BUSINESS DATA

ANALYSIS REPORT

Table of Contents

List of Figures

List of Tables

Problem1: Linear Regression

The comp-activ databases is a collection of a computer systems activity measures . The data was collected from a Sun Sparcstation 20/712 with 128 Mbytes of memory running in a multi-user university department. Users would typically be doing a large variety of tasks ranging from accessing the internet, editing files or running very cpu-bound programs.

As you are a budding data scientist you thought to find out a linear equation to build a model to predict 'usr'(Portion of time (%) that cpus run in user mode) and to find out how each attribute affects the system to be in 'usr' mode using a list of system attributes.

DATA DICTIONARY:

System measures used:

lread - Reads (transfers per second) between system memory and user memory
lwrite - writes (transfers per second) between system memory and user memory
scall - Number of system calls of all types per second
sread - Number of system read calls per second .
swrite - Number of system write calls per second .
fork - Number of system fork calls per second.
exec - Number of system exec calls per second.
rchar - Number of characters transferred per second by system read calls
wchar - Number of characters transfreed per second by system write calls
pgout - Number of page out requests per second
ppgout - Number of pages, paged out per second
pgfree - Number of pages per second placed on the free list.
pgscan - Number of pages checked if they can be freed per second
atch - Number of page attaches (satisfying a page fault by reclaiming a page in memory) per second
pgin - Number of page-in requests per second
ppgin - Number of pages paged in per second
pflt - Number of page faults caused by protection errors (copy-on-writes).
vflt - Number of page faults caused by address translation .
runqsz - Process run queue size (The number of kernel threads in memory that are waiting for a CPU to run.
Typically, this value should be less than 2. Consistently higher values mean that the system might be CPU-bound.)
freemem - Number of memory pages available to user processes
freeswap - Number of disk blocks available for page swapping.
---
usr - Portion of time (%) that cpus run in user mode

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

Ans

We will start analysing the data by going thru the basic steps like

1.Check head and tail of the dataset

2.Check info

3.Check summary

4.Check nulls

5.Check duplicates

Let us start by reading the data and extracting basic information.

There are 8192 observations and 22 columns .Let's see the top 5 records and bottom 5 records of the dataset

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

Figure 1: Top 5 records in the dataset

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

Figure 2: Bottom 5 records in the dataset

Let's see the column Datatypes

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

- Most of the columns in the data are numeric in nature ('int64' or 'float64' type).
- runqsz is string columns ('object' type).

Table 1: Column Datatypes

There are 22 variables in the dataset. There are 13 float types, 8 int types and 1 object type variable in the entire dataset . 'usr' is the dependent variable.

From the column Information above we see that there are null values in the column 'rchar' and 'wchar'.

There 104 null values in 'rchar' and 15 null values in 'wchar'.

There are no duplicates in the dataset

Let's see the data description of the numerical columns.

	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 2: Summary of the numerical columns

There seem to be outliers in the all the numerical variables and are mostly right skewed . runqsz is a categorical variable with 4331 records being non cpu bound and 3861 records being CPU bound.

50% of the variables values in pgfree, pgout,,ppgout,atch are 0. 75% of the values in pgscan are 0.

There seem to be no anomalies in the data in the numerical columns

```
Not_CPU_Bound     4331
CPU_Bound         3861
```

Univariate Analysis

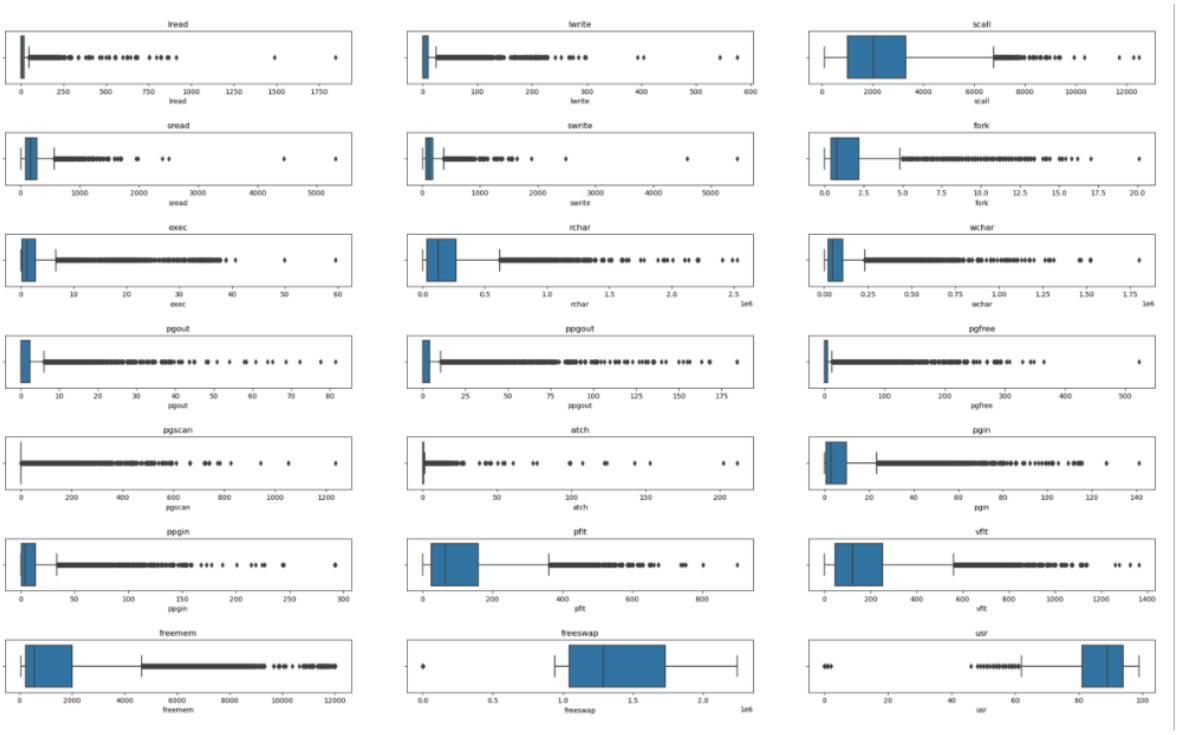


Figure 3: Box plot of the Numerical variables

As we can see from the boxplot of the numerical variables, there are outliers in all the variables

Bivariate Analysis

Heatmap of correlation between numerical variables

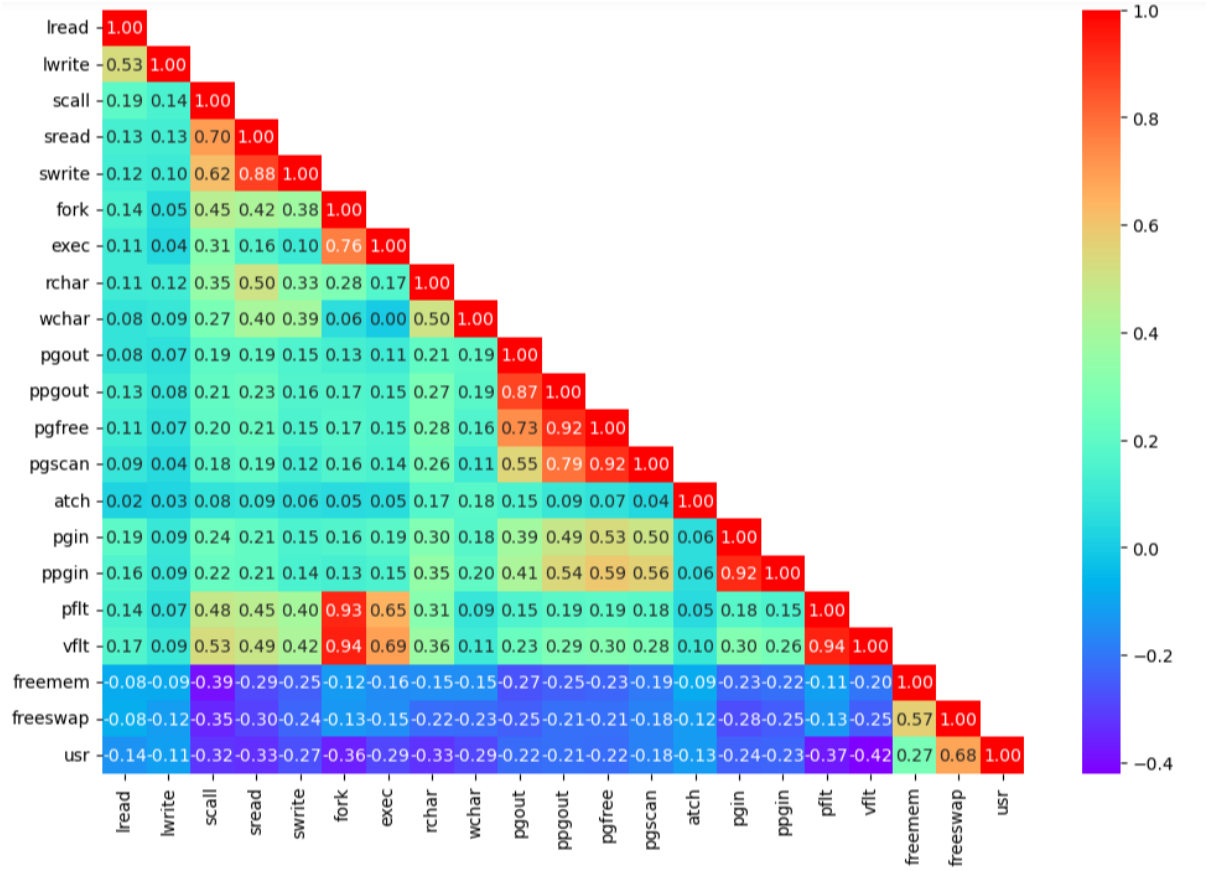


Figure 4:Heatmap of variables

We can see from the above heatmap that variable fork,pflt,vflt are highly correlated , scall is correlated to , sread. pgfree is highly correlated to ppgout, pgscan. sread is highly correlated to swrite .ppgout and pgout are highly correlated. Freemem,freeswap are moderately correlated, exec-fork,pflt.vflt are also moderately correlated .The target variable usr is moderately correlated with freeswap

Bivariate Analysis of numerical variable with categorical variable runqsz

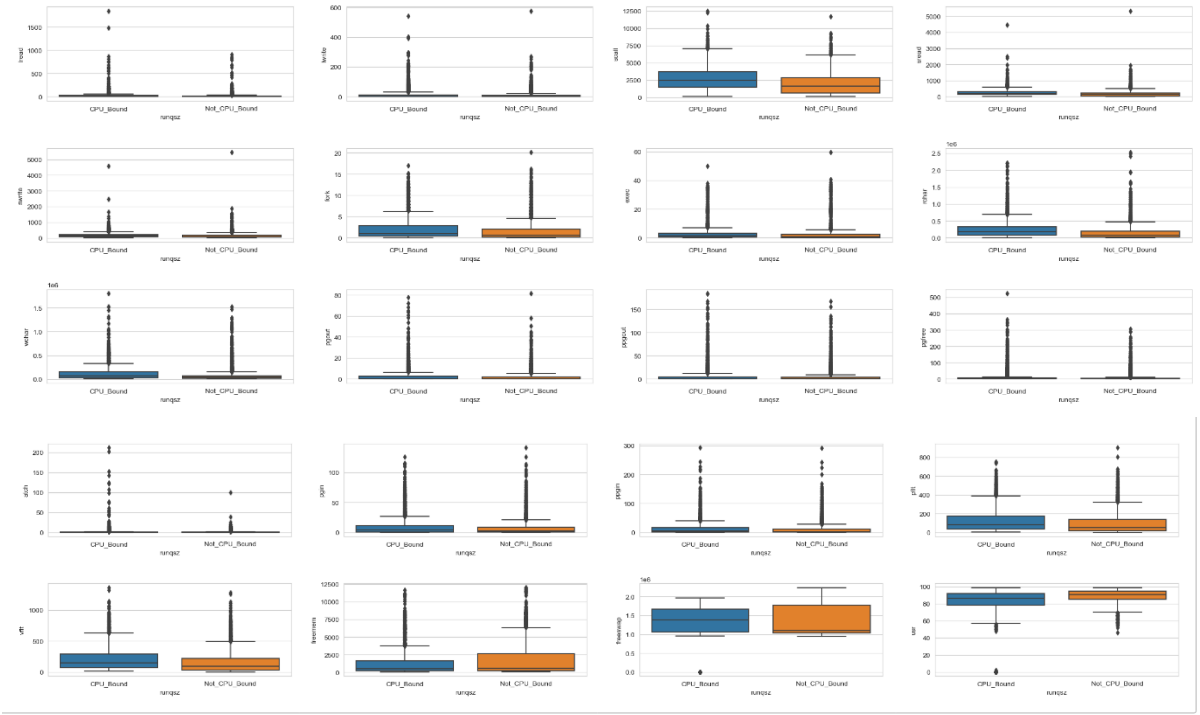
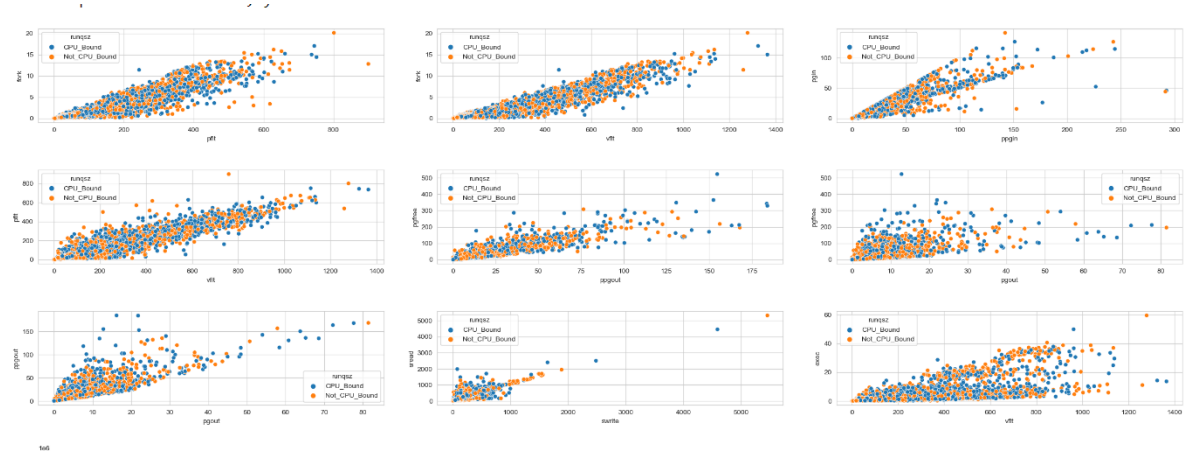


Figure 5: Boxplot of independent variables against categorical variables

We can infer from the above plot that mean values of variables freeswap, scall, wchar,rchar,pflt are very different for CPU bound and Non CPU bound. Mean 'usr' value for CPU_Bound is lower than Non_CPU_Bound

Multivariate Analysis

Let's analyse the scatterplot between the numerical variables with hue as CPU bound or Not CPU bound



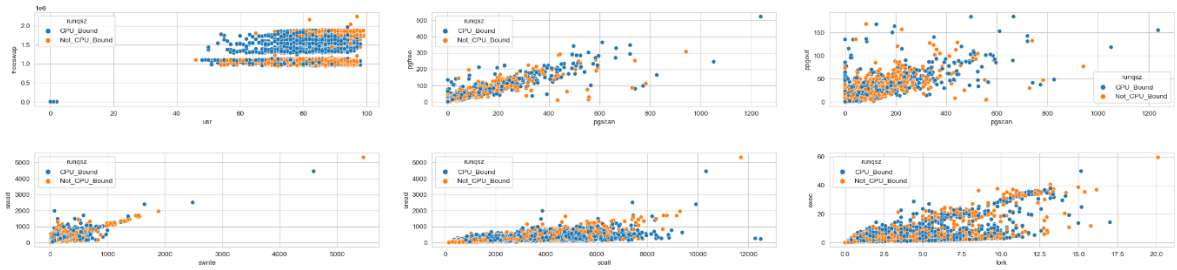


Figure 6: scatterplot between the numerical variables

We can see from the above plot that variable fork,pflt,vflt are highly correlated to each other , scall is moderately correlated to , sread. pgfree is highly correlated to ppgout, pgscan. .ppgout and pgout are highly correlated. Freemem, freeswap are moderately correlated. The target variable usr is moderately correlated with freeswap.

2. Impute null values if present, also check for the values which are equal to zero. Do they have any meaning or do we need to change them or drop them? Check for the possibility of creating new features if required. Also check for outliers and duplicates if there.

There are various ways to handle missing values. We can drop the rows, replace missing values with the mean/median of the available values, etc. Instead of dropping the rows, we will be replacing the missing values with median values.rchar and wchar variables have missing values and will be imputed with median values Before imputing the null values , let see the count of null values in each of the columns

```
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 3: Count of Null values

Only rchar and wchar see to be having null values , rhar has 104 and wchar has 15 null values

After Imputing the null values with median values , let print the null value count of columns again

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

Table 4: Count of Null values after imputing

As we can see there are no null values now

Let's see the Data summary after imputing the values

The null values of rchar and wchar are imputed to the median values

There are no duplicates in the dataset

Outliers

The linear regression model is very sensitive to outliers in the data set so we have to treat outliers.

Before the outlier treatment let's have a look at the boxplot of the variables

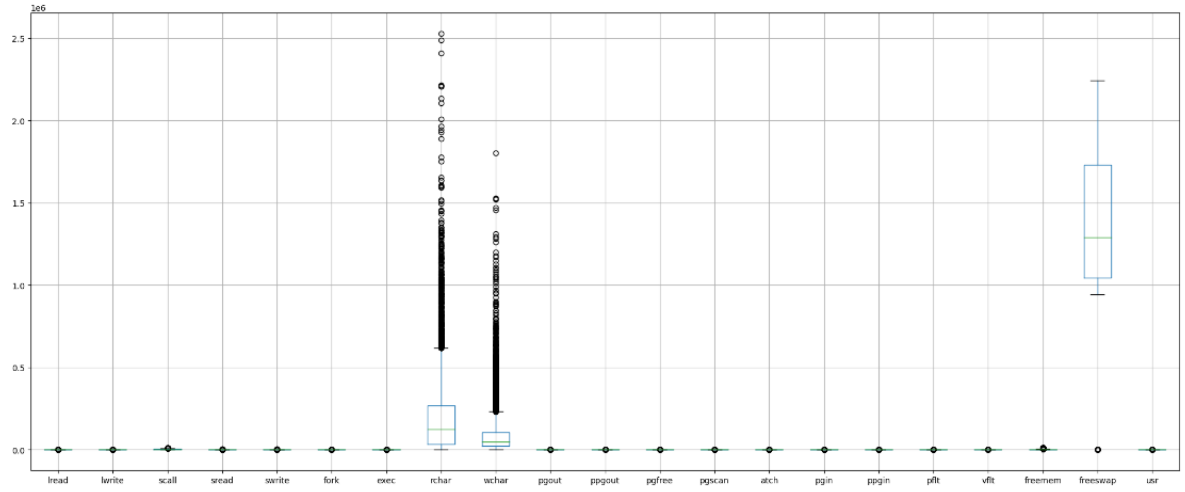


Figure 7:Boxplot of the numerical variables

As we can see , there are many outliers and to treat the outlier , we can use the cap and floor method to the cap the maximum value of the outlier value to Q3+1.5IQR and minimum value of the outlier value to Q1-1.5IQR.

After the outlier treatment

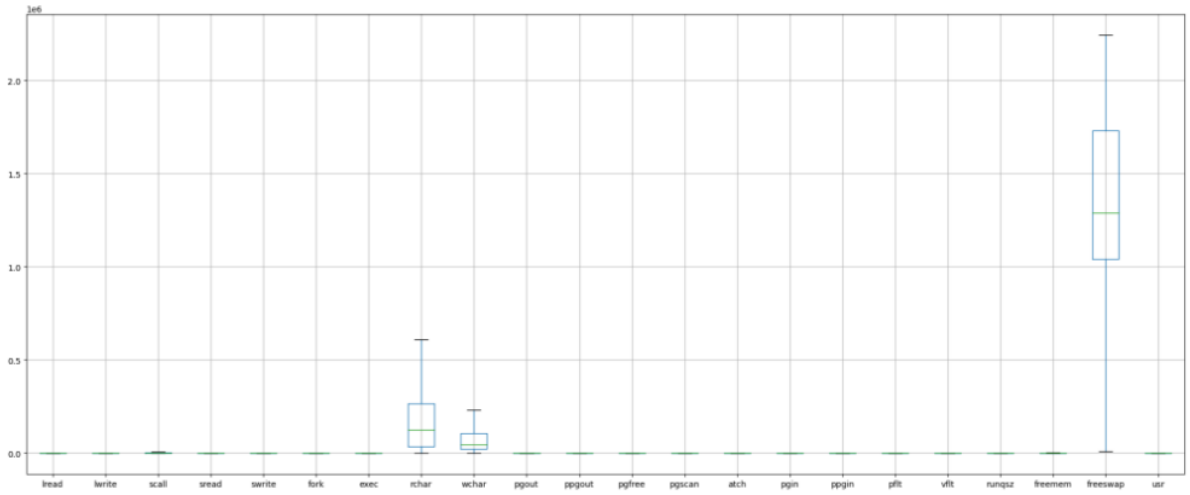


Figure 8: Boxplot of the numerical variables after outlier treatment

As we can see form the plot above , all the outliers have been treated .Let's see the data summary

	count	mean	std	min	25%	50%	75%	max
lread	8192.0	1.342285e+01	15.159741	0.0	2.00	7.0	20.000	4.700000e+01
lwrite	8192.0	6.657471e+00	9.291945	0.0	0.00	1.0	10.000	2.500000e+01
scall	8192.0	2.294484e+03	1593.093446	109.0	1012.00	2051.5	3317.250	6.775125e+03
sread	8192.0	1.997764e+02	146.758932	6.0	86.00	166.0	279.000	5.685000e+02
swrite	8192.0	1.379700e+02	97.141835	7.0	63.00	117.0	185.000	3.680000e+02
fork	8192.0	1.557771e+00	1.591220	0.0	0.40	0.8	2.200	4.900000e+00
exec	8192.0	1.931495e+00	2.028253	0.0	0.20	1.2	2.800	6.700000e+00
rchar	8192.0	1.788841e+05	174589.212910	278.0	34860.50	125473.5	265394.750	6.111961e+05
wchar	8192.0	7.564554e+04	71262.958027	1498.0	22977.75	46619.0	106037.000	2.306259e+05
pgout	8192.0	1.420901e+00	2.200251	0.0	0.00	0.0	2.400	6.000000e+00
ppgout	8192.0	2.560702e+00	4.037317	0.0	0.00	0.0	4.200	1.050000e+01
pgfree	8192.0	3.164586e+00	4.983345	0.0	0.00	0.0	5.000	1.250000e+01
pgscan	8192.0	0.000000e+00	0.000000	0.0	0.00	0.0	0.000	0.000000e+00
atch	8192.0	3.882788e-01	0.562937	0.0	0.00	0.0	0.600	1.500000e+00
pgin	8192.0	6.385262e+00	7.684420	0.0	0.60	2.8	9.765	2.351250e+01
ppgin	8192.0	9.140437e+00	11.160927	0.0	0.60	3.8	13.800	3.360000e+01
pflt	8192.0	1.056361e+02	101.548788	0.0	25.00	63.8	159.600	3.615000e+02
vflt	8192.0	1.756225e+02	162.497031	0.2	45.40	120.4	251.800	5.614000e+02
runqsz	8192.0	5.286865e-01	0.499207	0.0	0.00	1.0	1.000	1.000000e+00
freemem	8192.0	1.387625e+03	1605.763418	55.0	231.00	579.0	2002.250	4.659125e+03
freeswap	8192.0	1.328520e+06	420782.723746	10989.5	1042623.50	1289289.5	1730379.500	2.243187e+06
usr	8192.0	8.624622e+01	9.748585	61.5	81.00	89.0	94.000	9.900000e+01

Table 5: Data Summary after outlier treatment

We can see the after outlier treatment pgscan value is 0 for the max value too.

Since this variable is not contributing anything to the model, we can drop the variable .

3. Encode the data (having string values) for Modelling. Split the data into train and test (70:30). Apply Linear regression using scikit learn. Perform checks for significant variables using appropriate method from statsmodel. Create multiple models and check the performance of Predictions on Train and Test sets using Rsquare, RMSE & Adj Rsquare. Compare these models and select the best one with appropriate reasoning.

runqsz variable is a categorical variable with values Not_CPU_Bound and CPU_Bound, but categorical variable cannot be included to build a linear equation, so it has to be converted to numerical value. This can be done by label encoding or one hot encoding .

After converting the variable to a numerical variable , the Not_CPU_Bound value is encoded to 1 and CPU_Bound to 0.

.Before passing the data to a linear regression model, we have to split the data into training and testing.

Dependent variable is 'Usr' and other variables are independent variables .So we create two dataframes X and y. X will have all variables except 'usr' and y will have only the variable 'usr.

The train-test split is used to estimate the performance of machine learning algorithms that are applicable for prediction-based Algorithms/Applications. The Test set is split into 30 % of actual data and the training set is split into 70% of the actual data. We need to split a dataset into train and test sets to evaluate how well our machine learning model performs. The train set is used to fit the model, and the statistics of the train set are known. The second set is called the test data set, this set is solely used for predictions.

After splitting the training data has 5734 records and testing data has 2458 records .

Linear Regression using statsmodel(OLS)

Let's first see a model without the treatment of outliers

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.643
Model:	OLS	Adj. R-squared:	0.642
Method:	Least Squares	F-statistic:	489.6
Date:	Tue, 23 May 2023	Prob (F-statistic):	0.00
Time:	16:21:19	Log-Likelihood:	-21788.
No. Observations:	5734	AIC:	4.362e+04
Df Residuals:	5712	BIC:	4.377e+04
Df Model:	21		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	44.6380	0.746	59.831	0.000	43.175	46.101
lread	-0.0199	0.003	-6.214	0.000	-0.026	-0.014
lwrite	0.0048	0.006	0.795	0.427	-0.007	0.017
scall	0.0010	0.000	7.451	0.000	0.001	0.001
sread	-0.0005	0.002	-0.257	0.797	-0.004	0.003
swrite	-0.0020	0.002	-1.018	0.309	-0.006	0.002
fork	-1.7222	0.244	-7.052	0.000	-2.201	-1.244
exec	-0.0896	0.048	-1.879	0.060	-0.183	0.004
rchar	-4.062e-06	8.29e-07	-4.898	0.000	-5.69e-06	-2.44e-06
wchar	-1.164e-05	1.28e-06	-9.118	0.000	-1.41e-05	-9.14e-06
pgout	-0.1739	0.064	-2.717	0.007	-0.299	-0.048
ppgout	0.0989	0.037	2.701	0.007	0.027	0.171
pgfree	-0.0703	0.020	-3.508	0.000	-0.110	-0.031
pgscan	0.0086	0.006	1.362	0.173	-0.004	0.021
atch	-0.0786	0.027	-2.949	0.003	-0.131	-0.026
pgin	0.0913	0.029	3.103	0.002	0.034	0.149
ppgin	-0.0594	0.019	-3.128	0.002	-0.097	-0.022
pflt	-0.0415	0.004	-9.697	0.000	-0.050	-0.033
vflt	0.0223	0.003	6.665	0.000	0.016	0.029
runqsz	7.7908	0.303	25.693	0.000	7.196	8.385
freemem	-0.0016	7.53e-05	-21.489	0.000	-0.002	-0.001
freeswap	3.219e-05	4.54e-07	70.985	0.000	3.13e-05	3.31e-05

Omnibus:	1507.319	Durbin-Watson:	2.057
Prob(Omnibus):	0.000	Jarque-Bera (JB):	4768.238
Skew:	-1.333	Prob(JB):	0.00
Kurtosis:	6.585	Cond. No.	7.48e+06

Table 6:Linear Regression Model before outlier treatment

We see from the above summary that the R2 value is 64.3 meaning this model explains only 64.3 % of the variation .

Let's see a model after treating outliers

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.796
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1115.
Date:	Fri, 26 May 2023	Prob (F-statistic):	0.00
Time:	12:14:23	Log-Likelihood:	-16657.
No. Observations:	5734	AIC:	3.336e+04
Df Residuals:	5713	BIC:	3.350e+04
Df Model:	20		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.1217	0.316	266.106	0.000	83.502	84.741
lread	-0.0635	0.009	-7.071	0.000	-0.081	-0.046
lwrite	0.0482	0.013	3.671	0.000	0.022	0.074
scall	-0.0007	6.28e-05	-10.566	0.000	-0.001	-0.001
sread	0.0003	0.001	0.305	0.760	-0.002	0.002
swrite	-0.0054	0.001	-3.777	0.000	-0.008	-0.003
fork	0.0293	0.132	0.222	0.824	-0.229	0.288
exec	-0.3212	0.052	-6.220	0.000	-0.422	-0.220
rchar	-5.167e-06	4.88e-07	-10.598	0.000	-6.12e-06	-4.21e-06
wchar	-5.403e-06	1.03e-06	-5.232	0.000	-7.43e-06	-3.38e-06
pgout	-0.3688	0.090	-4.098	0.000	-0.545	-0.192
ppgout	-0.0766	0.079	-0.973	0.330	-0.231	0.078
pgfree	0.0845	0.048	1.769	0.077	-0.009	0.178
atch	0.6276	0.143	4.394	0.000	0.348	0.908
pgin	0.0200	0.028	0.703	0.482	-0.036	0.076
ppgin	-0.0673	0.020	-3.415	0.001	-0.106	-0.029
pflt	-0.0336	0.002	-16.957	0.000	-0.037	-0.030
vflt	-0.0055	0.001	-3.830	0.000	-0.008	-0.003
runqsz	1.6153	0.126	12.819	0.000	1.368	1.862
freemem	-0.0005	5.07e-05	-9.038	0.000	-0.001	-0.000
freeswap	8.832e-06	1.9e-07	46.472	0.000	8.46e-06	9.2e-06

Omnibus:	1103.645	Durbin-Watson:	2.016
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2372.553
Skew:	-1.119	Prob(JB):	0.00
Kurtosis:	5.219	Cond. No.	7.74e+06

Table 7: Linear Regression Model after outlier treatment

After treating outliers, we see that R-squared value has increased to 0.796. which means 79.6 % of the variance is explained by this model.

Let's look at the variance inflation factor to check which variables have very high multicollinearity

Multicollinearity exists when there is a correlation between multiple independent variables in a multiple regression model. This can adversely affect the regression results. Multicollinearity can be a problem in a regression model when using algorithms such as OLS (ordinary least squares) in statsmodels. This is because the estimated regression coefficients become unstable and difficult to interpret in the presence of multicollinearity.

There are several ways to detect multicollinearity in a dataset, such as using the Variance Inflation Factor (VIF) or calculating the correlation matrix of the independent variables. VIF (Variance Inflation Factors) is a measure of the amount of multicollinearity in regression analysis.

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

dtype: float64

Table 8: VIF values of the independent variables without dropping variables

- VIF starts at 1 and has no upper limit
- VIF = 1, no correlation between the independent variable and the other variables
- VIF exceeding 5 indicates high multicollinearity between this independent variable and the others

As can be seen there are many variables with VIF factor greater than 5. So, let's drop the variables one by one and see the impact on the R-squared .

After dropping variable 'ppgout' which has maximum VIF, the following is the model

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.796
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1174.
Date:	Sat, 27 May 2023	Prob (F-statistic):	0.00
Time:	17:49:47	Log-Likelihood:	-16658.
No. Observations:	5734	AIC:	3.336e+04
Df Residuals:	5714	BIC:	3.349e+04
Df Model:	19		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.1477	0.315	267.138	0.000	83.530	84.765
lread	-0.0635	0.009	-7.077	0.000	-0.081	-0.046
lwrite	0.0482	0.013	3.675	0.000	0.022	0.074
scall	-0.0007	6.28e-05	-10.575	0.000	-0.001	-0.001
sread	0.0003	0.001	0.303	0.762	-0.002	0.002
swrite	-0.0054	0.001	-3.782	0.000	-0.008	-0.003
fork	0.0325	0.132	0.247	0.805	-0.226	0.291
exec	-0.3225	0.052	-6.247	0.000	-0.424	-0.221
rchar	-5.166e-06	4.88e-07	-10.598	0.000	-6.12e-06	-4.21e-06
wchar	-5.45e-06	1.03e-06	-5.283	0.000	-7.47e-06	-3.43e-06
pgout	-0.4264	0.068	-6.286	0.000	-0.559	-0.293
pgfree	0.0477	0.029	1.634	0.102	-0.010	0.105
atch	0.6295	0.143	4.407	0.000	0.349	0.909
pgin	0.0212	0.028	0.745	0.456	-0.035	0.077
ppgin	-0.0685	0.020	-3.482	0.001	-0.107	-0.030
pflt	-0.0336	0.002	-16.957	0.000	-0.037	-0.030
vflt	-0.0055	0.001	-3.846	0.000	-0.008	-0.003
runqsz	1.6130	0.126	12.804	0.000	1.366	1.860
freemem	-0.0005	5.07e-05	-9.074	0.000	-0.001	-0.000
freeswap	8.824e-06	1.9e-07	46.472	0.000	8.45e-06	9.2e-06

Omnibus:	1102.077	Durbin-Watson:	2.016
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2366.754
Skew:	-1.118	Prob(JB):	0.00
Kurtosis:	5.216	Cond. No.	7.71e+06

Table 9: Linear regression model after dropping 'ppgout'

After dropping the 'ppgout', we see that the R-squared and Adjusted R-squared has not changed. Let's check the VIF values now

```
const       29.021961
lread        5.350387
lwrite       4.328325
scall        2.960379
sread        6.420135
swrite       5.597025
fork        13.027305
exec         3.239231
rchar        2.133614
wchar        1.580894
pgout        6.453978
pgfree       6.172847
atch         1.875553
pgin        13.784007
ppgin       13.898848
pflt        12.001460
vflt        15.966865
runqsz       1.156421
freemem      1.959267
freeswap     1.838167
dtype: float64
```

Table 10: VIF values of the independent variables after dropping variables 'ppgout'

We observe after dropping 'ppgout; the VIF values of 'pgout' and 'pgfree' has reduced considerably. The fact that these variables were highly correlated was also shown in the heatmap.

Next we drop 'vflt' that has the highest VIF values.

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.796
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1235.
Date:	Sat, 27 May 2023	Prob (F-statistic):	0.00
Time:	17:50:00	Log-Likelihood:	-16665.
No. Observations:	5734	AIC:	3.337e+04
Df Residuals:	5715	BIC:	3.349e+04
Df Model:	18		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.0090	0.313	268.139	0.000	83.395	84.623
lread	-0.0654	0.009	-7.281	0.000	-0.083	-0.048
lwrite	0.0491	0.013	3.735	0.000	0.023	0.075
scall	-0.0007	6.28e-05	-10.769	0.000	-0.001	-0.001
sread	-2.068e-05	0.001	-0.021	0.984	-0.002	0.002
swrite	-0.0053	0.001	-3.720	0.000	-0.008	-0.003
fork	-0.2082	0.116	-1.793	0.073	-0.436	0.019
exec	-0.3293	0.052	-6.376	0.000	-0.431	-0.228
rchar	-5.294e-06	4.87e-07	-10.871	0.000	-6.25e-06	-4.34e-06
wchar	-4.982e-06	1.03e-06	-4.858	0.000	-6.99e-06	-2.97e-06
pgout	-0.4205	0.068	-6.194	0.000	-0.554	-0.287
pgfree	0.0408	0.029	1.397	0.162	-0.016	0.098
atch	0.5868	0.143	4.116	0.000	0.307	0.866
pgin	0.0086	0.028	0.305	0.760	-0.047	0.064
ppgin	-0.0685	0.020	-3.476	0.001	-0.107	-0.030
pflt	-0.0373	0.002	-21.570	0.000	-0.041	-0.034
runqsz	1.6096	0.126	12.761	0.000	1.362	1.857
freemem	-0.0005	5.07e-05	-9.165	0.000	-0.001	-0.000
freeswap	8.945e-06	1.87e-07	47.712	0.000	8.58e-06	9.31e-06

Omnibus:	1058.324	Durbin-Watson:	2.014
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2225.362
Skew:	-1.085	Prob(JB):	0.00
Kurtosis:	5.145	Cond. No.	7.65e+06

Table 11: Linear regression model after dropping 'vflt'

After dropping 'vflt' , we see that the R-squared and Adjusted R-squared has not changed. Let's check the VIF values now.

VIF values:

const	28.641818
lread	5.335455
lwrite	4.327130
scall	2.952947
sread	6.374687
swrite	5.595777
fork	10.089700
exec	3.235396
rchar	2.123783
wchar	1.558923
pgout	6.450724
pgfree	6.149223
atch	1.864254
pgin	13.602134
ppgin	13.898845
pflt	9.131802
runqsz	1.156363
freemem	1.957966
freeswap	1.787695

dtype: float64

Table 12: VIF values of the independent variables after dropping variables 'vflt'

We observe that VIF values of 'fork' and 'pflt' values have reduced after dropping 'vflt' because they were highly correlated with 'vflt'

Next, we can drop 'ppgin' because it has the highest VIF value

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.795
Model:	OLS	Adj. R-squared:	0.795
Method:	Least Squares	F-statistic:	1305.
Date:	Sat, 27 May 2023	Prob (F-statistic):	0.00
Time:	17:50:14	Log-Likelihood:	-16671.
No. Observations:	5734	AIC:	3.338e+04
Df Residuals:	5716	BIC:	3.350e+04
Df Model:	17		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.0531	0.313	268.240	0.000	83.439	84.667
lread	-0.0678	0.009	-7.563	0.000	-0.085	-0.050
lwrite	0.0513	0.013	3.909	0.000	0.026	0.077
scall	-0.0007	6.29e-05	-10.692	0.000	-0.001	-0.001
sread	-4.826e-06	0.001	-0.005	0.996	-0.002	0.002
swrite	-0.0054	0.001	-3.732	0.000	-0.008	-0.003
fork	-0.1927	0.116	-1.659	0.097	-0.420	0.035
exec	-0.3290	0.052	-6.364	0.000	-0.430	-0.228
rchar	-5.506e-06	4.84e-07	-11.385	0.000	-6.45e-06	-4.56e-06
wchar	-4.978e-06	1.03e-06	-4.849	0.000	-6.99e-06	-2.97e-06
pgout	-0.4138	0.068	-6.091	0.000	-0.547	-0.281
pgfree	0.0311	0.029	1.070	0.284	-0.026	0.088
atch	0.5966	0.143	4.181	0.000	0.317	0.876
pgin	-0.0839	0.009	-8.848	0.000	-0.103	-0.065
pflt	-0.0374	0.002	-21.568	0.000	-0.041	-0.034
runqsz	1.6017	0.126	12.689	0.000	1.354	1.849
freemem	-0.0005	5.08e-05	-9.196	0.000	-0.001	-0.000
freeswap	8.922e-06	1.88e-07	47.572	0.000	8.55e-06	9.29e-06

Omnibus:	1052.296	Durbin-Watson:	2.014
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2207.367
Skew:	-1.081	Prob(JB):	0.00
Kurtosis:	5.137	Cond. No.	7.65e+06

Table 13: Linear regression model after dropping 'ppgin'

After dropping 'ppgin , we see that the R-squared has reduced and but Adjusted R-squared has not changed.

So, there is negligible impact of dropping the variable Let's check the VIF values now.

```
VIF values:

const        28.594882
lread         5.304009
lwrite        4.316362
scall         2.951826
sread         6.374556
swrite        5.595670
fork         10.074886
exec          3.235387
rchar         2.090401
wchar         1.558921
pgout         6.445478
pgfree        6.093623
atch          1.863536
pgin          1.529142
pflt          9.131545
runqsz        1.155990
freemem       1.957713
freeswap      1.785393
dtype: float64
```

Table 14: VIF values of the independent variables after dropping variables 'ppgin'

We observe that VIF values of 'pgin' has reduced after dropping 'ppgin' because they were highly correlated

Next we can drop the variable 'fork' because it has the highest VIF value.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    usr   R-squared:                       0.795
Model:                            OLS   Adj. R-squared:                  0.794
Method:                 Least Squares   F-statistic:                     1386.
Date:                Sat, 27 May 2023   Prob (F-statistic):              0.00
Time:                        17:50:23   Log-Likelihood:                -16672.
No. Observations:                5734   AIC:                         3.338e+04
Df Residuals:                    5717   BIC:                         3.349e+04
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         84.0913      0.313    269.048      0.000      83.479      84.704
lread         -0.0686      0.009     -7.675      0.000      -0.086      -0.051
lwrite         0.0527      0.013      4.025      0.000       0.027       0.078
scall         -0.0007   6.25e-05    -10.572      0.000      -0.001      -0.001
sread       1.712e-05      0.001      0.017      0.986      -0.002       0.002
swrite        -0.0058      0.001     -4.126      0.000      -0.009      -0.003
exec          -0.3583      0.049     -7.375      0.000      -0.454      -0.263
rchar      -5.524e-06   4.84e-07    -11.423      0.000   -6.47e-06   -4.58e-06
wchar      -4.854e-06   1.02e-06     -4.740      0.000   -6.86e-06   -2.85e-06
pgout         -0.4133      0.068     -6.084      0.000      -0.547      -0.280
pgfree         0.0307      0.029      1.054      0.292      -0.026       0.088
atch           0.6020      0.143      4.220      0.000       0.322       0.882
pgin          -0.0833      0.009     -8.789      0.000      -0.102      -0.065
pflt          -0.0396      0.001    -37.167      0.000      -0.042      -0.038
runqsz         1.5972      0.126     12.654      0.000       1.350       1.845
freemem       -0.0005   5.08e-05     -9.222      0.000      -0.001      -0.000
freeswap    8.91e-06   1.87e-07     47.536      0.000    8.54e-06    9.28e-06
==============================================================================
Omnibus:                     1044.060   Durbin-Watson:                   2.014
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2198.744
Skew:                          -1.071   Prob(JB):                         0.00
Kurtosis:                       5.148   Cond. No.                     7.62e+06
==============================================================================
```

Table 15: Linear regression model after dropping 'fork'

After dropping 'fork' , we see that the R-squared has not reduced and but Adjusted R-squared has dropped by .001. which is a negligible amount. So, there is negligible impact of dropping the variable Let's check the VIF values now.

const	28.440419
lread	5.285069
lwrite	4.298019
scall	2.914853
sread	6.373458
swrite	5.390263
exec	2.856973
rchar	2.089364
wchar	1.550686
pgout	6.445377
pgfree	6.093041
atch	1.862553
pgin	1.526800
pflt	3.458168
runqsz	1.155448
freemem	1.957226
freeswap	1.782829

dtype: float64

Table 16: VIF values of the independent variables after dropping variables 'fork'

'fork' was very highly correlated to 'pflt' and moderately correlated to 'exec', hence on dropping 'fork' we see that 'pflt', VIF has dropped considerably and 'exec' values also reduced.

Next, we can drop 'pgfree'

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.795
Model:	OLS	Adj. R-squared:	0.794
Method:	Least Squares	F-statistic:	1478.
Date:	Sat, 27 May 2023	Prob (F-statistic):	0.00
Time:	17:50:58	Log-Likelihood:	-16673.
No. Observations:	5734	AIC:	3.338e+04
Df Residuals:	5718	BIC:	3.348e+04
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.0915	0.313	269.046	0.000	83.479	84.704
lread	-0.0683	0.009	-7.646	0.000	-0.086	-0.051
lwrite	0.0522	0.013	3.989	0.000	0.027	0.078
scall	-0.0007	6.24e-05	-10.617	0.000	-0.001	-0.001
sread	2.832e-05	0.001	0.028	0.978	-0.002	0.002
swrite	-0.0058	0.001	-4.116	0.000	-0.009	-0.003
exec	-0.3567	0.049	-7.345	0.000	-0.452	-0.261
rchar	-5.517e-06	4.84e-07	-11.411	0.000	-6.47e-06	-4.57e-06
wchar	-4.869e-06	1.02e-06	-4.756	0.000	-6.88e-06	-2.86e-06
pgout	-0.3541	0.038	-9.282	0.000	-0.429	-0.279
atch	0.6056	0.143	4.246	0.000	0.326	0.885
pgin	-0.0820	0.009	-8.726	0.000	-0.100	-0.064
pflt	-0.0396	0.001	-37.173	0.000	-0.042	-0.038
runqsz	1.5953	0.126	12.640	0.000	1.348	1.843
freemem	-0.0005	5.07e-05	-9.326	0.000	-0.001	-0.000
freeswap	8.916e-06	1.87e-07	47.584	0.000	8.55e-06	9.28e-06

Omnibus:	1045.844	Durbin-Watson:	2.014
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2203.581
Skew:	-1.073	Prob(JB):	0.00
Kurtosis:	5.150	Cond. No.	7.62e+06

Table 17: Linear regression model after dropping 'pgfree'

After dropping 'pgfree , we see that the R-squared and Adjusted R-squared has not changed . So, there is no impact of dropping the variable Let's check the VIF values now.

```
VIF values:

const       28.440413
lread        5.279884
lwrite       4.292116
scall        2.910969
sread        6.372750
swrite       5.389795
exec         2.854013
rchar        2.089028
wchar        1.550381
pgout        2.031561
atch         1.861498
pgin         1.498879
pflt         3.458070
runqsz       1.155222
freemem      1.946405
freeswap     1.781403
dtype: float64
```

Table 18: VIF values of the independent variables after dropping variables 'pgfree'

'pgfree' was very highly correlated to 'pgout', hence on dropping 'pgfree' we see that VIF of 'pgout' has reduced considerably.

Next, we can drop 'sread',

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.795
Model:	OLS	Adj. R-squared:	0.794
Method:	Least Squares	F-statistic:	1584.
Date:	Sat, 27 May 2023	Prob (F-statistic):	0.00
Time:	17:51:11	Log-Likelihood:	-16673.
No. Observations:	5734	AIC:	3.338e+04
Df Residuals:	5719	BIC:	3.348e+04
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.0919	0.312	269.420	0.000	83.480	84.704
lread	-0.0684	0.009	-7.653	0.000	-0.086	-0.051
lwrite	0.0523	0.013	3.995	0.000	0.027	0.078
scall	-0.0007	5.96e-05	-11.111	0.000	-0.001	-0.001
swrite	-0.0058	0.001	-5.481	0.000	-0.008	-0.004
exec	-0.3568	0.049	-7.355	0.000	-0.452	-0.262
rchar	-5.511e-06	4.33e-07	-12.740	0.000	-6.36e-06	-4.66e-06
wchar	-4.872e-06	1.02e-06	-4.779	0.000	-6.87e-06	-2.87e-06
pgout	-0.3540	0.038	-9.287	0.000	-0.429	-0.279
atch	0.6055	0.143	4.247	0.000	0.326	0.885
pgin	-0.0820	0.009	-8.730	0.000	-0.100	-0.064
pflt	-0.0396	0.001	-37.292	0.000	-0.042	-0.038
runqsz	1.5953	0.126	12.641	0.000	1.348	1.843
freemem	-0.0005	5.06e-05	-9.328	0.000	-0.001	-0.000
freeswap	8.915e-06	1.87e-07	47.769	0.000	8.55e-06	9.28e-06

Omnibus:	1045.912	Durbin-Watson:	2.014
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2203.816
Skew:	-1.073	Prob(JB):	0.00
Kurtosis:	5.150	Cond. No.	7.61e+06

Table 19: Linear regression model after dropping 'sread'

After dropping 'sread' , we see that the R-squared and Adjusted R-squared has not changed . So, there is no impact of dropping the variable .Let's check the VIF values now.

```
VIF values:

const       28.366778
lread        5.272488
lwrite       4.282984
scall        2.653943
swrite       3.012451
exec         2.847353
rchar        1.672481
wchar        1.537067
pgout        2.029172
atch         1.860242
pgin         1.497984
pflt         3.436202
runqsz       1.155214
freemem      1.945888
freeswap     1.767780
dtype: float64
```

Table 20: VIF values of the independent variables after dropping variables 'sread'

'sread' was very highly correlated to 'swrite' and 'scall', hence on dropping 'sread' we see that VIF of 'swrite' and 'scall' has reduced considerably.

Next we can drop 'lread'

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.793
Model:	OLS	Adj. R-squared:	0.792
Method:	Least Squares	F-statistic:	1684.
Date:	Sat, 27 May 2023	Prob (F-statistic):	0.00
Time:	17:51:18	Log-Likelihood:	-16702.
No. Observations:	5734	AIC:	3.343e+04
Df Residuals:	5720	BIC:	3.353e+04
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	83.9943	0.313	267.987	0.000	83.380	84.609
lwrite	-0.0347	0.007	-5.325	0.000	-0.047	-0.022
scall	-0.0007	5.98e-05	-11.434	0.000	-0.001	-0.001
swrite	-0.0059	0.001	-5.567	0.000	-0.008	-0.004
exec	-0.3937	0.049	-8.118	0.000	-0.489	-0.299
rchar	-5.539e-06	4.35e-07	-12.740	0.000	-6.39e-06	-4.69e-06
wchar	-4.52e-06	1.02e-06	-4.416	0.000	-6.53e-06	-2.51e-06
pgout	-0.3554	0.038	-9.278	0.000	-0.431	-0.280
atch	0.5867	0.143	4.096	0.000	0.306	0.868
pgin	-0.0942	0.009	-10.132	0.000	-0.112	-0.076
pflt	-0.0415	0.001	-39.927	0.000	-0.044	-0.039
runqsz	1.6886	0.126	13.376	0.000	1.441	1.936
freemem	-0.0005	5.09e-05	-9.195	0.000	-0.001	-0.000
freeswap	8.998e-06	1.87e-07	48.055	0.000	8.63e-06	9.37e-06

Omnibus:	1034.508	Durbin-Watson:	2.010
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2155.036
Skew:	-1.067	Prob(JB):	0.00
Kurtosis:	5.114	Cond. No.	7.61e+06

Table 21: Linear regression model after dropping 'lread'

After dropping 'lread' , we see that the R-squared and Adjusted R-squared has reduced by 0.002 . So, there is negligible impact of dropping the variable .Let's check the VIF values now.

```
VIF values:

const       28.319429
lwrite       1.051857
scall        2.647850
swrite       3.011789
exec         2.819082
rchar        1.672367
wchar        1.533942
pgout        2.029125
atch         1.859694
pgin         1.454496
pflt         3.254499
runqsz       1.144436
freemem      1.945632
freeswap     1.761784
dtype: float64
```

Table 22: VIF values of the independent variables after dropping variables 'lread'

'lread' was very highly correlated to 'lwrite', hence on dropping 'lread' we see that VIF of 'lwrite' has reduced considerably.

Now, we can see that all the variables have values less than 5, so we have dropped variables that are highly correlated.

Now that we do not have multicollinearity in our data, the p-values of the coefficients have become reliable and we can remove the non-significant predictor variables. Non-significant predictors are having p-value>0.05 .But there are no predictors with p-value>0.05. so, we can stop dropping the variables at this point

Let's see how good the model is in predicting .Lets print the top 10 rows of the data frame with actual values, predicted values and their residuals.

	Actual Values	Fitted Values	Residuals
0	91.0	90.985214	0.014786
1	94.0	91.740525	2.259475
2	61.5	75.127299	-13.627299
3	83.0	80.546759	2.453241
4	94.0	97.487993	-3.487993
5	98.0	98.459931	-0.459931
6	79.0	77.912998	1.087002
7	87.0	96.898262	-9.898262
8	83.0	82.399996	0.600004
9	82.0	82.390489	-0.390489

Table 23: Actuals Vs Fitted Values and Residuals

Linearity and Independence of predictors

The residuals vs Fitted values plot is as shown below. There is no pattern in the data thus the assumption of linearity and independence of predictors satisfied

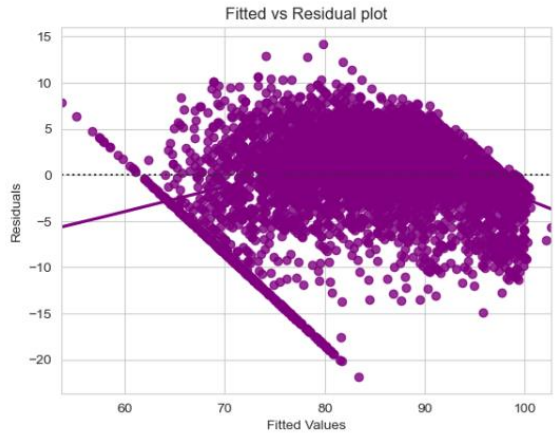


Figure 9: The residuals vs Fitted values plot

No pattern in the data thus the assumption of linearity and independence of predictors satisfied

Let's test for Normality of Residuals

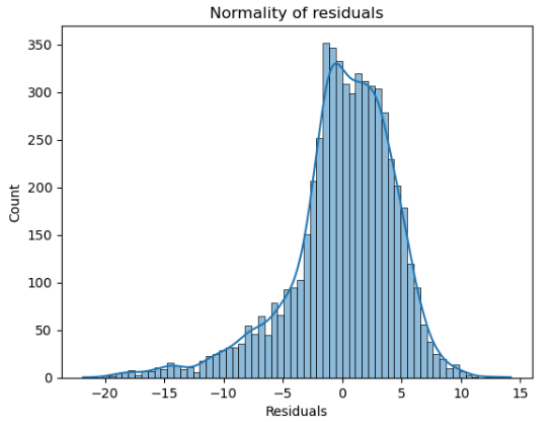


Figure 10: The residuals density plot

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

Null hypothesis - Data is normally distributed.

Alternate hypothesis - Data is not normally distributed.

ShapiroResult(statistic=0.9425298571586609, pvalue=1.4881789691129557e-42).

Since p-value < 0.05, we reject the null hypothesis that the residuals are normal as per shapiro test.
Also, the figure above proves that residual is not normal

Let's Test for Homoscedasticity using goldfeldquandt

The goldfeldquandt test can be used for checking the Homoscedasticity. The null and alternate hypotheses of the test are as follows:

Null hypothesis : Residuals are homoscedastic
Alternate hypothesis : Residuals have heteroscedasticity

The values are [('F statistic', 1.1138123553995614), ('p-value', 0.0020027008064391126)]
Since p-value < 0.05, we reject the null hypothesis that the residuals are homoscedastic as per goldfeldqua ndt test.
.QQ Plot residuals with a normal distribution will produce a straight-line plot. As we can see there is devi ation from the straight line indicating not a very normal distribution

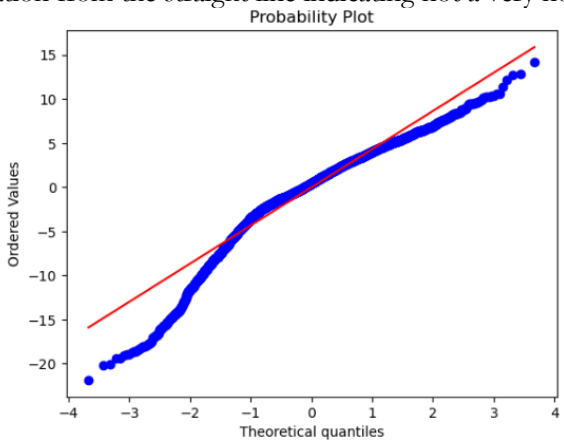


Figure 11:Q_Q plot

So, none of the assumption for the residuals are satisfied

The Root Mean Squared Error (RMSE)

RMSE is one of the main performance indicators for a regression model. It measures the average difference between values predicted by a model and the actual values. It provides an estimation of how well the model is able to predict the target value (accuracy).

RMSE for training data is 4.4544 and RMSE for testing data is 4.685. The lower the value of the **Root Mean Squared Error**, the better the model is. A perfect model (a hypothetic model that would always predict the exact expected value) would have a **Root Mean Squared Error** value of 0. We can improve the **Root Mean Squared Error** by adding more influencer variables in the training dataset.

The model equation will be as follows:

```
const       83.994323
lwrite      -0.034689
scall       -0.000684
swrite      -0.005904
exec        -0.393740
rchar       -0.000006
wchar       -0.000005
pgout       -0.355439
atch         0.586737
pgin        -0.094199
pflt        -0.041500
runqsz       1.688606
freemem     -0.000468
freeswap     0.000009
dtype: float64
```

Table 24: The model coefficient of variables

```
usr= 83.99432279722069 + -0.03468896257901462 * ( lwrite ) +  -0.000684284933243225 * ( scall ) +  -0.005904009677282263 * ( sw
rite ) +  -0.39373991015657783 * ( exec ) +  -5.5386991466067535e-06 * ( rchar ) +  -4.5202801228802394e-06 * ( wchar ) +  -0.3
5543894094213613 * ( pgout ) +  0.5867366021283322 * ( atch ) +  -0.09419918823792178 * ( pgin ) +  -0.041500224932716524 * ( p
flt ) +  1.6886056874993933 * ( runqsz ) +  -0.0004679344689752775 * ( freemem ) +  8.998425417637567e-06 * ( freeswap )
```

Linear Regression model using (sklearn)

Using Linear Regression, we get the coefficient of the variables as below

```
The coefficient for const is 0.0
The coefficient for lwrite is -0.03468896257897594
The coefficient for scall is -0.0006842849332436193
The coefficient for swrite is -0.005904009677276247
The coefficient for exec is -0.39373991015642895
The coefficient for rchar is -5.5386991466068O3e-06
The coefficient for wchar is -4.520280122845764e-06
The coefficient for pgout is -0.355438940942057
The coefficient for atch is 0.5867366021279309
The coefficient for pgin is -0.09419918823789823
The coefficient for pflt is -0.041500224932719106
The coefficient for runqsz is 1.6886056874994104
The coefficient for freemem is -0.00046793446897557846
The coefficient for freeswap is 8.998425417652394e-06
```

The intercept for our Linear Regression model is 83.99432279719366
Using this method, the R-Squared value we get for training data is 0.7928767063811195
Using this method, the R-Squared value we get for testing data is 0.7644517444502072
RMSE for trained data is 4.454425641530247
RMSE for testing data is 4.685030644797855

As we can see using both statsmodels and LinearRegression model both give same values of R-squared a nd RMSE values

To improve the performance of the model we can add new variables

R-squared is a function of your between-group sum of squares relative to your total sum of squares, or the ratio of the variance explained relative to the total variance in the model. When you add another variable, even if it does not significantly account additional variance, it will likely account for at least some. Thus, adding another variable into the model likely increases the between sum of squares, which in turn increases your R-squared value. Let's print the correlation map between the independent and dependent variables

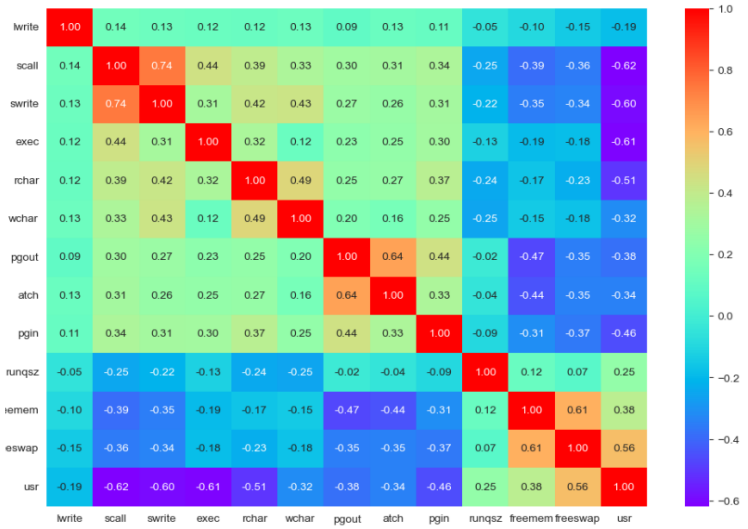


Table 25:Correlation Heatmap of the variables

As we can see the variables 'freeswap' , 'scall' and 'exec' are moderately correlated to target variable usr.

So, let's create new variables by transforming the columns by squaring them and see if the model performs better. After adding new columns freeswap_sq, exec_squared, the R-squared value has increased to 0.897 from the last model which has R-squared values 0.793

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    usr   R-squared:                       0.897
Model:                            OLS   Adj. R-squared:                  0.897
Method:                 Least Squares   F-statistic:                     3314.
Date:                Sun, 04 Jun 2023   Prob (F-statistic):               0.00
Time:                        12:25:38   Log-Likelihood:                -14704.
No. Observations:                5734   AIC:                         2.944e+04
Df Residuals:                    5718   BIC:                         2.955e+04
Df Model:                          15
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const           73.8250      0.259    285.120      0.000      73.317      74.333
lwrite          -0.0300      0.005     -6.526      0.000      -0.039      -0.021
scall           -0.0016   4.42e-05    -35.238      0.000      -0.002      -0.001
swrite          -0.0075      0.001     -9.884      0.000      -0.009      -0.006
exec             0.5840      0.086      6.824      0.000       0.416       0.752
rchar        -3.876e-06   3.09e-07    -12.534      0.000   -4.48e-06   -3.27e-06
wchar        -3.753e-06   7.23e-07     -5.195      0.000   -5.17e-06   -2.34e-06
pgout           -0.2117      0.027     -7.808      0.000      -0.265      -0.159
atch             0.3095      0.101      3.059      0.002       0.111       0.508
pgin            -0.1753      0.007    -26.357      0.000      -0.188      -0.162
pflt            -0.0394      0.001    -53.189      0.000      -0.041      -0.038
runqsz           0.1564      0.092      1.702      0.089      -0.024       0.336
freemem          0.0005   3.84e-05     13.292      0.000       0.000       0.001
freeswap      3.708e-05   4.05e-07     91.639      0.000    3.63e-05    3.79e-05
freeswap_sq  -1.349e-11   1.84e-13    -73.129      0.000   -1.39e-11   -1.31e-11
exec_squared    -0.1681      0.011    -14.692      0.000      -0.191      -0.146
==============================================================================
Omnibus:                      561.835   Durbin-Watson:                   1.954
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2844.505
Skew:                          -0.339   Prob(JB):                         0.00
Kurtosis:                       6.383   Cond. No.                     1.38e+13
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.38e+13. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Table 26: Linear regression model after adding 'freeswap_sq' and 'exec_squared'

Let's see the value of residuals and plot them

	Actual Values	Fitted Values	Residuals
0	91.0	93.013443	-2.013443
1	94.0	94.860814	-0.860814
2	61.5	63.026092	-1.526092
3	83.0	83.326246	-0.326246
4	94.0	97.352893	-3.352893
5	98.0	96.529872	1.470128
6	79.0	79.417442	-0.417442
7	87.0	93.865664	-6.865664
8	83.0	84.849790	-1.849790
9	82.0	82.446465	-0.446465

Table 27:Actuals Vs Fitted Values vs Residuals

The residuals vs Fitted values plot is as shown below.

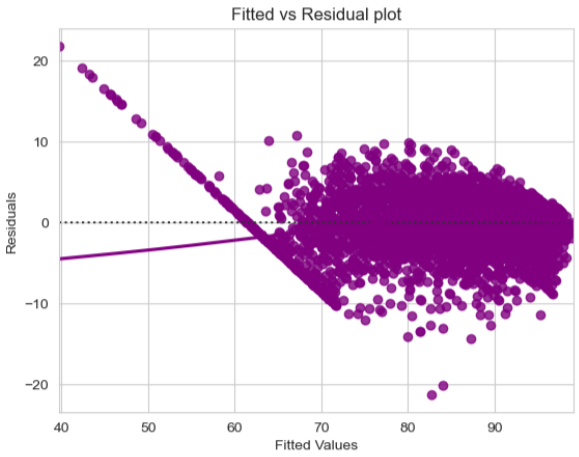


Figure 12: residuals vs Fitted values plot

There is no pattern in the data thus the assumption of linearity and independence of predictors satisfied

Let's test for Normality of Residuals.

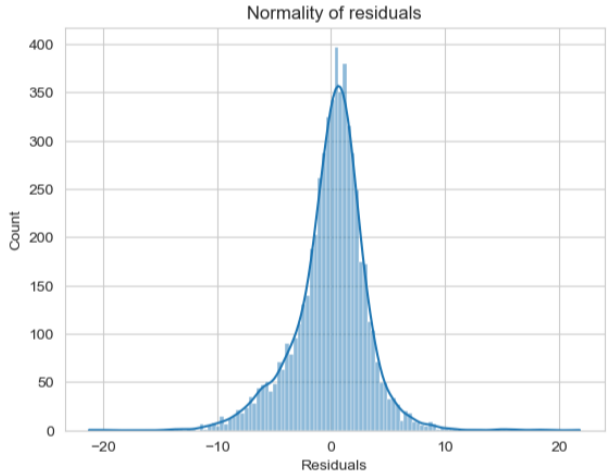


Figure 13:Density plot of residuals

ShapiroResult(statistic=0.9531151056289673, pvalue=1.9972090440697315e-39)
Since p-value < 0.05, the residuals are not normal as per shapiro test.
Also, the figure above proves that residual is not normal

Let's Test for Homoscedasticity using goldfeldquandt test
[('F statistic', 1.0105955302563507), ('p-value', 0.38923628029162954)]

Since p-value > 0.05, we can say the residuals are homoscedastic.

QQ Plot residuals with a normal distribution will produce a straight-line plot. As we can see there is devia tion from the straight line indicating not a very normal distribution , but it's better than the model before

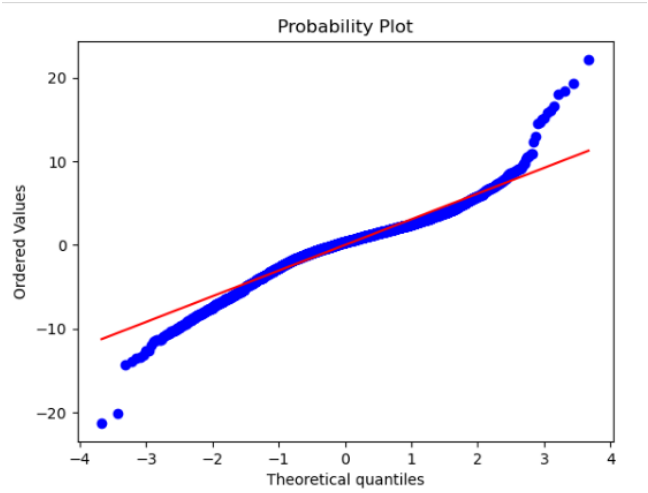


Figure 14:Q-Q plot

Let's see the coefficients of the independent variables with newer variables freeswap_sq and exec_squared

```
const           7.382496e+01
lwrite         -3.001324e-02
scall          -1.557416e-03
swrite         -7.464151e-03
exec            5.840004e-01
rchar          -3.876321e-06
wchar          -3.753404e-06
pgout          -2.116792e-01
atch            3.095073e-01
pgin           -1.753075e-01
pflt           -3.943711e-02
runqsz          1.563800e-01
freemem         5.101357e-04
freeswap        3.707648e-05
freeswap_sq    -1.349211e-11
exec_squared   -1.681209e-01
dtype: float64
```

Figure 15 Coefficients of the independent variables

The model equation will be as follows with new variables is as follows

```
usr= 73.8249647115637 + -0.030013244928522316 * ( lwrite ) +  -0.0015574161523834223 * ( scall
) +  -0.0074641505712279635 * ( swrite ) +  0.5840004128758598 * ( exec ) +  -3.87632110546956
e-06 * ( rchar ) +  -3.7534035767923584e-06 * ( wchar ) +  -0.2116792479802895 * ( pgout ) +
0.3095072704297266 * ( atch ) +  -0.17530749139195279 * ( pgin ) +  -0.03943711343400853 * ( p
flt ) +  0.15638001492770479 * ( runqsz ) +  0.0005101357337461945 * ( freemem ) +  3.70764848
8400133e-05 * ( freeswap ) +  -1.3492105560477411e-11 * ( freeswap_sq ) +  -0.1681209102559742
5 * ( exec_squared )
```

Let's see the RMSE of the new model

1.RMSE for training data is 3.143 and RMSE for testing data is 3.224.The lower the value of the **Root M ean Squared Error**, the better the model is. As we can see the RMSE is less compared to the previous model. Also, we can see that RMSE on the train and test sets are not very different .
So, our model is not suffering from overfitting.
2. MAE indicates that our current model is able to predict usr within a mean error of 2.3 units on the test data and 2.26 units on the train data.
3. Hence, we can conclude the model is good for prediction as well as inference purposes.

Using Linear Regression, we get the coefficient of the variables as below

```
The coefficient for const is 0.0
The coefficient for lwrite is -0.03001327004637352
The coefficient for scall is -0.00155741591228199
The coefficient for swrite is -0.007464154330819872
The coefficient for exec is 0.5840002082414866
The coefficient for rchar is -3.8763215097133115e-06
The coefficient for wchar is -3.7534022323724933e-06
The coefficient for pgout is -0.21167925085387382
The coefficient for atch is 0.30950735432883203
The coefficient for pgin is -0.175307617273429
The coefficient for pflt is -0.03943710682097712
The coefficient for runqsz is 0.1563794824000068
The coefficient for freemem is 0.0005101361249574727
The coefficient for freeswap is 3.707648274444476e-05
The coefficient for freeswap_sq is -1.349212485622496e-11
The coefficient for exec_squared is -0.16812091026580747
```

The intercept for our Linear Regression model is 73.82497875273377

Using this method, the R-Squared value we get for training data is
0.8968392119448401
Using this method, the R-Squared value we get for testing data is
0.8884323833002334
RMSE for trained data is 3.1436527486678236

RMSE for testing data is 3.224345469751469

As we can see using both statsmodels and LinearRegression model both give same values of R-squared a nd RMSE values

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

We can analyse the coefficients to understand how each attribute affects the system to be in 'usr' mode. A positive coefficient indicates that an increase in the attribute value leads to an increase in the 'usr' value, while a negative coefficient indicates that an increase in the attribute value leads to a decrease in the 'usr' value.

In summary, the linear equation to predict 'usr' based on the system attributes can be shown as below

```
usr= 73.8249647115637 + -0.030013244928522316 * ( lwrite ) +  -0.0015574161523834223 * ( scall
) +  -0.0074641505712279635 * ( swrite ) +  0.5840004128758598 * ( exec ) +  -3.87632110546956
e-06 * ( rchar ) +  -3.7534035767923584e-06 * ( wchar ) +  -0.2116792479802895 * ( pgout ) +
0.3095072704297266 * ( atch ) +  -0.17530749139195279 * ( pgin ) +  -0.03943711343400853 * ( p
flt ) +  0.15638001492770479 * ( runqsz ) +  0.0005101357337461945 * ( freemem ) +  3.70764848
8400133e-05 * ( freeswap ) +  -1.3492105560477411e-11 * ( freeswap_sq ) +  -0.1681209102559742
5 * ( exec_squared )
```

A unit increase in the writes (transfers per second) between system memory and user memory will result in a 0.03 unit decrease in the usr, all other variables remaining constant.

A unit increase in the Number of system exec calls per second will result in a 0.58 unit increase in the usr, all other variables remaining constant

A unit increase in the Number of pages out requests per second will result in a 0.21 unit decrease in the usr, all other variables remaining constant

A unit increase in the Number of pages attaches per second will result in a 0.30 unit increase in the usr, all other variables remaining constant

A unit increase in the Number of page-in requests per second will result in a 0.17 unit decrease in the usr, all other variables remaining constant

The usr of Not CPU Bounded will be 0.156 units higher than a usr of CPU Bounded, all other variables remaining constant.

The columns 'scall', 'swrite','freemem', 'freeswap', 'wchar', 'rchar','freeswap_square' has minimal impact on the usr. A unit increase in these columns, amount of time spent on usr decreases by a minuscule amount.There are many variables that are highly correlated and some predictors are not very helpful in predicting the outcome, these variables need not be considered for prediction while collecting the data

Problem2 : Logistic Regression, LDA and CART

You are a statistician at the Republic of Indonesia Ministry of Health and you are provided with a data of 1473 females collected from a Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of the survey.

The problem is to predict do/don't they use a contraceptive method of choice based on their demographic and socio-economic characteristics.

Dataset for Problem 2: Contraceptive_method_dataset.xlsx

Data Dictionary:

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

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

Let's see the top 5 and bottom 5 records of the dataset

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

Figure 16: Top 5 records of the dataset

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

Figure 17: Bottom 5 records of the dataset

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

Table 28: Columns with data types

As we can see there are 1473 observations in the dataset.

There are 10 columns of which 3 are numerical and 7 are object type variables.

Contraceptive_method_used is the dependent variable

Data Summary

Let's see the data summary of the numerical variables

	Wife_age	No_of_children_born	Husband_Occupation
count	1402.000000	1452.000000	1473.000000
mean	32.606277	3.254132	2.137814
std	8.274927	2.365212	0.864857
min	16.000000	0.000000	1.000000
25%	26.000000	1.000000	1.000000
50%	32.000000	3.000000	2.000000
75%	39.000000	4.000000	3.000000
max	49.000000	16.000000	4.000000

Table 29 Data summary of the numerical variables

Husband_Occupation is a categorical variable with categories 1,2,3,4

Wife_age and No_of_children_born are numerical variables with a slightly right skewed distribution .

Min age of wife is 16 and maximum is 49. Min number of children is 0 and max is 16.

There are outliers in variable No_of_children_born

Duplicate check

We see that there are 80 duplicate records in the dataset.

We can drop the duplicate rows and then do EDA

Null value check

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

There are 67 null values in wife_age and 21 null values in no_of_children_born variable

EDA

Let's look at the categorical variables and number of records in each type of category

```
Wife_ education
Tertiary          577
Secondary         410
Primary           334
Uneducated        152
```

Wife education has four levels uneducated, Primary, Secondary, Tertiary . Tertiary has the highest count with 577 records and Uneducated has the least count of 152

```
Husband_education
Tertiary          899
Secondary         352
Primary           178
Uneducated         44
```

Husband education has four levels uneducated, Primary, Secondary, Tertiary . Tertiary has the highest count with 899 records and Uneducated has the least count of 44.

```
Husband_Occupation
3                    585
1                    436
2                    425
4                     27
```

Husband Occupation has four levels 1,2,3,4 . 3 has the highest count with 585 records and 4 has the least count of 27.

```
Wife_religion
Scientology        1253
Non-Scientology     220
```

Wife religion has 2 categories Scientology and Non-scientology . Scientology has the highest count with 1253 records and non-scientology has the least count of 220.

```
Wife_Working
No               1104
Yes               369
```

Wife working has 2 categories No and Yes. No has the highest count with 1104 records and non-scientology has the least count of 369.

```
Standard_of_living_index
Very High                   684
High                        431
Low                         229
Very Low                    129
```

Standard of living Index has four levels Low, very Low, High, Very High . Very High has the highest count with 684 records and Very Low has the least count of 129.

```
Media_exposure
Exposed            1364
Not-Exposed         109
```

Media Exposure has 2 categories Exposed and Not-Exposed. Exposed has the highest count with 1364 records and Not-Exposed has the least count of 109.

```
Contraceptive_method_used
Yes                          844
No                           629
```

Contraceptive method used has 2 categories No and Yes. Yes has the highest count with 844 records and No has the least count of 629.

Univariate Analysis

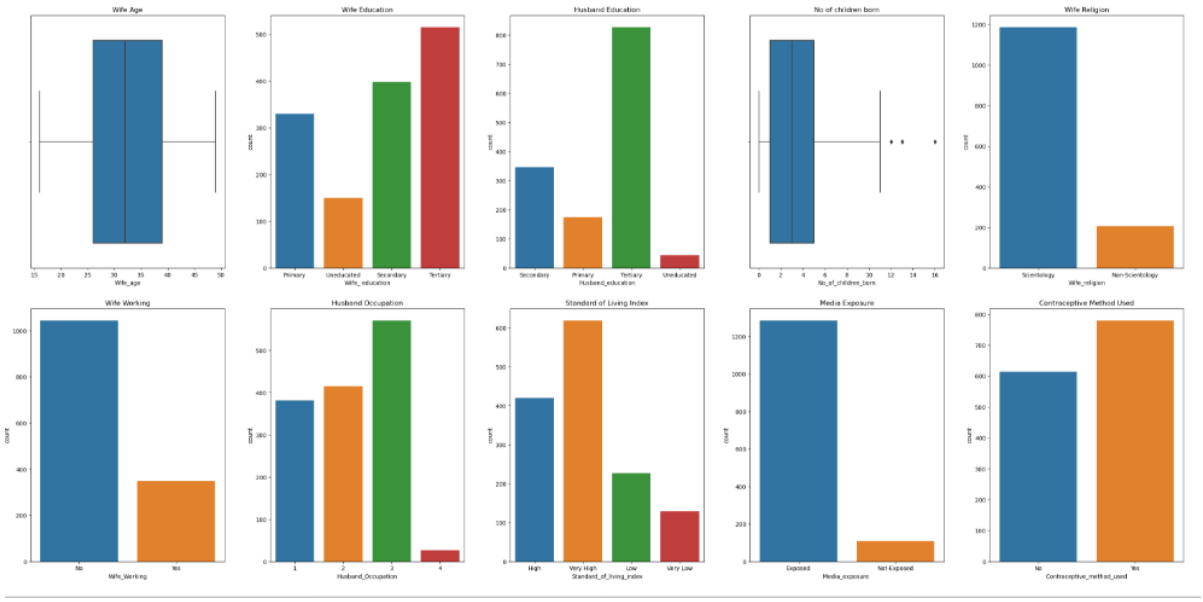


Figure 18: Univariate Analysis of the variables

We see from boxplot that Wife age variable is normally distributed .

No of children variable is right skewed with outliers.

Wife education has four levels uneducated, Primary, Secondary, Tertiary . Tertiary has the highest count with 577 records and Uneducated has the least count of 152

Husband education has four levels uneducated, Primary, Secondary, Tertiary . Tertiary has the highest count with 899 records and Uneducated has the least count of 44.

Husband Occupation has four levels 1,2,3,4 . 3 has the highest count with 585 records and 4 has the least count of 27.

Wife religion has 2 categories Scientology and Non-scientology . Scientology has the highest count with 1253 records and non-scientology has the least count of 220.

Wife working has 2 categories No and Yes. No has the highest count with 1104 records and non-scientology has the least count of 369.

Standard of living Index has four levels Low, very Low, High, Very High . Very High has the highest count with 684 records and Very Low has the least count of 129.

Media Exposure has 2 categories Exposed and Not-Exposed. Exposed has the highest count with 1364 records and Not-Exposed has the least count of 109.

Contraceptive method used has 2 categories No and Yes. Yes has the highest count with 844 records and No has the least count of 629

Bivariate Analysis

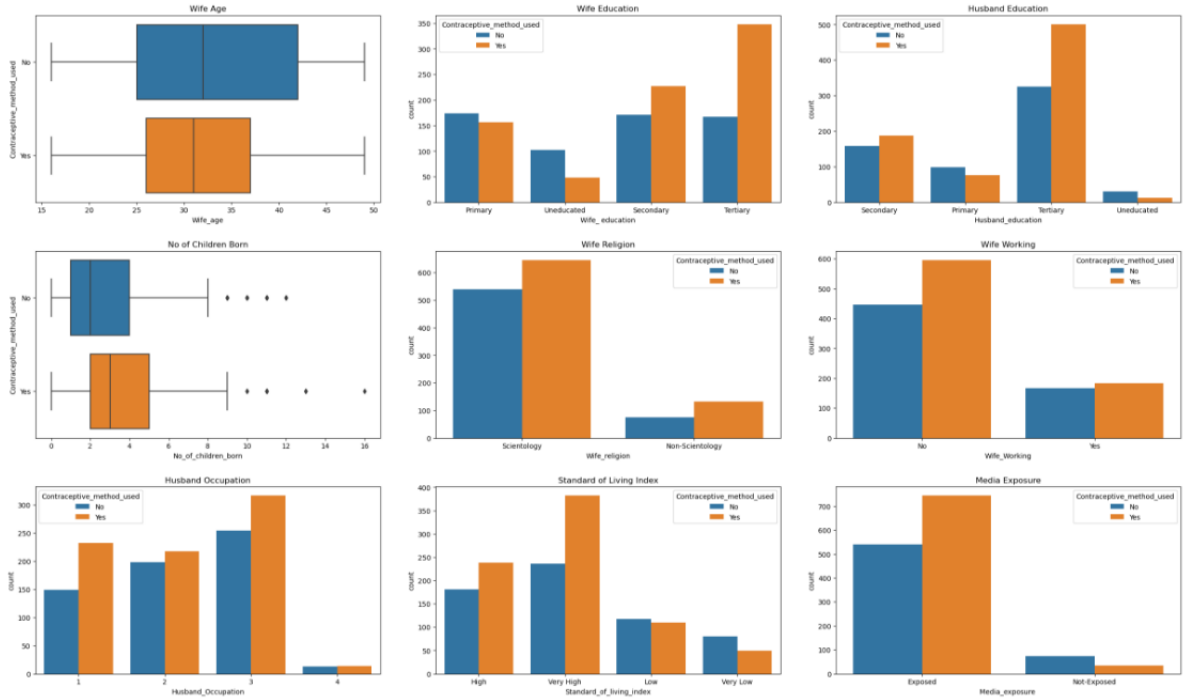


Figure 19:Bivariate Analysis of independent vs dependent variables

This is a bivariate analysis of the independent variables vis-à-vis the dependent variable

Mean age of Wife who uses contraceptive is 32 and Wife does who doesn't use contraceptive is 33

```
Contraceptive_method_used
No     33.0
Yes    32.0
```

Average No of children born to women who uses contraceptive is 3 and women who doesn't use contraceptive is 2

```
Contraceptive_method_used
No     2.0
Yes    3.0
Name: No_of_children_born,
```

Wife_ education	Contraceptive_method_used	
Primary	No	0.526946
	Yes	0.473054
Secondary	Yes	0.573171
	No	0.426829
Tertiary	Yes	0.696707
	No	0.303293
Uneducated	No	0.677632
	Yes	0.322368

Among the Uneducated Women only 32 % use contraceptive, among women with Tertiary education 70% use contraception ,among women with Primary education 47%, among women with Secondary education 57% use contraception

Husband_education	Contraceptive_method_used	
Primary	No	0.556180
	Yes	0.443820
Secondary	Yes	0.542614
	No	0.457386
Tertiary	Yes	0.624027
	No	0.375973
Uneducated	No	0.704545
	Yes	0.295455

Among women with husband's Uneducated only 29.5 % use contraceptive, women with husband's education Tertiary 62% use contraception , among women with husband's education Primary 44%, among women with husband's education secondary 54% use contraception

Husband_Occupation	Contraceptive_method_used	
1	Yes	0.637615
	No	0.362385
2	Yes	0.529412
	No	0.470588
3	Yes	0.558974
	No	0.441026
4	Yes	0.518519
	No	0.481481

Among the women with husband's occupation 1, 64 % use contraceptive, among women with husband's occupation 2, 53 % use contraceptive, among women with husband's occupation 3, 56 % use contraceptive, among women with husband's occupation 4, 52 % use contraceptive

Wife_religion	Contraceptive_method_used	
Non-Scientology	Yes	0.659091
	No	0.340909
Scientology	Yes	0.557861
	No	0.442139

Among the women with religion Scientology, 56% use contraceptive, Among the women with religion Non-Scientology, 66% use contraceptive

Wife_Working	Contraceptive_method_used	
No	Yes	0.584239
	No	0.415761
Yes	Yes	0.539295
	No	0.460705

Among the women working, 54% use contraceptive, Among the women who are not working, 58% use contraceptive

Standard_of_living_index	Contraceptive_method_used	
High	Yes	0.573086
	No	0.426914
Low	No	0.510917
	Yes	0.489083
Very High	Yes	0.637427
	No	0.362573
Very Low	No	0.620155
	Yes	0.379845

Among the women with standard of Living Index Very High, 64 % use contraceptive ,among the women with standard of Living Index High, 57 % use contraceptive, Among the women with standard of Living Index Very Low, 38 % use contraceptive, Among the women with standard of Living Index Low, 49 % use contraceptive

Media_exposure	Contraceptive_method_used	
Exposed	Yes	0.593109
	No	0.406891
Not-Exposed	No	0.678899
	Yes	0.321101

Among the women with Media exposure, 59 % use contraceptive ,among the women with No media exposure 32% use contraceptive .

Let's plot the variable number of children against other categorical variables

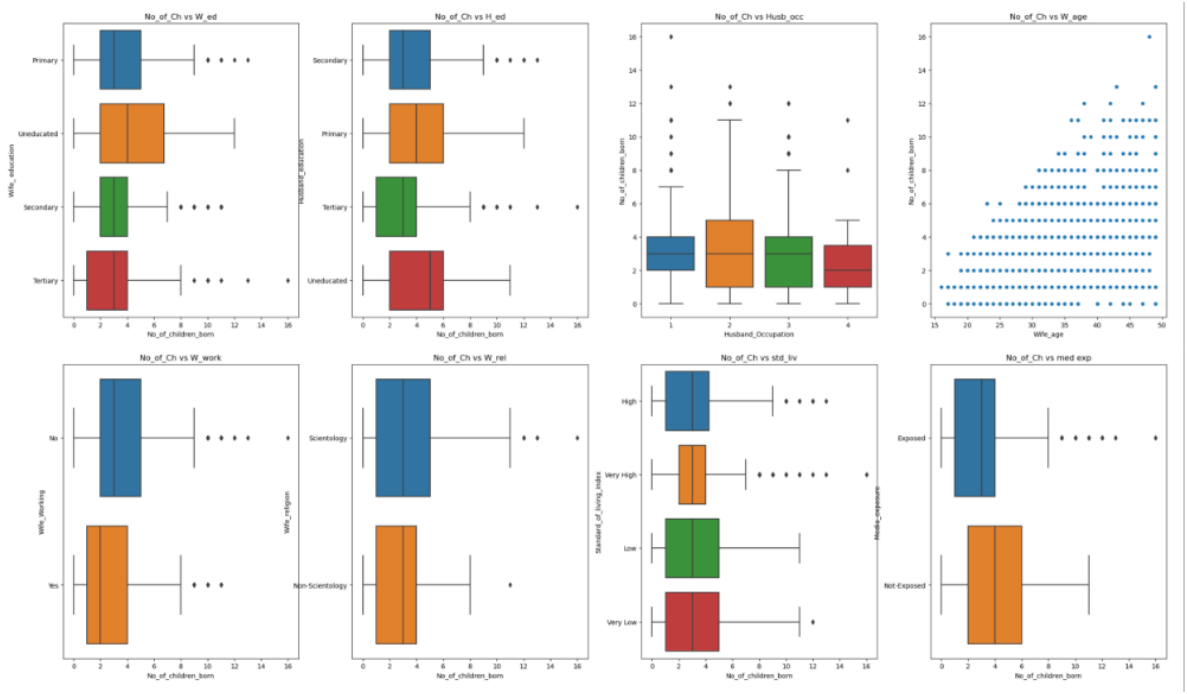


Figure 20: Bivariate Analysis of independent variables

```
Wife_ education
Primary       1.0
Secondary     2.0
Tertiary      1.0
Uneducated    4.0
```

As we can see women who are uneducated is more likely to having more children

```
Husband_education
Primary       2.0
Secondary     1.0
Tertiary      2.0
Uneducated    5.0
```

As we can see women with uneducated husband are more likely to having more children

```
Husband_Occupation
1    2.0
2    1.0
3    2.0
4    1.0
```

As we can see husband with occupation 1 and 3 more likely to have 2 children and husband with occupation 2 and 4 more likely to have 1 child

```
Wife_Working
No           2.0
Yes    [1.0, 2.0]
```

As we can see women not working is more likely to have 2 and women working are more likely to have 1 or 2 children

```
Wife_religion
Non-Scientology           3.0
Scientology          [1.0, 2.0]
```

As we can see women with religion non-scientology is more likely to have 3 children and women with religion scientology is more likely to have 1 or 2 children

```
Standard_of_living_index
High         2.0
Low          1.0
Very High    3.0
Very Low     1.0
```

As we can Women with Very high standard of living Index are more likely to have more children than other standards of living

```
Media_exposure
Exposed        1.0
Not-Exposed    3.0
```

As we can Women with no media exposure are more likely to have more children than women with exposure

Pair plot of the variables

Multivariate Analysis

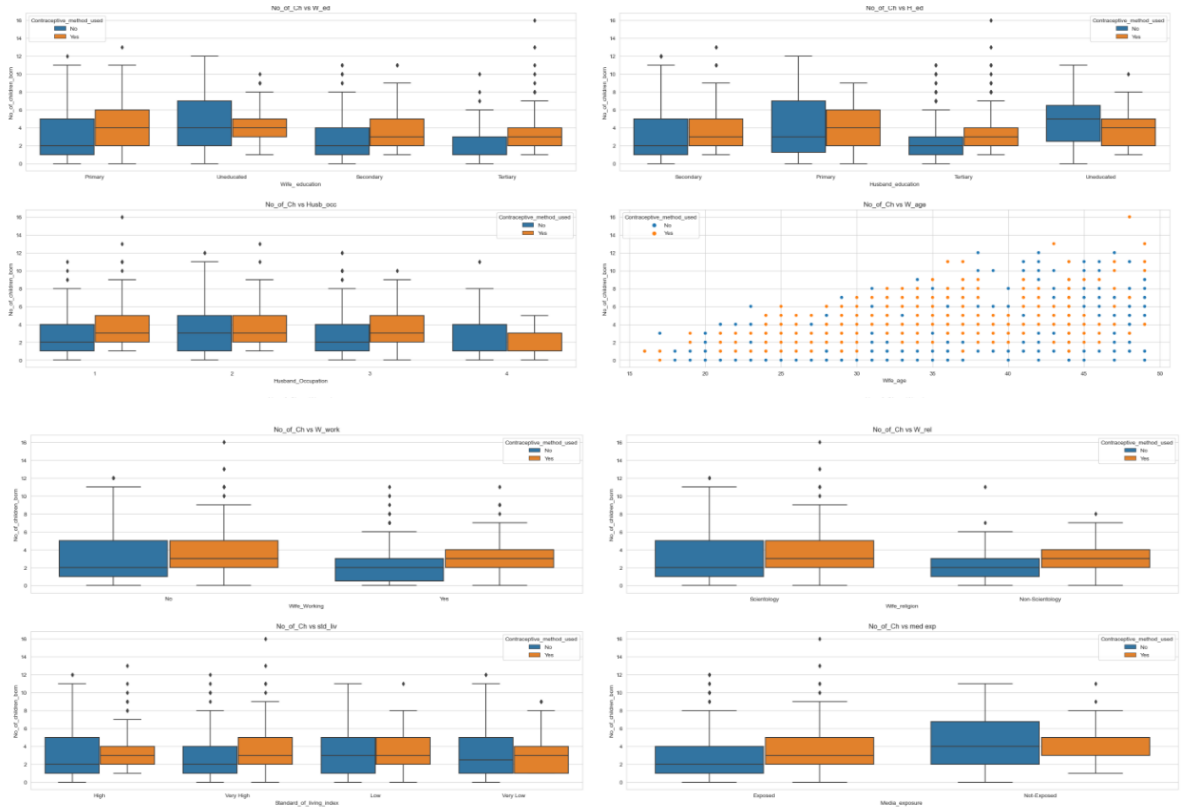


Figure 21: Multivariate Analysis of independent variables and dependent variables

As we can infer number of children born for different levels of education is different .The uneducated women has same mean number of children irrespective of whether they used contraceptive or not

As we can infer number of children born for women who do not use contraceptive with uneducated husband has more children

As we can infer women with religion scientology or non-scientology are equally likely to have 3 children if they use contraceptive

As we can infer women not working and using contraceptive are likely to have more children

As we can infer women with very high standard of living index and using contraceptive are likely to have more children

As we can infer women media exposure and no media exposure and using contraceptive are likely to have more children

As we saw earlier that there are null values in the variables wife_age and no_of_children_born

We can impute the null value with the median values and after imputing the null values we see the there are no Null values

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

Table 30:Null values count after imputing

.

2.2 Do not scale the data. Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Logistic Regression and LDA (linear discriminant analysis) and CART.

Before we procced with the model lets treat the outliers. Outliers produce extremely large residuals. These outliers can unduly influence the results of the analysis and lead to incorrect inferences .There are outliers in the data in variable no_of_children_born, we can treat them before feeding data into the model.

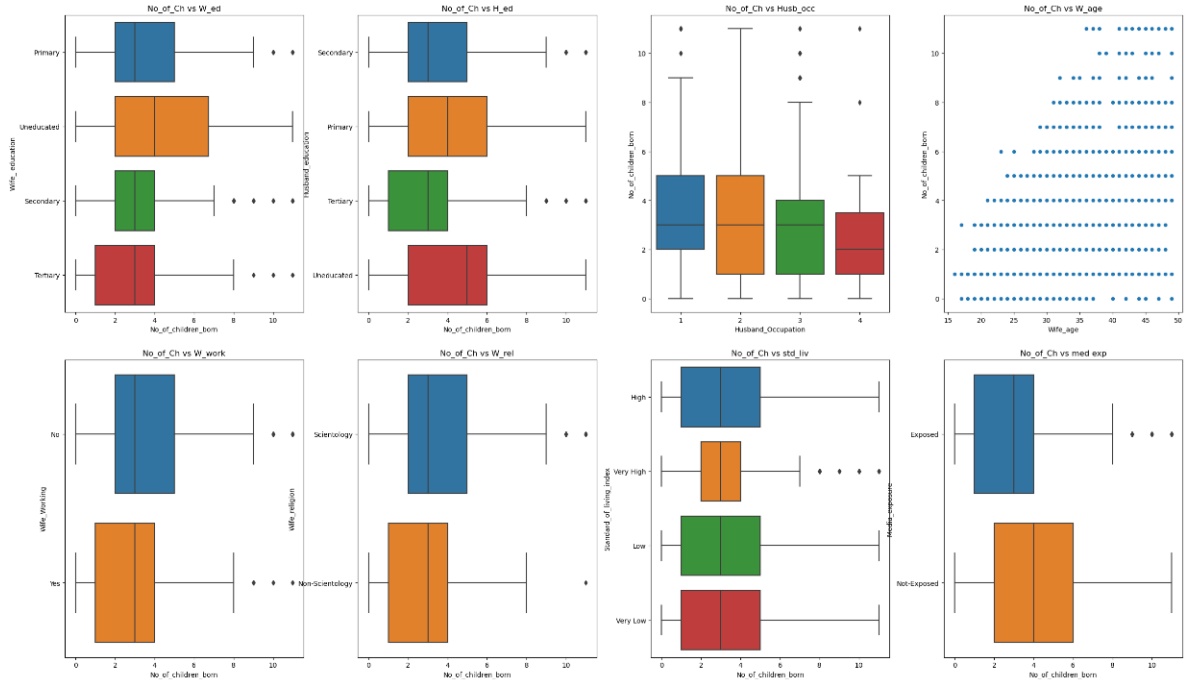


Figure 22:Boxplot of independent variables after outlier treatment

Encoding

Variables 'Wife_ education', 'Husband_education', 'Wife_religion','Wife_Working','Husband_Occupation' , 'Standard_of_living_index', 'Media_exposure ', 'Contraceptive_method_used' are categorical variables and needs to encoded to numerical variables in order to be fitted into the model

We have encoded the categorical variable as follows
Wife's education (categorical) -1=uneducated, 2=Primary, 3=Secondary, 4=Tertiary
Husband's education (categorical) -1=uneducated, 2=Primary, 3=Secondary, 4=Tertiary
Wife's religion (binary) -Non-Scientology-0, Scientology-1
Wife's now working? (binary) -Yes-1, No-0
Husband's occupation (categorical) -1, 2, 3, 4(random)
Standard-of-living index (categorical) -1=Very low, 2-Very High, 3-low, 4=high
Media exposure (binary)- No exposure-0, Exposed-1
Contraceptive method used -(class attribute) No-0,Yes-1
After encoding the categories as above we can see from the table below the categorical variable are now converted to int type variable

```
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Wife_age                  1393 non-null   float64
 1   Wife_ education           1393 non-null   int32
 2   Husband_education         1393 non-null   int32
 3   No_of_children_born       1393 non-null   float64
 4   Wife_religion             1393 non-null   int8
 5   Wife_Working              1393 non-null   int8
 6   Husband_Occupation        1393 non-null   int64
 7   Standard_of_living_index  1393 non-null   int32
 8   Media_exposure            1393 non-null   int32
 9   Contraceptive_method_used 1393 non-null   int8
dtypes: float64(2), int32(4), int64(1), int8(3)
memory usage: 69.4 KB
```

Table 31: Data Variables after encoding

Wife_ education is encoded as below

```
Wife_ education
4                515
3                398
2                330
1                150
```

Husband_education is encoded as below

```
Husband_education
4                  827
3                  347
2                  175
1                   44
```

Husband_occupation is encoded as below

```
Husband_Occupation
3                      570
2                      415
1                      381
4                       27
```

Wife_religion is encoded as below

```
Wife_religion
1               1186
0                207
```

Wife_Working is encoded as below

```
Wife_Working
0               1043
1                350
```

Standard_of_living_index is encoded as below

```
Standard_of_living_index
2                          618
4                          419
3                          227
1                          129
```

Media_exposure is encoded as below

```
Media_exposure
1                  1284
0                   109
```

Contraceptive_method_used is encoded as below

```
Contraceptive_method_used
1                           779
0                           614
```

After the data has been encoded let's see the Dataset again

	0	1	2	3	4	5	6	7	8	9
Wife_age	24.0	45.0	43.0	42.0	36.0	19.0	38.0	21.0	27.0	45.0
Wife_ education	2.0	1.0	2.0	3.0	3.0	4.0	2.0	3.0	2.0	1.0
Husband_education	3.0	3.0	3.0	2.0	3.0	4.0	3.0	3.0	3.0	1.0
No_of_children_born	3.0	10.0	7.0	9.0	8.0	0.0	6.0	1.0	3.0	8.0
Wife_religion	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Wife_Working	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Husband_Occupation	2.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	2.0
Standard_of_living_index	4.0	2.0	2.0	4.0	3.0	4.0	3.0	3.0	2.0	3.0
Media_exposure	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0
Contraceptive_method_used	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 32:data summary after encoding

As we can see all the categorical variables are now numerical

Let's see the pair plot of the variables

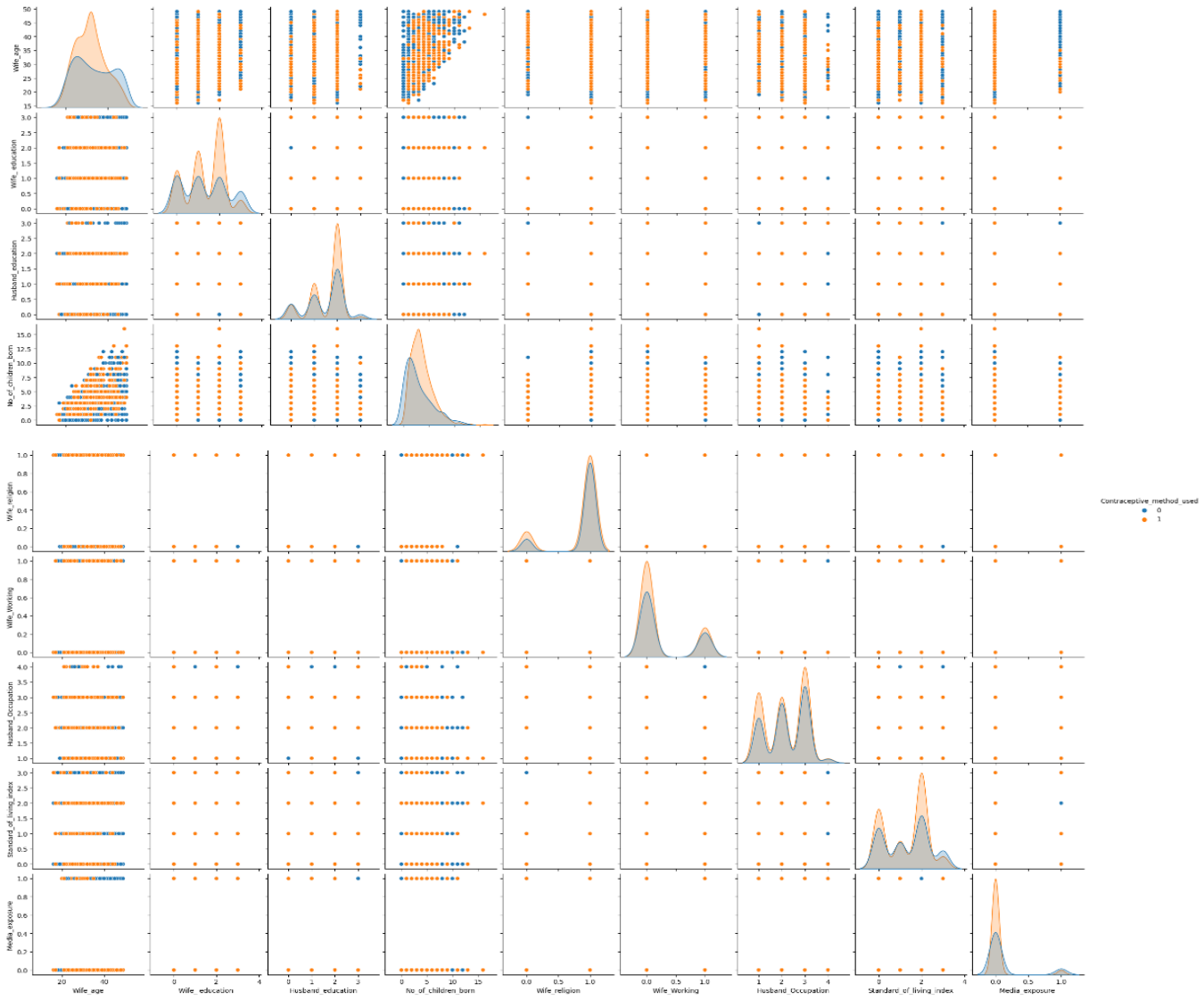


Figure 23:Pairplot of numerical variables

As we can see from the above plot the variables wife age and no of children born show the separation between 2 classes (Contraceptive used or not used) to some extent , whereas other variables the plots are overlapping, hence they are not very good variables to separate the two classes

Before passing the data to a model, we have to split the data into training and testing.

The train-test split is used to estimate the performance of machine learning algorithms that are applicable for prediction-based Algorithms/Applications. The Test set is split into 30 % of actual data and the training set is split into 70% of the actual data. We need to split a dataset into train and test sets to evaluate how well our machine learning model performs. The train set is used to fit the model, and the statistics of the train set are known. The second set is called the test data set, this set is solely used for predictions.

An important consideration is that rows are assigned to the train and test sets randomly. This is done to ensure that datasets are a representative sample of the original dataset, which in turn, should be a representative sample of observations from the problem domain. When comparing machine learning algorithms, it is desirable that they are fit and evaluated on the same subsets of the dataset. This can be achieved by fixing the seed for the pseudo-random number generator used when splitting the dataset. This can be achieved by setting the "random_state" to an integer value in this case we chose 1

Split of the dataset into train and test sets should be in a way that preserves the same proportions of examples in each class as observed in the original dataset. We can achieve this by setting the "stratify" argument to the y component of the original dataset. Setting "stratify" as "Y" in train_test_split() function ensures that both the train and test sets have the proportion of examples in each class that is present in the provided "y" array

Logistic Regression

Now we have prepared our dataset by removing duplicates, imputing null values and treating outliers now we can train the dataset using the training set. For fitting the model to the training set, we will use the **Logistic Regression** class of the **sklearn** library. We will create a classifier object and use it to fit the model to the logistic regression.

There are various parameters passed in the logistic regression function, we can choose the best parameters to get a regularized model. There are various algorithms to use in the optimization problem ***'lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'saga'***. Default is 'lbfgs'.

Penalty has options 'l1', 'l2', 'elasticnet', None}, default='l2'

The parameters passed in this logistic regression function would max_iter=10000, n_jobs=2, penalty='l2', solver='newton-cg'

Predicting the Test Result

Our model is trained on the training set, so we can now predict the result by using test set data.

The actual vs predicted is as below

	Actual	Predictions
0	0	1
1	0	0
2	0	1
3	1	1
4	0	1
...	...	...
413	1	1
414	0	1
415	0	1
416	1	0
417	1	1

Table 33:Actuals Vs Predicted Values in training dataset

Test Accuracy of the result

Now we will create the confusion matrix here to check the accuracy of the classification. It takes two parameters, mainly the actual values and the predicted .

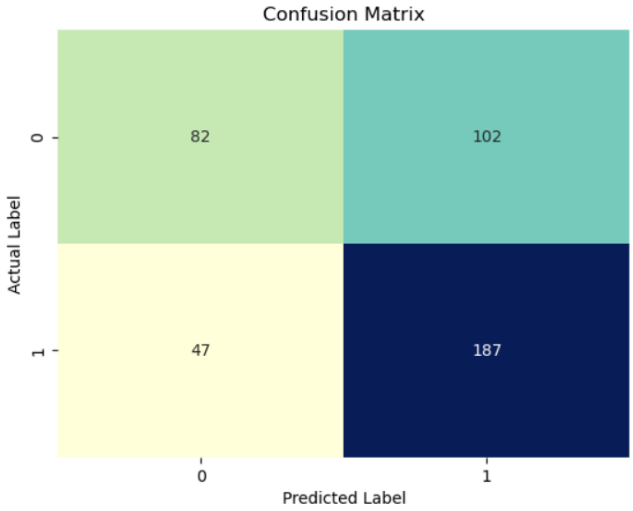


Table 34:Confusion matrix of the logistic regression model on Test data

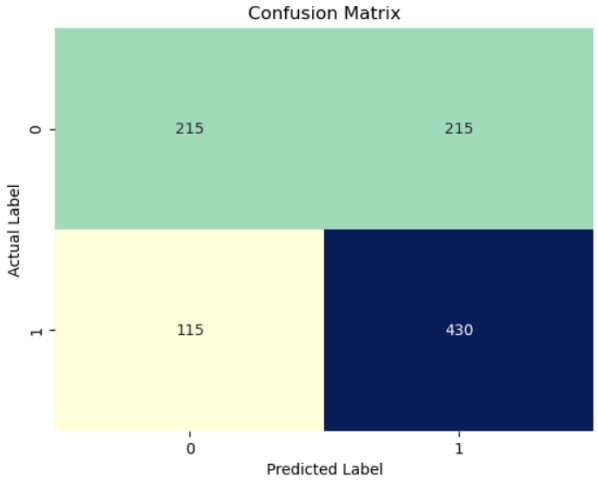


Table 35 Confusion matrix of the logistic regression model on trained data

Another performance evaluation metric in machine learning classification report . It is used to show the precision, recall, F1 Score, and support of your trained classification model.

	precision	recall	f1-score	support
0	0.64	0.45	0.52	184
1	0.65	0.80	0.72	234
accuracy			0.64	418
macro avg	0.64	0.62	0.62	418
weighted avg	0.64	0.64	0.63	418

Table 36:Classification report of logistic regression model on Test Data

	precision	recall	f1-score	support
0	0.65	0.50	0.57	430
1	0.67	0.79	0.72	545
accuracy			0.66	975
macro avg	0.66	0.64	0.64	975
weighted avg	0.66	0.66	0.65	975

Table 37 Classification report of logistic regression model on Train Data

The above figure shows the precision, recall, accuracy of the testing data .

As we can see accuracy of training data is 0.66 meaning the model is predicting the labels with an accuracy of 66%. But what we can also see is that recall score of labels 1 is 0.79 and of label 0 is .50 indicting that there are more 1's (cases where Contraeceptive_used as Yes)in the data meaning this model is good at predicting 1's than 0's. The precision of label 0 is 0.65 and of 1's is 0.67.

As we can see accuracy of testing data is 0.64 meaning the model is predicting the labels with an accuracy of 64%. But what we can also see is that recall score of labels 1 is 0.80 and of label 0 is .45 indicting that there are more 1's (cases where Contraeceptive_used as Yes)in the data meaning this model is good at predicting 1's than 0's. The precision of label 0 is 0.64 and of 1's is 0.65.

Classification report on training data:

The confusion matrix of train data shows that the model predicted 215 True negative values, 215 false negative values, 115 false positive values, and 430 True positive values.

Classification report on testing data:

The confusion matrix of test data shows that the model predicted 82 True negative values, 102 false negative values, 47 false positive values, and 187 True positive values.

We can use GridSearchCV for performing hyperparameter tuning in order to determine the optimal values for a given model. We pass predefined values for hyperparameters to the GridSearchCV function. We do this by defining a dictionary in which we mention a particular hyperparameter along with the values it can take. GridSearchCV tries all the combinations of the values passed in the dictionary and evaluates the model for each combination using the Cross-Validation method. Hence after using this function, we get accuracy/loss for every combination of hyperparameters and we can choose the one with the best performance.

We find from GridSearchCV that 'penalty'as 'none' , 'solver 'as 'lbfgs' and 'tol': 0.0001 would give the be st model

After applying the best model obtained from GridSearchCV we obtain the following classification report and confusion matrix.

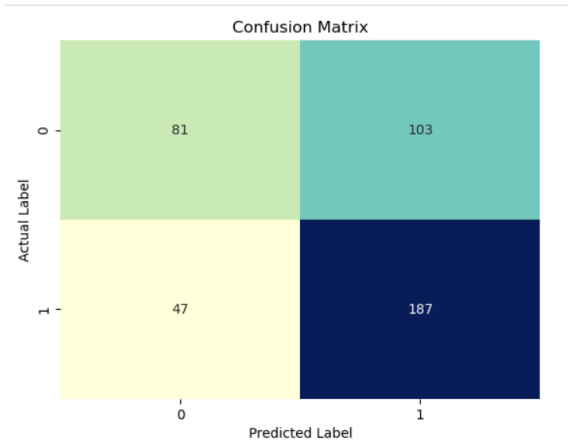


Figure 24: Confusion matrix of the logistic regression improved model

	precision	recall	f1-score	support
0	0.63	0.44	0.52	184
1	0.64	0.80	0.71	234
accuracy			0.64	418
macro avg	0.64	0.62	0.62	418
weighted avg	0.64	0.64	0.63	418

Table 38: Classification report of logistic regression improved model

After fitting the data into best fit model, we see that , there is not much improvement in the accuracy, precision or recall scores from previous model

LDA

LDA helps reduce dimensions of the feature set but at same time retain the information that discriminates output classes. LDA determines a decision boundary around each cluster of a class projects the data points to new dimensions in a way that the clusters are as separate from each other as possible and the individual elements within a cluster are as close to the centroid of the cluster as possible. These new dimensions form the linear discriminants of the feature set.

First step is to divide dataset into features and corresponding labels and then divide the resultant dataset into training and test sets. The Test set is split into 30 % of actual data and the training set is split into 70% of the actual data.

The LinearDiscriminantAnalysis class of the sklearn library can be used to Perform LDA in Python. We execute the fit methods to actually retrieve the linear discriminants.

Once we've fit the model using our data, we can evaluate how well the model performed . Below is the classification report of LDA model. We can see the precision, recall, F1 Score, and support of the trained classification model.

	precision	recall	f1-score	support
0	0.65	0.44	0.53	184
1	0.65	0.82	0.72	234
accuracy			0.65	418
macro avg	0.65	0.63	0.62	418
weighted avg	0.65	0.65	0.64	418

Table 39: Classification report of LDA model of Testing Data

	precision	recall	f1-score	support
0	0.66	0.49	0.56	430
1	0.67	0.80	0.73	545
accuracy			0.66	975
macro avg	0.66	0.64	0.64	975
weighted avg	0.66	0.66	0.65	975

Table 40: Classification report of LDA model of Training Data

As we can see accuracy of testing data is 0.65 meaning the model is predicting the labels with an accuracy of 65%. But what we can also see is that recall score of labels 1 is 0.82 and of label 0 is .44 indicting that there are more 1's in the data meaning this model is good at predicting 1's than 0's. The precision of label 0 is 0.65 and of 1's is 0.65.

As we can see accuracy of training data is 0.66 meaning the model is predicting the labels with an accuracy of 66%. But what we can also see is that recall score of labels 1 is 0.80 and of label 0

is .49 indicting that there are more 1's in the data meaning this model is good at predicting 1's than 0's. The precision of label 0 is 0.66 and of 1's is 0.67.

Evaluation of LDA model-Performance measurement of classification algorithms is judged by confusion matrix which comprise the classification count values of actual and predicted labels. The confusion matrix for binary classification is given by.

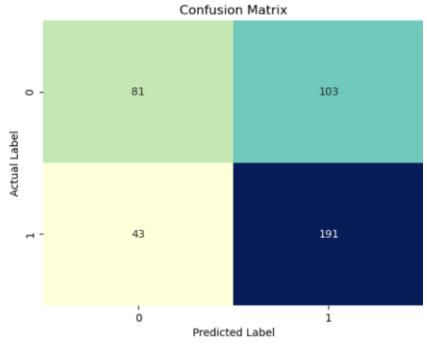


Table 41: Confusion matrix of LDA model testing data

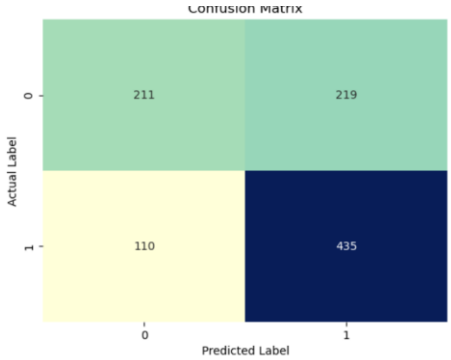


Table 42 Confusion matrix of LDA model training data

Classification report on training data:

The confusion matrix of train data shows that the model predicted 211 True negative values, 219 false negative values, 110 false positive values, and 435 True positive values.

Classification report on testing data:

The confusion matrix of test data shows that the model predicted 81 True negative values, 103 false negative values, 43 false positive values, and 191 True positive values.

Probability function and Linear Discriminat function

LDA can be derived from simple probabilistic models which model the class conditional distribution of the data for each class . Predictions can then be obtained by using Bayes' rule, for each training sample.

Using Probalistic models 948 rows are classified as 1(Contraceptive use as Yes) and 445 rows are classifie d as 0(Contraceptive use as No)

Other approach is to use Linear Discriminant function(LDF) to separate the classes using LDF we get th e model intercept as -0.71555201 and model coefficient are as below
array([[-0.06732334, 0.54684355, 0.11521417, 0.31966728, -0.52130767,
-0.11443342, 0.10297826, -0.03507941, 0.53364837]])
Using the above intercept and coefficient we can build a rule using linear discriminant function Classifica tion Rule :

if LDF>=0 then Classify as 1
else if LDF <0 then Classify as 0 On applying the rule

948 rows are classified as Contraceptive use as Yes and 445 rows are classified as Contraceptive use as No

As we can see both LDF and probability function has classified the same numbers of 0's and 1's

CART

CART is a predictive algorithm used in Machine learning and it explains how the target variable's values can be predicted based on other matters. It is a decision tree where each fork is split into a predictor variable and each node has a prediction for the target variable at the end.
In the decision tree, nodes are split into sub-nodes on the basis of a threshold value of an attribute. The root node is taken as the training set and is split into two by considering the best attribute and threshold value. Further, the subsets are also split using the same logic. This continues till the last pure sub-set is found in the tree or the maximum number of leaves possible in that growing tree.

- The best split point of each input is obtained.
- Based on the best split points of each input in Step 1, the new "best" split point is identified.
- Split the chosen input according to the "best" split point.
- Continue splitting until a stopping rule is satisfied or no further desirable splitting is available.

CART algorithm uses Gini Impurity to split the dataset into a decision tree .It does that by searching for the best homogeneity for the sub nodes, with the help of the Gini index criterion.

First step is to divide dataset into features and corresponding labels and then divide the resultant dataset into training and test sets. The Test set is split into 30 % of actual data and the training set is split into 70% of the actual data.

The DecisionTreeClassifier class of the sklearn,tree library can be used to perform CART model in Python. DecisionTreeClassifier is a class capable of performing multi-class classification on a dataset.

DecisionTreeClassifier takes as input two arrays: an array X holding the training samples, and an array Y of integer values holding the class labels for the training samples. After being fitted, the model can then be used to predict the class of samples:

Once we've fit the model using our data, we can evaluate how well the model performed . Below is the classification report of CART model. We can see the precision, recall, F1 Score, and support of the test model.

	precision	recall	f1-score	support
0	0.51	0.52	0.52	184
1	0.62	0.61	0.61	234
accuracy			0.57	418
macro avg	0.56	0.56	0.56	418
weighted avg	0.57	0.57	0.57	418

Table 43 Classification report of CART model of Testing Data

The above figure shows the precision, recall, accuracy of the testing data .

As we can see accuracy of testing data is 57 %

	precision	recall	f1-score	support
0	0.98	1.00	0.99	430
1	1.00	0.98	0.99	545
accuracy			0.99	975
macro avg	0.99	0.99	0.99	975
weighted avg	0.99	0.99	0.99	975

Table 44: Classification report of CART model of Training Data

The above figure shows the precision, recall, accuracy of the training data .

As we can see accuracy of testing data is 0.99%.

This is a clear case of overfitting , so we use max_depth to control the size of the tree to prevent overfitting. Also, we can use min_samples_split or min_samples_leaf to ensure that multiple samples inform every decision in the tree, by controlling which splits will be considered. While min_samples_split can create arbitrarily small leaves, min_samples_leaf guarantees that each leaf has a minimum size, avoiding low-variance, over-fit leaf nodes in regression problems.

After passing parameters with max_depth as 7 , min_samples_leaf as 50, we get a regularized model with following classification report

	precision	recall	f1-score	support
0	0.68	0.51	0.58	184
1	0.68	0.82	0.74	234
accuracy			0.68	418
macro avg	0.68	0.66	0.66	418
weighted avg	0.68	0.68	0.67	418

Table 45: Classification report of CART improved model of Testing Data

	precision	recall	f1-score	support
0	0.78	0.58	0.67	430
1	0.72	0.87	0.79	545
accuracy			0.74	975
macro avg	0.75	0.73	0.73	975
weighted avg	0.75	0.74	0.74	975

Table 46: Classification report of CART improved model of Training Data

As we can see accuracy of testing data is 0.68 meaning the model is predicting the labels with an accuracy of 68%. But what we can also see is that recall score of labels 1 is 0.82 and of label 0 is .51 indicting that there are more 1's in the data meaning this model is good at predicting 1's than 0's. The precision of label 0 is 0.68 and of 1's is 0.68

As we can see accuracy of training data is 0.74 meaning the model is predicting the labels with an accuracy of 74%. But what we can also see is that recall score of labels 1 is 0.87 and of label 0 is .58 indicting that there are more 1's in the data meaning this model is good at predicting 1's than 0's. The precision of label 0 is 0.78 and of 1's is 0.72

Evaluation of CART model-Performance measurement of classification algorithms are judged by confusion matrix which comprise the classification count values of actual and predicted labels. The confusion matrix for binary classification is given by.

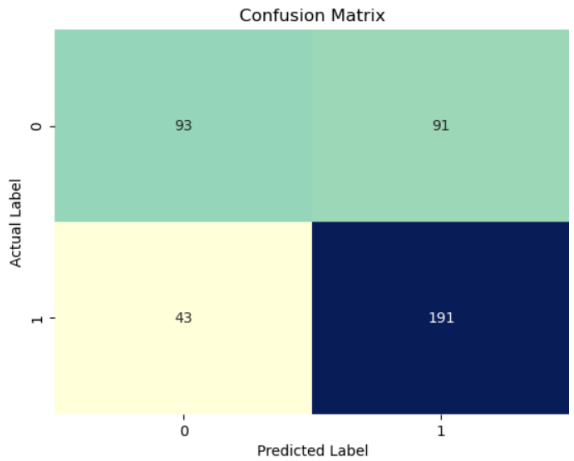


Table 47: :Confusion matrix of CART model testing data

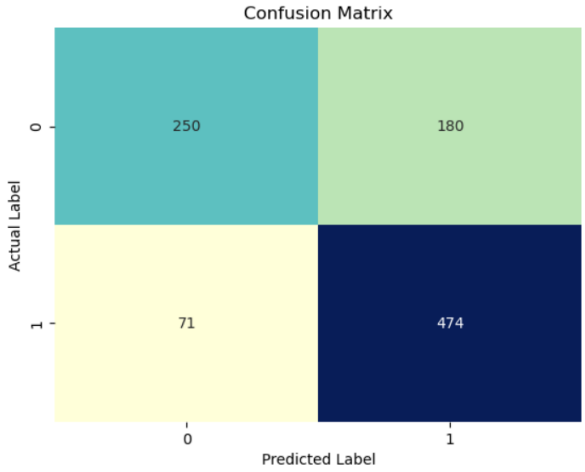


Table 48:Confusion matrix of CART model training data

Classification report on training data:

The confusion matrix of train data shows that the model predicted 250 True negative values, 180 false negative values, 71 false positive values, and 474 True positive values.

Classification report on testing data:

The confusion matrix of test data shows that the model predicted 93 True negative values, 91 false negative values, 43 false positive values, and 191 True positive values.

As an alternative to outputting a specific class, the probability of each class can be predicted, which is the fraction of training samples of the class in a leaf:

```
936  rows classified as 1
457  rows classified as 0
```

Using the dataset, we can construct a tree as follows:

Once trained, you can plot the tree with the plot_tree function and export the tree in Graphviz format. Below is an example graphviz export of the above tree trained on the training dataset; the results are saved in an output file

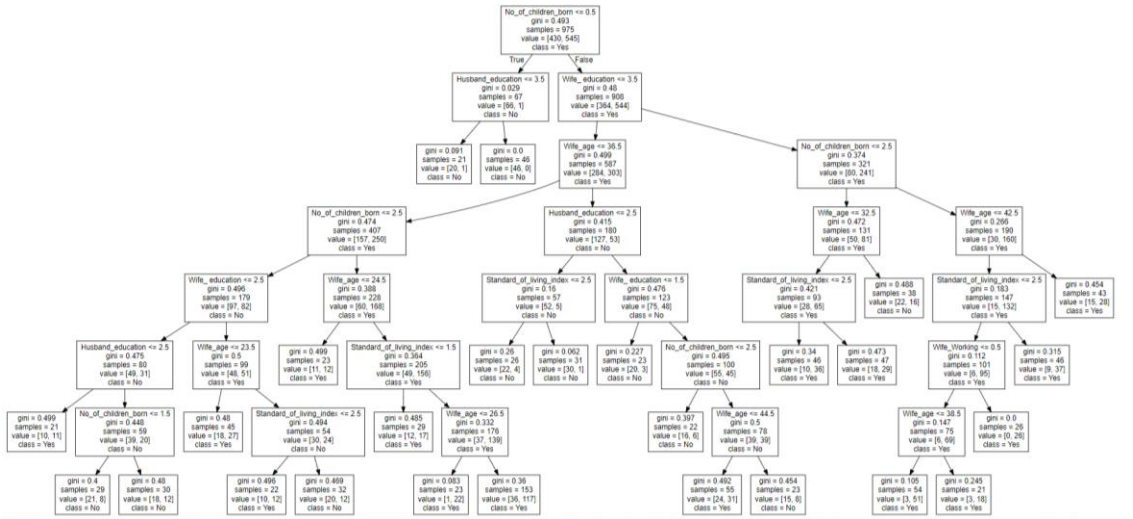
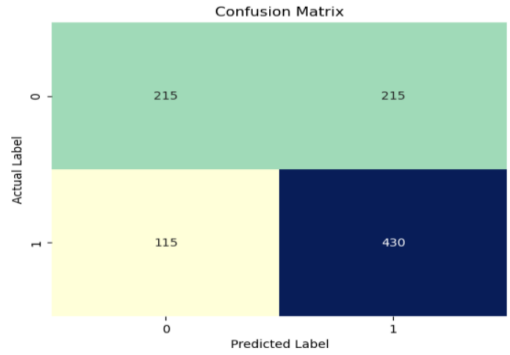


Figure 25:Decision Tree

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

Model Evaluation and Performance To check performance of Predictions of every model built on Train and Test datasets, Accuracy score is calculated. A Confusion Matrix, ROC curve and AUC-ROC score has been devised as well. We have considered the 'Contraceptive method used' i.e., both 0, 1 as the interest classes. Therefore, we will also look at the Accuracy scores of all models.

Comparing Confusion matrix of Logistic Regression Train data and Testing Data



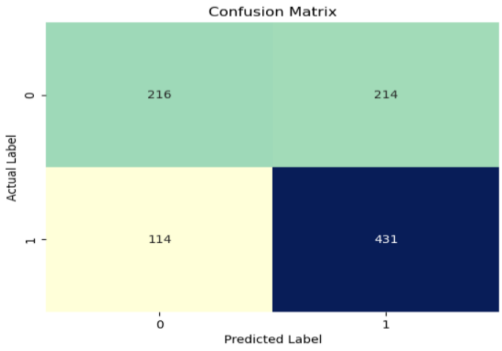
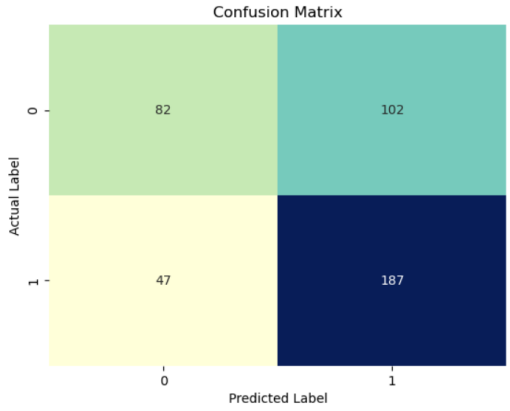


Table 49 Confusion matrix Logistic Regression Train Data Table 50 Confusion matrix Logistic Regression Train Data(Improved)



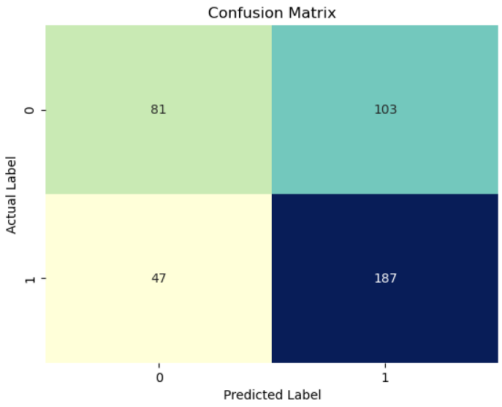


Table 51 Confusion matrix Logistic Regression Test Data *Table 52 Confusion matrix Logistic Regression Test Data(Improved)*

Comparing Confusion matrix of LDA Train data and Testing Data

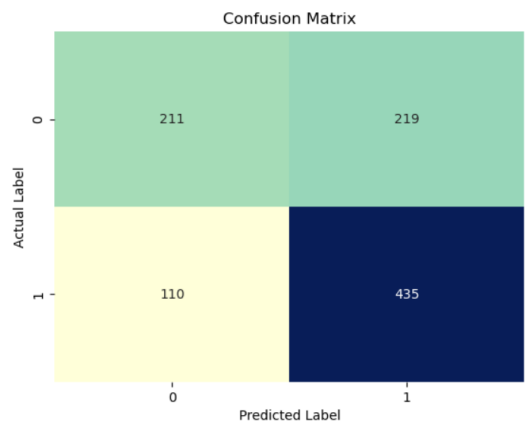


Table 53 Confusion matrix LDA Train data

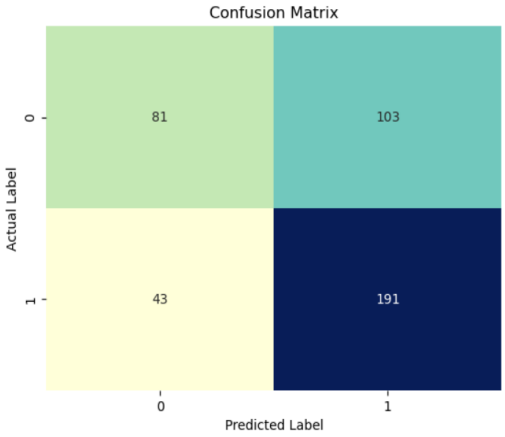


Table 54 Confusion matrix LDA Test data

Comparing Confusion matrix of CART Train data and Testing Data

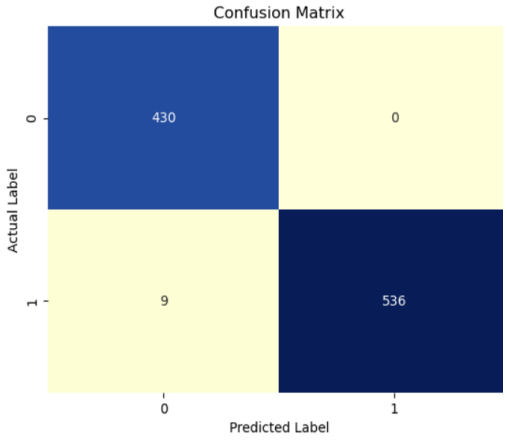


Table 55 Confusion matrix CART Train data

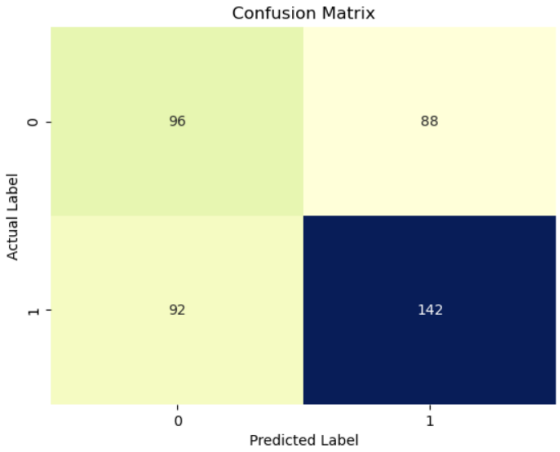


Table 56 Confusion matrix CART test data

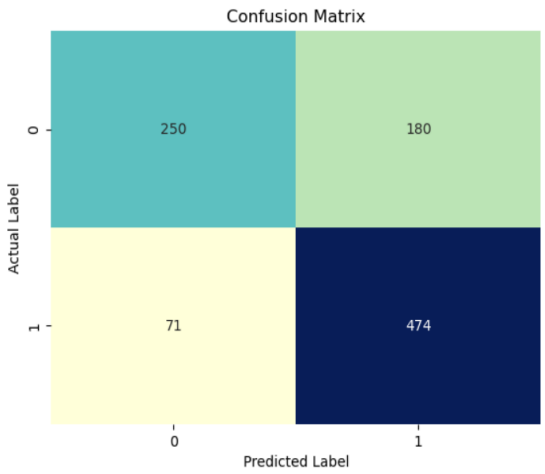


Table 57 Confusion matrix CART Train data(Regularized)

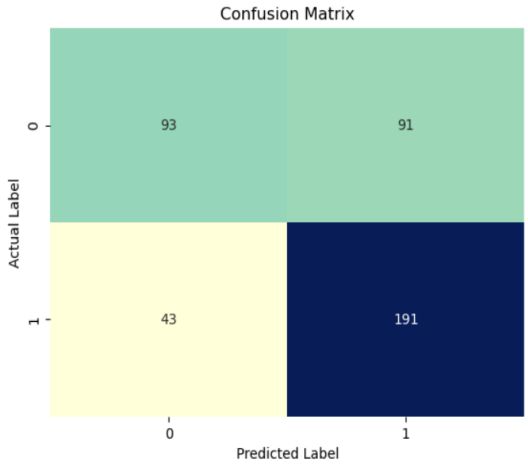


Table 58 Confusion matrix CART Test data(Regularized)

Table 59

AUC -ROC Curve

Logistic Regression AUC -ROC Curve of Training and Testing Dataset

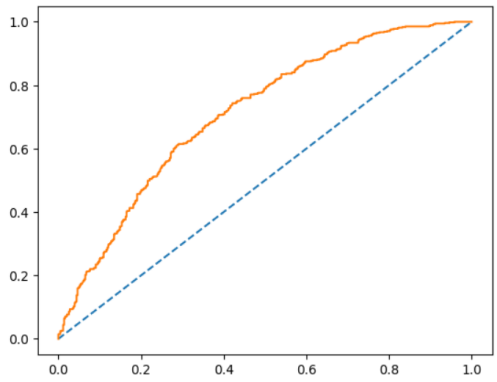


Figure 26: Logistic Regression AUC -ROC Curve of Training Data

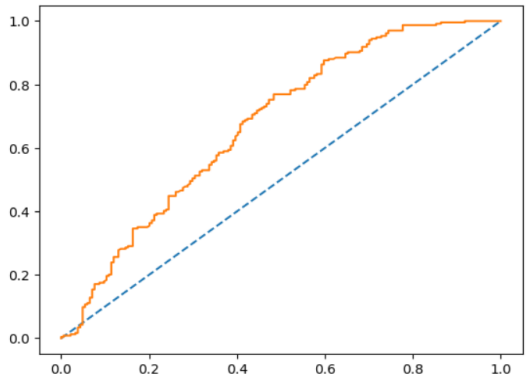


Figure 27 Logistic Regression AUC -ROC Curve of Testing Data

LDA AUC -ROC Curve of Training and Testing Dataset

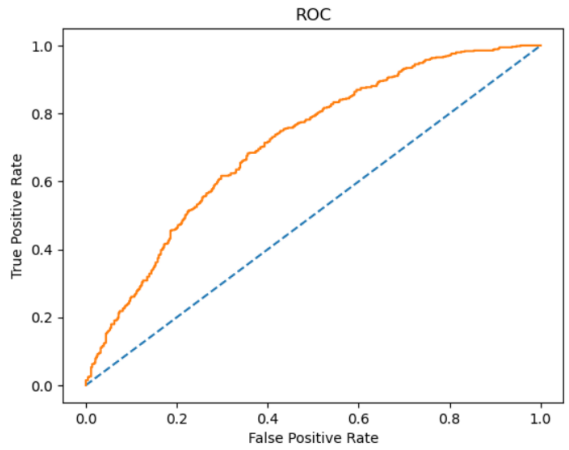


Figure 28 LDA AUC -ROC Curve of Training Dataset

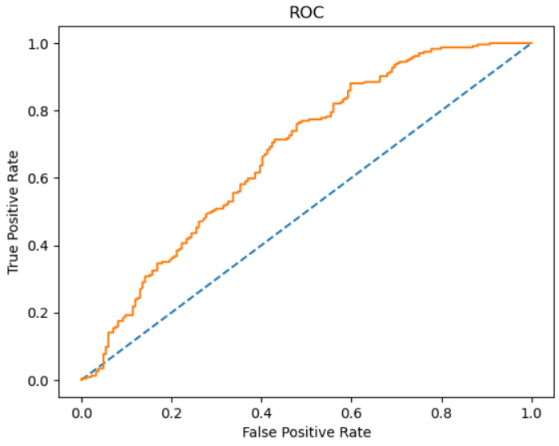


Figure 29 LDA AUC -ROC Curve of Testing Dataset

CART AUC -ROC Curve of Regularized Model

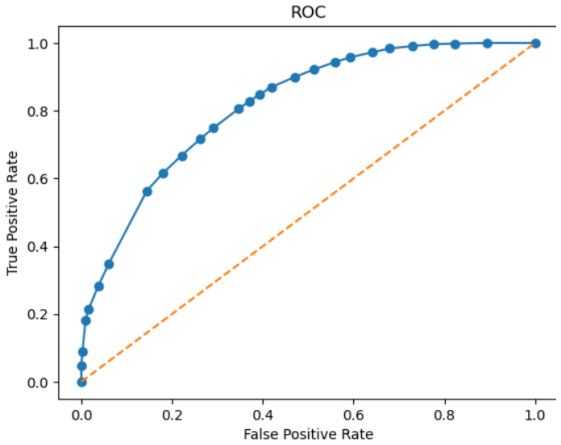


Figure 30 CART AUC -ROC Curve of Training Dataset

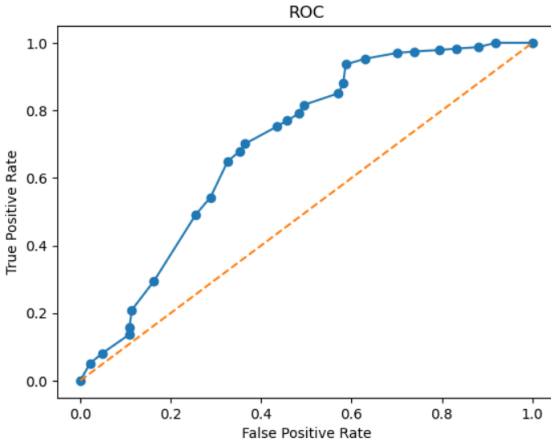


Figure 31 CART AUC -ROC Curve of testing Dataset

Model	Accuracy		Recall-1		Recall-0		Precision-1		Precision-0		AUC score	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Logistic Regression	0.66	0.64	0.79	0.80	0.50	0.45	0.67	0.65	0.65	0.64	0.712	0.674
LDA	0.66	0.65	0.80	0.82	0.49	0.44	0.67	0.65	0.66	0.65	0.712	0.674
Regularized CART	0.74	0.68	0.87	0.82	0.58	0.51	0.72	0.68	0.78	0.68	0.817	0.703

From all the inferences above, we see that mostly all the models have similar performance. The Accuracy score for all the models is above 60% for both test and train data.

Best model selection: We can see that the regularized CART model has performed best compared to the rest of the models. With an Accuracy value of 68% on testing dataset, it is predicting the highest percentage of both our classes of interest. The recall value is also high compared to both LDA and Logistic Regression.

The AUC score of both training and testing data is highest for CART model

Next best is LDA with 65% accuracy on testing data followed by Logistic regression model with 64% accuracy.

Accuracy, AUC, Precision and Recall for test data is almost in line with training data. This shows there is no overfitting or underfitting and the model is a good for classification.

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

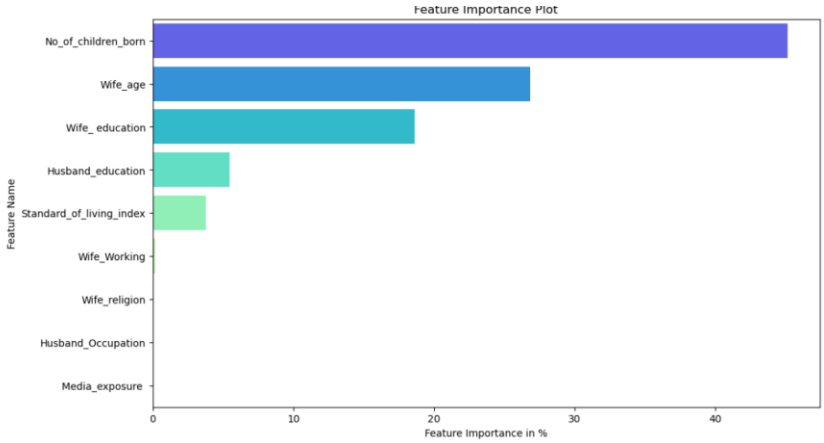


Figure 32 Feature Importance Plot

•The feature importance's of the variables are as follows

No_of_children_born	0.434211
Wife_age	0.258082
Wife_ education	0.182001
Standard_of_living_index	0.063380
Husband_education	0.052572
Husband_Occupation	0.008180
Wife_Working	0.001574
Wife_religion	0.000000
Media_exposure	0.000000

Table 60 feature importance's index

Inferences

From the feature importance's index of the CART model, we can see that Wife's age , No of Children are two most important parameters followed by wife's education and Standard of Living Index in deciding whether the women will use contraceptive methods or not. Wife religion and Media exposure are not important predictors.

The below boxplot also indicates that Wife's age and No of Children show the separation of the classes more clearly than any other plots.

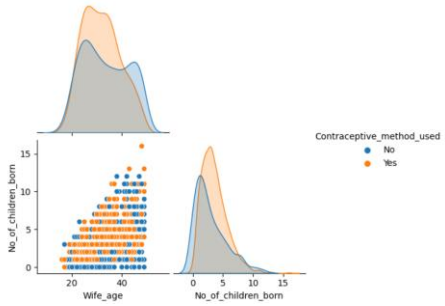


Figure 33 Density plot of Variables

1.The education level of the husband and wife also plays a major role in deciding whether wife will use contraceptive methods or not. Women with tertiary education have highest usage of contraceptive

2.Standard of living ia an important predictor .Women with Standard of living very high have highest usage of contraceptive

3.Working and Non- working women both use contraceptives

4.Media Exposures and Wife Religion is not a great indicator in predicting whether a woman will use contraceptive or not

Recommendations

- Republic of Indonesia Ministry of Health should focus on educating men and women because uneducated men and women seem to have more children than rest

- We saw women with or without media exposure using contraceptive are having more children .May be the contraceptive methods used are not very reliable , so the Republic of Indonesia Ministry of Health should educate the women on right usage of contraceptive .

We see that women as young as 16 and as old as 49 are having children meaning there is a large reproductive age range , Republic of Indonesia Ministry of Health should educate the women to bear

children at the right age window for health reasons which might also decrease the incidence of having more children .

**********End of Report**********