0005	7.28e-05	-6.793	0.000	-0.001	-0.000
swrite	-0.0094	0.001	-8.140	0.000	-0.012	-0.007
exec	-0.3853	0.034	-11.446	0.000	-0.451	-0.319
rchar	-5.831e-06	4.86e-07	-11.995	0.000	-6.78e-06	-4.88e-06

wchar	-5.222e-06	8.07e-07	-6.470	0.000	-6.8e-06	-3.64e-06
pgout	-0.2058	0.027	-7.487	0.000	-0.260	-0.152
pgscan	0.0018	0.002	1.074	0.283	-0.001	0.005
atch	0.0279	0.046	0.612	0.540	-0.061	0.117
pgin	-0.0860	0.008	-10.832	0.000	-0.102	-0.070
pflt	-0.0354	0.001	-30.636	0.000	-0.038	-0.033
freemem	9.91e-05	3.77e-05	2.628	0.009	2.52e-05	0.000
freeswap	4.696e-06	2.87e-07	16.376	0.000	4.13e-06	5.26e-06
runqsz_Not_CPU_Bound	2.0237	0.150	13.510	0.000	1.730	2.317

Omnibus:	3655.526	**Durbin-Watson:**	2.007
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	32007.315
Skew:	-2.567	**Prob(JB):**	0.00
Kurtosis:	12.532	**Cond. No.**	9.78e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.78e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

Table 15: Model Summary Table

In the regression summary table, another important parameter is the p-value. The p-value measures the hypothesis that the coefficient value for an attribute is actually 0. If the p-value is above 0.05, we are unable to reject this hypothesis and conclude that the specific variable has no significant effect on the dependent variable. In such cases, that particular attribute should be dropped. Since, in this case, there are multiple attributes with a p-value over 0.05, we will drop one attribute at a time, starting with the one with the highest p-value and moving down the order, checking p-values after dropping each variable until p-values of all the remaining variables are below 0.05.

OLS Regression Results

Dep. Variable:	usr	**R-squared:**	0.693
Model:	OLS	**Adj. R-squared:**	0.693
Method:	Least Squares	**F-statistic:**	1137.
Date:	Sun, 09 Jun 2024	**Prob (F-statistic):**	0.00
Time:	16:42:54	**Log-Likelihood:**	-20482.
No. Observations:	6553	**AIC:**	4.099e+04
Df Residuals:	6539	**BIC:**	4.109e+04
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	88.5517	0.464	190.781	0.000	87.642	89.462
lread	-0.0578	0.007	-8.367	0.000	-0.071	-0.044
lwrite	0.0212	0.005	3.994	0.000	0.011	0.032
scall	-0.0005	7.28e-05	-6.792	0.000	-0.001	-0.000
swrite	-0.0094	0.001	-8.132	0.000	-0.012	-0.007
exec	-0.3829	0.034	-11.394	0.000	-0.449	-0.317
rchar	-5.775e-06	4.82e-07	-11.972	0.000	-6.72e-06	-4.83e-06
wchar	-5.312e-06	8.04e-07	-6.610	0.000	-6.89e-06	-3.74e-06
pgout	-0.1850	0.022	-8.495	0.000	-0.228	-0.142
pgin	-0.0847	0.008	-10.862	0.000	-0.100	-0.069
pflt	-0.0354	0.001	-30.665	0.000	-0.038	-0.033
freemem	9.449e-05	3.75e-05	2.518	0.012	2.09e-05	0.000
freeswap	4.679e-06	2.86e-07	16.341	0.000	4.12e-06	5.24e-06
runqsz_Not_CPU_Bound	2.0244	0.150	13.516	0.000	1.731	2.318

Omnibus:	3657.115	**Durbin-Watson:**	2.006
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	32036.727
Skew:	-2.569	**Prob(JB):**	0.00
Kurtosis:	12.536	**Cond. No.**	9.75e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.75e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

Table 16: Model Summary Table

Finally, we have optimized our model by bring the VIF scores for all the variables in the model below 5 and dropping the attributes which were not contributing towards the model.

We built the regression model using updated dataset which is made by dropping 'const' column as sklearn package has intercept by default, we will check for R-square, RMSE and MAE value on test and train data on to understand stability of the model.

For train data

- 69.33% of the variance in 'usr' is explained by predictors in the model.
- RMSE score is: 5.511
- MAE score is: 3.467

For test data

- 69.95% of the variance in usr is explained by predictors in the model.
- RMSE score is: 5.436
- MAE score is: 3.371

Since all the values are almost similar for both test and train data, we can conclude that we have a stable model which is our first model. However, in the model summary table above we had a very high condition number implying that there is still high multicollinearity present in the data which could have an adverse effect on the model, to overcome that effect built a model by applying PCA to original data (X) containing independent attributes.

PCA Model

In order to do PCA data has to be scaled, we applied Z scaling method to scale the data which ensures that mean for all attributes is 0 and standard deviation is 1

Statistical summary of scaled data

	count	mean	std	min	25%	50%	75%	max
lread	8192.0	-1.778092e-17	1.000061	-0.771303	-0.670828	-0.419641	0.233446	2.644843
lwrite	8192.0	-4.943962e-17	1.000061	-0.561833	-0.561833	-0.519233	-0.135842	2.249707
scall	8192.0	1.734723e-18	1.000061	-1.399624	-0.816365	-0.144938	0.672626	2.901510
sread	8192.0	-8.673617e-18	1.000061	-1.343130	-0.784374	-0.225619	0.563624	2.582130
swrite	8192.0	6.505213e-18	1.000061	-1.315336	-0.759085	-0.222701	0.452747	2.508887
fork	8192.0	1.301043e-17	1.000061	-0.653360	-0.653360	-0.653360	0.238917	2.469609
exec	8192.0	-2.081668e-17	1.000061	-0.632684	-0.632684	-0.310401	0.011881	2.912423
rchar	8192.0	7.806256e-18	1.000061	-1.006180	-0.813415	-0.308336	0.471595	2.586204
wchar	8192.0	-1.691355e-17	1.000061	-0.844052	-0.636232	-0.407499	0.167379	2.642836
pgout	8192.0	-1.604619e-17	1.000061	-0.528450	-0.528450	-0.528450	0.027306	2.528206
ppgout	8192.0	3.469447e-17	1.000061	-0.510495	-0.510495	-0.510495	-0.164856	2.254618
pgfree	8192.0	2.645453e-17	1.000061	-0.514480	-0.514480	-0.514480	-0.330558	2.060436
pgscan	8192.0	-2.602085e-18	1.000061	-0.520069	-0.520069	-0.520069	-0.520069	1.922823
atch	8192.0	1.647987e-17	1.000061	-0.453100	-0.453100	-0.453100	-0.453100	2.381197
pgin	8192.0	1.734723e-17	1.000061	-0.685380	-0.685380	-0.494347	0.174266	2.657685
ppgin	8192.0	1.908196e-17	1.000061	-0.673417	-0.673417	-0.485798	0.139600	2.641191
pflt	8192.0	-1.734723e-17	1.000061	-1.038884	-0.791517	-0.415519	0.534371	2.533098
vflt	8192.0	4.336809e-18	1.000061	-1.058995	-0.789945	-0.341528	0.441707	2.492467
freemem	8192.0	-1.474515e-17	1.000061	-0.694307	-0.624610	-0.486798	0.076823	2.280518
freeswap	8192.0	-5.204170e-18	1.000061	-1.222752	-0.931782	-0.214228	1.068909	2.560673
runqsz_Not_CPU_Bound	8192.0	-4.250073e-17	1.000061	-1.059118	-1.059118	0.944182	0.944182	0.944182

Table 17: Statistical Summary

Applying PCA

On applying the PCA we get eigen vector and eigen value where eigen vector represents the direction of maximum variance in the data explained by a PC for each feature and eigen value represents the percentage of variance in data explained by a PC.

Initially for applying we take principal components equal to the number of numeric columns which in this case is 21 and get the eigen vector and eigen values, the output received is in the form of array.

```
array([[-2.97, -3.96, -0.41, ...,  1.11,  2.35, -1.93],
       [ 0.07, -0.48,  0.62, ..., -0.56, -1.01,  0.35],
       [-0.18, -1.45, -0.38, ...,  1.55,  1.09, -0.75],
       ...,
       [ 0.  , -0.01,  0.04, ..., -0.02, -1.27,  0.03],
       [ 0.05, -0.01,  0.05, ..., -0.04, -0.11,  0.07],
       [ 0.01,  0.04, -0.11, ...,  0.26, -0.12, -0.  ]])
```

Table 18: Eigen vectors

```
array([7.39638722, 2.99572523, 1.83601076, 1.60903442, 1.32364427,
       1.17767411, 0.89070238, 0.78574404, 0.64195312, 0.48652962,
       0.38714062, 0.34492286, 0.31078056, 0.1986425 , 0.14305005,
       0.1294473 , 0.09674215, 0.08062925, 0.06448604, 0.05724168,
       0.04607563])
```

Table 19: Eigen values

We made a scree plot using eigen values to identify optimum number of principal components

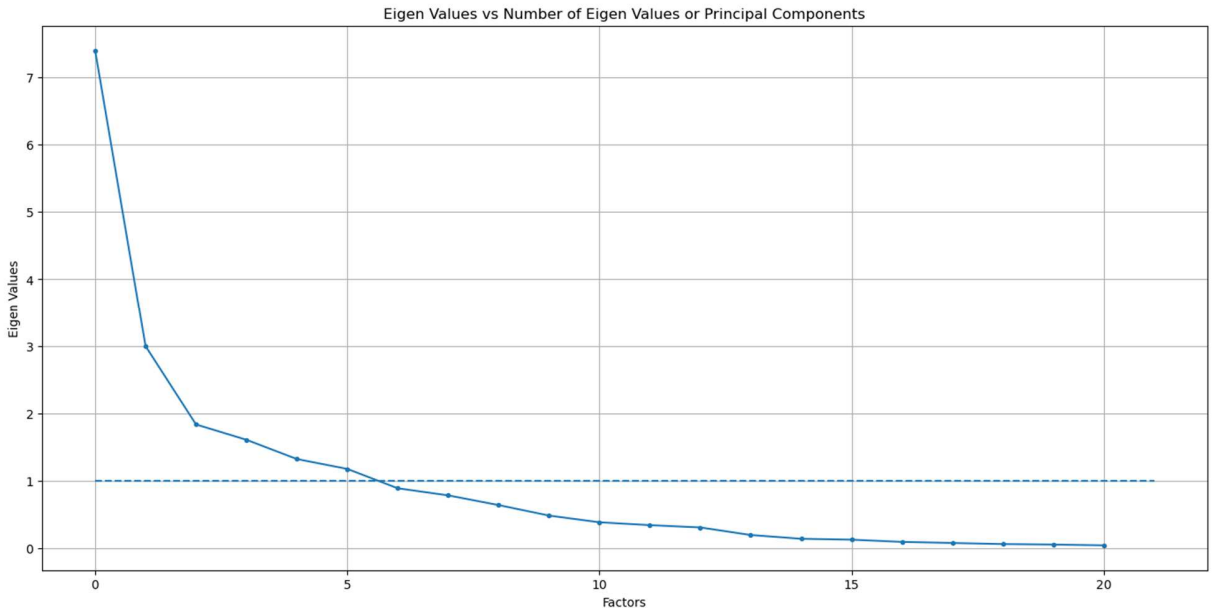


Figure 8: Scree plot

In PCA, two methods are commonly used to determine the optimum number of components:

1. Eigenvalues: In this method, eigenvalues are plotted, and the number of components with eigenvalues above 1 is considered the optimum number of components. In our case, this method suggests that 5 components are optimal.
2. Explained Variance: This method examines how much variance each principal component explains. The cumulative sum of the explained variances is then calculated, and the optimum number of components is determined based on a predefined threshold.

For our analysis, we will set the threshold at 95%, meaning we will select the components that collectively explain at least 95% of the variance in the data.

```
explained variance ratio

array([0.35216592, 0.14263617, 0.08741841, 0.07661133, 0.06302299,
       0.05607287, 0.04240922, 0.03741182, 0.03056546, 0.02316525,
       0.01843302, 0.01642289, 0.01479727, 0.00945801, 0.00681108,
       0.0061634 , 0.00460621, 0.00383902, 0.00307039, 0.00272546,
       0.00219381])

cumulative variance explained ratio

array([0.35216592, 0.49480209, 0.5822205 , 0.65883183, 0.72185482,
       0.77792769, 0.82033692, 0.85774873, 0.8883142 , 0.91147944,
       0.92991246, 0.94633535, 0.96113262, 0.97059063, 0.97740171,
       0.98356511, 0.98817132, 0.99201034, 0.99508073, 0.99780619,
       1.        ])
```

Table 20: Explained variance ratio

from cumulative variance explained ratio we can find that first 5 components which had eigen value of above 1 can explain only 72% variance but our threshold is 95%, to find the optimum number of principal components we will make a plot based on explained variance and cumulative explained variance.

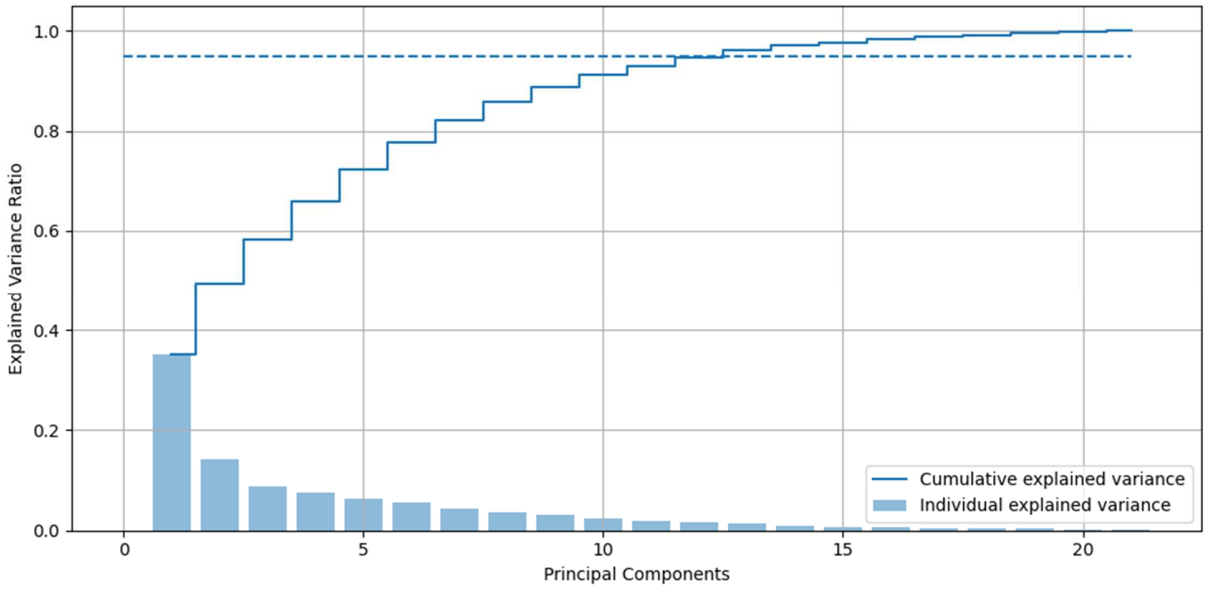


Figure 9: Scree plot

In the above plot 95% is marked by the horizontal line and optimum number of components are those for which cumulatively explained variance is above 95% which as per the above plot is 13. We will continue by taking number of components as 13.

Based on the above results we again applied PCA this time taking number of components as 13 and we loaded the components of 13 PCs for each of the columns of the original data set where values in cells of the table below represents how much variance of a columns is explained by that PC.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
lread	0.177682	0.094365	0.098504	0.629964	0.009214	-0.051109	-0.009508	-0.040567	0.011846	-0.012475	0.029949	0.015414	-0.028255
lwrite	0.077871	0.042738	0.172898	0.708744	-0.022112	-0.117221	0.061515	-0.059808	-0.007220	-0.025703	-0.035827	-0.011834	0.043067
scall	0.260331	0.165057	0.211680	-0.116155	-0.179008	0.029390	0.011837	-0.161198	0.232656	0.210106	0.530819	0.175654	0.087579
sread	0.276037	0.181206	0.255850	-0.155744	-0.066634	-0.062724	0.203091	-0.167725	0.188910	-0.138356	0.040672	0.021176	0.049169
swrite	0.255637	0.189966	0.258015	-0.152509	-0.114499	-0.064151	0.226043	-0.255327	0.239170	0.123263	-0.181584	-0.109841	0.065416
fork	0.236591	0.325807	-0.316553	-0.027686	0.001762	-0.007703	0.011313	-0.003250	-0.057830	0.000243	-0.259001	-0.071235	-0.020959
exec	0.208435	0.222278	-0.330159	0.054486	0.002969	0.072082	-0.217689	0.206039	-0.257880	0.261131	0.531748	0.099087	0.153461
rchar	0.215137	0.076143	0.190893	-0.107195	0.325188	-0.165722	0.121725	0.310146	-0.181001	-0.704068	0.181384	0.173602	0.102696
wchar	0.136698	0.010695	0.378901	-0.111349	0.295179	-0.276922	0.183573	0.222529	-0.487763	0.513769	-0.099783	-0.196530	-0.069951
pgout	0.235553	-0.328848	-0.126150	-0.015441	-0.065046	-0.212108	0.044600	-0.005553	0.010123	0.118323	-0.186143	0.552869	-0.019578
ppgout	0.249095	-0.341481	-0.151854	-0.015767	-0.018362	-0.200811	0.032520	-0.138643	-0.069326	0.045721	-0.053721	0.281808	-0.011553
pgfree	0.250081	-0.336167	-0.165297	-0.018612	-0.028653	-0.180762	0.009766	-0.208189	-0.093041	-0.040442	0.074647	-0.102952	-0.021473
pgscan	0.231609	-0.305003	-0.151990	-0.016915	-0.051673	-0.111446	-0.027529	-0.245015	-0.102253	-0.181604	0.231141	-0.644013	-0.023632
atch	0.167825	-0.172966	-0.043189	0.020878	-0.154424	-0.225229	-0.036700	0.706205	0.544163	0.094819	-0.018778	-0.229920	-0.011083
pgin	0.225706	-0.178488	0.042608	0.029755	0.403330	0.465022	-0.046685	0.004108	0.159319	0.097137	-0.054682	-0.004810	0.040426
ppgin	0.226944	-0.194817	0.041232	0.033782	0.415799	0.434763	-0.030606	-0.008993	0.146589	0.065499	-0.048869	-0.026399	0.075079
pflt	0.248365	0.321046	-0.289666	-0.019579	0.017736	-0.016849	0.005799	0.016679	-0.034555	-0.021417	-0.239471	-0.036678	-0.050121
vflt	0.286990	0.265537	-0.246743	-0.013056	0.015696	0.056713	0.001470	0.017358	-0.013450	-0.055964	-0.206625	-0.046823	-0.123054
freemem	-0.183364	0.101535	-0.232588	0.037223	0.442802	-0.227121	0.311503	-0.120921	0.284400	0.073504	0.279558	0.046425	-0.605224
freeswap	-0.201324	0.085685	-0.208192	0.006825	0.396800	-0.385083	-0.005919	-0.167751	0.224671	0.098019	-0.037981	-0.049379	0.701414
nqsz_Not_CPU_Bound	-0.078260	-0.099430	-0.239455	0.081412	-0.188117	0.274871	0.837940	0.150007	-0.111925	0.028622	0.082638	-0.021295	0.242256

Table 21: Components of selected PCs for original dataset columns

Using these component values for each attribute we replaced these attributes with the principal components.

A glimpse at the updated data frame

	const	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
1305	1.0	-0.247809	-1.402192	-0.418225	-0.395010	-1.203423	-0.134458	-1.543361	-1.296092	-0.684898	-0.507836	0.517966	-1.730049	-0.619269
6407	1.0	-3.739769	-0.391546	-1.266267	-0.098442	1.182155	-0.596927	0.942310	0.037499	0.375898	-0.497502	0.495248	0.175025	-0.277796
961	1.0	-3.982823	-0.561836	-1.270427	0.055792	0.997941	-0.533322	0.863985	-0.122201	0.310053	0.343209	0.320785	-0.052692	-0.421106
4988	1.0	0.106925	-2.022571	1.028733	-1.170920	0.700780	-1.538064	-0.707249	-0.814107	-1.846345	0.928471	0.072008	-2.053900	0.192579
3017	1.0	7.620300	-0.644178	-1.091430	2.539405	2.086697	0.208043	-0.917508	-0.264161	-0.976318	-0.560377	0.729111	1.014764	0.268395

Table 22: Data overview

We also checked the correlation between principal components

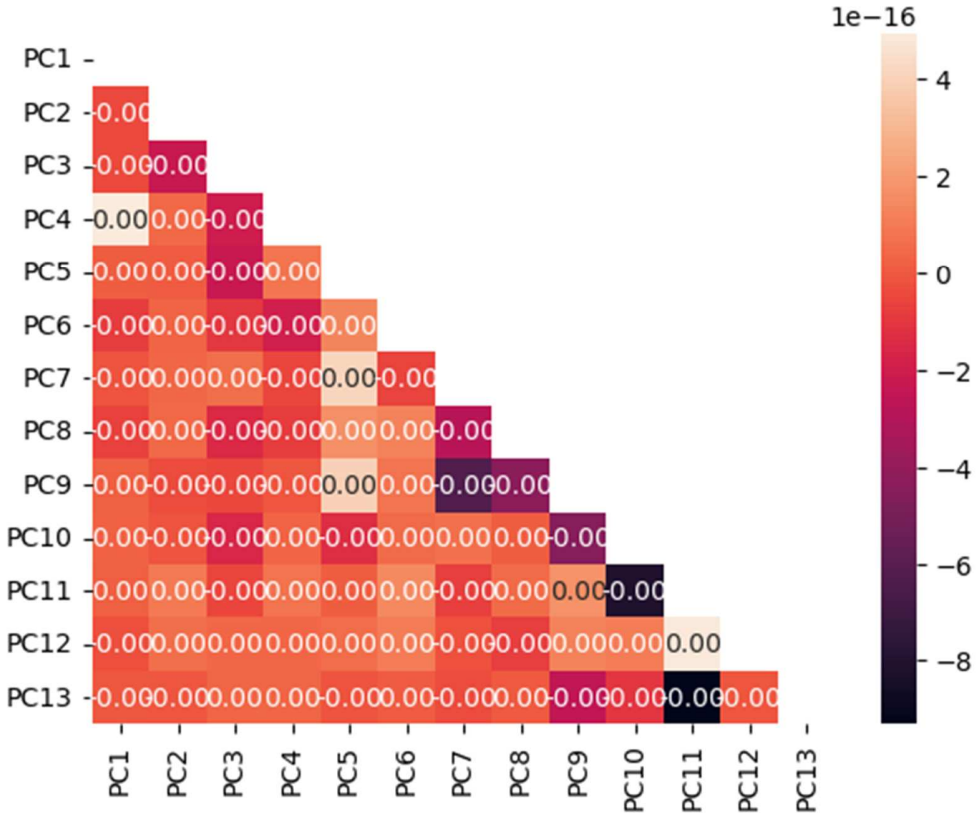


Figure 10: Heatmap

There is no correlation, we will now proceed towards making regression model from this dataset by splitting it into train and test as per the pre-defined proportion of 80:20.

Model using Statsmodel

OLS Regression Results

Dep. Variable:	usr	**R-squared:**	0.694
Model:	OLS	**Adj. R-squared:**	0.694
Method:	Least Squares	**F-statistic:**	1143.
Date:	Sun, 09 Jun 2024	**Prob (F-statistic):**	0.00
Time:	16:42:56	**Log-Likelihood:**	-20470.
No. Observations:	6553	**AIC:**	4.097e+04
Df Residuals:	6539	**BIC:**	4.106e+04
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	86.1806	0.068	1266.479	0.000	86.047	86.314
PC1	-2.8501	0.025	-114.012	0.000	-2.899	-2.801
PC2	-1.5082	0.039	-38.430	0.000	-1.585	-1.431
PC3	0.0271	0.050	0.537	0.591	-0.072	0.126
PC4	0.1331	0.053	2.491	0.013	0.028	0.238
PC5	-0.1111	0.059	-1.888	0.059	-0.227	0.004
PC6	-0.3703	0.063	-5.911	0.000	-0.493	-0.248
PC7	0.7361	0.072	10.221	0.000	0.595	0.877
PC8	-0.5307	0.076	-6.955	0.000	-0.680	-0.381
PC9	0.6364	0.084	7.539	0.000	0.471	0.802
PC10	-0.0046	0.097	-0.048	0.962	-0.194	0.185
PC11	0.2931	0.109	2.679	0.007	0.079	0.507
PC12	-0.6312	0.115	-5.472	0.000	-0.857	-0.405
PC13	1.1048	0.122	9.037	0.000	0.865	1.344

Omnibus:	3693.169	**Durbin-Watson:**	2.006
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	32136.590
Skew:	-2.606	**Prob(JB):**	0.00
Kurtosis:	12.515	**Cond. No.**	4.89

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Table 23: Model summary table

Checking the VIF score

```
const    1.000676
PC1      1.000389
PC2      1.000413
PC3      1.000757
PC4      1.000398
PC5      1.000469
PC6      1.000339
PC7      1.000620
PC8      1.000810
PC9      1.000291
PC10     1.000356
PC11     1.000455
PC12     1.000459
PC13     1.000228
dtype: float64
```

Table 24: VIF score

For the model made using PCA data we checked for VIF score which for all the components is nearly 1 meaning there is no multicollinearity, however, as per the model summary table there are components for whom p-value is above 0.05, we will drop them one by one check the p-value until it is below 0.05 for all attributes.

OLS Regression Results

Dep. Variable:	usr	**R-squared:**	0.694
Model:	OLS	**Adj. R-squared:**	0.694
Method:	Least Squares	**F-statistic:**	1485.
Date:	Sun, 09 Jun 2024	**Prob (F-statistic):**	0.00
Time:	16:42:56	**Log-Likelihood:**	-20472.
No. Observations:	6553	**AIC:**	4.097e+04
Df Residuals:	6542	**BIC:**	4.104e+04
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	86.1809	0.068	1266.594	0.000	86.048	86.314
PC1	-2.8503	0.025	-114.019	0.000	-2.899	-2.801
PC2	-1.5084	0.039	-38.433	0.000	-1.585	-1.431
PC4	0.1328	0.053	2.487	0.013	0.028	0.238
PC6	-0.3706	0.063	-5.915	0.000	-0.493	-0.248
PC7	0.7368	0.072	10.231	0.000	0.596	0.878

PC8	-0.5318	0.076	-6.970	0.000	-0.681	-0.382
PC9	0.6370	0.084	7.546	0.000	0.472	0.803
PC11	0.2930	0.109	2.679	0.007	0.079	0.507
PC12	-0.6310	0.115	-5.470	0.000	-0.857	-0.405
PC13	1.1019	0.122	9.013	0.000	0.862	1.342

Omnibus:	3661.606	**Durbin-Watson:**	2.006
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	31210.393
Skew:	-2.585	**Prob(JB):**	0.00
Kurtosis:	12.358	**Cond. No.**	4.89

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Table 25: Model summary table

in the above table, for all the components the p-value is below 0.05 meaning we have an optimum model based on PCA data using which we created the model using LinearRegression from sklearn to evaluate its performance on train and test data.

Model performance on train data

- 69.42% of the variance in 'usr' is explained by predictors in the model.
- RMSE score is: 5.502
- MAE score is: 3.492

Model performance on test data

- 69.93% of the variance in 'usr' is explained by predictors in the model.
- RMSE score is: 5.439
- MAE score is: 3.436

Key metrics scores for both the test and train data are almost identical meaning we have a stable model which will be our second model.

In the first summary table displayed in Statsmodel, coefficient values for some attributes were very small due to which scientific notations of e-06 or e-05 required to be used meaning they had minimalistic effect on the model and some of these coefficients were present even in the optimized model. We feel that there are chances that these coefficients have no effect on the model and to understand it better we will build a model applying regularization techniques.

Regularization

Regularization is a technique that helps reduce the effect of noise in a model helping make more consistent model by adding penalty term to loss function. There are two types of penalty:

1. Ridge also referred as l2

2. Lasso referred as l1

Using Ridge Method

We have to first find the right penalty value which we did using RidgeCV.

Optimal alpha for Ridge: 14.8

Using this optimum penalty value referred to as alpha we created a regression model using Ridge from sklearn.linear_model.

Model performance for Ridge model

For train data

- 69.76% of the variance in 'usr' is explained by predictors in the model.
- RMSE score is: 5.472
- MAE score is: 3.432

For test data

- 70.33% of the variance in 'usr' is explained by predictors in the model.
- RMSE score is: 5.387
- MAE score is: 3.344

Model performance for both train and test data are almost similar meaning we have a stable model which is our third model.

Using Lasso Method

We have to first find the right penalty value which we did using LassoCV.

Optimal alpha for Lasso: 0.05

Using this optimum penalty value referred to as alpha we created a regression model using Lasso from sklearn.linear_model.

Model performance for Lasso model

For train data

- 69.74% of the variance in 'usr' is explained by predictors in the model.
- RMSE score is: 5.473
- MAE score is: 3.413

For test data

- 70.31% of the variance in 'usr' is explained by predictors in the model.
- RMSE score is: 5.389
- MAE score is: 3.33

Model performance for both train and test data are almost similar meaning we have a stable model which is our fourth model.

1.11 Model Comparison

We have created 4 models using different techniques compared each models performance for test and train data using key metrics and have found that all the models are stable now we will compare these models with each other to find the best model based on R-squared value.

	RSquared
Ridge	0.693461
Lasso	0.693455
PCA	0.691831
Model1	0.690863

Table 26: Model comparision

From the above table, we can conclude that the model created by regularizing the data using the Ridge method has the highest R-square value. Therefore, we can identify it as the best model.

Next, we will print the regression equation for the Ridge method and display the attributes along with their coefficients, sorted in descending order of coefficient values.

```
Ridge Regression Equation:
usr = 89.5666 + (-0.0707 * lread) + (0.0373 * lwrite) + (-0.0003 * scall) + (-0.0020 * sread) + (-0.0106 * swrite) + (-0.2016 *
fork) + (-0.4311 * exec) + (-0.0000 * rchar) + (-0.0000 * wchar) + (-0.2632 * pgout) + (0.0675 * ppgout) + (-0.0053 * pgfree) +
(-0.0087 * pgscan) + (0.0928 * atch) + (-0.0488 * pgin) + (-0.0137 * ppgin) + (-0.0162 * pflt) + (-0.0095 * vflt) + (0.0001 * f
reemem) + (0.0000 * freeswap) + (1.7379 * runqsz_Not_CPU_Bound)
```

Equation 4: Ridge regression equation

Coefficient value by attributes sorted in descending order

	Coefficient
Intercept	89.566590
runqsz_Not_CPU_Bound	1.737918
atch	0.092785
ppgout	0.067453
lwrite	0.037346
freemem	0.000085
freeswap	0.000004
wchar	-0.000005
rchar	-0.000006
scall	-0.000294
sread	-0.001977
pgfree	-0.005323
pgscan	-0.008694
vflt	-0.009460
swrite	-0.010640
ppgin	-0.013735
pflt	-0.016207
pgin	-0.048839
lread	-0.070672
fork	-0.201552
pgout	-0.263236
exec	-0.431057

Table 27: Coefficient values

1.12 Conclusion

1. Final regression equation is: usr = 89.5666 + (-0.0707 * lread) + (0.0373 * lwrite) + (-0.0003 * scall) + (-0.0020 * sread) + (-0.0106 * swrite) + (-0.2016 * fork) + (-0.4311 * exec) + (-0.0000 * rchar) + (-0.0000 * wchar) + (-0.2632 * pgout) + (0.0675 * ppgout) + (-0.0053 * pgfree) + (-0.0087 * pgscan) + (0.0928 * atch) + (-0.0488 * pgin) + (-0.0137 * ppgin) + (-0.0162 * pflt) + (-0.0095 * vflt) + (0.0001 * freemem) + (0.0000 * freeswap) + (1.7379 * runqsz_Not_CPU_Bound)

Equation 5: Ridge regression equation

2. As per this regression equation we can conclude

- In terms of coefficient value impact of runqsz_Not_CPU_Bound is highest that is if 'runsqz' is 'Not_CPU_Bound' then 'usr' value increases by 1.738 meaning CPUs run 1.738% Portion of time more in user mode provided all other attributes remain constant.
- While 'runsqz_Not_CPU_Bound' is a categorical attribute meaning its value can either be 0 or 1, for continuous variables whose values can differ significantly we find that in absolute terms 'exec' has highest impact followed by 'pgout' and their impact is negative meaning provided that other attributes remain constant every unit increase in 'exec' reduces CPU run time in user mode by 0.431% and for 'pgout' this decrease is 0.263%.
- For continuous attribute which increase CPU run time in user mode, 'atch' has the highest coefficient value of 0.093, that is for every one unit increase in 'atch' if others remain constant then CPU run time in user mode increases by 0.093%.

Key Takeaways

While we have discussed the impact of different attributes in predicting 'usr' in terms of the regression equation and coefficient values, it is essential to consider the business context. The scale of values for some variables varies significantly, as highlighted by the statistical summary. For many attributes, the median value is 0, while for a few, it is in the thousands. This range difference indicates that some attributes have higher variance. Despite having lower coefficient values, these attributes can have a significant impact on the equation when multiplied by their respective attribute values as they are highly sensitive. Consequently, they will significantly affect the 'usr' value and play a larger role in determining the percentage of CPU run time in user mode.

In a real-world environment, multiple attributes change together, and the magnitude of change for these sensitive attributes will be significantly higher, thus affecting 'usr' considerably more.

We will examine this effect with the help of a table that outlines the impact each variable has according to the five-point summary.

	Coefficient	min	25%	50%	75%	max
freeswap	0.000004	3.766998	4.161733	4.529277	6.912509	8.964651
runqsz_Not_CPU_Bound	1.737918	0.000000	0.000000	1.737918	1.737918	1.737918
freemem	0.000085	0.004658	0.019139	0.048101	0.166404	0.640804
lwrite	0.037346	0.000000	0.000000	0.037346	0.373462	2.464848
pgout	-0.263236	-0.000000	-0.000000	-0.000000	-0.526471	-2.895591
atch	0.092785	0.000000	0.000000	0.000000	0.000000	0.463926
pgscan	-0.008694	-0.000000	-0.000000	-0.000000	-0.000000	-1.173705
pgfree	-0.005323	-0.000000	-0.000000	-0.000000	-0.026617	-0.372633
ppgout	0.067453	0.000000	0.000000	0.000000	0.269811	2.158484
fork	-0.201552	-0.000000	-0.000000	-0.000000	-0.403105	-1.410867
ppgin	-0.013735	-0.000000	-0.000000	-0.041204	-0.192287	-0.727944
pgin	-0.048839	-0.000000	-0.000000	-0.097678	-0.439551	-1.709366
wchar	-0.000005	-0.007628	-0.117599	-0.237414	-0.539690	-1.842918
sread	-0.001977	-0.011859	-0.171957	-0.330078	-0.555401	-1.120686
exec	-0.431057	-0.000000	-0.000000	-0.431057	-0.862114	-4.741625
lread	-0.070672	-0.000000	-0.141344	-0.494706	-1.413445	-4.805712
scall	-0.000294	-0.032063	-0.300330	-0.606248	-0.985411	-1.990823
rchar	-0.000006	-0.001552	-0.195720	-0.701476	-1.513416	-3.599326

pflt	-0.016207	-0.000000	-0.388962	-1.037231	-2.576872	-5.850633
vflt	-0.009460	-0.000000	-0.425711	-1.135229	-2.383982	-5.619386
swrite	-0.010640	-0.074480	-0.670321	-1.255522	-1.979043	-4.170887
Intercept	89.566590	NaN	NaN	NaN	NaN	NaN

Table 28: Attributes magnitude of change

As expected, 'freeswap', which has the lowest coefficient value in absolute terms, will have the highest values in the regression equation due to its sensitive nature. According to our data, 'freeswap' has a minimum value of 3.766, a maximum of 8.96, and a median of 4.53, meaning it accounts for 3.766% to 8.96% of CPU run time in user mode.

As per the above table, the top three attributes that are pushing up the CPU run time in user mode in median terms are:

1. freeswap
2. runqsz_Not_CPU_Bound
3. freemem

and bringing CPU run time down in user mode down in median terms are:

1. swrite
2. vflt
3. pflt

Problem 2

2.1 Background Information

The Ministry of Health of the Republic of Indonesia has launched an initiative aimed at gaining a better understanding of contraceptive use among married women and identifying the key factors that influence their decisions. This study will provide valuable insights towards informed policy decisions aimed at improving public health.

2.2 Business Context

Study different demographic and socio-economic factor and create a predictive model that could help identify whether married women (who are either not pregnant or are uncertain of their pregnancy status) choose to use a contraceptive method. This insight would help the Ministry of Health of the Republic of Indonesia in making informed policy decisions aimed at improving public health.

2.3 Problem Statement

The goal is to identify key demographic and socio-economic factor that could help classify whether married women (who are either not pregnant or are uncertain of their pregnancy status) opt for a contraceptive method of choice. Using this understanding build classification models using different

techniques that could help predict whether these women opt for a contraceptive method of choice and identify the best model based on model evaluation metrics.

2.4 METHODOLOGY

Import the libraries - Load the data - Check the structure of the data - Check the types of the data – Check for and treat missing values - Check the statistical summary - Check for and treat (if needed) data irregularities – Univariate Analysis – Bivariate Analysis – Data Encoding – Data Splitting – Apply classification models – Evaluate models – Compare models– Identify important features – Conclusion.

Key Points

1. **Data Collection**: Data was provided by The Ministry of Health of the Republic of Indonesia.
2. **Data Cleaning and Pre-processing:** Dataset was checked for duplicates, missing values, bad data and outliers. There were missing values and duplicates in the data, additionally, column names for some columns were not inline with common nomenclature practice, all these anomalies were treated during pre-processing.
3. **Univariate Analysis:** All attributes were analyzed using boxplot and histogram to understand distribution, central tendency and variability of variables.
4. **Bivariate Analysis:** All the variables were examined with the aim of gaining deeper insights about demographic and socio-economic features.
5. **Visualization Techniques:** In the report we have used histograms, boxplot and countplot for univariate analysis, in bivariate analysis, to understand correlation between numeric variables heatmap and pair plot are used, violin plot and tables are used to understand relationship between categorical and numeric variable.
6. **Tools and Software:** We have carried out the analysis using programming language python on Jupyter notebook. For this analysis Python libraries Numpy, Pandas, Matplotlib, Seaborn, Scikit-learn, math and statistics were used.

2.5 Data Overview

1. **Data Description:** Dataset has 1473 rows and 10 columns

```
shape of the dataset
--------------------------------------------------------------------------------

(1473, 10)
```

Table 29: Dataset shape

2. **Data Information:** In the 10 columns 7 are object type, 2 are float and 1 is int64 type.

```
information of features
--------------------------------------------------------------------------------
<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 30: Dataset information

3. **Missing Data:** There are missing values in the dataset.

```
missing values
--------------------------------------------------------------------------------

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

Table 31: Missing values

4. **Duplicate Data:** There are duplicate values in the dataset

```
checking for duplicates
--------------------------------------------------------------------------------
number of duplicate rows: 80
```

Table 32: Data duplicates

5. **Statistical Summary:**

	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 33: Statistical Summary

6. **Frequency Distribution of Categorical Columns:**

```
value counts for Wife_ education
--------------------------------------------------------------------------------
Wife_ education
Tertiary      577
Secondary     410
Primary       334
Uneducated    152
Name: count, dtype: int64


value counts for Husband_education
--------------------------------------------------------------------------------
Husband_education
Tertiary      899
Secondary     352
Primary       178
Uneducated     44
Name: count, dtype: int64
```

```
value counts for Wife_religion
--------------------------------------------------------------------------------
Wife_religion
Scientology        1253
Non-Scientology     220
Name: count, dtype: int64


value counts for Wife_Working
--------------------------------------------------------------------------------
Wife_Working
No     1104
Yes     369
Name: count, dtype: int64


value counts for Standard_of_living_index
--------------------------------------------------------------------------------
Standard_of_living_index
Very High    684
High         431
Low          229
Very Low     129
Name: count, dtype: int64


value counts for Media_exposure
--------------------------------------------------------------------------------
Media_exposure
Exposed        1364
Not-Exposed     109
Name: count, dtype: int64


value counts for Contraceptive_method_used
--------------------------------------------------------------------------------
Contraceptive_method_used
Yes    844
No     629
Name: count, dtype: int64
```

Table 34: Frequency Distribution categorical columns

Key Observations

1. Dataset has 1473 rows and 10 columns. According to the data dictionary, we expected 6 columns to contain numeric data and 4 to contain object data. However, only 3 columns have numeric data. Despite this discrepancy, the impact is minimal as the columns in question are all categorical in nature.
2. From the statistical summary we can conclude that mean and median values are almost similar meaning that there are no significant outliers.

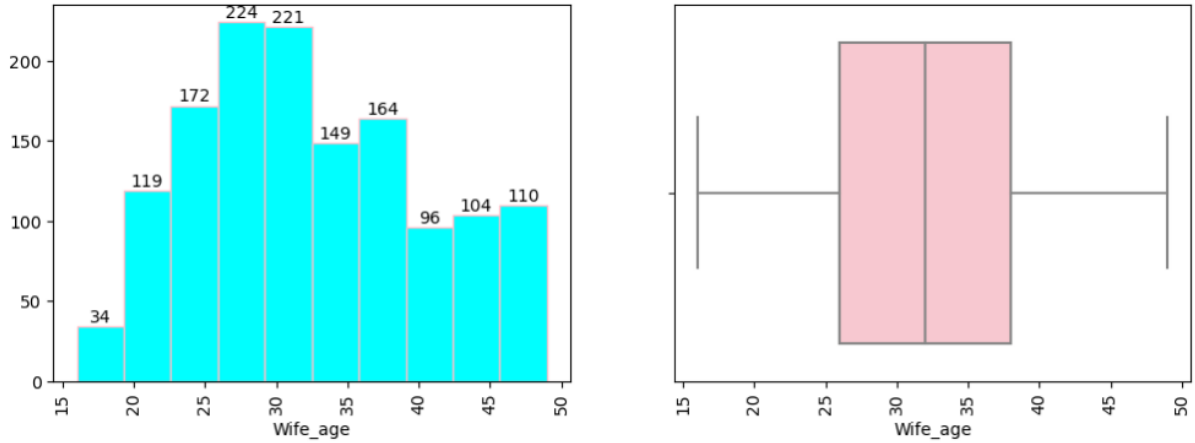
3. There are duplicates and missing values in the dataset, we will have to remove the duplicates and impute missing values during pre-processing phase.
4. There are no bad data in the dataset, however, column name for columns 'Wife_ education ', 'Wife_Working' and 'Husband_Occupation' are not in line with the column nomenclature practice, we will have renamed these columns to 'Wife_education', 'Wife_working' and 'Husband_occupation'.
5. Since the numeric data does not exhibit significant variance in range and the units are similar, scaling will not be required.

2.6 Exploratory Data Analysis

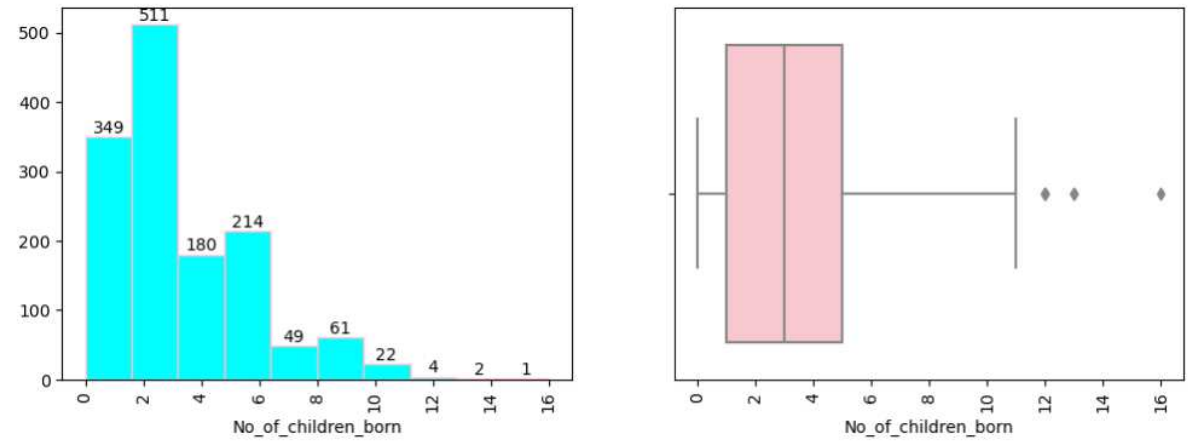
2.6.1 Univariate Analysis

Analysis of Numeric Columns

```
Skewness of Wife_age:  0.27590764346223623
Distribution of Wife_age
--------------------------------------------------------------------------------
```




```
Skewness of No_of_children_born:  1.101235327654049
Distribution of No_of_children_born
--------------------------------------------------------------------------------
```



```
Skewness of Husband_occupation:  -0.15494229804229273
Distribution of Husband_occupation
--------------------------------------------------------------------------------
```

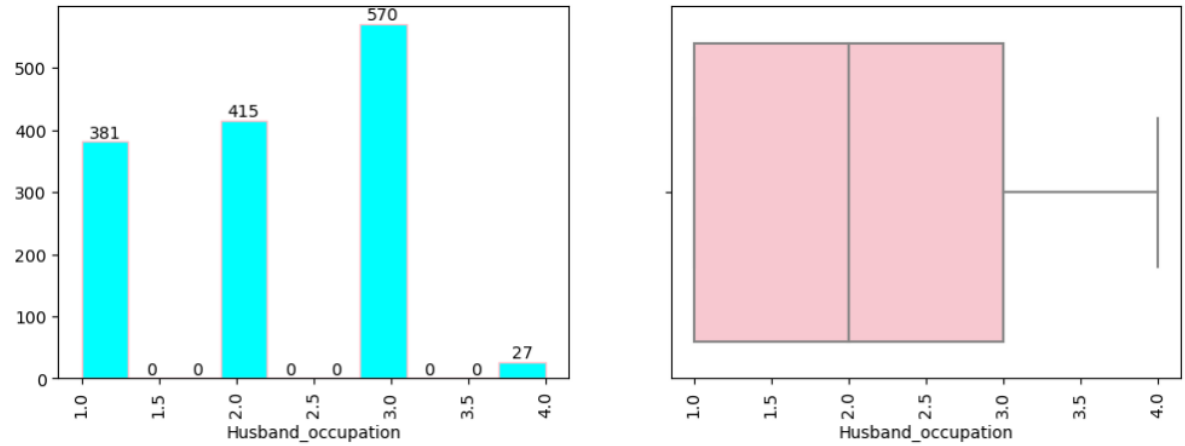
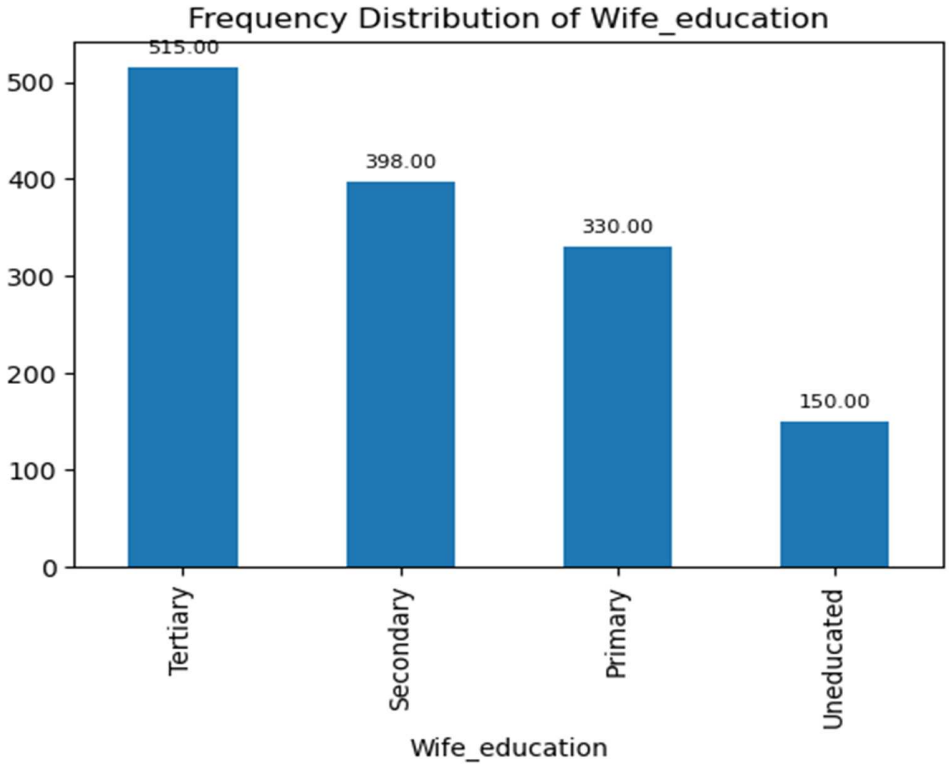


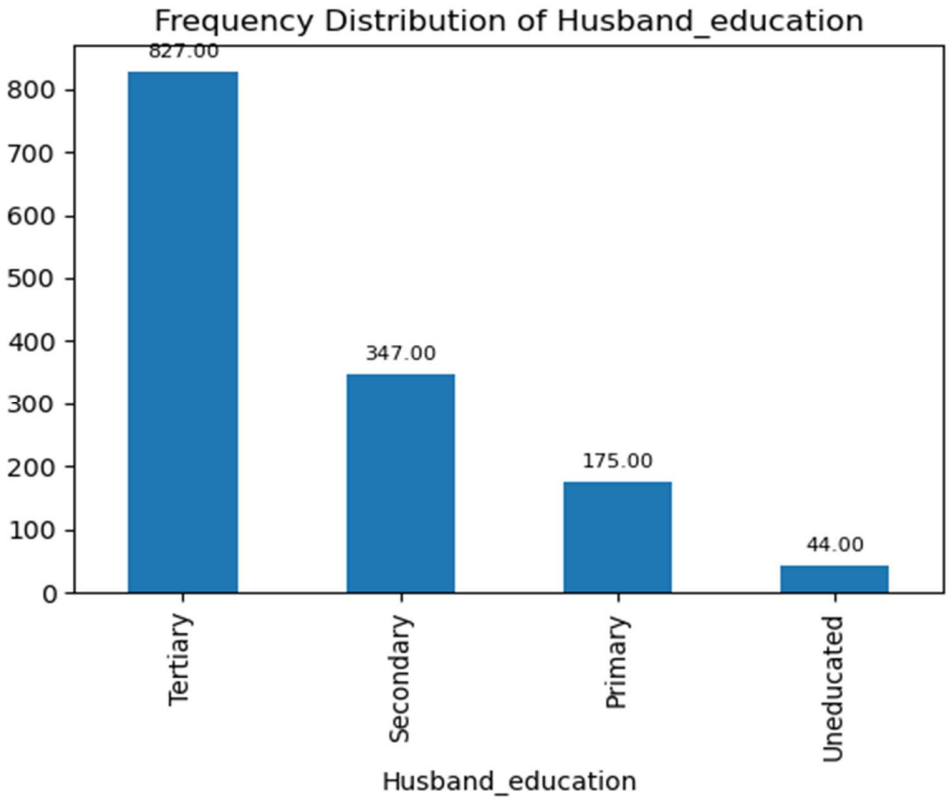
Figure 11: Univariate analysis numeric columns

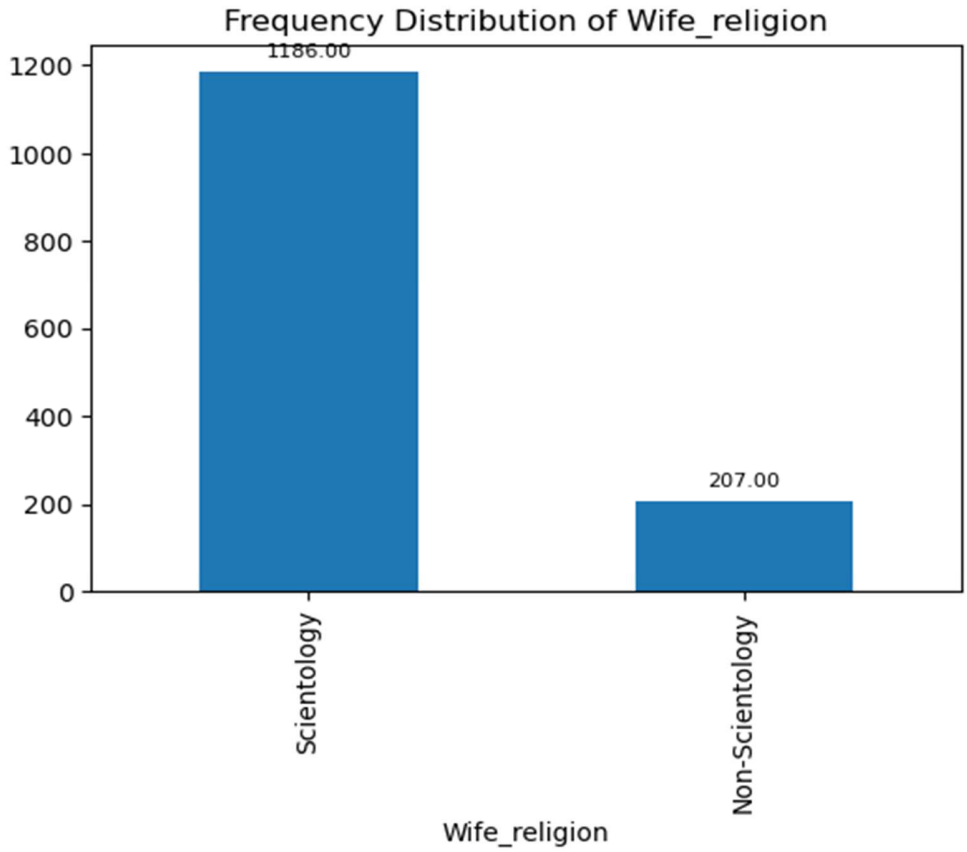
Key Observations

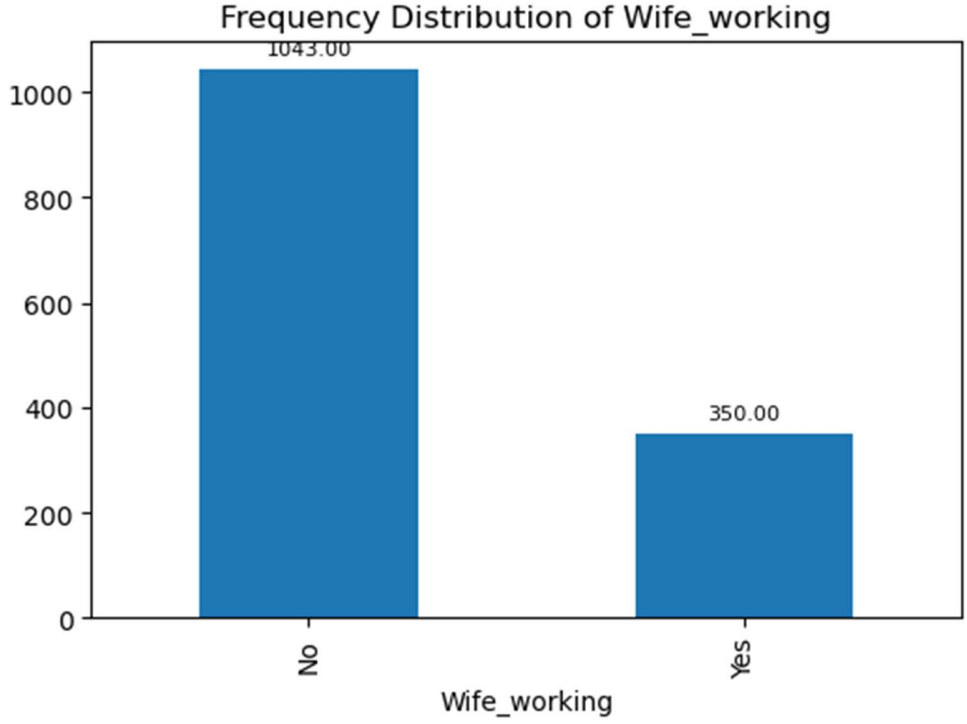
1. There are outliers present only in No_of_children_born, however, we believe outlier treatment is not required as median and mean values are almost similar.
2. Data is not heavily skewed.

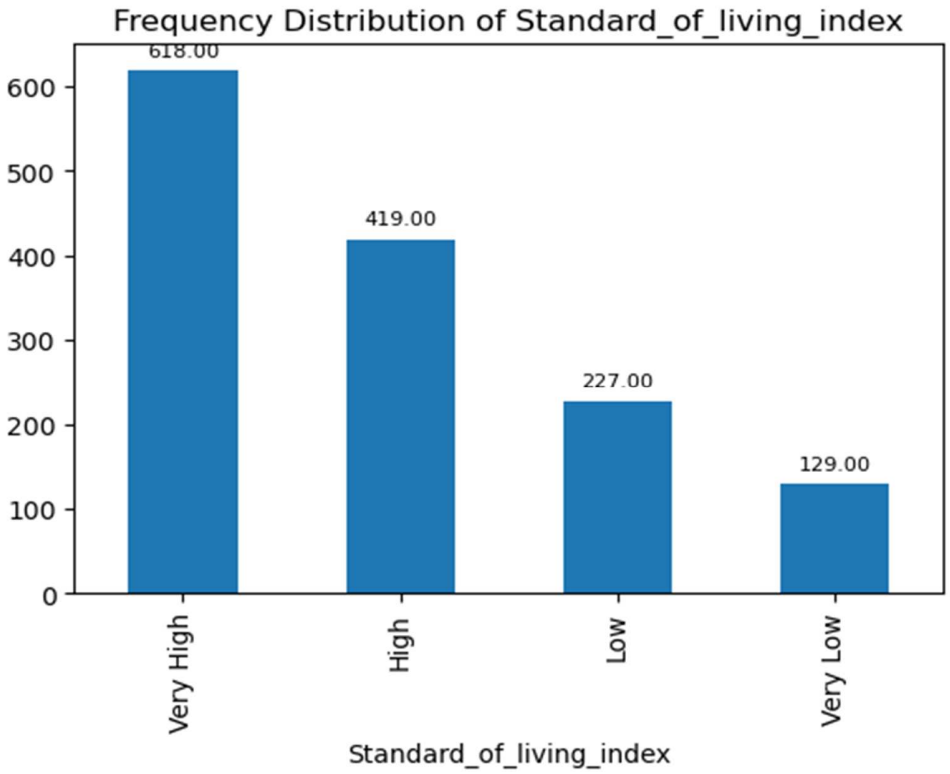
Analysis of Categorical Column

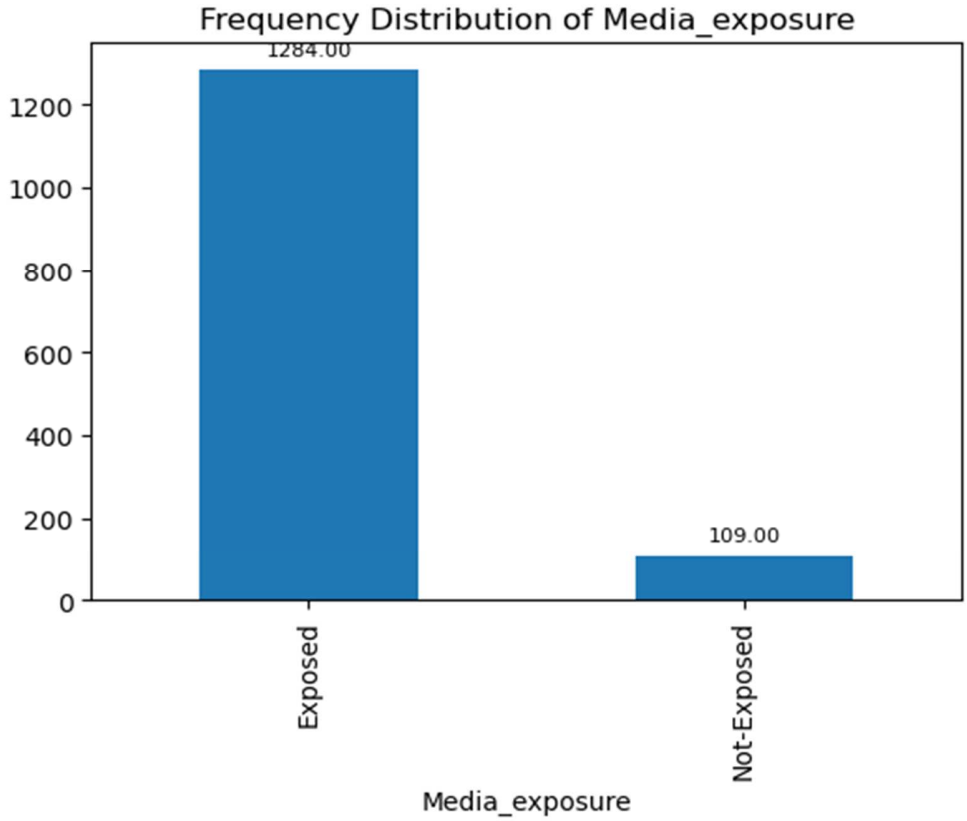



Frequency Distribution of Husband_education
827.00
347.00
175.00
44.00
800
700
600
500
400
300
200
100
0
Tertiary
Secondary
Primary
Uneducated
Husband_education


Frequency Distribution of Wife_religion
1186.00
207.00
1200
1000
800
600
400
200
0
Scientology
Non-Scientology
Wife_religion


Frequency Distribution of Wife_working
1043.00
350.00
1000
800
600
400
200
0
No
Yes
Wife_working


Frequency Distribution of Standard_of_living_index
618.00
419.00
227.00
129.00
600
500
400
300
200
100
0
Very High
High
Low
Very Low
Standard_of_living_index


Frequency Distribution of Media_exposure
1284.00
109.00
1200
1000
800
600
400
200
0
Exposed
Not-Exposed
Media_exposure

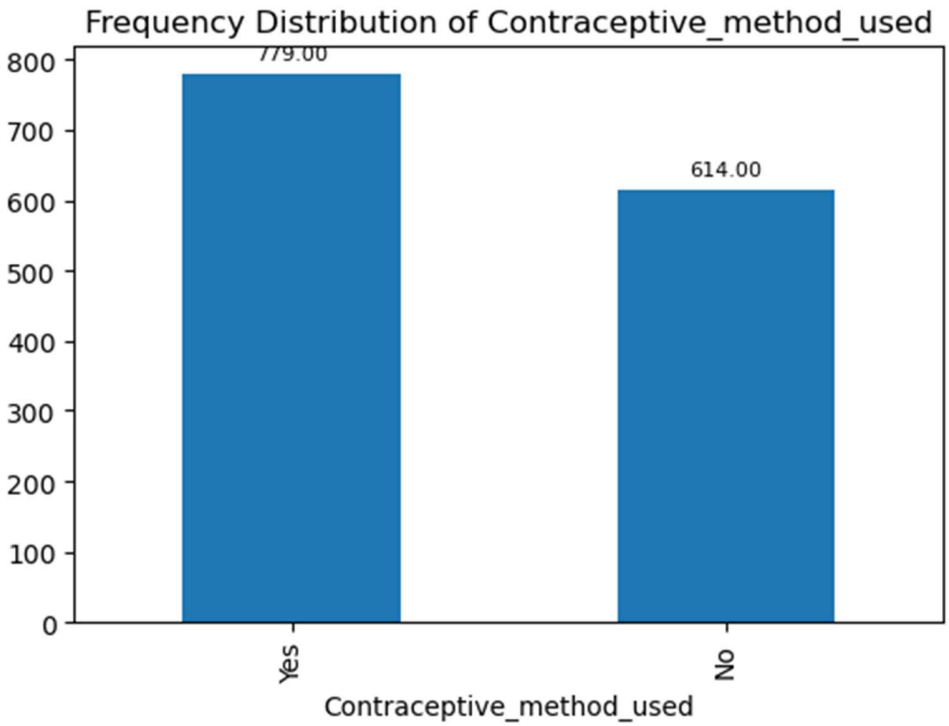


Figure 12: Univariate analysis categorical columns

Key Observations

1. In terms of education for both husbands and wives, the majority have attained some level of education and as the level of education increases, the number of individuals also increases. Standard of living is also showing similar trend where as the standard of living increases, the number of individuals also increases. There might be some correlation between education and standard of living.
2. For Wife_working, Wife_religion and Media_exposure, data in all of which are binary, around 70% wifes are not working, 80% have Scientology as religion and over 90% have media exposure.
3. For Contraceptive_method_used which is the target variable, here number of women using contraceptive method are more as compared to those not using.

2.6.2 Bivariate Analysis

Relation between numeric variables

Pair plot showing data spread between numeric features

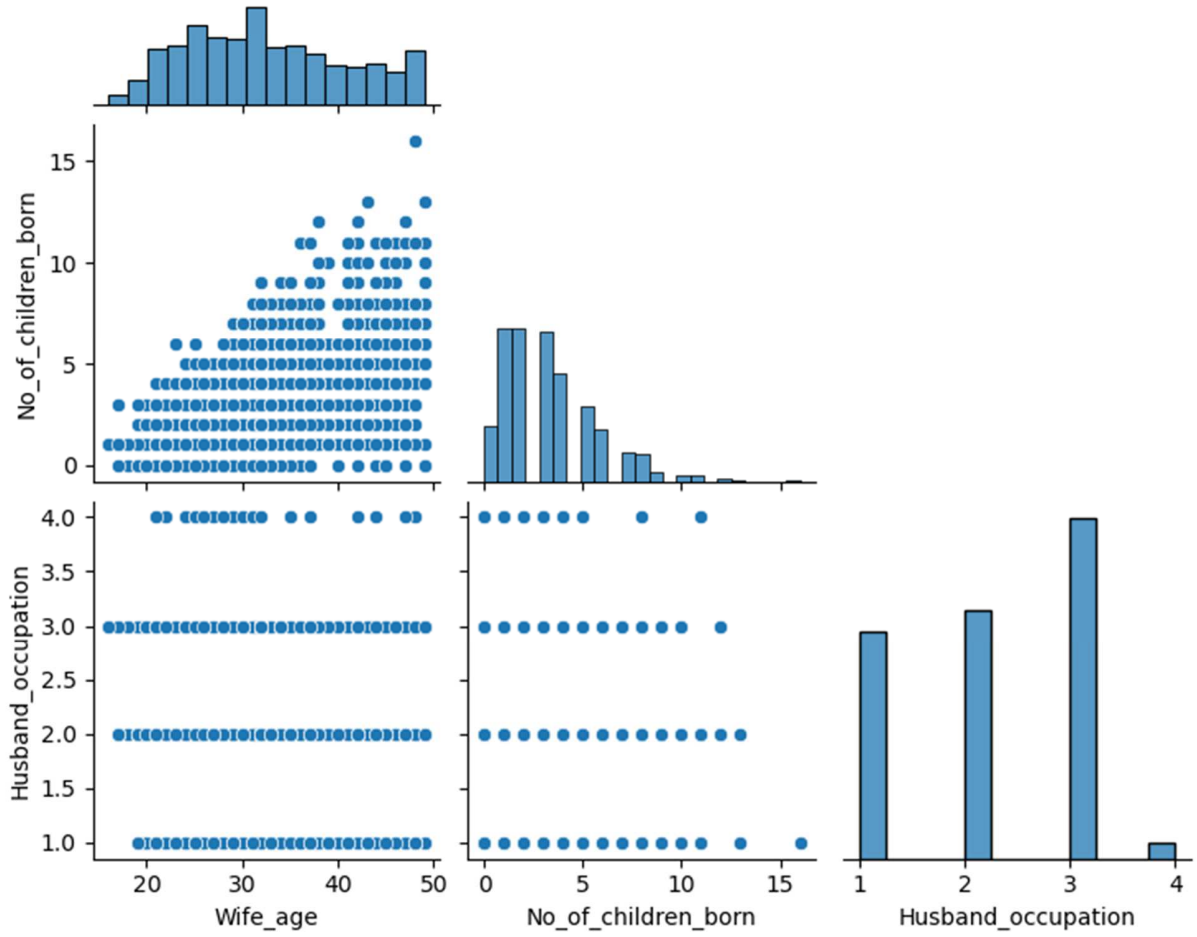


Figure 13: Pair plot

Heatmap showing correlation between numeric variables

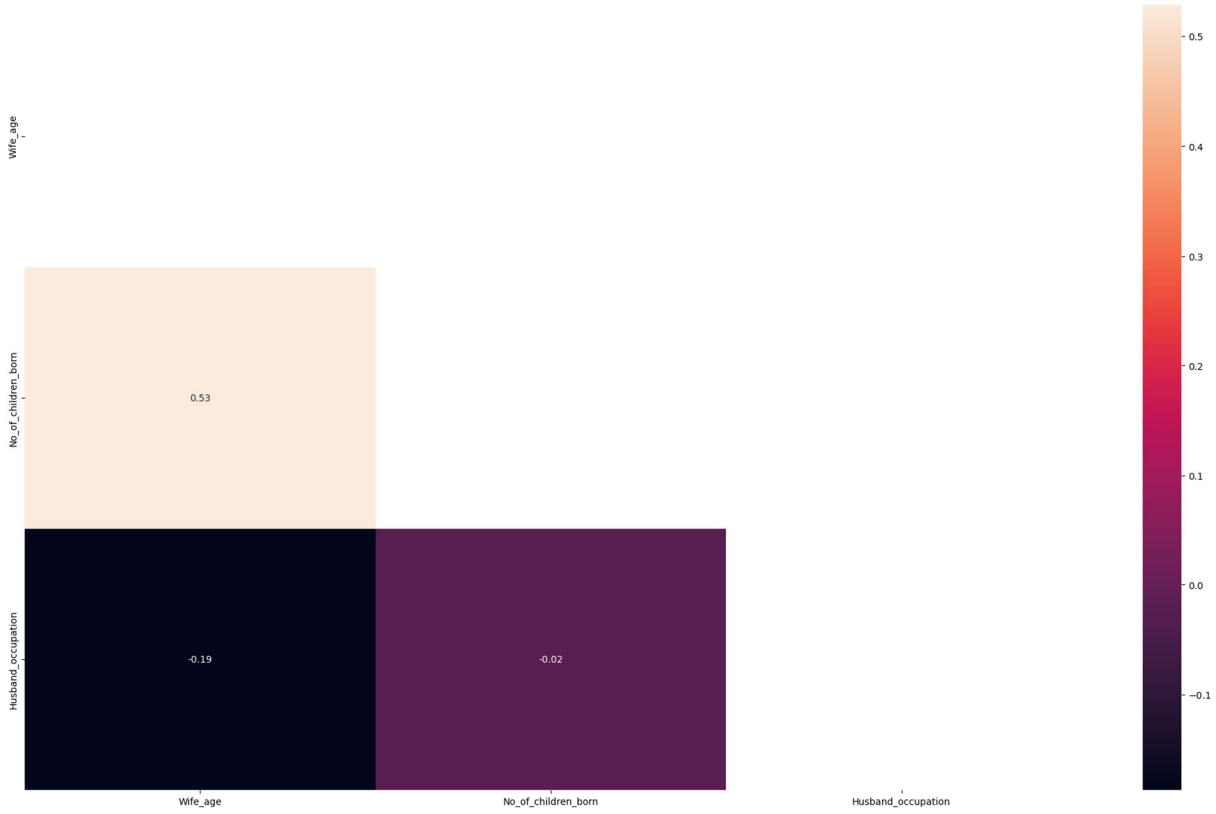


Figure 14: Heatmap

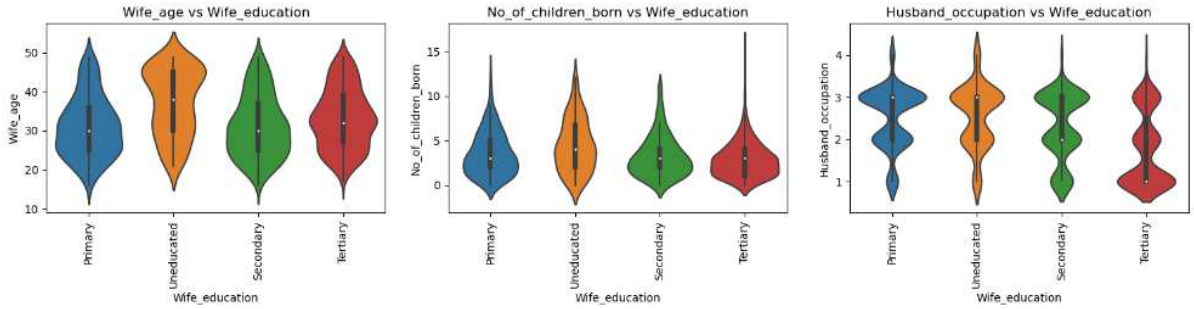
Key Observations

1. Husband_occupation is a categorical variable so it has very weak correlation with other 2 variables.
2. Number of children born and wife age have a moderate correlation though this corelation is very natural.

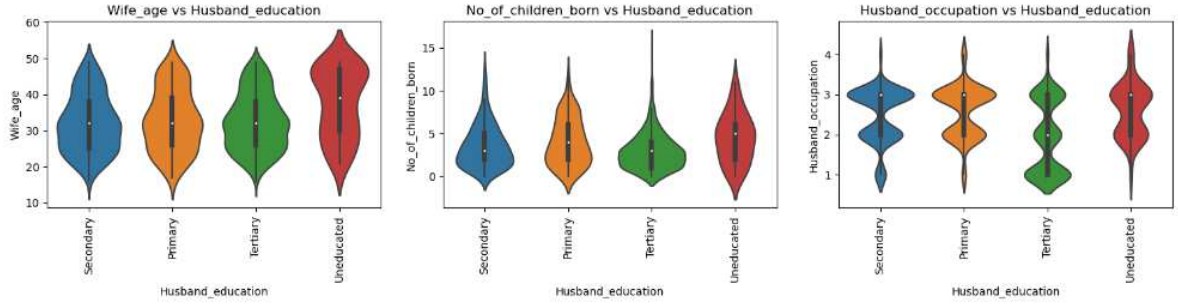
Relation between categorical and numeric variables

Bivariate Analysis for State

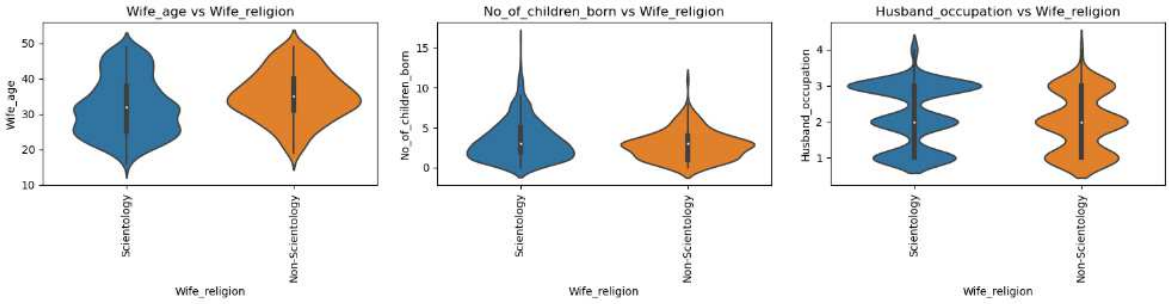
Relationship between Wife_education with numeric columns



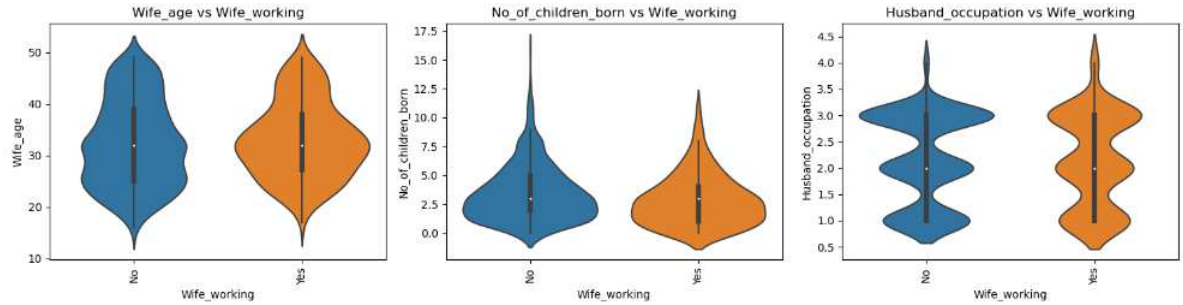
Relationship between Husband_education with numeric columns



Relationship between Wife_religion with numeric columns



Relationship between Wife_working with numeric columns



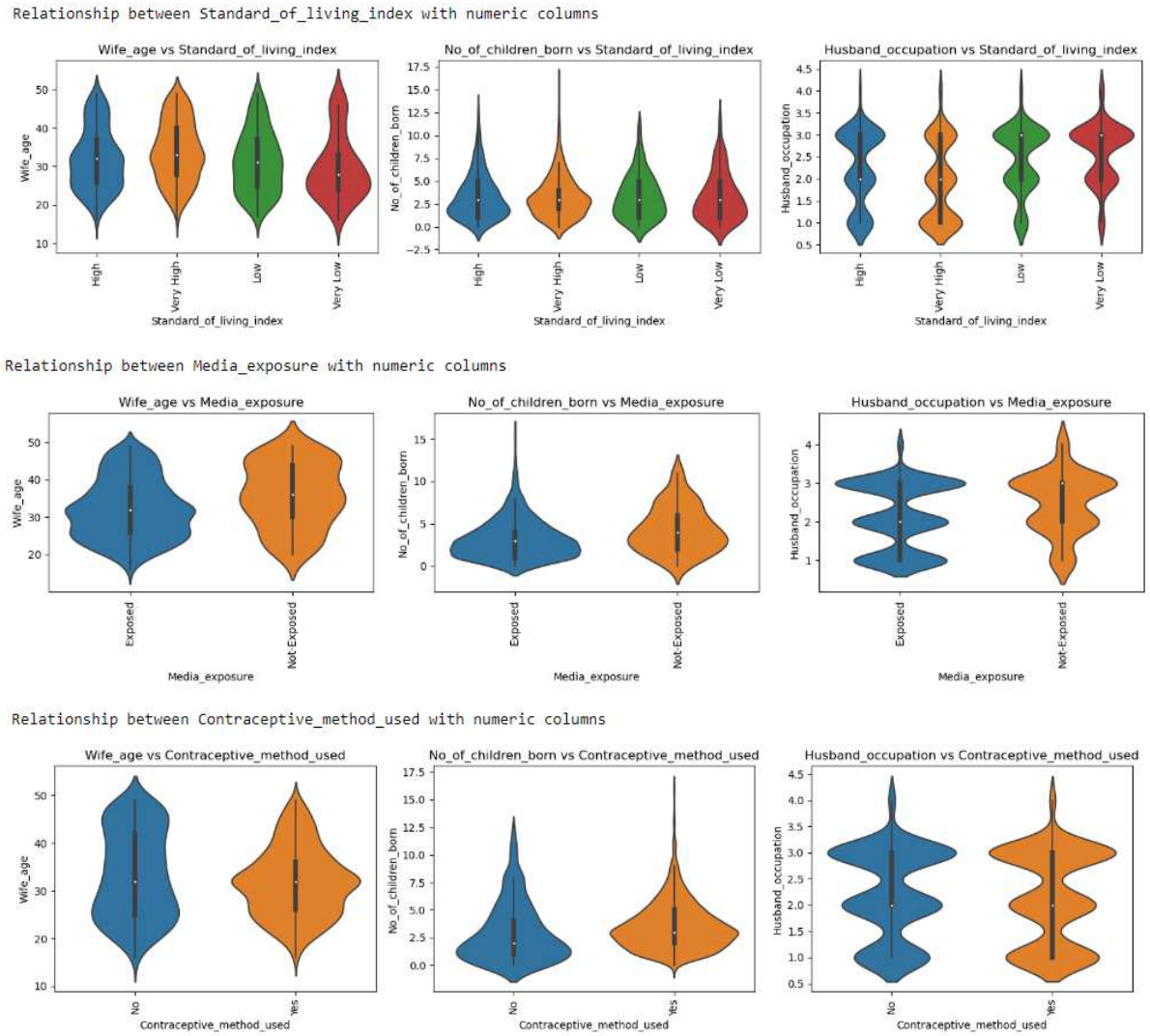


Figure 15: Bivariate Analysis between categorical and numeric variables

Key Observations

1. On analysing wife age with different categorical variables, there are few interesting insights
   - As the level of education for both wife and husband are increasing median wife age is coming down from which we might infer that with time education levels are increasing.
   - Wife age and wife religion also show an interesting trend where median age for wife with non-scientology religion is higher and even the data spread show variance with women having scientology religion have higher data spread under 35 years of age while women with non-scientology have higher data spread between 30 and 40 years.
   - Wife age also has highlightable relationship with standard of living and media exposure whereas the age of wife increases standard of living also increases on the contrary higher aged wife have lower media exposed numbers with median age of media exposed women significantly lower than non-media exposed women.
2. Analysing number of children born with different categorical variables

- Education has a significant impact on number of education where husband's education level has more prominent effect. As the level of education increases amongst husband median number of children they have is coming down.
- Media exposure is also showing some relationship with number of children born as median number of children born are lower for those with media exposure when compared to those with no media exposure.

3. Analysing husband occupation with different categorical variables

- Education is also playing a prominent role with husband occupation level, here wife's education level is affecting husband's occupation level where with the change in wife's education level, husband's occupation level is also changing. Wife with higher education levels have more husbands with lower numeric levels of husband occupation.
- Husband occupation and standard of living are also showing meaningful relation where husbands occupation has been assigned level from 1 to 4 and as we move up the levels in numeric order in husband occupation the standard of living is coming down, since, there is no explanation provided about the labelling of occupation levels we are considering that the labelling is done based income from different occupation meaning with each increase in level of occupation income level are coming down which is causing this decline in standard of living. Also wife with higher education levels are with husbands having lower numeric levels of occupation meaning they have husbands with higher income.
- Husband occupation and media exposure are also related where non media exposed have higher occupation levels.

4. Based on this analysis we can interpret that education is coming out as a single most important demographic factor as it is impacting all other attributes directly or indirectly.
5. On analysing the numeric attributes with target variable contraceptive method used, number of children born is only showing a differentiable relationship where wife using contraceptive methods have higher median value of number of children born. From this we can infer is that women with more children are more likely to use contraceptive methods.

Relationship between categorical column

```
Cross-tabulation between Wife_education and Husband_education
```

Husband_education Wife_education	Primary	Secondary	Tertiary	Uneducated
Primary	87	128	103	12
Secondary	25	148	220	5
Tertiary	4	28	482	1
Uneducated	59	43	22	26

Cross-tabulation between Wife_education and Wife_religion

Wife_religion Wife_education	Non-Scientology	Scientology
Primary	22	308
Secondary	52	346
Tertiary	130	385
Uneducated	3	147

Cross-tabulation between Wife_education and Wife_working

Wife_working Wife_education	No	Yes
Primary	255	75
Secondary	310	88
Tertiary	363	152
Uneducated	115	35

Cross-tabulation between Wife_education and Standard_of_living_index

Standard_of_living_index Wife_education	High	Low	Very High	Very Low
Primary	100	71	104	55
Secondary	134	81	146	37
Tertiary	140	38	329	8
Uneducated	45	37	39	29

Cross-tabulation between Wife_education and Media_exposure

Media_exposure Wife_education	Exposed	Not-Exposed
Primary	302	28
Secondary	380	18
Tertiary	509	6
Uneducated	93	57

Cross-tabulation between Wife_education and Contraceptive_method_used

Contraceptive_method_used	No	Yes
Wife_education		
Primary	174	156
Secondary	171	227
Tertiary	167	348
Uneducated	102	48

Cross-tabulation between Husband_education and Wife_religion

Wife_religion	Non-Scientology	Scientology
Husband_education		
Primary	8	167
Secondary	27	320
Tertiary	170	657
Uneducated	2	42

Cross-tabulation between Husband_education and Wife_working

Wife_working	No	Yes
Husband_education		
Primary	134	41
Secondary	260	87
Tertiary	619	208
Uneducated	30	14

Cross-tabulation between Husband_education and Standard_of_living_index

Standard_of_living_index	High	Low	Very High	Very Low
Husband_education				
Primary	46	48	43	38
Secondary	111	83	109	44
Tertiary	250	84	458	35
Uneducated	12	12	8	12

Cross-tabulation between Husband_education and Media_exposure

Media_exposure	Exposed	Not-Exposed
Husband_education		
Primary	133	42
Secondary	320	27
Tertiary	801	26
Uneducated	30	14

Cross-tabulation between Husband_education and Contraceptive_method_used

Contraceptive_method_used	No	Yes
Husband_education		
Primary	98	77
Secondary	159	188
Tertiary	326	501
Uneducated	31	13

Cross-tabulation between Wife_religion and Wife_working

Wife_working	No	Yes
Wife_religion		
Non-Scientology	143	64
Scientology	900	286

Cross-tabulation between Wife_religion and Standard_of_living_index

Standard_of_living_index	High	Low	Very High	Very Low
Wife_religion				
Non-Scientology	44	15	143	5
Scientology	375	212	475	124

Cross-tabulation between Wife_religion and Media_exposure

Media_exposure	Exposed	Not-Exposed
Wife_religion		
Non-Scientology	199	8
Scientology	1085	101

Cross-tabulation between Wife_religion and Contraceptive_method_used

Contraceptive_method_used	No	Yes
Wife_religion		
Non-Scientology	74	133
Scientology	540	646

Cross-tabulation between Wife_working and Standard_of_living_index

Standard_of_living_index	High	Low	Very High	Very Low
Wife_working				
No	323	170	441	109
Yes	96	57	177	20

Cross-tabulation between Wife_working and Media_exposure

Media_exposure	Exposed	Not-Exposed
Wife_working		
No	961	82
Yes	323	27

Cross-tabulation between Wife_working and Contraceptive_method_used

Contraceptive_method_used	No	Yes
Wife_working		
No	447	596
Yes	167	183

Cross-tabulation between Standard_of_living_index and Media_exposure

Media_exposure	Exposed	Not-Exposed
Standard_of_living_index		
High	397	22
Low	187	40
Very High	600	18
Very Low	100	29

Cross-tabulation between Standard_of_living_index and Contraceptive_method_used

Contraceptive_method_used	No	Yes
Standard_of_living_index		
High	181	238
Low	117	110
Very High	236	382
Very Low	80	49

Cross-tabulation between Media_exposure and Contraceptive_method_used

Contraceptive_method_used	No	Yes
Media_exposure		
Exposed	540	744
Not-Exposed	74	35

Table 35: Cross tab

Key Observations

1. Comparing wife education with other categorical variables, it is showcasing interesting insights and trends.

- Most wife have husbands with similar or higher education levels
- Wife following non-scientology religion are more educated and have higher level of education in proportion terms when compared to those following Scientology religion.

- Wife education have an impact on standard of living where with the increase in education levels, standard of living increases.
- Educated wife are more exposed to media than uneducated where exposure increases with increase in education levels.
- As the education levels are increasing the use of contraceptive are also increasing.

2. On comparing husband education with other categorical variables, we can take following insights:

- Husband whose wife are following non-scientology religion are more educated and have higher level of education in proportion terms when compared to those following Scientology religion.
- Interestingly, for uneducated husband, around 35% have working partners this proportion is considerably lower for educated husbands where for each level this proportion is below 25% mark.
- As the education level of husband increases likelihood of having higher standard of living also increases.
- Educated husband are more exposed to media than uneducated where exposure increases with increase in education levels.
- As the education levels are increasing the use of contraceptive are also increasing.
- As education level in husband increases wife are more likely to use contraceptive methods.

3. On comparing wife religion with remaining categorical variables, we can take following insights

- Wife following non-scientology religion are more likely to have high or very high standard of living with around 90% in these categories, while proportion for those following Scientology having high or very high standard of living is around 70%.
- In percentage terms use of contraceptives by wife following non-scientology religion is far greater than those following Scientology.

4. The likelihood of having a higher standard of living increases if the wife is working. Our data shows that the proportion of cases where the wife is working, when compared in percentage terms, rises as the standard of living increases.
5. When comparing standard of living index with media exposure and contraceptive method we find that those with higher standard of living are more exposed to media and have higher likelihood of using contraceptive methods.
6. Wife who are exposed to media are far more likely to use contraceptive methods than those not exposed to media.

2.7 Data Encoding

Here all categorical variables which were object type were encoded numerically, first the attributes like 'Wife_education', 'Husband_education' and ‘Standard_of_living_index’, since, these features had ordinal data they were encoded accordingly.

Data overview after encoding ordinal attributes

	Wife_age	Wife_education	Husband_education	No_of_children_born	Wife_religion	Wife_working	Husband_occupation	Standard_of_living_index	Media_expo
0	24.0	2	3	3.0	Scientology	No	2	3	Expo
1	45.0	1	3	10.0	Scientology	No	3	4	Expo
2	43.0	2	3	7.0	Scientology	No	3	4	Expo
3	42.0	3	2	9.0	Scientology	No	3	3	Expo
4	36.0	3	3	8.0	Scientology	No	3	2	Expo

Figure 36: Data overview

Target variable was encoded using LabelEncoder from sklearn.preprocessing, where 1 was assigned to the class of interest which in this case is use of contraceptive method.

```
Contraceptive_method_used
1    779
0    614
Name: count, dtype: int64
```

Table 37: Target variable value count

Remaining variables were encoded using dummy variable.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1393 entries, 0 to 1392
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Wife_age                    1393 non-null   int32
 1   Wife_education              1393 non-null   int32
 2   Husband_education           1393 non-null   int32
 3   No_of_children_born         1393 non-null   int32
 4   Husband_occupation          1393 non-null   int32
 5   Standard_of_living_index    1393 non-null   int32
 6   Contraceptive_method_used   1393 non-null   int32
 7   Wife_religion_Scientology   1393 non-null   int32
 8   Wife_working_Yes            1393 non-null   int32
 9   Media_exposure_Not-Exposed  1393 non-null   int32
dtypes: int32(10)
memory usage: 54.5 KB
```

Figure 38: Dataset information

2.8 Splitting Data

The data is converted into X and Y where X contains independent variables and Y contains class labels. X and Y are further split into train and test data in the ratio of 70:30

Train Data

```
Contraceptive_method_used
1    0.561026
0    0.438974
Name: proportion, dtype: float64
```

Table 39: Target variable proportion

Test Data

```
Contraceptive_method_used
1    0.555024
0    0.444976
Name: proportion, dtype: float64
```

Table 40: Target variable proportion

For target label though both the classes are not completely balanced, however, for classification modelling if the higher class represents less than 75% data than we consider it to be balanced and by that yardstick our data appears balanced.

2.9 Classification Modelling

We will build models using different classification techniques namely logistic regression, linear discriminant analysis and decision tree using CART and compare their performance to identify the best model.

Logistic Regression

A logistic regression model was created using LogisticRegression from sklearn which was than evaluated using multiple metrics

Model Evaluation

Model accuracy for train data
0.6635897435897435

Model accuracy for test data
0.6770334928229665

Model accuracy for both test and train data are almost identical meaning we have no issues for underfitting or over fitting and the model is stable.

Using AUC and ROC Method

```
for training data
AUC: 0.700
```

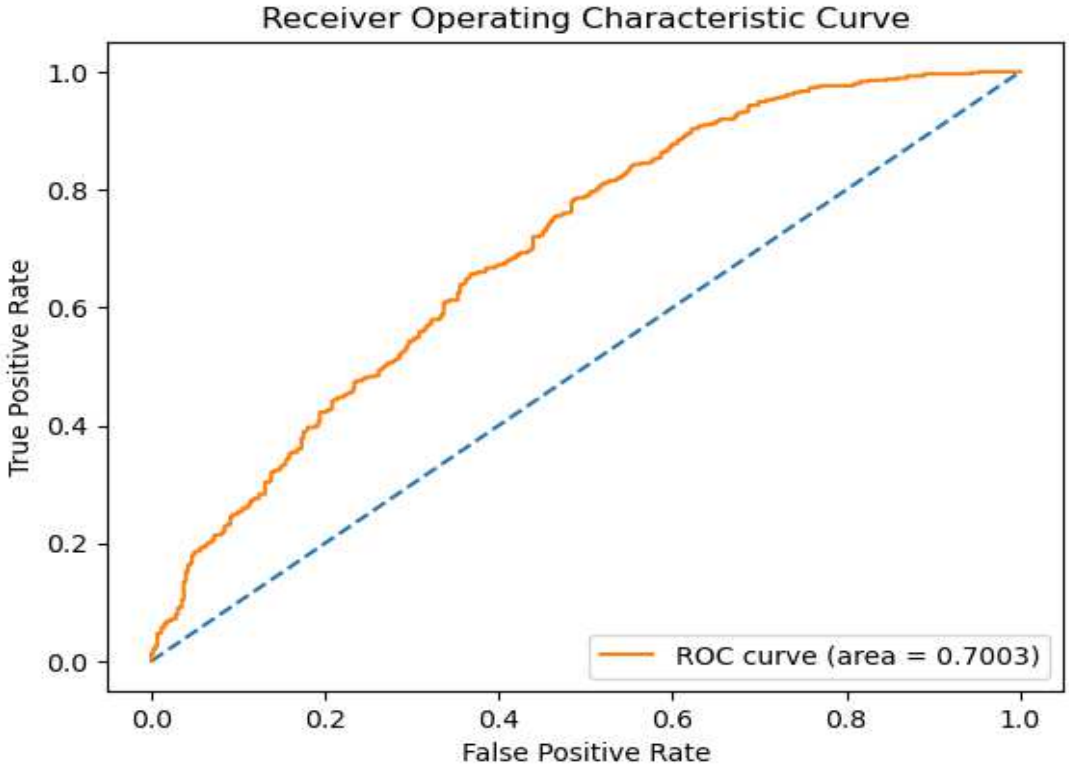


Figure 16: AUC – ROC curve

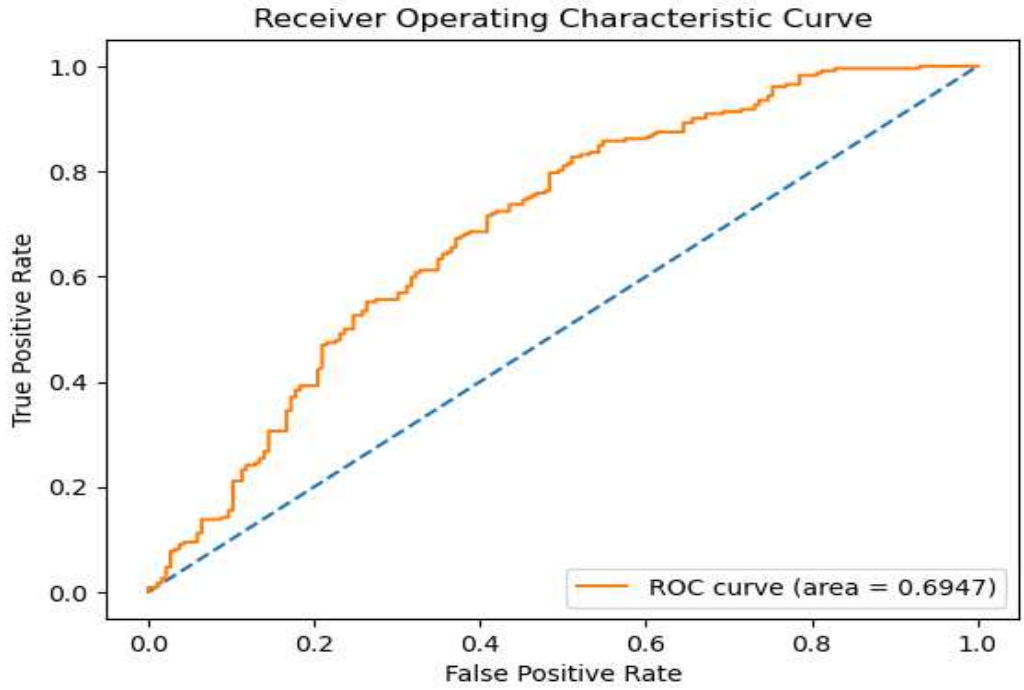


Figure 17: AUC – ROC curve

Using Confusion Matrix

```
for train data
-----------------------------------------------------------------------

array([[211, 217],
       [111, 436]], dtype=int64)
```

Table 41: Confusion matrix

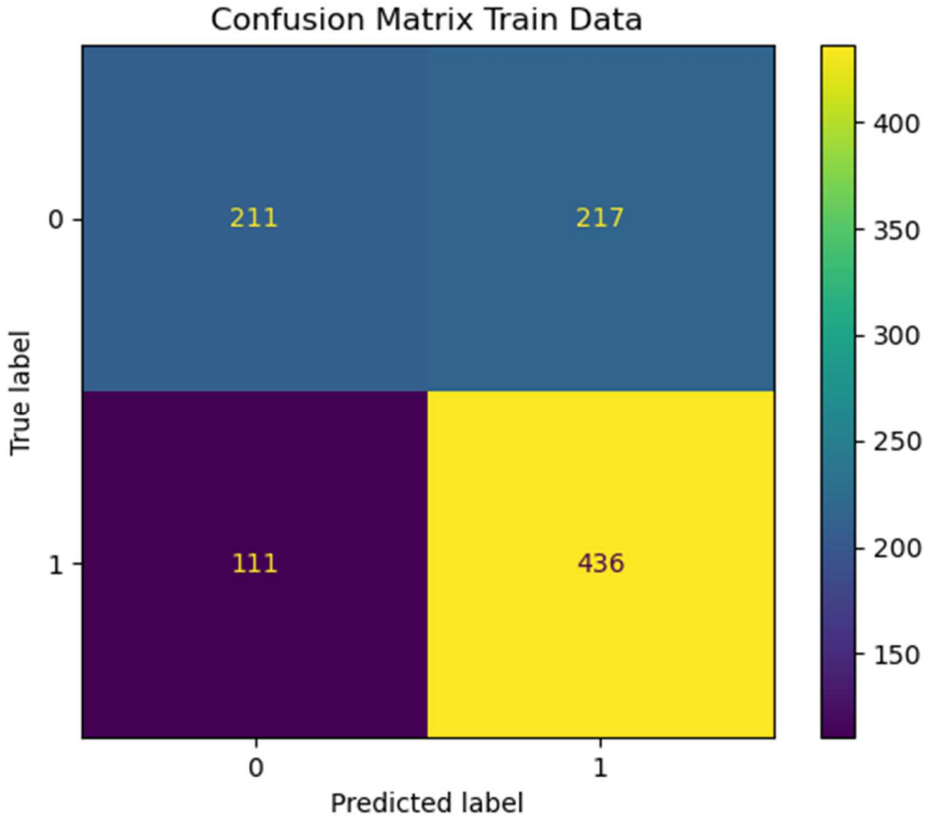


Figure 18: Confusion Matrix

```
Classification report train data
-----------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.66      0.49      0.56       428
           1       0.67      0.80      0.73       547

    accuracy                           0.66       975
   macro avg       0.66      0.65      0.64       975
weighted avg       0.66      0.66      0.65       975
```

Table 42: Classification report

```
for test data
----------------------------------------------------------------------

array([[ 84, 102],
       [ 33, 199]], dtype=int64)
```

Table 43: Confusion matrix

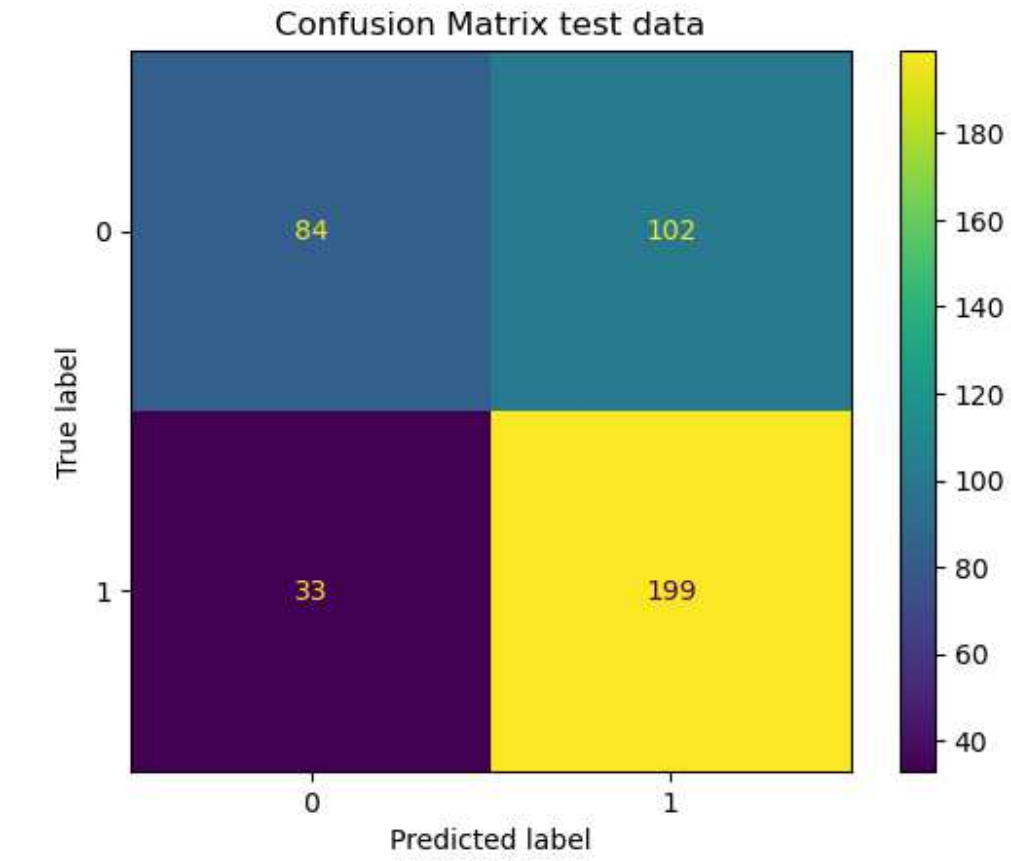


Figure 19: Confusion Matrix

```
Classification report test data data
----------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.72      0.45      0.55       186
           1       0.66      0.86      0.75       232

    accuracy                           0.68       418
   macro avg       0.69      0.65      0.65       418
weighted avg       0.69      0.68      0.66       418
```

Table 44: Classification report

1. The AUC score and f1-score for model is almost identical which reiterates the fact that model is stable.
2. Model has an AUC score of 0.7 on train data and 0.694 on test data meaning model is doing a fair performance.

Linear Discriminant Analysis

A linear discriminant analysis model was created using LinearDiscriminantAnalysis from sklearn which was then evaluated using multiple metrics.

Model Evaluation

Model accuracy for train data
0.6646153846153846

Model accuracy for test data
0.6626794258373205

Model accuracy for both test and train data are almost identical meaning we have no issues for underfitting or over fitting and the model is stable.
Using AUC and ROC Method

```
for training data
AUC: 0.701
```

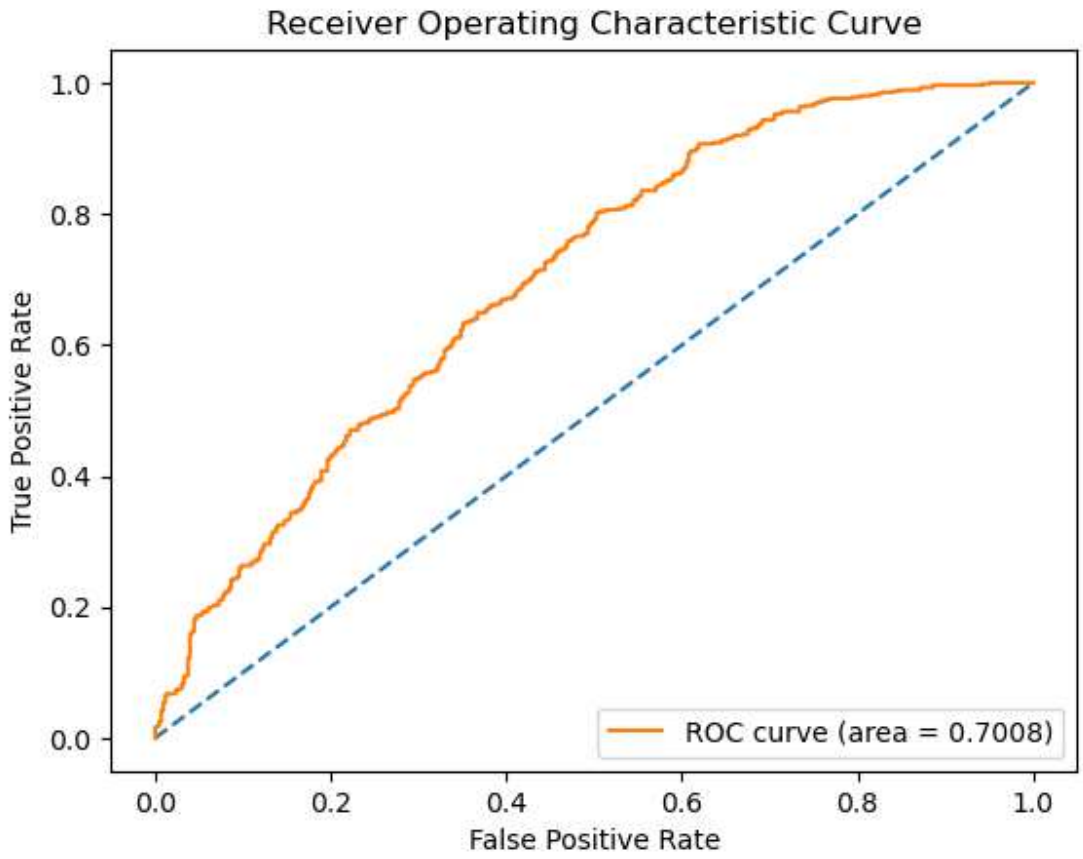


Figure 20: AUC – ROC Matrix

```
for test data
AUC: 0.693
```

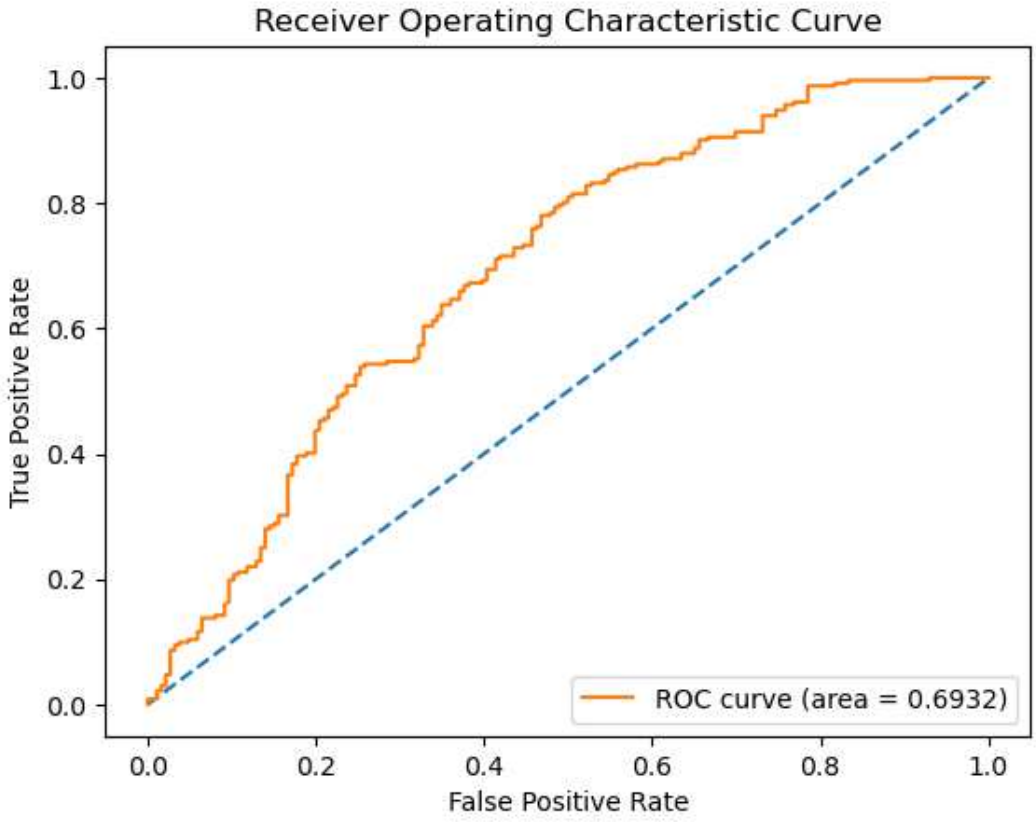


Figure 21: AUC – ROC Matrix

Using Confusion Matrix

```
for training data
------------------------------------------------------------------------

array([[207, 221],
       [106, 441]], dtype=int64)
```

Table 45: Confusion matrix

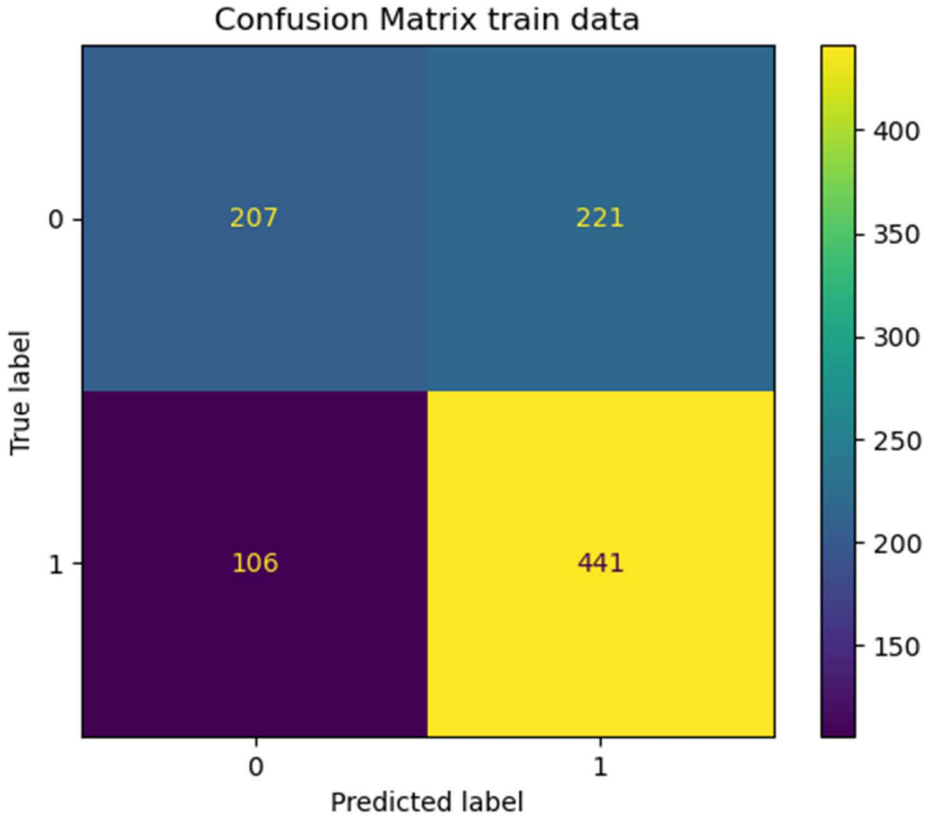


Figure 22: Confusion Matrix

Classification report train data

	precision	recall	f1-score	support
0	0.66	0.48	0.56	428
1	0.67	0.81	0.73	547
accuracy			0.66	975
macro avg	0.66	0.64	0.64	975
weighted avg	0.66	0.66	0.65	975

Table 46: Classification report

for test data

```
array([[ 78, 108],
       [ 33, 199]], dtype=int64)
```

Table 47: Confusion matrix

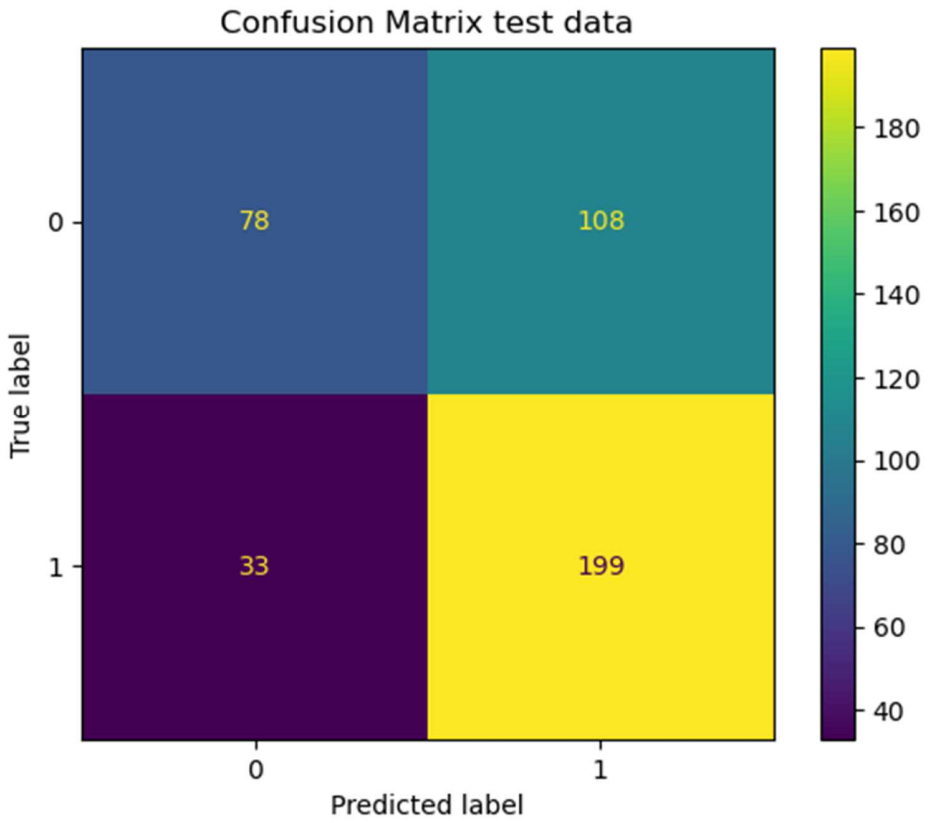


Figure 23: Confusion Matrix

Classification report test data

	precision	recall	f1-score	support
0	0.70	0.42	0.53	186
1	0.65	0.86	0.74	232
accuracy			0.66	418
macro avg	0.68	0.64	0.63	418
weighted avg	0.67	0.66	0.64	418

Table 48: Classification report

1. The AUC score and f1-score for model is almost identical which reiterates the fact that model is stable.
2. Model has an AUC score of 0.701 on train data and 0.693 on test data meaning model is doing a fair performance.

Decision Tree

A linear discriminant analysis model was created using DecisionTreeClassifier from sklearn which was then evaluated using multiple metrics.

Model Evaluation

Model accuracy for train data
0.9835897435897436

Model accuracy for test data
0.6411483253588517

Model accuracy for test and train data are considerably different which means there is an issue of overfitting in the model, we will have to prune the model. For pruning we will use GridSearch to find best parameters.

Pruning using GridSearch

Using GridSearchCV we passed a list of values each for maximum depth, minimum samples split and minimum samples leaf to find the best pruning parameters which as per it were:

```
Best parameters found:  {'max_depth': 5, 'min_samples_leaf': 3, 'min_samples_split': 20}
```

Table 49: Best parameters

Using these parameters again the model was made and metrics for which are:

Model accuracy for train data
0.7312820512820513

Model accuracy for test data
0.6698564593301436
Though still there is still a difference in the test and train data scores, we have bought down the difference significantly, we will continue with this model.

Using AUC and ROC Method

```
for training data
AUC: 0.779
```

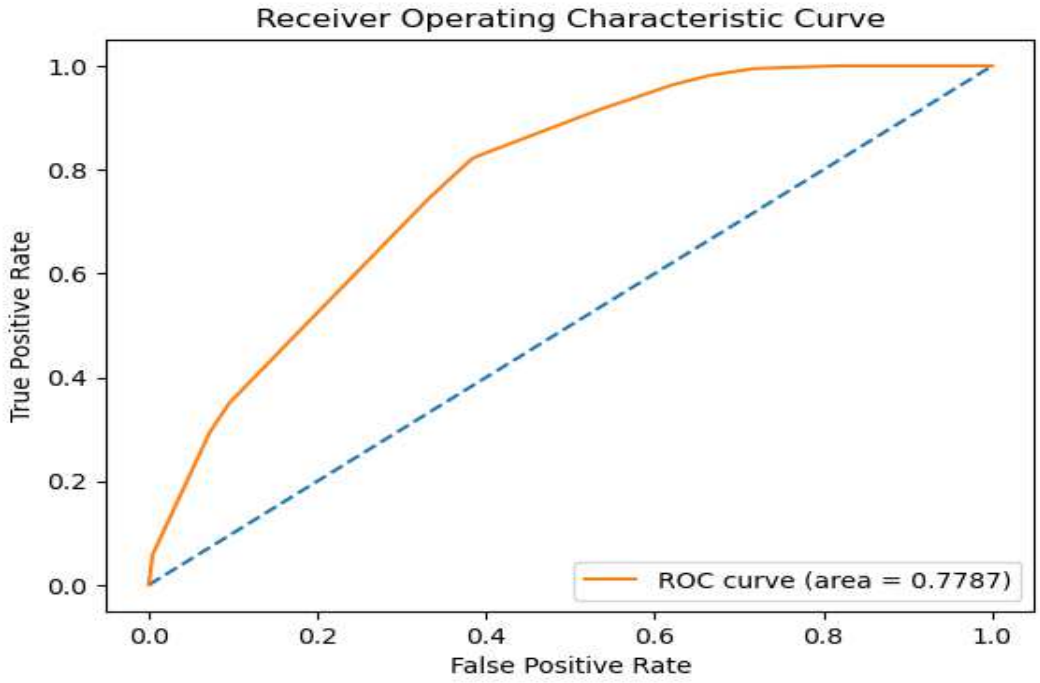


Figure 24: AUC – ROC Curve

```
for test data
AUC: 0.701
```

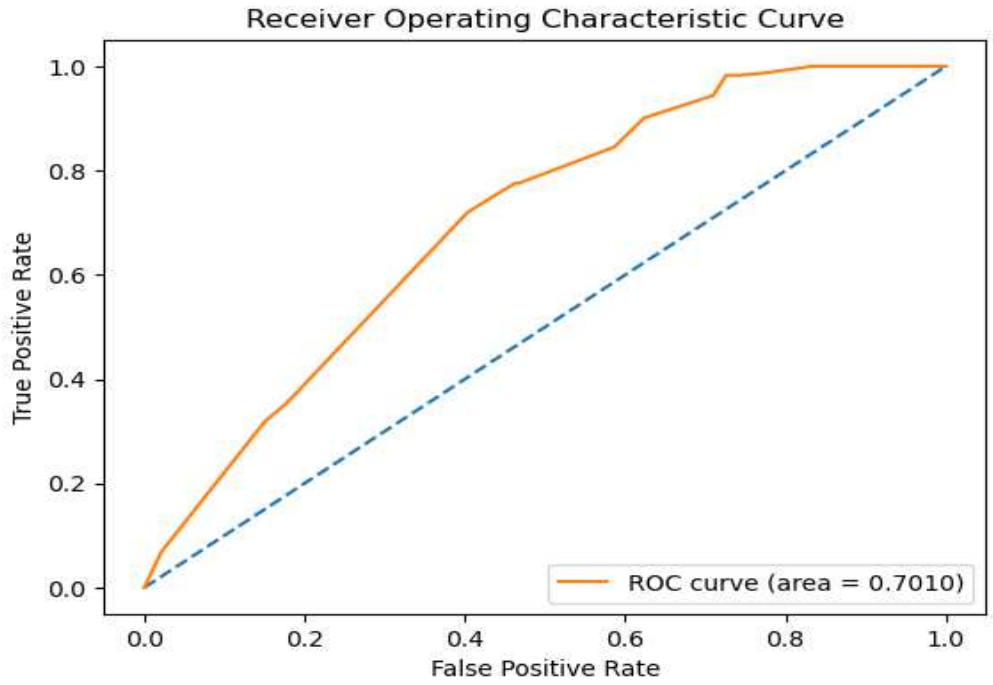


Figure 25: AUC – ROC Curve

Using confusion matrix

```
for training data
------------------------------------------------------------------------

array([[264, 164],
       [ 98, 449]], dtype=int64)
```

Table 50: Confusion matrix

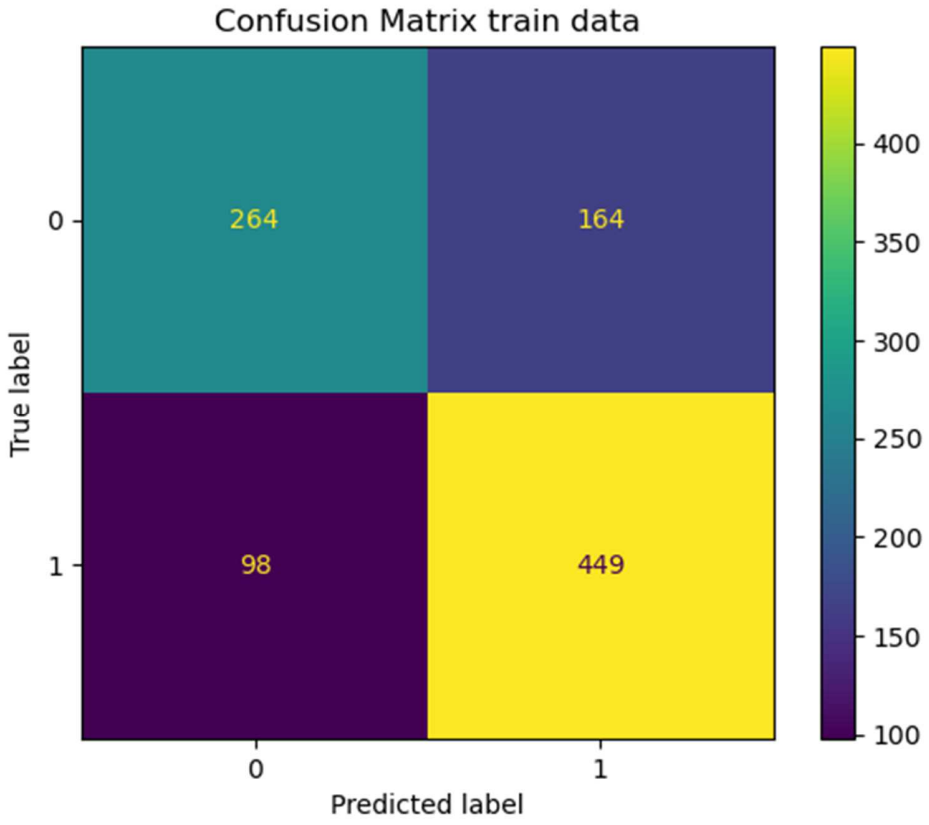


Figure 26: Confusion Matrix

```
Classification report train data
------------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.73      0.62      0.67       428
           1       0.73      0.82      0.77       547

    accuracy                           0.73       975
   macro avg       0.73      0.72      0.72       975
weighted avg       0.73      0.73      0.73       975
```

Table 51: Classification report

```
for test data
----------------------------------------------------------------------

array([[100,  86],
       [ 52, 180]], dtype=int64)
```

Table 52: Confusion matrix

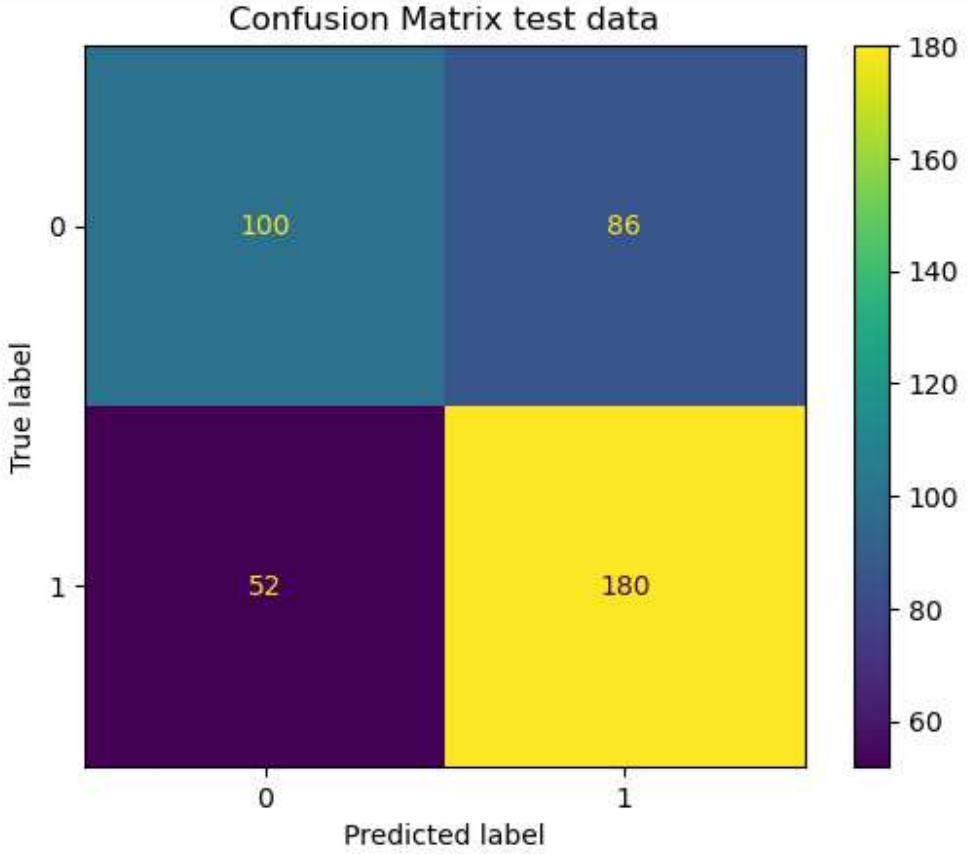


Figure 27: Confusion Matrix

```
Classification report test data
----------------------------------------------------------------------
              precision    recall  f1-score   support

           0       0.66      0.54      0.59       186
           1       0.68      0.78      0.72       232

    accuracy                           0.67       418
   macro avg       0.67      0.66      0.66       418
weighted avg       0.67      0.67      0.66       418
```

Table 53: Classification report

1. The AUC score and f1-score for model show some difference which we had already expected based on accuracy score still we consider it as a stable model as the difference is around 10% only.
2. Model has an AUC score of 0.779 on train data and 0.701 on test data meaning model is doing a fair performance.

2.10 Model Comparison

We will compare the models based on their accuracy score and f1-score for train and test data

	Train data score	Test data score	f1 score train data	f1 score test data
Logistic Regression	0.663590	0.677033	0.726667	0.746717
Decision Tree	0.731282	0.669856	0.774138	0.722892
Linear Discriminant Analysis	0.664615	0.662679	0.729529	0.738404

Table 54: Model comparision

From the above table, as per the train data decision tree model is performing the best in terms of both accuracy and f1 score. However, when we consider the test data, logistic regression model is performing best, in fact, for logistic regression model the test and train data performance is almost identical which means it is a better generalised model which is not the case for decision tree model. Thus, considering that logistic regression model is better generalised model we consider it as the best model.

2.11 Feature Importance

In logistic regression we would use coefficient values to determine the most important features.

	imp
Wife_education	0.441862
No_of_children_born	0.272439
Standard_of_living_index	0.190277
Husband_education	0.134508
Husband_occupation	0.123461
Wife_age	-0.078586
Wife_working_Yes	-0.168278
Wife_religion_Scientology	-0.397523
Media_exposure_Not-Exposed	-0.508125

Table 55: Feature importance

Based on the sign of coefficient values we have we can divide them into 2 categories:

1. Positive Coefficients: The value of these features increase the log odds of positive outcome that is being labelled as class 1.

- For positive coefficients 'Wife_education' has the highest value which means that higher the level of education is amongst wife more likely they are going to use the contraceptive methods.

2. Negative Coefficients: The value of these features increase the log odds of positive outcome decreasing that is being labelled as class 1.

- For negative coefficients 'Media_exposure_Not-Exposed' has the highest impact meaning women who are not exposed to media are more likely of not using conraceptive methods.

2.12 Conclusion

1. Based on the evaluation of various classification techniques for predictive modeling, the logistic regression model demonstrated the best generalization performance on the test data, achieving an accuracy score of 67.7%. This indicates that the model correctly predicts the class labels of the target variable 67.7% of the time.

2. Based on coefficient values of the features, the regression equation is:
   log odds (Contraceptive method used) = Intercept + 0.441862Wife_education + 0.272439No_of_children_born + 0.190277Standard_of_living_index + 0.134508Husband_education + 0.123461Husband_occupation - 0.078586Wife_age - 0.168278Wife_working_Yes - 0.397523Wife_religion_Scientology - 0.508125*Media_exposure_Not-Exposed

   Equation 6: Logistic regression equation

   Where intercept value of model is: [-0.24755177]

Here, using the log odds value the probability is calculated using equation

p-value = 1/1 + e^-log-odds

Equation 7: P-value equation

where if log odds value is greater than 0 than p-value > 0.5 and vice versa for log odds < 0.

Key Takeaways

Based on the model coefficient values most important demographic and socio-economic features for classification are:

1. 'Wife_education' has the highest positive coefficient value of 0.4418 followed by 'No_of_children_born' with value of 0.2724 meaning provided other features remain constant with every increase in level of education amongst wife log-odds value will increase by 0.4418 and with every additional child born log-odds value will increase by 0.2724.
2. 'Media_exposure_Not-Exposed' has the highest negative coefficient value of -0.5081 followed by 'Wife_religion_Scientology' with value of -0.3975 meaning provided other features remain constant if there is no media exposure the log-odds value will decrease by 0.5081 and if 'WIfe_religion' is Scientology the log-odds value will decrease by 0.3975.

In simple terms, if there is no media exposure it is highly unlikely that married women will use contraceptive method and with increase in education levels the probability that a married woman will use contraceptive method also increases.