PREDICTIVE MODELING PROJECT REPORT

By Apoorva P

Sl.No	Topic	Pg.No
1	**Problem 1**	**3**
1.1	**Define the problem and perform exploratory Data Analysis**	**4**
1.2	**Data Pre-processing**	**8**
1.3	**Model Building - Linear regression**	**10**
1.4	**Business Insights & Recommendations**	**12**
2	**Problem 2**	**14**
2.1	**Define the problem and perform exploratory Data Analysis**	**14**
2.2	**Data Pre-processing**	**16**
2.3	**Model Building and Compare the Performance of the Models**	**21**
2.4	**Business Insights & Recommendations**	**23**

PROBLEM - 1

CONTEXT

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

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

DATA DESCRIPTION :

System measures used:

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

1.1. DEFINE THE PROBLEM AND PERFORM EXPLORATORY DATA ANALYSIS

```
<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
```

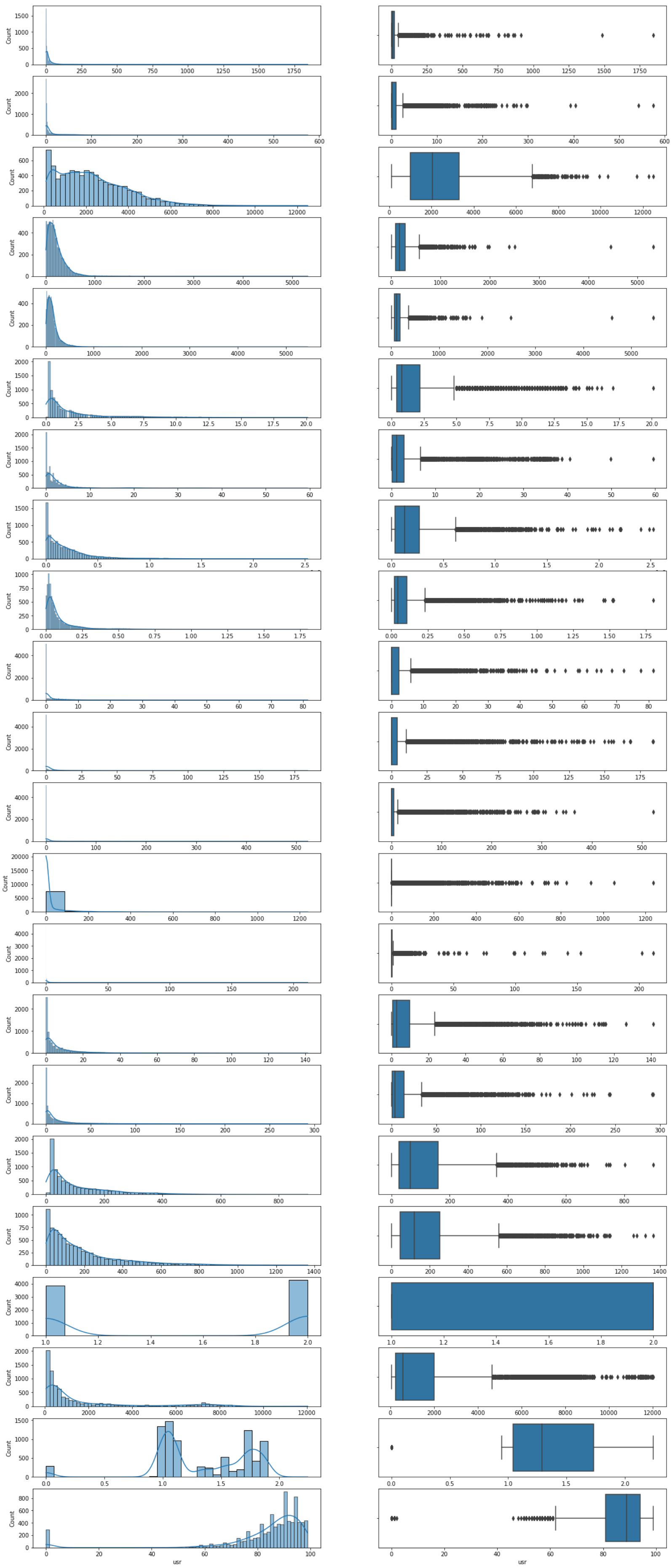
- Data set contains 8192 entries and 22 columns
- It has one column freeman which is object data type
- It has 13 float data types column 8 integer types and 1 objects

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgfree	pgscan	atch	pgin	ppgin	pflt	vflt	freemem	freeswap	usr
count	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8.088000e+03	8.177000e+03	8192.000000	...	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8192.000000	8.192000e+03	8192.000000
mean	19.559692	13.106201	2306.318237	210.479980	150.058228	1.884554	2.791998	1.973857e+05	9.590299e+04	2.285317	...	11.919712	21.526849	1.127505	8.277960	12.388586	109.793799	185.315796	1763.456299	1.328126e+06	83.968872
std	53.353799	29.891726	1633.617322	198.980146	160.478980	2.479493	5.212456	2.398375e+05	1.408417e+05	5.307038	...	32.363520	71.141340	5.708347	13.874978	22.281318	114.419221	191.000603	2482.104511	4.220194e+05	18.401905
min	0.000000	0.000000	109.000000	6.000000	7.000000	0.000000	0.000000	2.780000e+02	1.498000e+03	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.200000	55.000000	2.000000e+00	0.000000
25%	2.000000	0.000000	1012.000000	86.000000	63.000000	0.400000	0.200000	3.409150e+04	2.291600e+04	0.000000	...	0.000000	0.000000	0.000000	0.600000	0.600000	25.000000	45.400000	231.000000	1.042624e+06	81.000000
50%	7.000000	1.000000	2051.500000	166.000000	117.000000	0.800000	1.200000	1.254735e+05	4.661900e+04	0.000000	...	0.000000	0.000000	0.000000	2.800000	3.800000	63.800000	120.400000	579.000000	1.289290e+06	89.000000
75%	20.000000	10.000000	3317.250000	279.000000	185.000000	2.200000	2.800000	2.678288e+05	1.061010e+05	2.400000	...	5.000000	0.000000	0.600000	9.765000	13.800000	159.600000	251.800000	2002.250000	1.730380e+06	94.000000
max	1845.000000	575.000000	12493.000000	5318.000000	5456.000000	20.120000	59.560000	2.526649e+06	1.801623e+06	81.440000	...	523.000000	1237.000000	211.580000	141.200000	292.610000	899.800000	1365.000000	12027.000000	2.243187e+06	99.000000

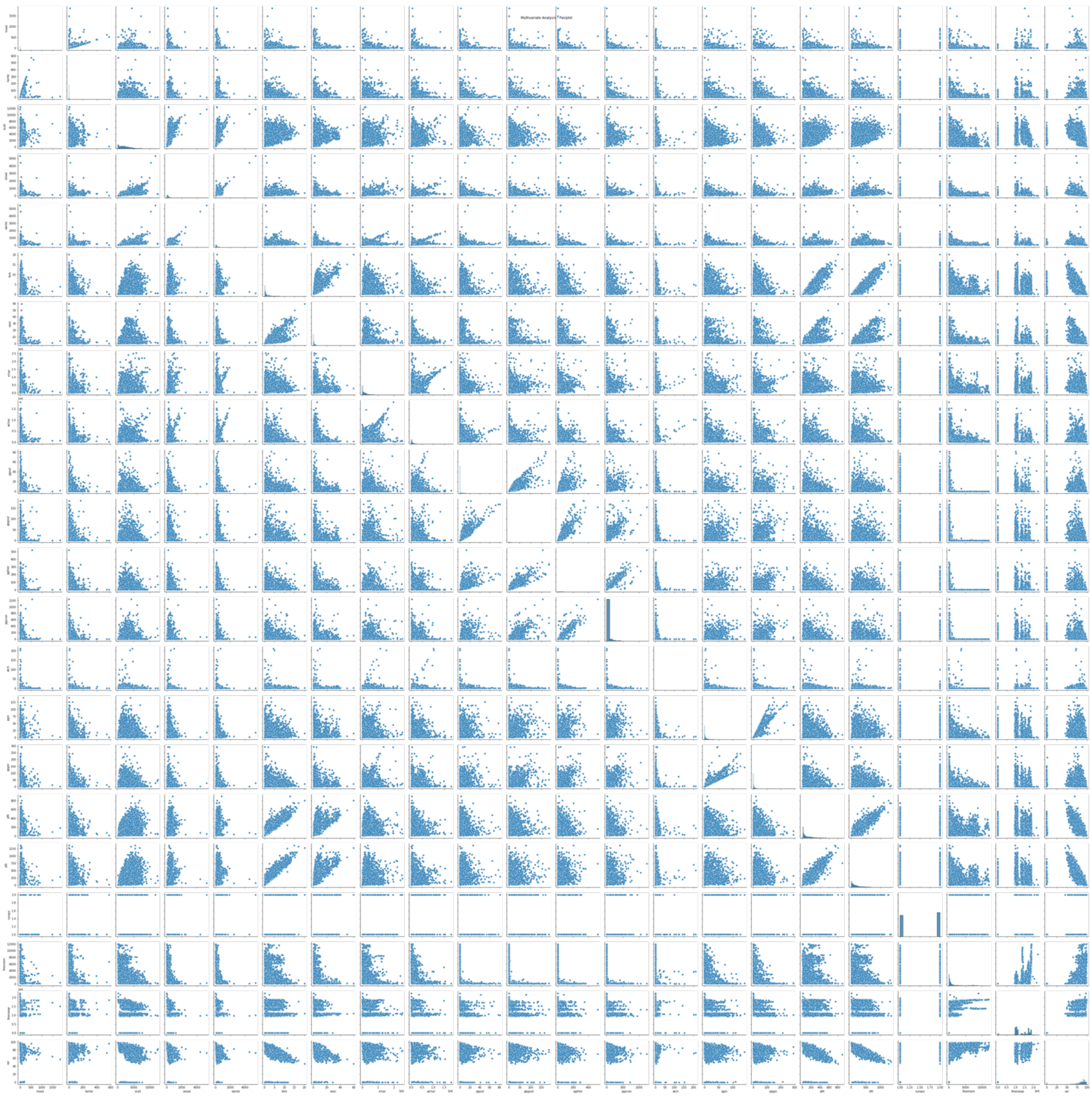
- There is no duplicated column, data set doesn't have duplicate rows as well
- Data set doesn't have null values except rchar and wchar columns.

```
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
```

- The individual histogram and box plot is shown below.


Count
usr
usr

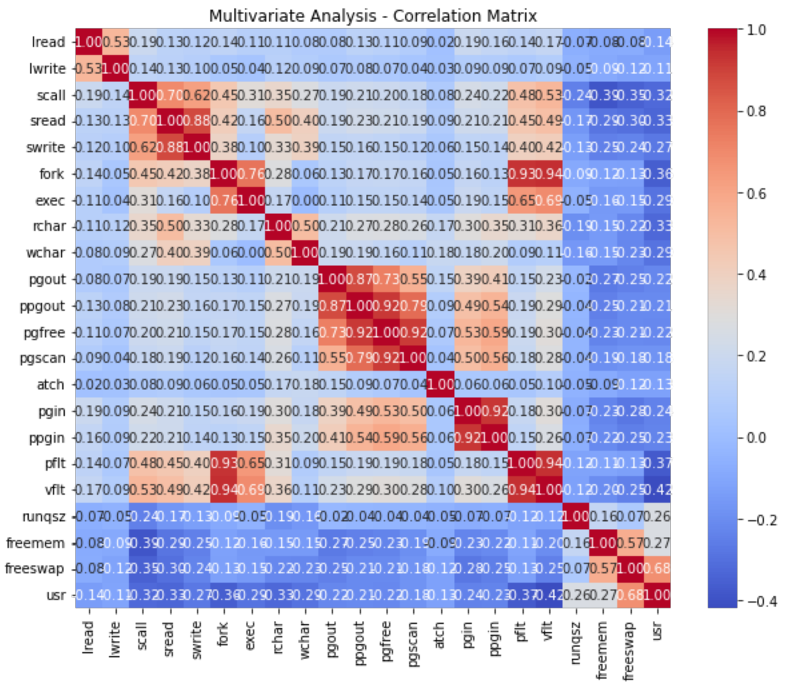
- Pairplot shows the relationship between the variables in the form of scatterplot and the distribution of the variable in the form of histogram



- As the given data set contains huge numbers of columns the pair plot is looking little messy. And as the plot we can see some columns having the positive correlation between them. Some having no correlation and some columns have negative correlation as well.
- Bivariate and multivariate analysis suggests that there is a strong positive correlation between the target variable 'usr' and the independent variables freemem and freeswap.

1.2. Data Pre-processing

- Let us use the 'For loop ' to treat these null values by replace with median values
- After the treatment null values in the data set was clear, no disturbance in data set. Linear regression sensitve to the null values.
- ENCODING
- Linear regression model requires only numerical values, but the data set have one object variable ,we can encode the object as numerical variable
- In data set there is a column 'runqsz' as object data type. Now Converting the columns as numerical by using the Label encoding method and replacing the 'Cpu_bound' as 1 and 'Notcpu_bound' as 2.
- OUTLIERS:
- Every column having the outliers. As the Linear regression is sensitive for outliers, but in my opinion is outliers treatment is not quite good because each and every data is unique with his own entry.
- And Treating the outliers will affect the original value of the data and it may lead to wrong prediction also. So, we will proceed the data with the outliers. Here in every column '0' place an important role as its showing huge difference in the range of the data. If we treat the 0 ,there will be change in data also (like null values) as the real data may have 0, so we will proceed with these.
- Removing the records with 0 values is not mandatory, as it might have no impact on the model building.Even upon dropping variables which lots of zeros there is no change in multicollinearity. Hence no need to drop the variable or change it because changing the variable could change the whole meaning of the variable. So we should keep it as they are.



- The correlation matrix with respect to usr is shown below corr_matrix

```
usr          1.000000
freeswap     0.678526
freemem      0.270308
runqsz       0.261980
lwrite      -0.111213
atch        -0.125074
lread       -0.141394
pgscan      -0.181488
ppgout      -0.212295
pgfree      -0.216278
pgout       -0.221877
ppgin       -0.233682
pgin        -0.241720
swrite      -0.272252
exec        -0.288526
wchar       -0.289036
scall       -0.323188
rchar       -0.326262
sread       -0.332160
fork        -0.363277
pflt        -0.372495
vflt        -0.420685
Name: usr, dtype: float6
```

- Let us create the x and y variable data with respect to 'usr' column as the targetvariable. Now x having every data except the target variable and y having only the targetvariable .
- Using stats model api as SM to intercept the X variable.
- Using sklearn to split the data into x_tain and y_train.Now x_train data having the follows,

	freeswap	freemem	runqsz	lwrite	atch	lread	pgscan	ppgout	pgfree	pgout	...	pgin	swrite	exec	wchar	scall	rchar	sread	fork	pflt	vflt
2949	1306851	956	1	4	0.0	13	0.0	0.0	0.0	0.0	...	20.40	150	2.2	93098.0	5183	237284.0	200	1.6	154.60	312.40
3281	1775705	4033	1	0	0.0	2	0.0	0.0	0.0	0.0	...	0.20	50	1.6	21646.0	431	45640.0	55	1.4	76.05	102.79
7961	13	86	1	2	0.4	20	44.4	2.0	22.6	1.8	...	31.60	195	0.8	259833.0	1169	312017.0	194	0.4	35.00	75.40
4507	1435354	236	1	2	1.6	10	0.0	2.4	2.4	1.8	...	0.80	306	2.2	14080.0	3408	196093.0	397	5.2	249.40	417.40
3124	1044506	614	1	78	0.0	56	0.0	0.0	0.0	0.0	...	2.79	112	2.4	71594.0	1815	103043.0	94	1.6	115.37	168.86
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4931	1004142	302	2	0	0.0	1	0.0	5.0	5.0	2.6	...	1.40	173	0.2	45906.0	1310	58730.0	187	0.2	16.00	47.80
3264	1720094	667	2	19	0.0	14	0.0	0.0	0.0	0.0	...	0.00	141	0.2	12351.0	2283	4429.0	97	0.2	20.40	17.80
1653	1038861	855	1	0	0.0	1	0.0	0.0	0.0	0.0	...	22.20	200	5.0	459666.0	3341	191162.0	132	2.4	65.00	155.40
2607	1834560	5757	1	1	0.0	1	0.0	0.0	0.0	0.0	...	0.40	41	0.2	245749.0	353	263443.0	154	0.2	15.60	16.80

2732	1832568	5663	2	21	0.0	13	0.0	0.0	0.0	0.0	...	2.60	37	0.2	7024.0	1036	46946.0	58	0.2	16.40	17.00

6553 rows × 21 columns

	freeswap	**freemem**	**runqsz**	**lwrite**	**atch**	**lread**	**pgscan**	**ppgout**	**pgfree**	**pgout**	**...**	**pgin**	**swrite**	**exec**	**wchar**	**scall**	**rchar**	**sread**	**fork**	**pflt**	**vflt**
2310	1051227	2728	2	0	0.00	1	0.00	0.00	0.00	0.00	...	0.40	167	2.40	93523.0	2164	249529.0	166	0.60	50.40	61.00
1916	998441	151	1	54	4.98	42	59.96	0.40	15.34	0.40	...	6.97	336	2.79	28167.0	3844	380591.0	490	4.98	304.58	518.92
3585	1008875	139	2	3	0.00	4	7.78	7.98	11.38	3.99	...	0.00	107	0.60	61464.0	1357	44343.0	150	0.40	23.75	45.31
7404	949230	317	1	6	6.40	9	0.00	17.60	17.60	11.00	...	28.40	79	3.40	55307.0	1042	135900.0	109	2.00	155.00	267.60
5278	1042942	337	1	2	0.80	4	0.00	1.80	1.80	1.80	...	7.41	190	1.00	53346.0	2821	650097.0	320	0.80	66.53	113.23
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4731	7	93	1	0	0.00	3	0.00	0.00	0.00	0.00	...	1.60	116	0.60	32924.0	943	34217.0	119	0.80	58.40	130.00
6739	1310246	343	1	10	4.80	21	0.80	12.80	8.40	1.40	...	17.80	266	2.40	94229.0	3862	111286.0	239	2.40	96.80	223.80
4265	1709802	165	1	0	0.00	1	0.00	0.60	0.60	0.40	...	0.80	64	0.20	12370.0	1493	18589.0	47	0.20	17.60	25.60
6072	1703808	337	1	0	0.00	0	0.00	0.40	0.40	0.20	...	10.00	119	0.20	69327.0	1512	114263.0	106	0.20	15.60	22.20
6485	1537662	914	1	5	0.20	24	0.00	0.00	0.00	0.00	...	1.40	137	1.40	50462.0	2198	49285.0	134	1.40	80.00	98.00

1.3. **Model Building - Linear regression**

- As the Train and the test data split up we can process with creating the linear model. Now for creating the OLS model, we can use the .ols from stats model api package. And Fit the data with x_train and y_train.

```
                            OLS Regression Results
==================================================================================
Dep. Variable:                    usr   R-squared:                       0.640
Model:                            OLS   Adj. R-squared:                  0.638
Method:                 Least Squares   F-statistic:                     551.7
Date:                Sun, 17 Dec 2023   Prob (F-statistic):               0.00
Time:                        22:51:28   Log-Likelihood:                -24985.
No. Observations:                6553   AIC:                         5.001e+04
Df Residuals:                    6531   BIC:                         5.016e+04
Df Model:                          21
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const         35.4048      0.836     42.336      0.000      33.765      37.044
freeswap    3.284e-05   4.25e-07     77.317      0.000     3.2e-05    3.37e-05
freemem       -0.0016   7.08e-05    -23.205      0.000      -0.002      -0.002
runqsz         7.9612      0.287     27.784      0.000       7.399       8.523
lwrite         0.0094      0.005      1.806      0.071      -0.001       0.020
atch          -0.0469      0.023     -2.038      0.042      -0.092      -0.002
lread         -0.0220      0.003     -7.536      0.000      -0.028      -0.016
pgscan         0.0117      0.005      2.265      0.024       0.002       0.022
ppgout         0.1142      0.036      3.164      0.002       0.043       0.185
```

```
pgfree        -0.0768      0.017     -4.436      0.000      -0.111      -0.043
pgout         -0.2251      0.063     -3.602      0.000      -0.348      -0.103
ppgin         -0.0268      0.017     -1.588      0.112      -0.060       0.006
pgin           0.0381      0.026      1.454      0.146      -0.013       0.089
swrite        -0.0017      0.002     -0.863      0.388      -0.005       0.002
exec          -0.0216      0.045     -0.477      0.633      -0.111       0.067
wchar       -1.07e-05   1.21e-06     -8.808      0.000   -1.31e-05   -8.32e-06
scall          0.0010      0.000      7.993      0.000       0.001       0.001
rchar      -3.482e-06   7.98e-07     -4.364      0.000   -5.05e-06   -1.92e-06
sread      -3.324e-05      0.002     -0.019      0.985      -0.004       0.003
fork          -2.0201      0.234     -8.642      0.000      -2.478      -1.562
pflt          -0.0387      0.004     -9.769      0.000      -0.047      -0.031
vflt           0.0232      0.003      7.574      0.000       0.017       0.029
==============================================================================
Omnibus:                     1628.441   Durbin-Watson:                   1.962
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             4800.439
Skew:                          -1.287   Prob(JB):                         0.00
Kurtosis:                       6.310   Cond. No.                     8.91e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
ecified.
[2] The condition number is large, 8.91e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

- After splitting the dataset into training set and the test set. Then, we represent the regression_model and fit it on the training set with the fit method. In this step, the model learned the relationships between the training data (X_train, y_train). Now the model is ready to make predictions on the test data (X_test). Hence, we predict on the test data using the predict method.
- Regression metrics for model performance For regression problems, there are two ways to compute the model performance. They are RMSE (Root Mean Square Error) and R-Squared Value.
- These are explained below:- RMSE
- RMSE is the standard deviation of the residuals. So, RMSE gives us the standard deviation of the unexplained variance by the model. It can be calculated by taking square root of Mean Squared Error. RMSE is an absolute measure of fit. It gives us how spread the residuals are, given by the standard deviation of the residuals. The more concentrated the data is around the regression line, the lower the residuals and hence lower the standard deviation of residuals. It results in lower values of RMSE. So, lower values of RMSE indicate better fit of data.
- R2 Score R2 Score is another metric to evaluate performance of a regression model. It is also called coefficient of determination. It gives us an idea of goodness of fit for the linear regression models. It indicates the percentage of variance that is explained by the model. Mathematically, R2 Score = Explained Variation/Total Variation In general, the higher the R2 Score value, the better the model fits the data. Usually, its value ranges from 0 to 1. So, we want its value to be as close to 1. Its value can become negative if our model is wrong.

- The R-square value tells that the model can explain 64.1 %of the variance in the training set.
- Adjusted R-square value is 63.83%
  And
- RMSE on training set: 10.954819738366961
- RMSE on test set: 11.386257747878616

- R-squared is the percentage of the response variable variation that is explained by a linear model. It is always between 0 and 100%. R-squared is a statistical measure of how close the data are to the fitted regression line.
- For the same data set, higher R-squared values represent smaller differences between the observed data and the fitted values. R-squared is always between 0 and 100%:
- 0% represents a model that does not explain any of the variation in the response variable around its mean.
- 100% represents a model that explains all the variation in the response variable around its mean.
- In business decisions, the benchmark for the R-squared score value is 0.7. It means if R squared score value >= 0.7, then the model is good enough to deploy on unseen data whereas if R squared score value < 0.7, then the model is not good enough to deploy. In this case our R squared score value for both train and test is 0.64 and 0.63 respectively. It means that this model explains 63% of the variance in our dependent variable.
- So, the R squared score value confirms that the model is not good enough to deploy because it does not provide good fit to the data.
- The RMSE value for train and test has been found to be 10.81 and 11.59 respectively. It means the standard deviation for our prediction is approx. 11.So, sometimes we expect the predictions to be off by more than 11 and other times we expect less than 11. So, the model is not good fit to the data.

1.4. Business Insights & Recommendations

1. Data consists of both categorical and numerical variables.
2. There are total 8192 rows and 22 columns in the dataset. Out of 22 columns only 1 column is of object data type, 8 columns are of integer type and remaining 13 are of float data type.
3. “usr” is the target variable and all other are predictor variables.
4. Bivariate and multivariate analysis suggests that there is a strong positive correlation between the target variable ‘usr’ and the independent variables freemem and freeswap.
5. Data has null (missing) values in two fields, namely ‘rchar’, ‘wchar’.
6. Missing values got treated by imputing median values.
7. Outliers are present in almost all numeric features.
8. Records with zero values were not removed, as it might not have an impact on model bulding.
9. There are no duplicates records in the given data set.
10. Using the p>|t| result, we can say that the variables like lwrite, sread, swrite, pgscan are statistically insignificant variables as ther p-value is greater than 0.05.
11. Omnibus test checks the normality of the residuals once the model is deployed. Here prob(omnibus) is 0 indicating that there is 0% chance that the residuals are normally distributed. For a model to be robust the residual distribution is also required to be normal ideally apart from checking rsquared and other parameters.
12. This indicates our model is not robust and not fit.

13. Also there are very strong multicollinearity present in the dataset.
14. The final Linear Regression equation is: (35.40) * const + (-0.022) * lread + (0.0094) * lwrite + (0.001) * scall + (-3.324e-05) * sread + (-0.0017) * swrite + (-2.0201) * fork +(-0.0216) * exec + (-3.482e-06) * rchar + (-1.07e-05) * wchar + (-0.2251) * pgout + (-0.0268) * ppgout + (0.1142) * pgfree + (0.0117) * pgscan + (-0.0469) * atch + (0.0381) * pgin + (-0.0268) * ppgin + (-0.0387) * pflt + (0.0232) * vflt + (7.9612) * runqsz + (-0.0016) * freemem + (3.284e-05) * freeswap

PROBLEM - 2

OBJECTIVE

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

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

DATA DESCRIPTION

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

2.1. Define the problem and perform exploratory Data Analysis

- DEFINITION:
  The percentage of women aged 15-49 years, married or in-union, who are currently using, or whose sexual partner is using, at least one method of contraception, regardless of the method used.
- OBJECTIVE:
  The objective of this study is to predict whether they do/don't use a contraceptive method of choice based on their demographic and socio-economic characteristics.
- DATASET:
  The dataset contains 10 features including demographic and socio-economic characteristics of 1473 married women in Indonesia, which is obtained from National Indonesia Contraceptive Prevalence Survey. The dataset has 9 descriptive features and one target variable.

```
<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
```

TARGET FEATURE:
The response variable is "Contraceptive method used" having two classes.
Yes or No
The first 5 rows

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

The last 5 rows

	Wife_age	Wife_education	Husband_education	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Standard_of_living_index	Media_exposure	Contraceptive_method_used
1468	33.0	Tertiary	Tertiary	NaN	Scientology	Yes	2	Very High	Exposed	Yes
1469	33.0	Tertiary	Tertiary	NaN	Scientology	No	1	Very High	Exposed	Yes
1470	39.0	Secondary	Secondary	NaN	Scientology	Yes	1	Very High	Exposed	Yes

1471	33.0	Secondary	Secondary	NaN	Scientology	Yes	2	Low	Exposed	Yes
1472	17.0	Secondary	Secondary	1.0	Scientology	No	2	Very High	Exposed	Yes

There are three different datatypes. dtypes: float64(2), int64(1), object(7)

	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

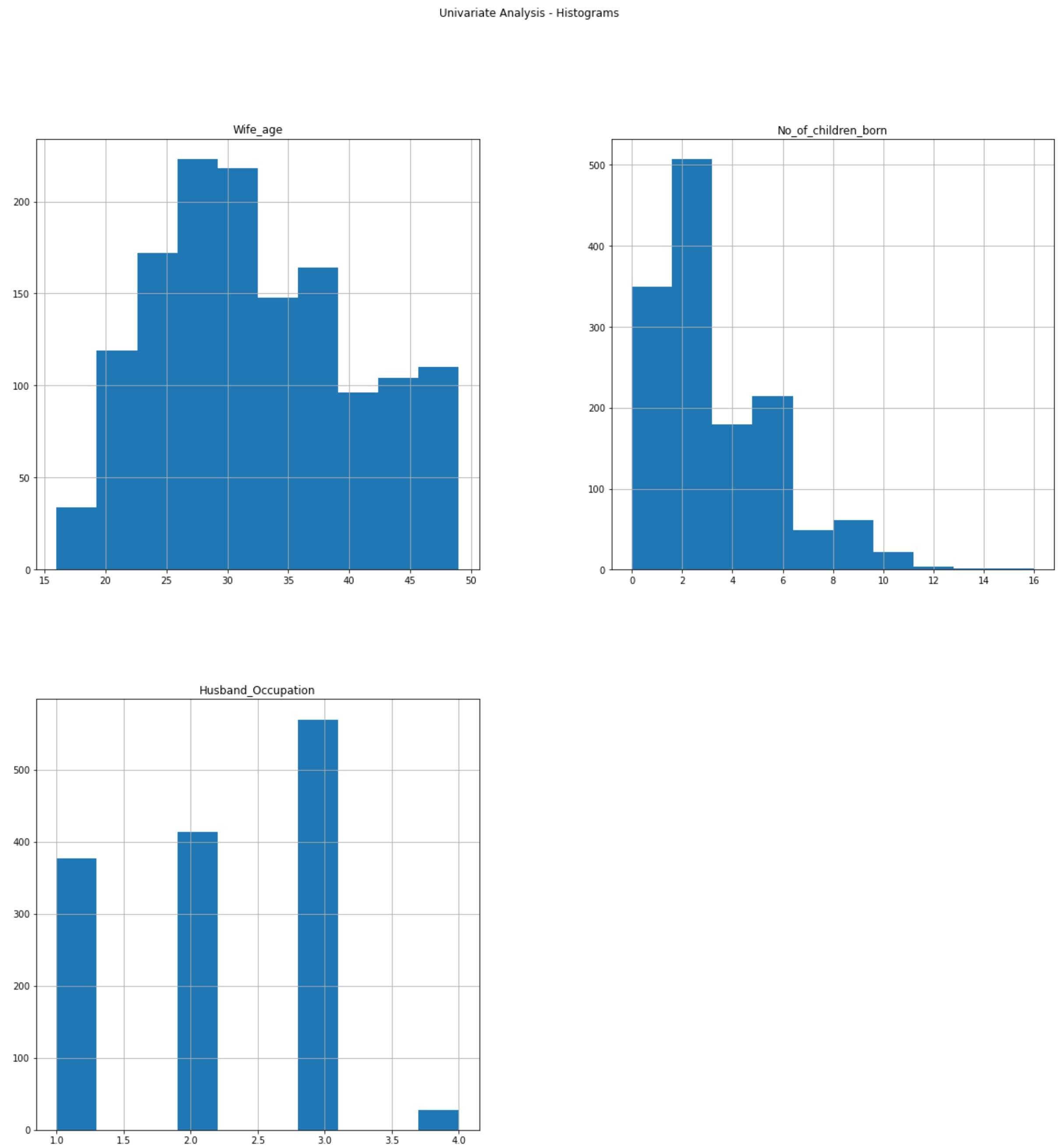
```
Wife_age                     71
Wife_ education               0
Husband_education             0
No_of_children_born          21
Wife_religion                 0
Wife_Working                  0
Husband_Occupation            0
Standard_of_living_index      0
Media_exposure                0
Contraceptive_method_used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

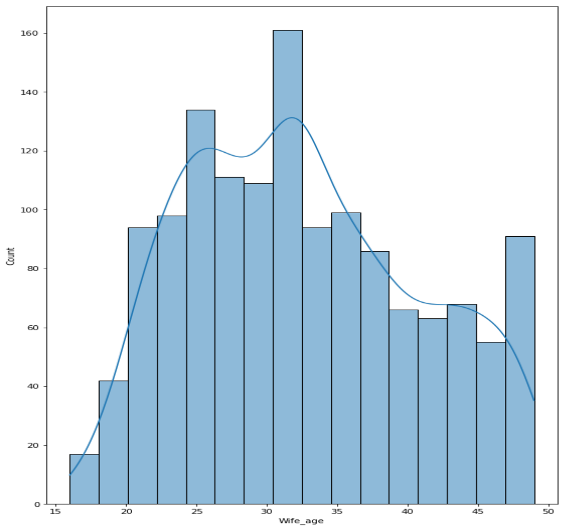
There are missing values present in the “wife’s age” and “No. of children born” variables of the dataset. Approx. 5% of missing values are there in wife’s age field and 1% in the latter. Missing values can confuse the model. Here we solve this missing value problem by replacing the NAN values with the Median.

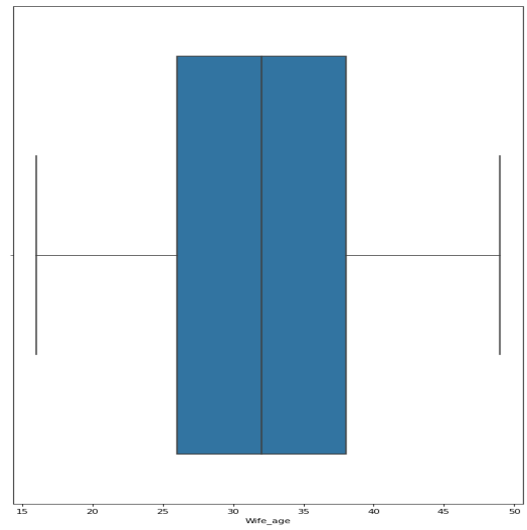
Clearly there are 80 rows in the data containing duplicate values Datasets that contain duplicates may contaminate the training data with the test data or vice versa. An entry appearing more than once receives disproportionate weight during training. Duplicate entries can ruin the split between train, validation, and test sets where identical entries are not all in the same set. This can lead to biased performance estimates that result in disappointing the model in production. There are many possible causes for duplicate entries in databases, such as processing steps that were rerun anywhere in the data pipeline. While the existence of duplicates hurt the learning process greatly, it is relatively easy to fix. This can be done easily with Pandas’ drop_duplicates function.

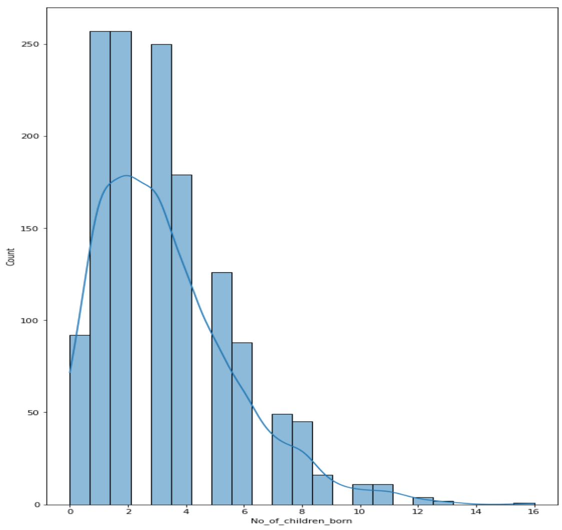
2.2. Data Pre-processing

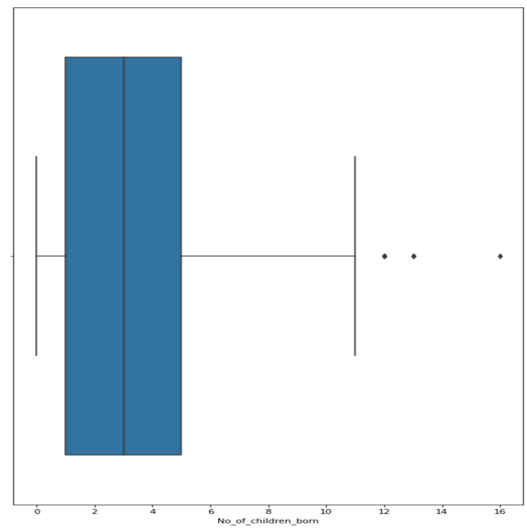
Checking for outliers.
Created a new data frame containing only numeric variables called contra_data_num which includes float and integer datatypes. This contra_data_num will be used further for plotting boxplots and outliers treatment.

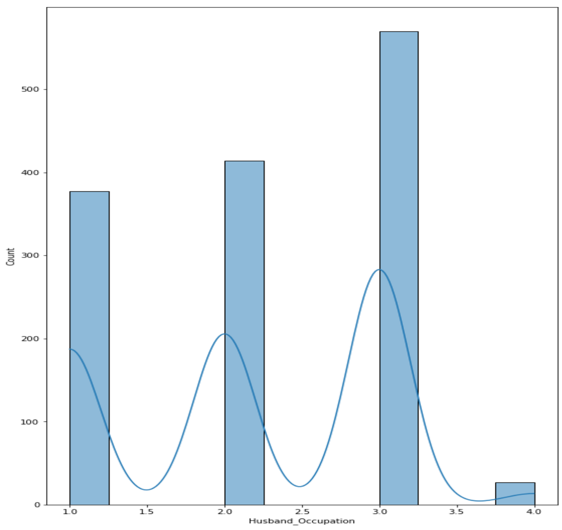


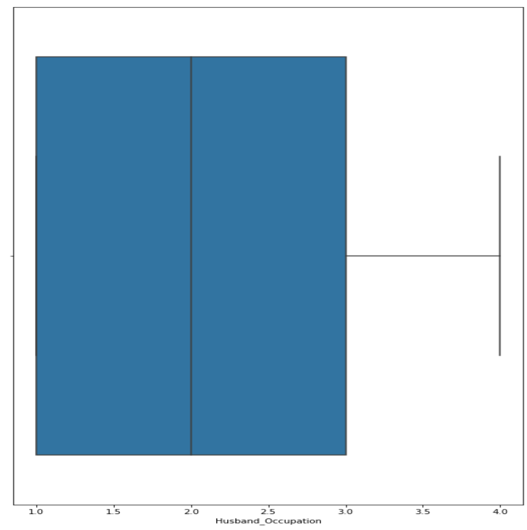






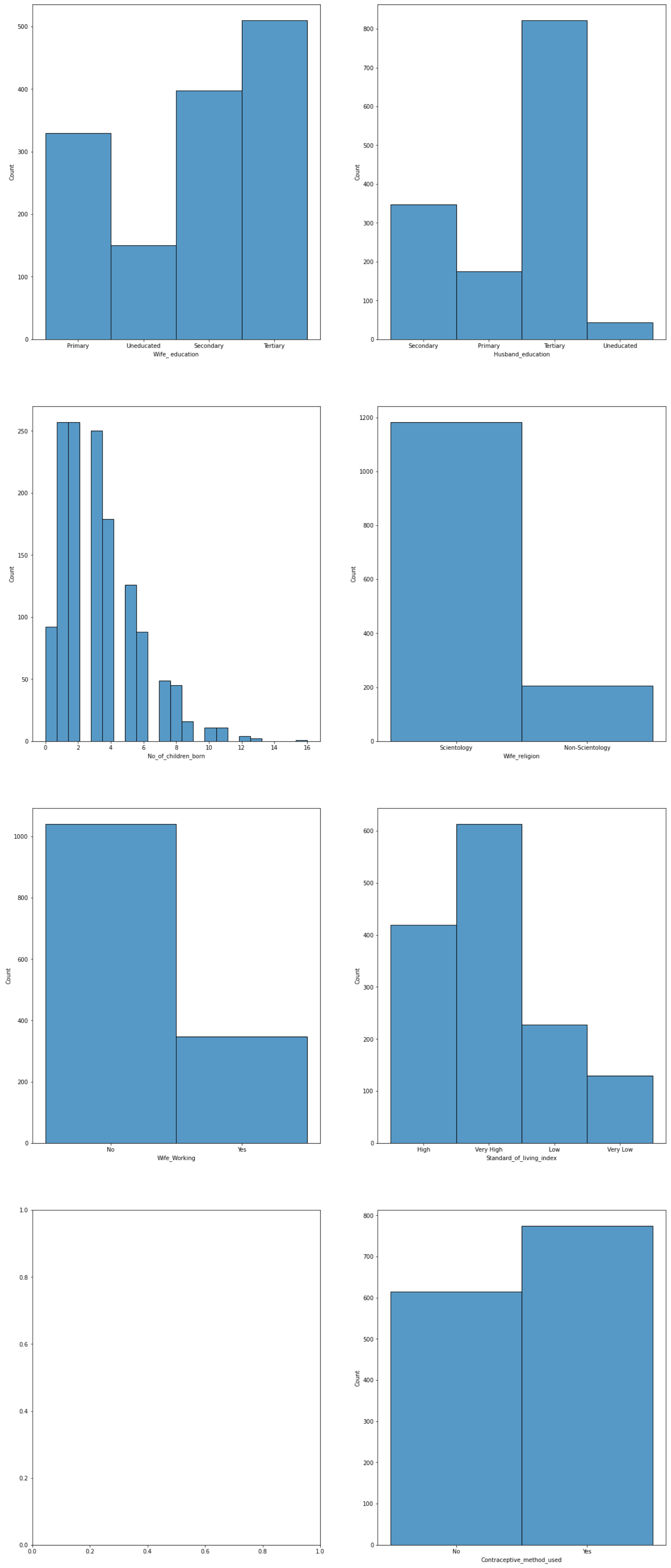






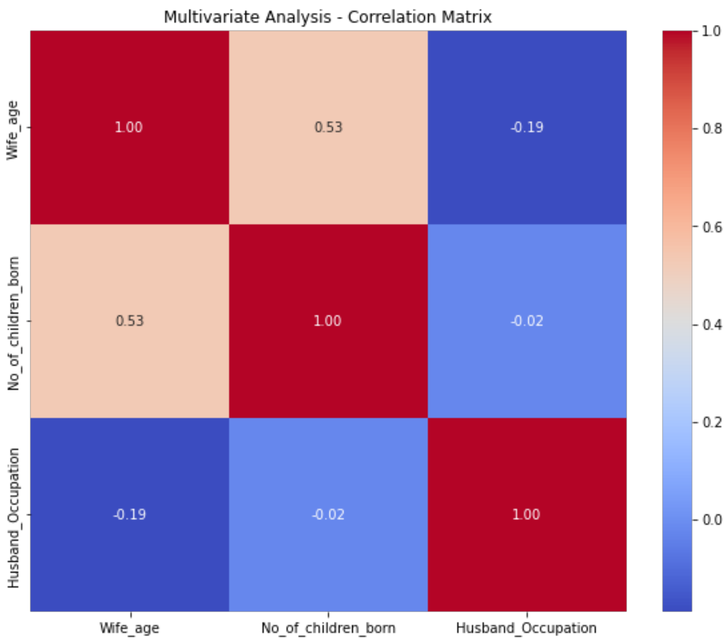
Boxplots are the best tool to visualise the outliers on the data. Upon plotting boxplots for all numeric features we can see there are outliers only in the No. of children born field. As we know outliers will undermine the training process so we need to treat them.

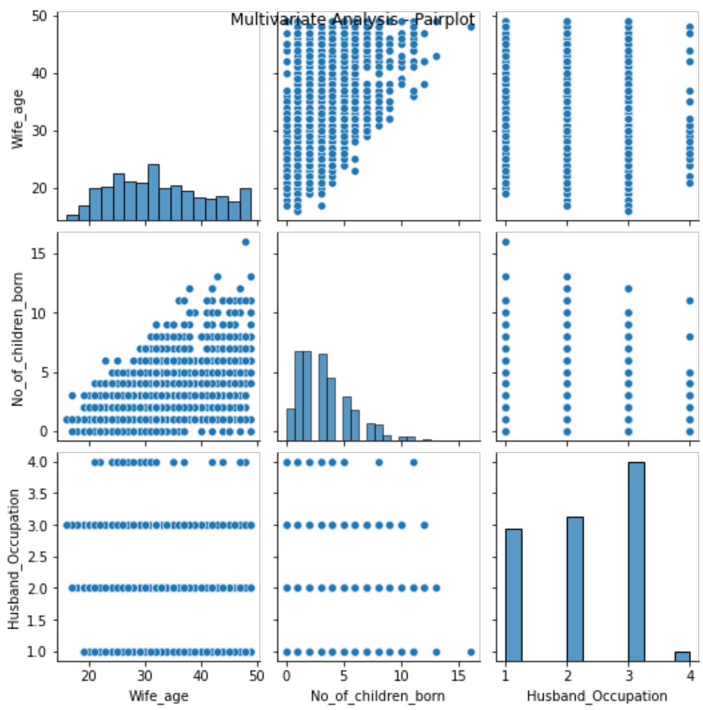
There are several ways to treat outliers. Here I have used capping and flooring technique to treat them.



Both wife and husband tertiary education level is more.
Scientologist wives are more.
Majority wives are not working.
Most of the males or husband has category 3 occupation.
Media exposure is more

Bivariate analysis





Data consists of both categorical and numerical values.
There are total of 1473 rows and 10 columns in the dataset. Out of 22, 7 columns are of object type, 1 columns of integer type and remaining 2 are of float type data.

'contraceptive used' is the target variable and all other are predictor variables.
Looking into the fields in the univariate analysis, we see outliers is present only in the field number of children.
Looking in to the boxplot between target variable contraceptive method used and the no_of_children_born, we see that, No_of_children_born is high in the case of use of contraception used.
Bivariate and multivariate analysis indicates that there is strong positive correlation between the field's wife age and no_of_children_born
We also notice that there are 80 duplicates records in the given data set and has been removed.
Null values identified has been imputed with median.

2.3. Model Building and Compare the Performance of the Models

All descriptive features in the dataset(including response or target variable and descriptive features) need to be converted into numeric features in order to use the dataset in Scikit-learn functions. Converted the target variable "Contraceptive_method_used" into numeric by using Label Encoder function from Sklearn by defining the function label encoder.
For other descriptive features having more than two levels Data=pd.get_dummies(Data) function is used. drop_first option has been set to 'True' to encode the variable into a single column of 0 or 1.

	Wife_age	No_of_children_born	Wife_religion	Wife_Working	Husband_Occupation	Contraceptive_method_used	education_
0	24.0	3.0	2	1	2	1	
1	45.0	10.0	2	1	3	1	
2	43.0	7.0	2	1	3	1	
3	42.0	9.0	2	1	3	1	
4	36.0	8.0	2	1	3	1	
...	...	...	...	...	...	...	
1466	42.0	3.0	2	1	2	2	
1468	33.0	3.0	2	2	2	2	
1470	39.0	3.0	2	2	1	2	
1471	33.0	3.0	2	2	2	2	
1472	17.0	1.0	2	1	2	2	

1388 rows × 20 columns

Decision tree classifier:
It is a class capable of performing multi-class classification on a dataset. takes as input two arrays: an array X, sparse or dense, of shape (n_samples, n_features) holding the training samples, and an array Y of integer values, shape (n_samples,), holding the class labels for the training samples: After being fitted, the model can then be used to predict the class of samples: Made models using Decision Tree Classifier , Logistic Regression and LDA and comparing the Accuracy to find the best model. Looks like Decision Tree Classifier, is under-fitting because train accuracy > test accuracy ., Let's Grid Search to get the best parameters or prune the tree

```
Logistic Regression Classification Report:
```

	precision	recall	f1-score	support
1	0.64	0.42	0.51	119
2	0.66	0.82	0.73	159
accuracy			0.65	278
macro avg	0.65	0.62	0.62	278
weighted avg	0.65	0.65	0.63	278

Linear Discriminant Analysis Classification Report:

	precision	recall	f1-score	support
1	0.64	0.40	0.49	119
2	0.65	0.83	0.73	159
accuracy			0.65	278
macro avg	0.65	0.62	0.61	278
weighted avg	0.65	0.65	0.63	278

Pruned CART Classification Report:

	precision	recall	f1-score	support
1	0.67	0.56	0.61	119
2	0.71	0.79	0.75	159
accuracy			0.69	278
macro avg	0.69	0.68	0.68	278
weighted avg	0.69	0.69	0.69	278

2.4. Business Insights & Recommendations

INSIGHTS FROM LOGISTIC REGRESSION:
For predicting the target variable " Contraceptive_method_used " is "No"(label 0) Precision : tells us how many predictions are actually positive out of all the total positive predicted. Precision (64%) – 65% of the people predicted are actually not using contraceptions out of all families predicted to have been not using contraceptions. Recall : how many observations of positive class are actually predicted as positive. Recall (45%) – out of all the people not using contraceptions, 51% of families have been predicted correctly. For predicting the target variable " Contraceptive_method_used " is "Yes" (label 1) .Precision (63%) – 63% of the people predicted are actually not using contracept ions out of all families predicted to have been not using contraceptions. Recall (79%) – out of all the people not using contraceptions, 79% of families have been predicted correctly. Overall accuracy of the model – 63 % of total predictions are correct.
INSIGHTS FROM LDA:
For predicting the target variable " Contraceptive_method_used " is "No"(label 0) .Precision (65%) – 65% of the people predicted are actually not using contracept ions out of all families predicted to have been not using contraceptions. Recall (45%) – out of all the people not using contraceptions, 45% of families have been predicted correctly. For predicting the target variable " Contraceptive_method_used " is "Yes" (label 1) .Precision (68%) – 68% of the people predicted are actually not using contracept ions out of all families predicted to have been not using contraceptions. Recall (82%) – out of all the people not using contraceptions, 82% of families ha ve been predicted correctly. Overall accuracy of the model – 64 % of total predictions are correct.
INSIGHTS FROM CART:
For predicting the target variable " Contraceptive_method_used " is "No"(label 0) .Precision (70%) – 70% of the people predicted are actually not using contracept ions out of all families predicted to have been not using contraceptions. Recall (54%) – out of all the people not using contraceptions, 54% of families have been predicted correctly. For predicting the target variable " Contraceptive_method_used " is "Yes" (label 1) .Precision (67%) – 67% of the people predicted are actually not using contracept ions out of all families predicted to have been not using contraceptions. Recall (80%) – out of all the people not using contraceptions, 80% of families have been predicted correctly. Overall accuracy of the model – 68 % of total predictions are correct. Accuracy of test data is comparatively more in CART(0.68) ,followed by LDA(0.6 4) and LOGISTIC model(0.63). AUC is also almost same for all the three models i.e. 0.718 Accuracy, AUC, Precision and Recall for test data is almost inline with training data. This proves no overfitting or underfitting has happened, and overall all the model can be considered suitable for classification.