PROJECT REPORT

SOMDEB PRAMANIK

CONTENTS

LIST OF TABLES

LIST OF FIGURES

PROBLEM 1: LINEAR REGRESSION ON COMPACTIVE DATASET

The comp-activ dataset 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.

In this dataset information related to various parameters are given in various columns that can affect the percentage of the time for which the cpu of a particular system runs in the user mode. We know that the cpu can run in two different modes, the user mode and the kernel mode.

The objective of this analysis is to build a linear regression model based on the values of the independent attributes and the target feature using which one can predict the fraction / percentage of the time for which the cpu of a system runs in the user mode given the values of the independent parameters for that system. We would also like to bring out the relative importances of the different independent attributes in being able to predict the value of the dependant feature.

The explanation of each independent feature and also the dependant feature was taken from the data dictionary provided along with the dataset.

The entire analysis was performed in python. The step by step process has been discussed in detail below and we have grouped the steps that have been executed into buckets according to the questions given in the rubrics.

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.

The basic libraries for data manipulation , data visualisation and mathematical computations (pandas, matplotlib and seaborn, numpy respectively) were imported. The other libraries required for model building and other specific tasks were imported later as and when required.

The relevant dataset which was provided in the form of an excel file was imported into the jupyter notebook in the form of a dataframe. The top 5 records and the last 5 records were printed to check whether the data had been loaded properly without any issues or not. It was observed at that stage that the total number of columns in the dataset was quite large, i.e 22 which was exceeding the maximum column display capacity of the notebook’s output window.

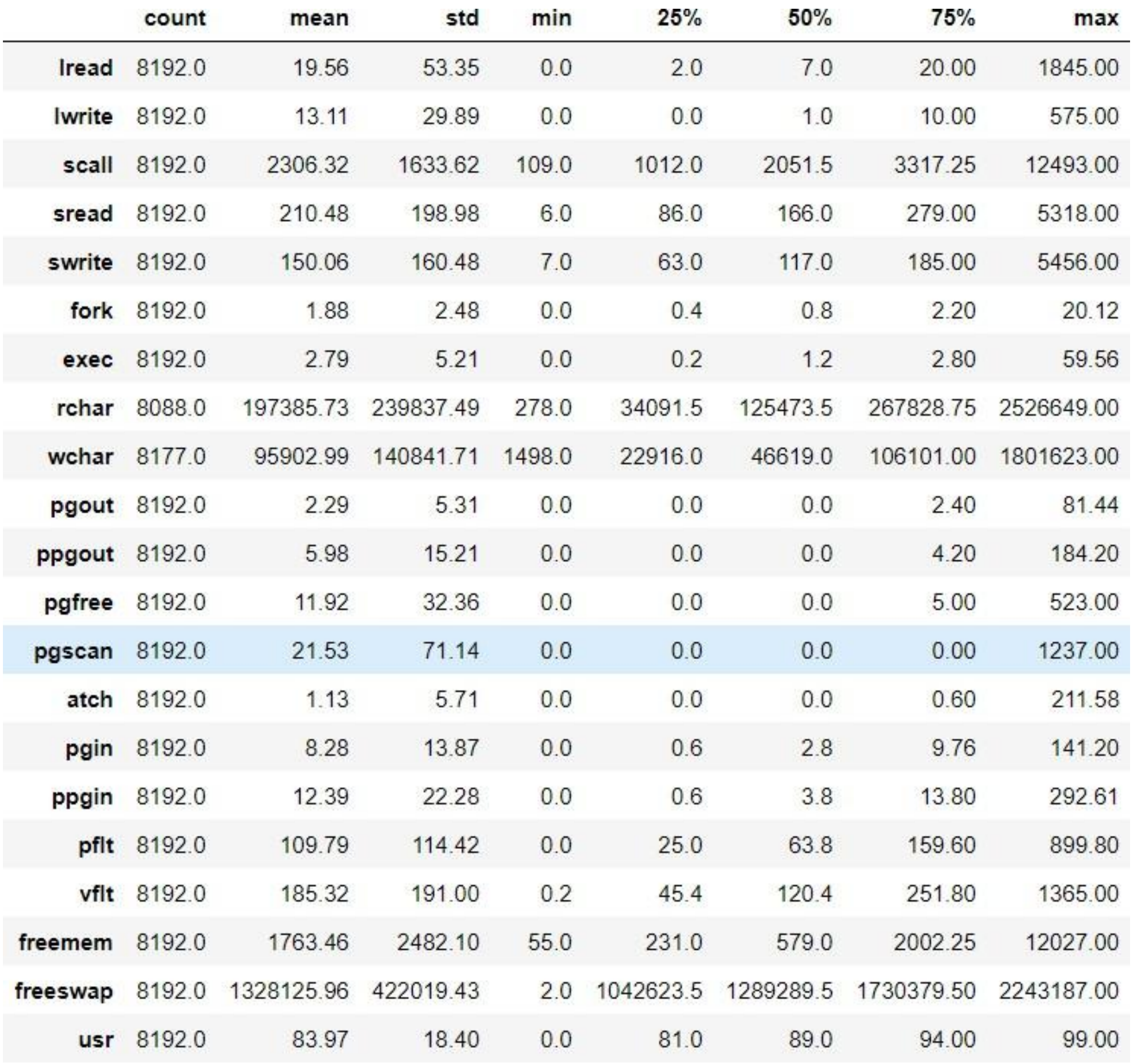
So some of the columns were compressed. In order to get rid of this issue the maximum display capacity of the jupyter notebook's output window was increased to 25. Thereafter all the columns were visible for the top 5 records and the last 5 records. It was observed that the data had been loaded properly.

Thereafter the shape of the dataset was checked and it was found to consist of 8192 rows and 22 columns. Each of the 8192 rows referred to one observation, i.e one computer system and each such system was described in terms of 22 features or attributes whose names were present along the 22 columns of the dataset. Out of these 22 features the last feature i.e the 'usr' was the dependant / target feature and the remaining were the independent / explanatory / predictor variables.

The datatypes of the columns were extracted and it was found that out of the 22 features, 21 (including the target feature 'usr) were of the numeric datatype (int or float) and only 1 feature 'runqsz' was of object datatype. So before considering this feature for building linear regression model, it has to be encoded because algorithms always work on numbers and not strings / characters.

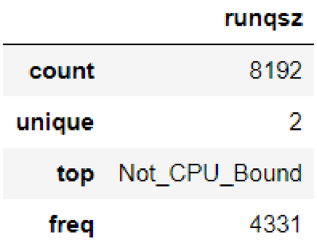
Some null / missing values were found to be present in the two features 'rchar' (104) and 'wchar' (15). Discussion on their imputation will be taken up later in this document. However there were no duplicate records in the dataset and each and every row was found to be unique.

The descriptive statistics of all the numeric columns and also the lone categorical feature was studied both non – visually and visually. From the non – visual part of the analysis some key points that were noted are as follows:



	count	mean	std	min	25%	50%	75%	max
lread	8192.0	19.56	53.35	0.0	2.0	7.0	20.00	1845.00
lwrite	8192.0	13.11	29.89	0.0	0.0	1.0	10.00	575.00
scall	8192.0	2306.32	1633.62	109.0	1012.0	2051.5	3317.25	12493.00
sread	8192.0	210.48	198.98	6.0	86.0	166.0	279.00	5318.00
swrite	8192.0	150.06	160.48	7.0	63.0	117.0	185.00	5456.00
fork	8192.0	1.88	2.48	0.0	0.4	0.8	2.20	20.12
exec	8192.0	2.79	5.21	0.0	0.2	1.2	2.80	59.56
rchar	8088.0	197385.73	239837.49	278.0	34091.5	125473.5	267828.75	2526649.00
wchar	8177.0	95902.99	140841.71	1498.0	22916.0	46619.0	106101.00	1801623.00
pgout	8192.0	2.29	5.31	0.0	0.0	0.0	2.40	81.44
ppgout	8192.0	5.98	15.21	0.0	0.0	0.0	4.20	184.20
pgfree	8192.0	11.92	32.36	0.0	0.0	0.0	5.00	523.00
pgscan	8192.0	21.53	71.14	0.0	0.0	0.0	0.00	1237.00
atch	8192.0	1.13	5.71	0.0	0.0	0.0	0.60	211.58
pgin	8192.0	8.28	13.87	0.0	0.6	2.8	9.76	141.20
ppgin	8192.0	12.39	22.28	0.0	0.6	3.8	13.80	292.61
pflt	8192.0	109.79	114.42	0.0	25.0	63.8	159.60	899.80
vflt	8192.0	185.32	191.00	0.2	45.4	120.4	251.80	1365.00
freemem	8192.0	1763.46	2482.10	55.0	231.0	579.0	2002.25	12027.00
freeswap	8192.0	1328125.96	422019.43	2.0	1042623.5	1289289.5	1730379.50	2243187.00
usr	8192.0	83.97	18.40	0.0	81.0	89.0	94.00	99.00

TABLE 1.1 : Statistical summary of numerical features



	runqsz
count	8192
unique	2
top	Not_CPU_Bound
freq	4331

TABLE 1.2 : Statistical summary of categorical features

- From the descriptive statistics of the numerical columns, by comparing the mean and the median for almost all features it can be said that most distributions are skewed and the skew is towards the right in most cases.
- There are possibilities of existence of outliers towards the higher side for most of the numerical features.
- There are quite a few features where the majority of the data points are 0s.
- These observations will be verified later through visual means (boxplot).
- For the lone categorical feature it was found that for majority of the records the 'runqsz' was of the type non-CPU bound. This will again be backed up by visual analysis (countplot).

VISUAL ANALYSIS

UNIVARIATE ANALYSIS OF NUMERICAL FEATURES

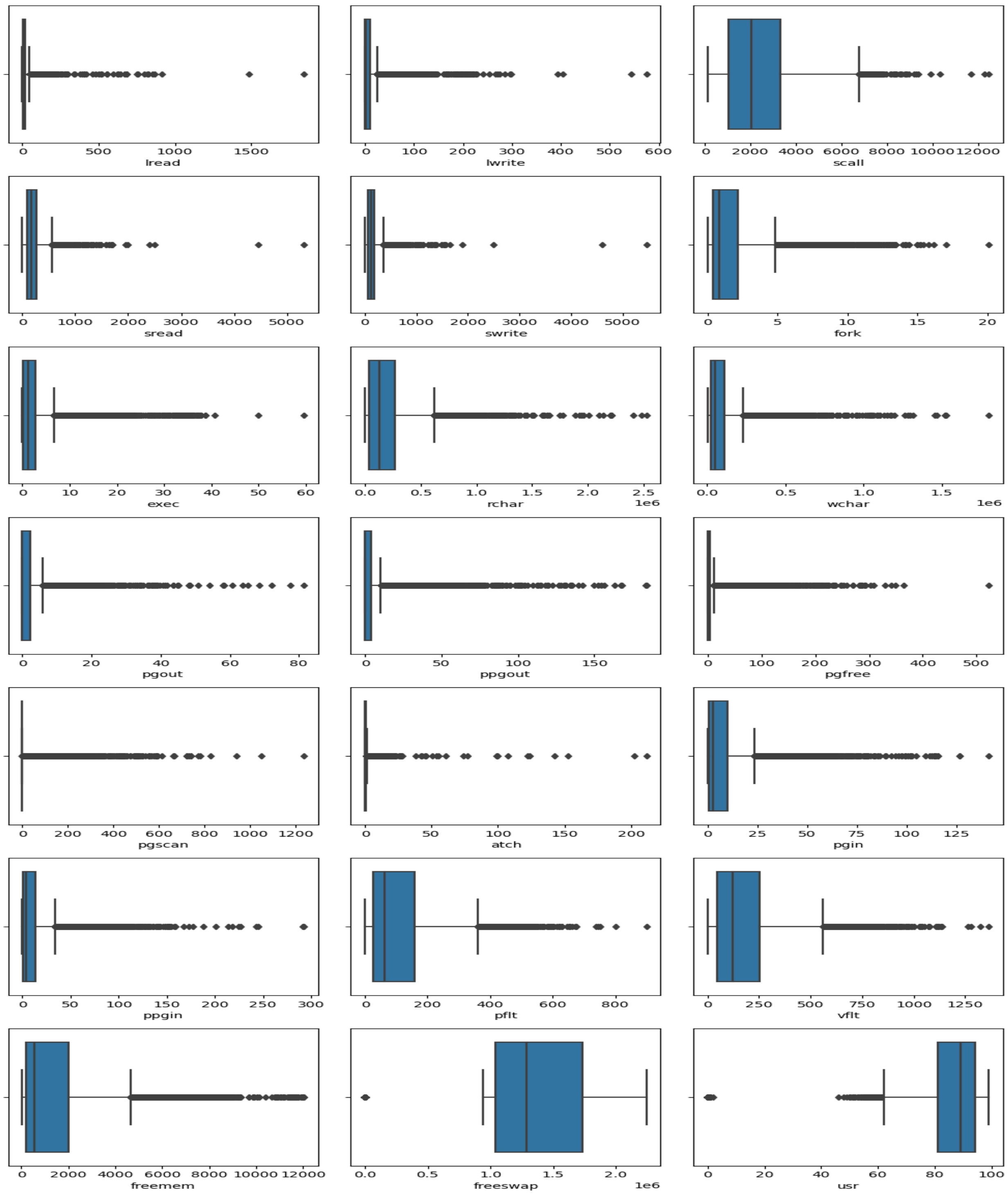


FIGURE 1.1 : Univariate analysis of numerical features through boxplot

It can be observed from the above boxplots that most of the numeric features have a right skewed distribution with a large number of outliers towards the right tail.

The dependant feature ‘usr’ however has a left skewed distribution.

We shall again refer to these boxplots again during the outlier treatment.

UNIVARIATE ANALYSIS OF THE CATEGORICAL FEATURE

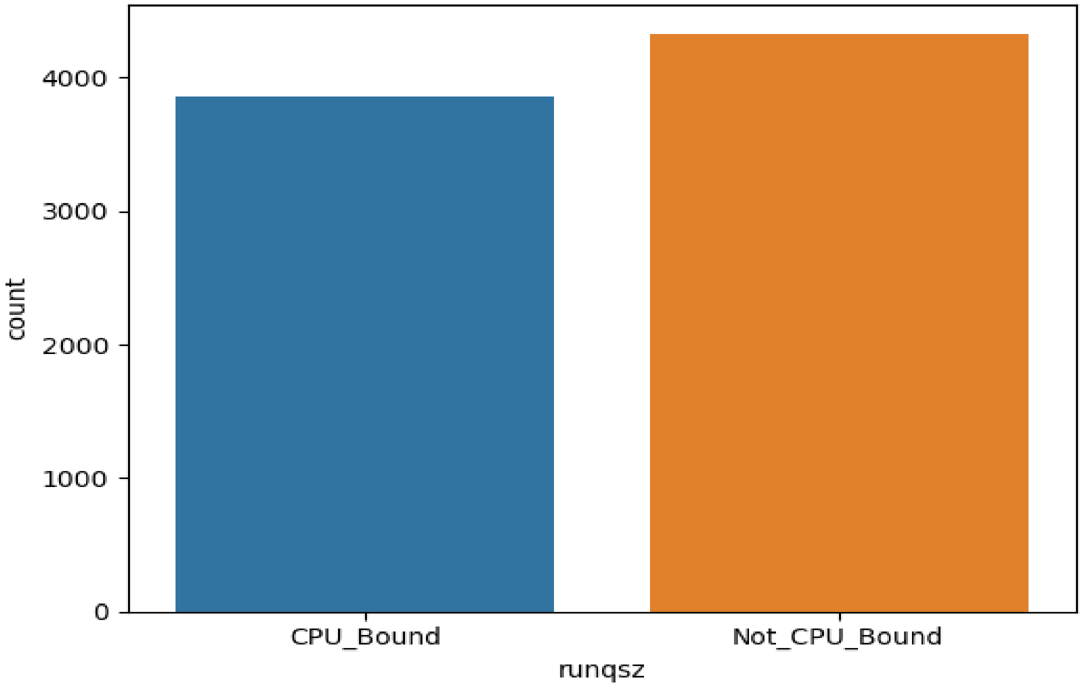


FIGURE 1.2 : Univariate analysis of categorical features through countplot

For a greater number of records in the dataset, the feature ‘runqsz’ is non cpu bound but the difference with the number of cpu bound ones is not too large as is evident from the above countplot.

BIVARIATE ANALYSIS OF NUMERICAL FEATURES

As a part of bivariate analysis of the numerical features we shall look at the pairplot and the heatmap / correlation matrix.

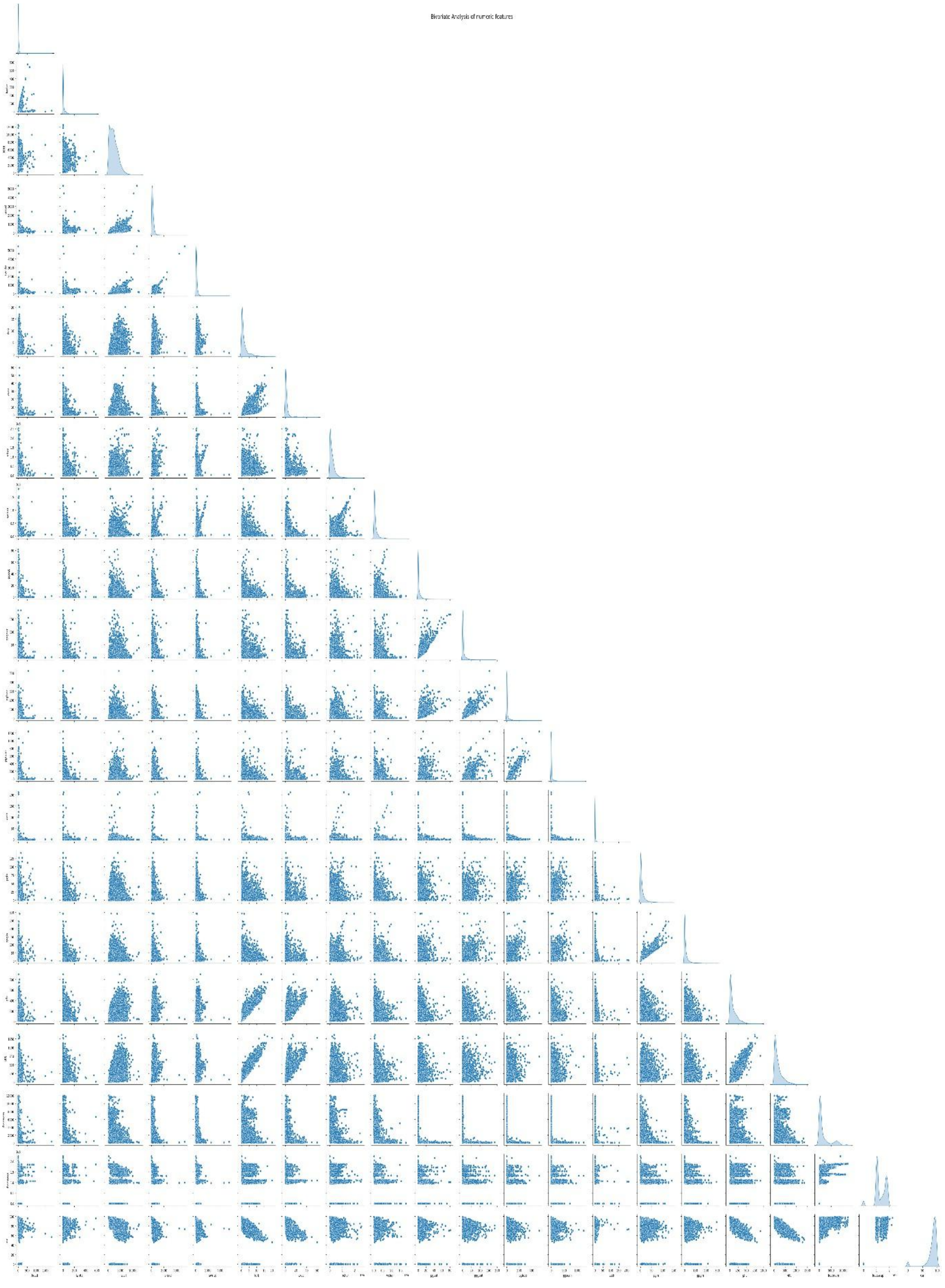


FIGURE 1.3 : Bivariate analysis of numerical features through pairplot (please see attached code file for better view)

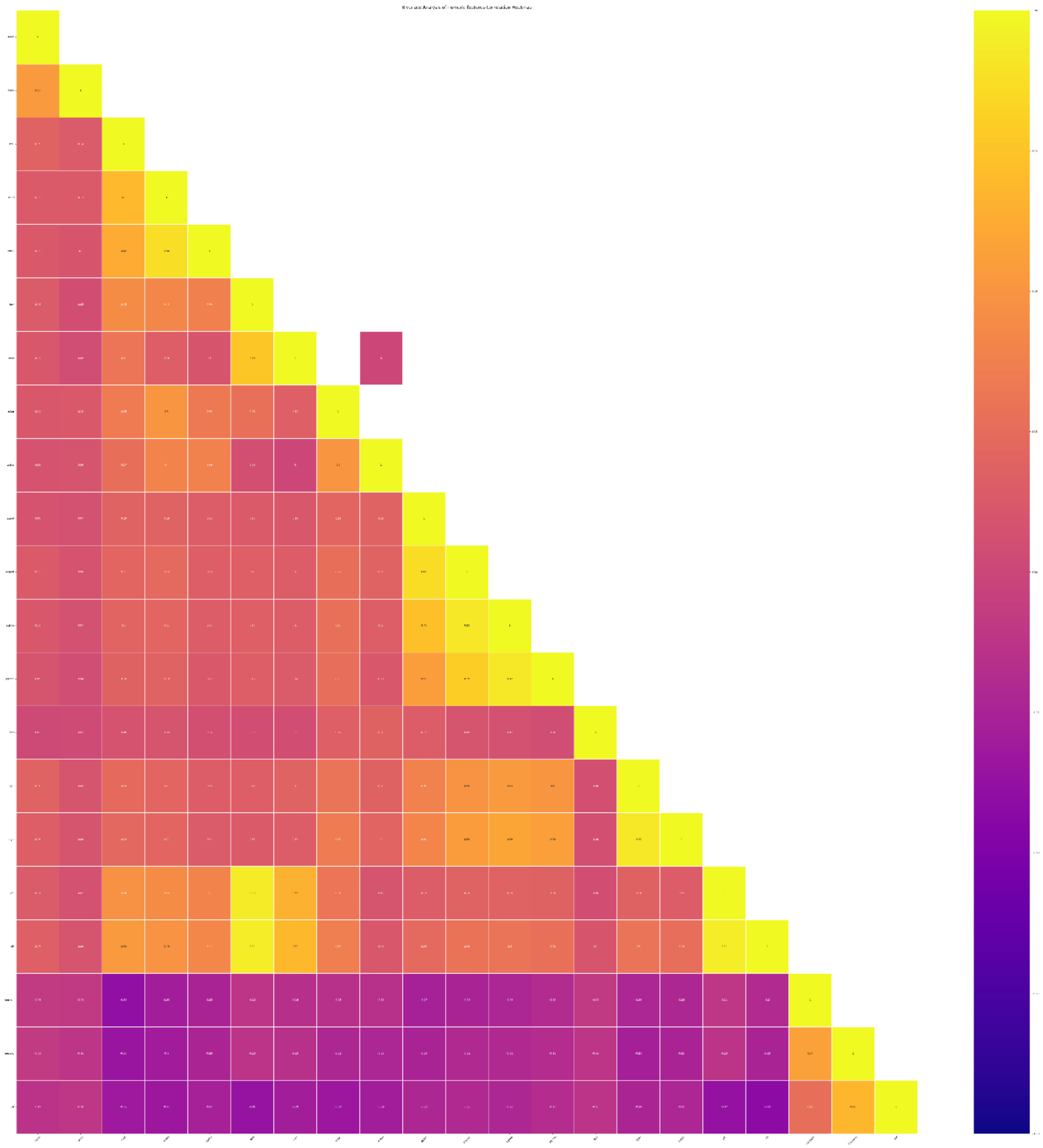


FIGURE 1.4 : Bivariate analysis of numerical features through heatmap (please see attached code file for better view)

From the pairplots it was observed that most of the independent features don’t have a strong correlation with the target feature (‘usr’) and that was confirmed from the heatmap / correlation matrix as well. Therefore we don’t expect even before the model is built, the presence of large positive or negative coefficients in the linear regression model. This is because most of the variables provided appear to be weak predictors of the target feature.

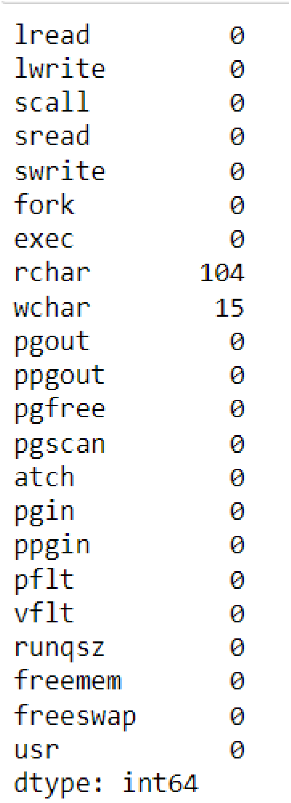
Moreover some pairs of independent variables appear to be internally correlated, which shows clear signs of multicollinearity.

However all these observations will be verified once again after the outlier treatment and the missing value imputation since the data will change to a certain extent after that.

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.

NULL VALUE IMPUTATION

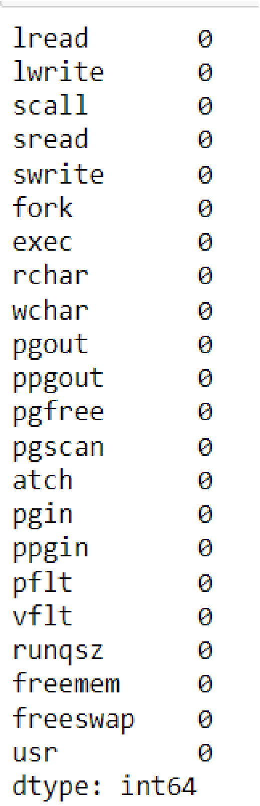
Two features viz ‘rchar’ and ‘wchar’ have 104 and 15 missing values respectively.



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

TABLE 1.3 : Null value counts for all features

Since both these features were of numeric type and both had lot of outliers (seen from boxplot), the imputation was done using the median values of the respective features and the relevant check was performed to see whether all the null values have been treated properly.



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

TABLE 1.4 : Null value counts for all features after imputation

We can see that no more null values are present in any column anymore.

DUPLICATE RECORDS

No two records in the dataset were found to be duplicate. Each and every record was unique.

OUTLIER TREATMENT

The dependant feature ‘usr’ was not subjected to outlier treatment.

There are some features which due to a high concentration of 0 values may get converted to effectively constant value features. Such zero variance features do not play an important role in prediction. These features were also left out from the list of features to which outlier treatment was applied. Therefore the features subjected to outlier treatment were 'freemem', 'freeswap', 'scall', 'vflt', 'pflt', 'ppgin', 'pgin', 'pgout', 'ppgout', 'wchar', 'rchar', 'exec', 'sread', 'swrite', and 'fork'.

The outlier treatment was done by capping the outliers on the upper end to the upper bound and those on the lower end to the lower bound, i.e by the traditional method. Now let us once again take a look at the boxplots of all the numeric features which were subjected to outlier treatment to see whether the outlier treatment was properly done.

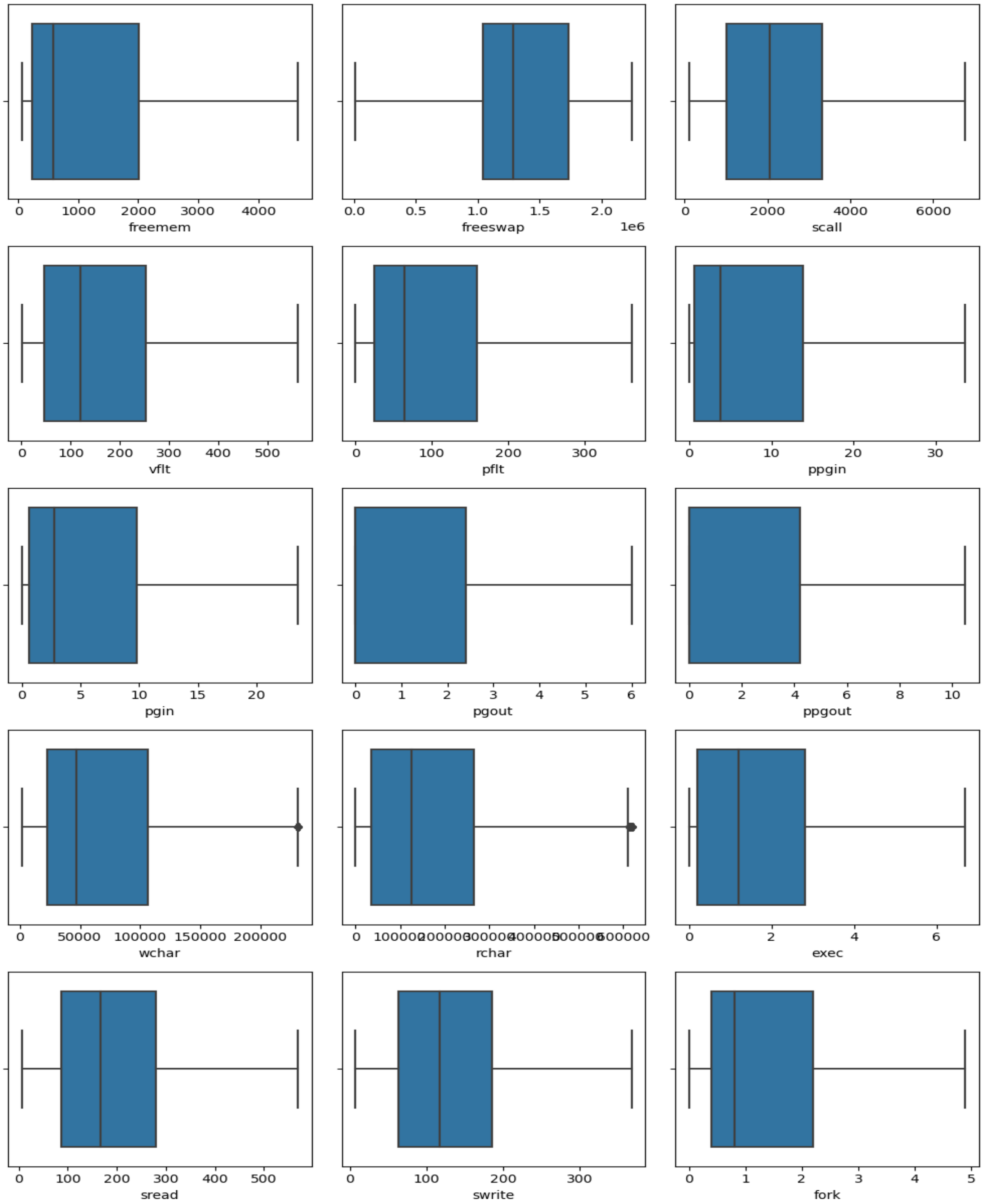


FIGURE 1.5 : Boxplots of numerical features after outlier treatment and missing value imputation

The above boxplots indicate the outlier treatment was done properly.

BIVARIATE ANALYSIS AFTER OUTLIER TREATMENT AND MISSING VALUE IMPUTATION

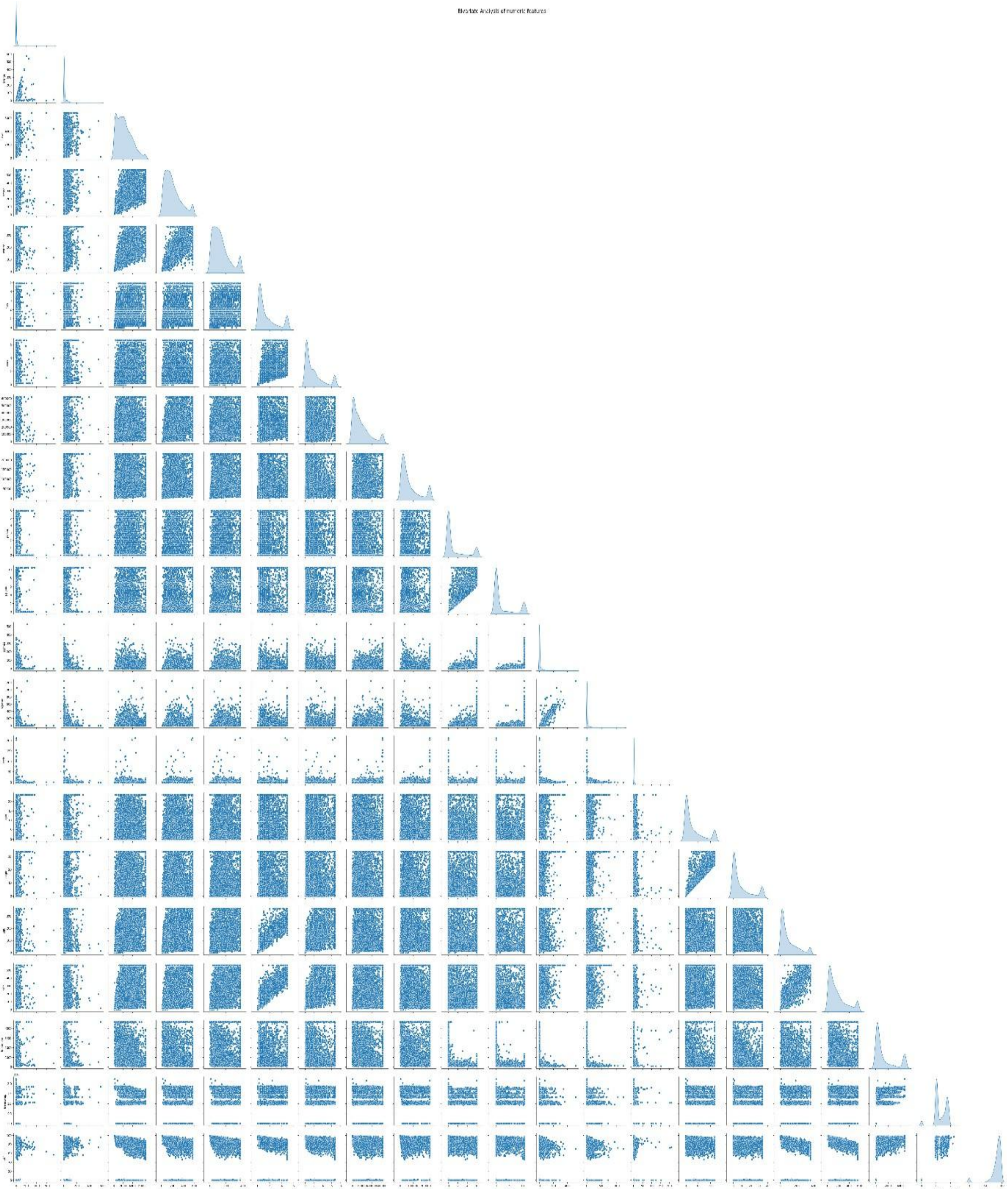


FIGURE 1.6 : Pairplots of numerical features after outlier treatment and missing value imputation

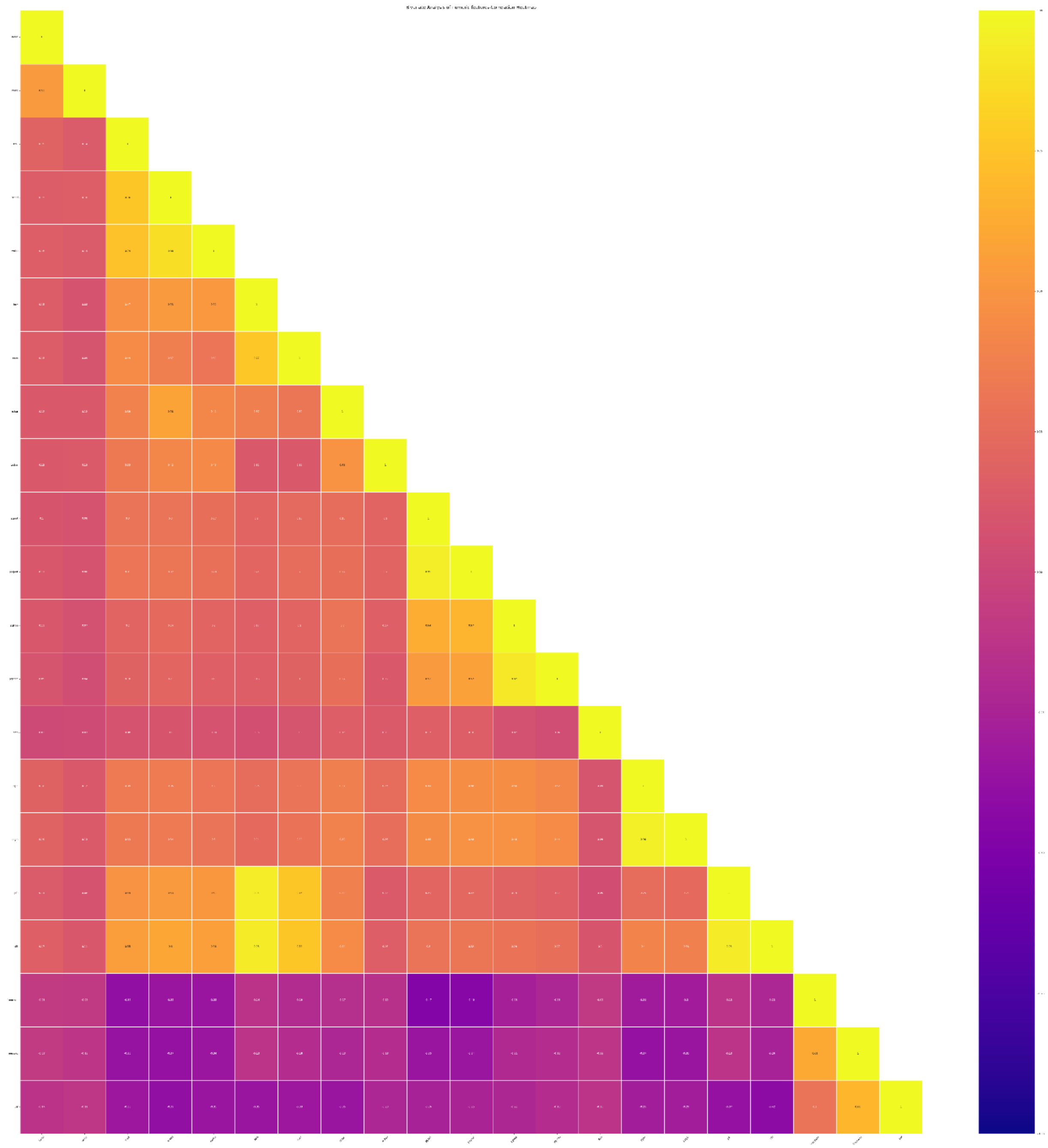


FIGURE 1.7 : Heatmap of numerical features after outlier treatment and missing value imputation

The previous version of the conclusions drawn from the bivariate analysis did not change much due to outlier treatment and missing value imputation.

The predictor variables continue to be weakly associated with the target variable whereas there are some strong positive and negative correlations among few pairs of independent variables. This is referred to as multicollinearity.

Now if we drop out of these pairs of linearly and strongly correlated variables the one having weak linear association with the target variable, and also if we try to remove independent features which appear to be weak predictors of the target variable, we may end up having no independent variable at our disposal for building the model. This is because the heatmap and the pairplot suggest that almost all independent features are weakly associated with the target variable. Therefore it would be a better approach to go ahead with the model building exercise and then address the issue of multicollinearity and unimportant predictor variables and thereby try to fine tune the model as model building is an iterative exercise and not a waterfall.

Quite a few features were found to have a large number of 0 values. These values are perfectly fine because there may be some systems which were used less compared to other systems which might have been the contributing factor for the 0 values in those features. If we change or drop them, the data will change and we will also lose a lot of other important information.

For creating new features out of the existing ones for better predictive modelling, domain knowledge is of utmost importance. Therefore it would be better to do this in collaboration with the client and not from our end based on our own understanding of the business objective as a standalone criterion.

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.

ENCODING OF CATEGORICAL FEATURES

There was only one feature in the entire dataset ('runqsz') that belonged to the 'object' datatype. It was encoded using the one hot encoding method and out of the two dummy variables thus created corresponding to its two categorical levels, only one was retained using the 'drop_first' parameter and setting its value to 2.

SEPARATING THE INDEPENDENT AND DEPENDANT FEATURES

From the original dataframe (df) two separate dataframes were created, one containing only the independent features (X) from the original dataframe and the other containing the dependant feature 'usr' (y).

SPLITTING THE DATA INTO TRAINING SET AND TESTING SET

Now both the independent and dependant features were split into two dataframes, viz the training set and the testing set. The records were randomly assigned to these two sets and 70% of the observations from the original dataset populated the training set and the remaining 30% of the records were part of the test set. To ensure repeatability / to freeze the randomness of the creation of training and testing sets across notebooks or for different executions of the code in the same notebook the value of the random_state parameter was set as 1.

MODEL BUILDING (TRAINING), PREDICTION AND MODEL EVALUATION USING SKLEARN

The linear regression model was initially built using the scikit learn library. The model was trained using the training dataset described above. Then the predictions of the built model were obtained on the training data as well as the testing data (data to which the model was unexposed while it was being built).

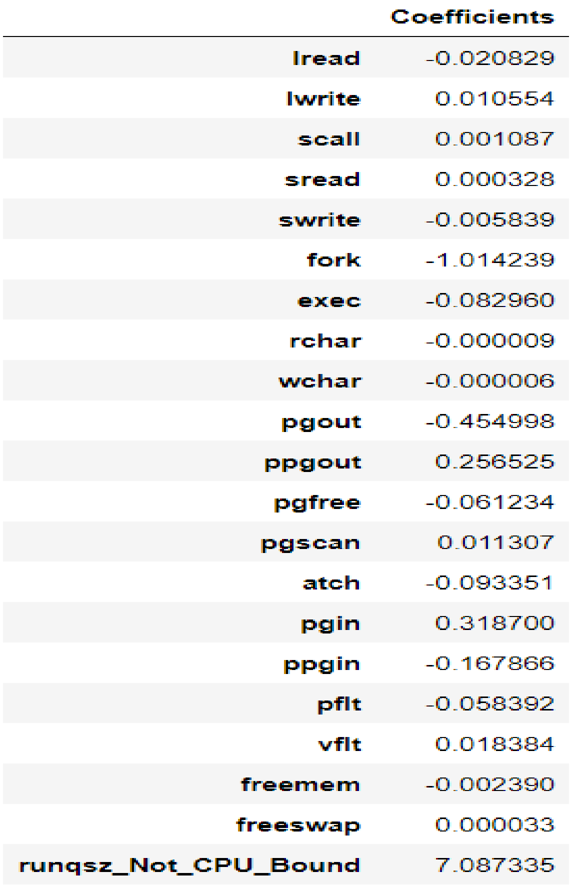
The rsquared values were computed for the models predictions and the values were 0.624 and 0.614 for the training data and testing data respectively. So the model, although not overfit (performance metrics for training and testing data were not having a difference of more that

10%) cannot be called a good model because it is able to explain (SSE) only around 62 % of the variation in the value of the dependant variable which was unexplained in the absence of the model (SST).

The entire process of model building, obtaining predictions and evaluating the model in terms of rsquared values was repeated without using any intercept term in the model, but the performance went further down as the rsquared value for the training set was 0.41 and that for the testing set was 0.45.

STRUCTURAL ANALYSIS OF THE MODEL

Therefore after being convinced that the model using a non zero value of the intercept is yielding better results, the structural analysis of the model was performed. The intercept term (value of the target variable for zero value of all independent features) was found to be 44.96 and the coefficients of the independent variables can be obtained from the following dataframe.



	Coefficients
lread	-0.020829
lwrite	0.010554
scall	0.001087
sread	0.000328
swrite	-0.005839
fork	-1.014239
exec	-0.082960
rchar	-0.000009
wchar	-0.000006
pgout	-0.454998
ppgout	0.256525
pgfree	-0.061234
pgscan	0.011307
atch	-0.093351
pgin	0.318700
ppgin	-0.167866
pflt	-0.058392
vflt	0.018384
freemem	-0.002390
freeswap	0.000033
runqsz_Not_CPU_Bound	7.087335

TABLE 1.5 : Coefficients of independent attributes in linear regression model built by using scikit learn

LINEAR REGRESSION EQUATION

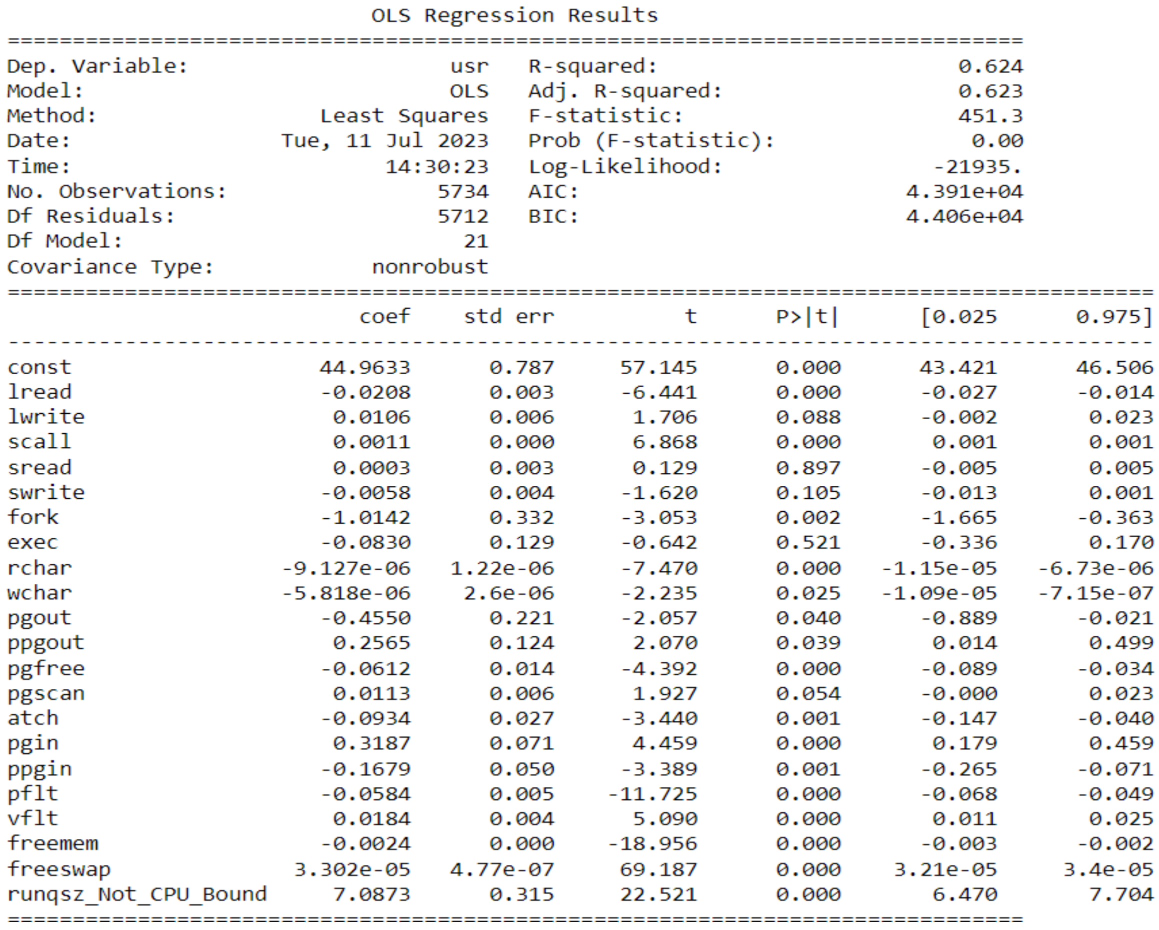
Now we can construct the linear regression model equation as follows.

$$\begin{aligned} usr = 44.96 - 0.02(lread) + (0.01)lwrite + (0.001)scall + (0.003)sread \\ + (-0.05)swrite + (-1.014)fork + (-0.08)exec + (-0.45)pgout \\ + (0.26)ppgout + (-0.06)pgfree + (0.01)pgscan + (-0.09)atch \\ + (0.32)pgin + (-0.17)ppgin + (-0.06)pflt + (0.018)vflt \\ + (-0.002)freemem + (7.087)runsqz\_not\_cpu\_bouns \end{aligned}$$

Some variables having extremely low coefficients in the above table have been intentionally omitted from the equation of the model because of their low predictive ability according to the model. According to this model, the ‘runsqz’ variable has the highest coefficient. However this being an encoded categorical variable, the interpretation of its coefficient is different from that of the remaining variables which are numerical. The portion of time the cpu runs in user mode is greater for systems with ‘runqsz’ non cpu bound by 7.087 % than those systems for which the ‘runqsz’ is cpu bound. Moreover the statistical significance of this model and that of each coefficient remains unknown. Therefore we went forward and repeated the process of model building and evaluation using the statsmodels library. However model building being an iterative exercise we have tried out different ways and means of improving the model.

As per the requirement of the statsmodels library the intercept was added to the dataframe containing the independent features.

The same training set and the testing set was used to train the linear regression model and the summary of the output is given below.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    usr   R-squared:                       0.624
Model:                            OLS   Adj. R-squared:                  0.623
Method:                 Least Squares   F-statistic:                     451.3
Date:                Tue, 11 Jul 2023   Prob (F-statistic):               0.00
Time:                        14:30:23   Log-Likelihood:                -21935.
No. Observations:                5734   AIC:                         4.391e+04
Df Residuals:                    5712   BIC:                         4.406e+04
Df Model:                          21
Covariance Type:            nonrobust
=========================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                  44.9633      0.787     57.145      0.000      43.421      46.506
lread                  -0.0208      0.003     -6.441      0.000      -0.027      -0.014
lwrite                  0.0106      0.006      1.706      0.088      -0.002       0.023
scall                   0.0011      0.000      6.868      0.000       0.001       0.001
sread                   0.0003      0.003      0.129      0.897      -0.005       0.005
swrite                 -0.0058      0.004     -1.620      0.105      -0.013       0.001
fork                   -1.0142      0.332     -3.053      0.002      -1.665      -0.363
exec                   -0.0830      0.129     -0.642      0.521      -0.336       0.170
rchar               -9.127e-06   1.22e-06     -7.470      0.000   -1.15e-05   -6.73e-06
wchar               -5.818e-06    2.6e-06     -2.235      0.025   -1.09e-05   -7.15e-07
pgout                  -0.4550      0.221     -2.057      0.040      -0.889      -0.021
ppgout                  0.2565      0.124      2.070      0.039       0.014       0.499
pgfree                 -0.0612      0.014     -4.392      0.000      -0.089      -0.034
pgscan                  0.0113      0.006      1.927      0.054      -0.000       0.023
atch                   -0.0934      0.027     -3.440      0.001      -0.147      -0.040
pgin                    0.3187      0.071      4.459      0.000       0.179       0.459
ppgin                  -0.1679      0.050     -3.389      0.001      -0.265      -0.071
pflt                   -0.0584      0.005    -11.725      0.000      -0.068      -0.049
vflt                    0.0184      0.004      5.090      0.000       0.011       0.025
freemem                -0.0024      0.000    -18.956      0.000      -0.003      -0.002
freeswap             3.302e-05   4.77e-07     69.187      0.000    3.21e-05     3.4e-05
runqsz_Not_CPU_Bound    7.0873      0.315     22.521      0.000       6.470       7.704
==============================================================================
```

TABLE 1.6 : Summary of the initial linear regression model built by using statsmodels

From the p-value (0.00 < 0.05) of the F-statistic for the entire model, we can conclude that the results are statistically significant on an overall basis.

However the R-squared (0.624) and the adjusted R-squared (0.623) values are quite low which means that the model is not a very good model. This may be due to the fact that we have started with all the variables inspite of noticing that most of the variables are not having a strong correlation with the dependant variable. But the R-squared values are not lower than what we observed with our model built on scikit learn.

For some of the variables like ‘sread’, ‘lwrite’, ‘exec’, etc. the p-values are quite high (>0.05) which leads to the conclusion that their coefficients are statistically significantly from zero.

Moreover from a direct observation the coefficients of several independent features are very small.

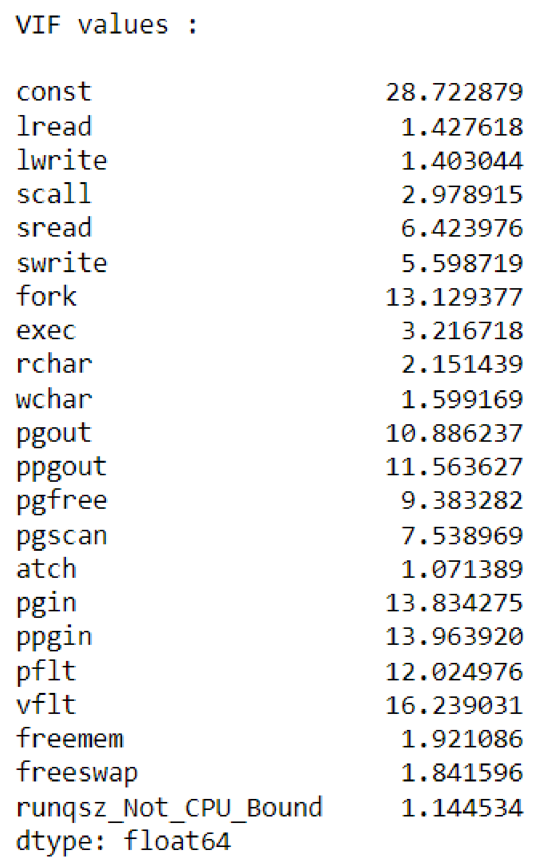
There may be multicollinearity (strong correlation among the independent features, evident from the previously plotted correlation matrix / heatmap) in the data also which can be tested by calculating the VIF scores for all the independent features.

In order to improve the model or at least to simplify it, the following approach was taken.

Initially we looked for multicollinearity in the data and tried to omit a few variables that were having the highest VIF scores (ensuring that the drop does not affect the r-squared values and the adjusted r-squared values by a significant amount). We also omitted some variables based on high p-values (> 0.05, indicative of their coefficients not being being statistically significantly different from 0).

Let us look at the detailed procedure that was followed.

The VIF scores calculated for the independent features after the first run of the linear regression model building (by statsmodels library) are as follows.



```
VIF values :

const                  28.722879
lread                   1.427618
lwrite                  1.403044
scall                   2.978915
sread                   6.423976
swrite                  5.598719
fork                   13.129377
exec                    3.216718
rchar                   2.151439
wchar                   1.599169
pgout                  10.886237
ppgout                 11.563627
pgfree                  9.383282
pgscan                  7.538969
atch                    1.071389
pgin                   13.834275
ppgin                  13.963920
pflt                   12.024976
vflt                   16.239031
freemem                 1.921086
freeswap                1.841596
runqsz_Not_CPU_Bound    1.144534
dtype: float64
```

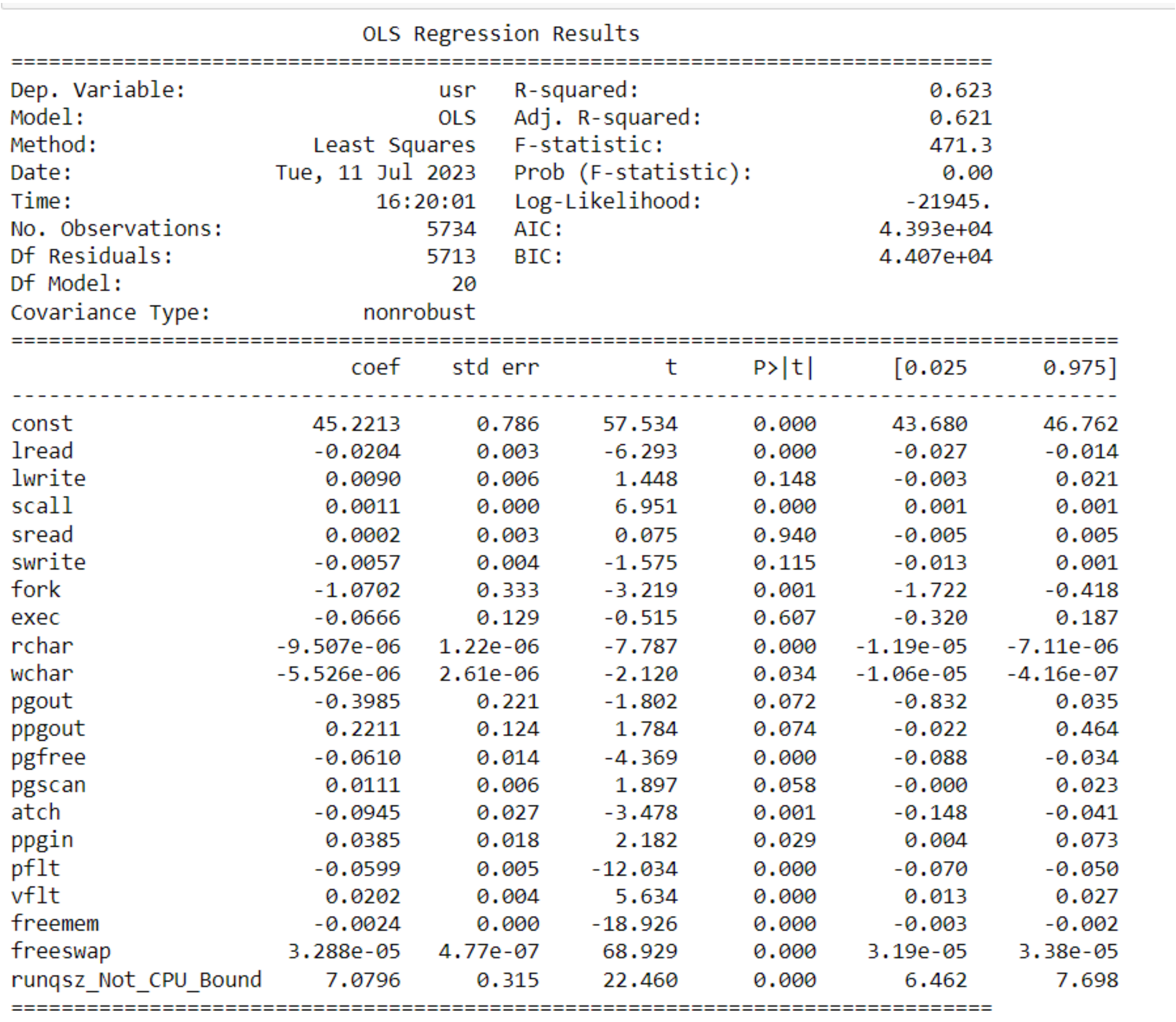
TABLE 1.7 : VIF scores based on the initial linear regression model built by using statsmodels

Quite a few variables have high VIF scores (>5) which means that the information brought in by these variables is also brought in by other variables. These variables are contributing the most towards the presence of multicollinearity in the data which makes the interpretation of the coefficients in the linear model untrustable.

Let us drop the variable with the highest VIF and see the effect of dropping it on the rsquared and adjusted rsquared values of the model.

On dropping 'vflt' (due to high VIF = 16.239) from the training set and reconstructing the model, the effect on R-squared and adjusted R-squared values was minimal. The same was the case with 'ppgin' (VIF = 13.96), 'pgin' (VIF = 13.83) and 'fork' (VIF = 13.13). However all these variables were dropped one at a time and while dropping the next, the previous was added back to the test set.

'pgin' was permanently dropped from the original training set and the regression was done all over again and multicollinearity was again checked for through the computation of VIFs of the remaining variables. The output was as follows.



```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                    usr   R-squared:                       0.623
Model:                            OLS   Adj. R-squared:                  0.621
Method:                 Least Squares   F-statistic:                     471.3
Date:                Tue, 11 Jul 2023   Prob (F-statistic):               0.00
Time:                        16:20:01   Log-Likelihood:                -21945.
No. Observations:                5734   AIC:                         4.393e+04
Df Residuals:                    5713   BIC:                         4.407e+04
Df Model:                          20                                         
Covariance Type:            nonrobust                                         
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
const                   45.2213      0.786     57.534      0.000      43.680      46.762
lread                   -0.0204      0.003     -6.293      0.000      -0.027      -0.014
lwrite                   0.0090      0.006      1.448      0.148      -0.003       0.021
scall                    0.0011      0.000      6.951      0.000       0.001       0.001
sread                    0.0002      0.003      0.075      0.940      -0.005       0.005
swrite                  -0.0057      0.004     -1.575      0.115      -0.013       0.001
fork                    -1.0702      0.333     -3.219      0.001      -1.722      -0.418
exec                    -0.0666      0.129     -0.515      0.607      -0.320       0.187
rchar                -9.507e-06   1.22e-06     -7.787      0.000   -1.19e-05   -7.11e-06
wchar                -5.526e-06   2.61e-06     -2.120      0.034   -1.06e-05   -4.16e-07
pgout                   -0.3985      0.221     -1.802      0.072      -0.832       0.035
ppgout                   0.2211      0.124      1.784      0.074      -0.022       0.464
pgfree                  -0.0610      0.014     -4.369      0.000      -0.088      -0.034
pgscan                   0.0111      0.006      1.897      0.058      -0.000       0.023
atch                    -0.0945      0.027     -3.478      0.001      -0.148      -0.041
ppgin                    0.0385      0.018      2.182      0.029       0.004       0.073
pflt                    -0.0599      0.005    -12.034      0.000      -0.070      -0.050
vflt                     0.0202      0.004      5.634      0.000       0.013       0.027
freemem                 -0.0024      0.000    -18.926      0.000      -0.003      -0.002
freeswap              3.288e-05   4.77e-07     68.929      0.000    3.19e-05    3.38e-05
runqsz_Not_CPU_Bound     7.0796      0.315     22.460      0.000       6.462       7.698
==============================================================================
```

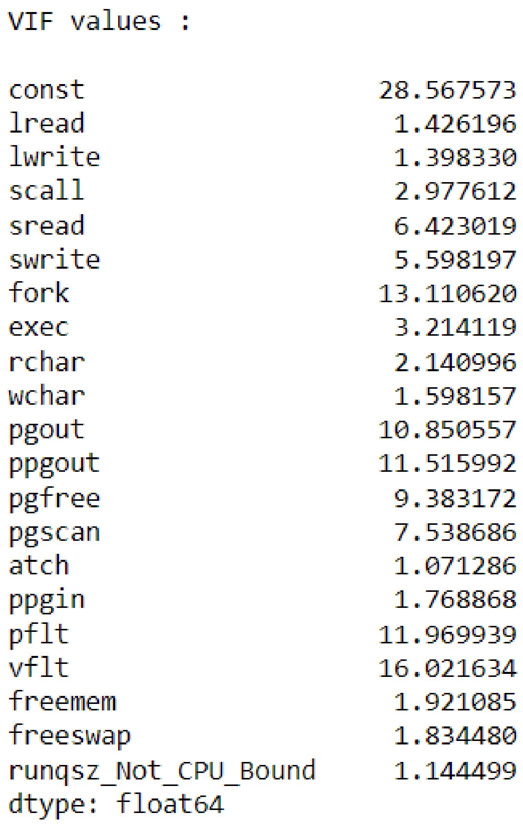
TABLE 1.8 : Summary of the revised model after dropping the ‘pgin’ feature

The r-squared and adjusted r-squared values did not go down drastically.

The output is statistically significant (p-value = 0.00 < 0.05).

The p-values of most of the coefficients are less than 0.05.

Let us take a look at the VIFs at this stage.



```
VIF values :

const                    28.567573
lread                     1.426196
lwrite                    1.398330
scall                     2.977612
sread                     6.423019
swrite                    5.598197
fork                     13.110620
exec                      3.214119
rchar                     2.140996
wchar                     1.598157
pgout                    10.850557
ppgout                   11.515992
pgfree                    9.383172
pgscan                    7.538686
atch                      1.071286
ppgin                     1.768868
pflt                     11.969939
vflt                     16.021634
freemem                   1.921085
freeswap                  1.834480
runqsz_Not_CPU_Bound      1.144499
dtype: float64
```

TABLE 1.9 : VIF scores based on the model after dropping the 'pgin' feature

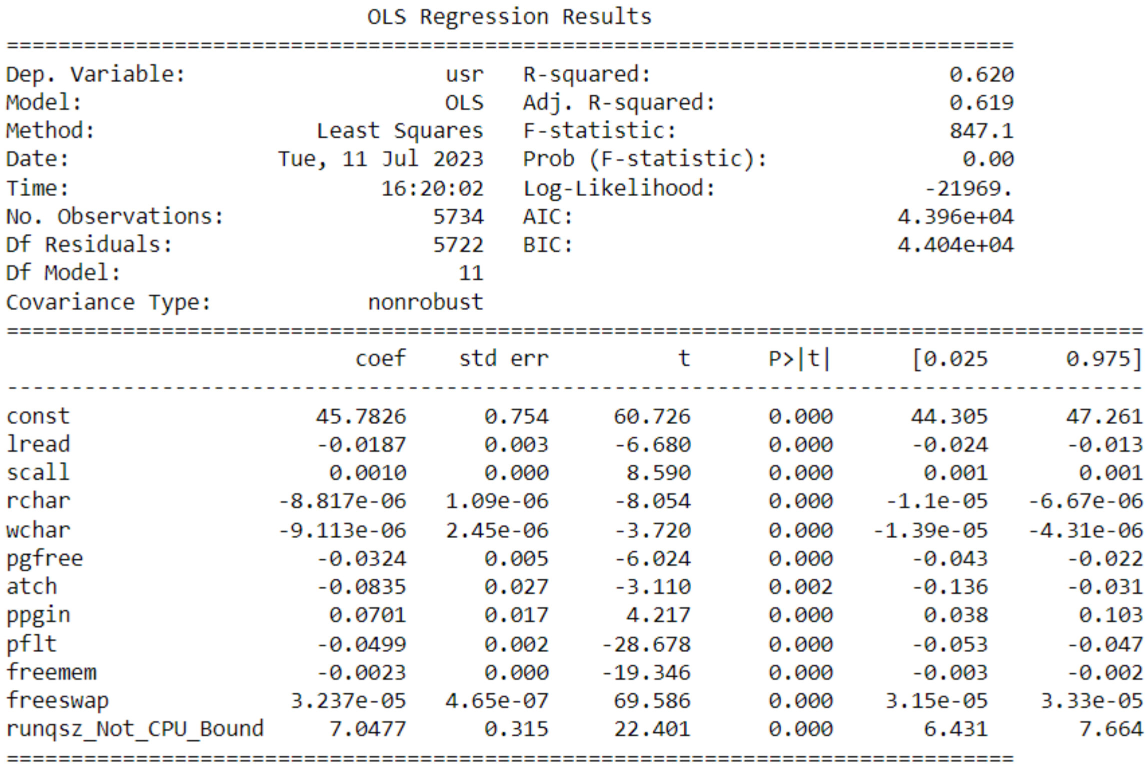
Still there are quite a few variables with high value of VIF.

So the process of dropping the variables with high values of VIFs , checking the effect on r-squared and adjusted r-squared values of the models after modification, identifying the variable causing minimum drop of r-squared and adjusted r-squared values, dropping it permanently from the training data and training the model with the modified training data and then again evaluating the VIF scores to check for possible existence of multicollinearity continued on an iterative manner (please see the attached code file for details).

In one of the iterations the independent variables with high p- values were dropped (since there is no evidence in the data which suggested that they are having coefficients significantly different from 0). These variables were ['lwrite','sread','exec','pgout','ppgout','pgscan'].

Then again the regression was done.

Finally after several iterations we came up with the following output.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    usr   R-squared:                       0.620
Model:                            OLS   Adj. R-squared:                  0.619
Method:                 Least Squares   F-statistic:                     847.1
Date:                Tue, 11 Jul 2023   Prob (F-statistic):               0.00
Time:                        16:20:02   Log-Likelihood:                -21969.
No. Observations:                5734   AIC:                         4.396e+04
Df Residuals:                    5722   BIC:                         4.404e+04
Df Model:                          11
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                    45.7826      0.754     60.726      0.000      44.305      47.261
lread                    -0.0187      0.003     -6.680      0.000      -0.024      -0.013
scall                     0.0010      0.000      8.590      0.000       0.001       0.001
rchar                 -8.817e-06   1.09e-06     -8.054      0.000    -1.1e-05   -6.67e-06
wchar                 -9.113e-06   2.45e-06     -3.720      0.000   -1.39e-05   -4.31e-06
pgfree                   -0.0324      0.005     -6.024      0.000      -0.043      -0.022
atch                     -0.0835      0.027     -3.110      0.002      -0.136      -0.031
ppgin                     0.0701      0.017      4.217      0.000       0.038       0.103
pflt                     -0.0499      0.002    -28.678      0.000      -0.053      -0.047
freemem                  -0.0023      0.000    -19.346      0.000      -0.003      -0.002
freeswap               3.237e-05   4.65e-07     69.586      0.000    3.15e-05    3.33e-05
runqsz_Not_CPU_Bound      7.0477      0.315     22.401      0.000       6.431       7.664
==============================================================================
```

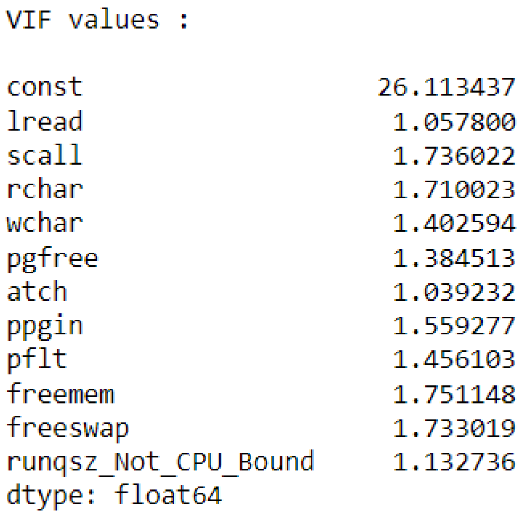
TABLE 1.10 : Summary of the final revised model

Here we find that the r-squared and adjusted r-squared values are close to the original ones.

The test results are statistically significant (p-value = 0.00 < 0.05).

The p- value of each feature is < 0.05 (means each one of their coefficients is statistically significantly different from 0).

Now let us check if multicollinearity still exists through the computation of the VIF scores.



```
VIF values :

const                  26.113437
lread                   1.057800
scall                   1.736022
rchar                   1.710023
wchar                   1.402594
pgfree                  1.384513
atch                    1.039232
ppgin                   1.559277
pflt                    1.456103
freemem                 1.751148
freeswap                1.733019
runqsz_Not_CPU_Bound    1.132736
dtype: float64
```

TABLE 1.11 : VIF scores based on the final revised model

Now all the VIF scores are very close to 1 which is an indication that we have successfully got rid of the multicollinearity issues. So the Linear regression model will be safe for prediction as well as interpretation.

The predictions of this model were obtained for the training as well as the testing data.

Using the predicted and the actual values, the rsquared values and the rmse scores were obtained for the model for both training and testing set.

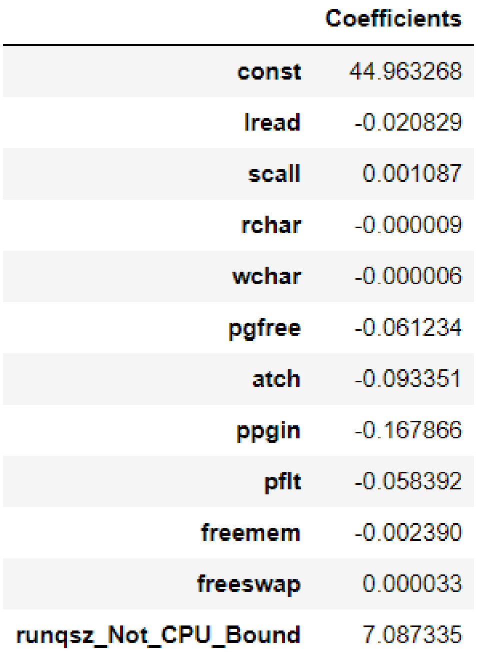
The values are summarised in the following table.

	Training Set	Testing Set
R squared	0.619	0.611
RMSE	11.16	11.91

TABLE 1.12 : R-squared and rmse values for the final model on training and testing data

The above values of the performance metrics suggest that the model, although not a very good one (due to low value of r-squared) is certainly not an overfit model. At least the performance is not going to degrade drastically in production.

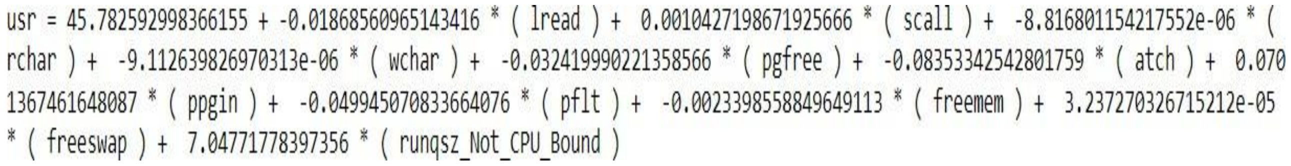
The values of the coefficients for the different parameters can be observed in the table / dataframe given below.



	Coefficients
const	44.963268
lread	-0.020829
scall	0.001087
rchar	-0.000009
wchar	-0.000006
pgfree	-0.061234
atch	-0.093351
ppgin	-0.167866
pflt	-0.058392
freemem	-0.002390
freeswap	0.000033
runqsz_Not_CPU_Bound	7.087335

TABLE 1.13 : Coefficients of independent attributes in the final model

The equation of linear regression is as follows.



```
usr = 45.782592998366155 + -0.01868560965143416 * ( lread ) +  0.0010427198671925666 * ( scall ) +  -8.816801154217552e-06 * (
rchar ) +  -9.112639826970313e-06 * ( wchar ) +  -0.032419990221358566 * ( pgfree ) +  -0.08353342542801759 * ( atch ) +  0.070
1367461648087 * ( ppgin ) +  -0.049945070833664076 * ( pflt ) +  -0.0023398558849649113 * ( freemem ) +  3.237270326715212e-05
* ( freeswap ) +  7.04771778397356 * ( runqsz_Not_CPU_Bound )
```

One problem with this model is that it reveals that none of the predictor variables appear to be a strong predictor of the dependant variable other than ‘runqsz’ . However that interpretation is to a large extent consistent with the findings of the EDA section where we found lack of any strong correlation between any of the independent variables and the target feature ‘usr’.

Inference: Basis on these predictions, what are the business insights and recommendations.

SUMMMARY OF THE VARIOUS STEPS PERFORMED IN THIS PROJECT

- The necessary libraries were loaded followed by the given dataset into the jupyter notebook.
- The data description was performed through both non-visual and visual techniques after the identification of the independent and dependant features, and the datatype of each feature.
- The missing value treatment and the outlier treatment was carried out and then the visual analysis was repeated.
- The main thing that was noticed during this exploratory data analysis was the fact that almost none of the independent features was having a good correlation with the target feature which is a prerequisite for a good linear regression model and there was the presence of multicollinearity also.
- The categorical independent feature was encoded using the one hot encoding method before proceeding with the model building.
- The independent and the dependant variables were separated out and both the sets were further split into the training set and the testing set in the ratio 70:30 with respect to the number of records.
- Initially the linear regression model was built using the scikit learn library and then it was again constructed using the statsmodels library. In both cases the predictions of the model were evaluated in terms of training and testing data and the models were found to be reasonably free from overfitting issues but the values for the evaluation metrics were not indicative of a very good model.
- The presence of multicollinearity was checked by calculating the VIF scores.
- The variables having highest VIF scores were dropped one by one and the effect on the performance parameters of the model were checked. VIF scores were recaulculated.
- The above step was repeated a number of times and the variables having statistically insignificant coefficients (large p values) were also dropped.
- In this way finally we arrived at a reasonably improved version of the model both in terms of prediction and interpretation (free of multicollinearity and statistically insignificant predictor variables) and also reasonably free from overfitting issue.

- The performance metrics were evaluated for the model for both training and testing data and the mathematical equation of the linear regression model was also obtained.

BUSINESS INSIGHTS AND RECOMMENDATIONS

Based on the results of exploratory data analysis and the linear regression model parameters it was observed that the independent features which are provided in the dataset are not really good predictors of the dependant feature ‘usr’. So if the business objective is to know the fraction / percentage of the time for which the a particular system runs in the user mode, better features have to be used. These may be totally new features or formed by suitable combinations of the independent features used in this analysis. But this feature engineering part is largely based on domain knowledge / expertise but is expected to bring the desired results which the current model built on the given data could not yield. Among the independent features used in this analysis only the ‘runqsz’ feature turned out to be a strong predictor of the target feature. Based on the current linear regression model one can say that if the run queue size of the processes is low / non CPU-bound type, the portion of the time the system will run in the user mode will be around 7 % higher than when the run queue size is high / CPU – bound type.

PROBLEM 2: CLASSIFICATION USING CONTRACEPTIVE METHOD DATATSET

Republic of Indonesia Ministry of Health conducted a survey on a sample of 1473 females regarding the prevalence of contraceptive use in the country. The women who were a part of the sample on which the survey was executed, were either not pregnant or were unaware of the pregnancy status (positive / negative) at the time of survey. The demographic and socio economic characteristics of each woman is present in the dataset provided. The objective is to build a classification model based on the given data which will be able to predict whether a new female citizen external to those in this dataset uses any contraceptive method of her choice or avoids using any such contraceptive method, based on her demographic and socio economic characteristics. The relevance of such a model could be in population control. If it is known whether a married female person uses contraceptive methods or not, then those not using any of these methods may be brought under some kind of awareness campaigns and the required knowledge regarding the necessities of using contraceptives may be imparted to them from time to time.

The explanation of each independent feature and also the dependant feature was taken from the data dictionary provided along with the dataset.

The entire analysis was performed in python. The step by step process has been discussed in detail below and we have grouped the steps that have been executed into buckets according to the questions given in the rubrics.

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.

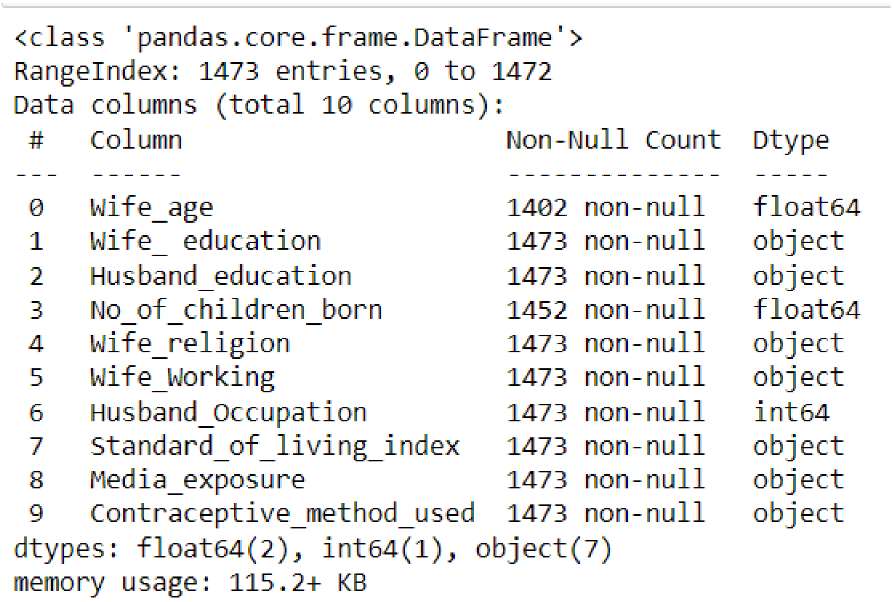
The basic libraries for data manipulation , data visualisation and mathematical computations (pandas, matplotlib and seaborn, numpy respectively) were imported. The other libraries required for model building and other specific tasks were imported later as and when required.

The relevant dataset which was provided in the form of an excel file was imported into the jupyter notebook in the form of a dataframe. The top 5 records and the last 5 records were printed to check whether the data had been loaded properly without any issues or not.

It was observed that the data had been loaded properly.

Thereafter the shape of the dataset was checked and it was found to consist of 1473 rows and 10 columns. Each of the 1473 rows referred to one observation, i.e one married female citizen of Indonesia and each person was described in terms of 10 demographic and socio-economic features or attributes whose names were present along the 10 columns of the dataset. Out of these 10 features the last feature i.e the ‘Contraceptive_method_used’ (Yes / No) was the dependant / target feature and the remaining were the independent / explanatory / predictor variables. This is a binary classification problem since we are interested in the prediction of a categorical variable (‘Contraceptive_method_used’) which has only two values (‘Yes’ and ‘No’) and each of these two values represents a separate class of married female citizens (Yes – married females using contraceptives and No – married females using no contraceptive).

The datatypes of the columns were extracted and it was found that out of the 10 features, 7 features including the target feature (‘Contraceptive_method_used’) belonged to ‘object’ datatype and the remaining three features belonged to the numeric datatype (‘float’ and ‘int’).



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

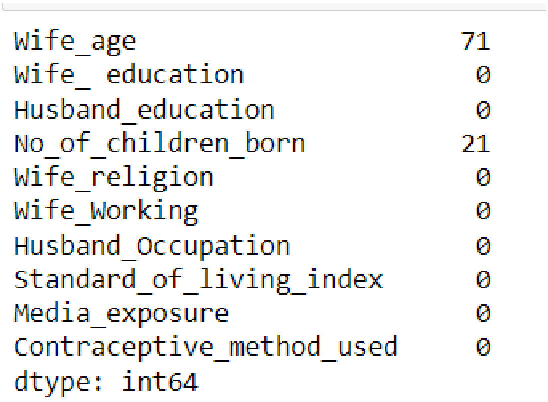
TABLE 2.1 Datatypes of the features in the raw dataset

The datatypes of almost all the features were found to be consistent with the kind of values that populated the feature. Only some issue was found with ‘Husband_Occupation’ column which consists of only four values 1,2,3 and 4. Since the number of unique values in this feature is very limited it is expected to behave like a categorical feature. So for the time being it can be considered as an encoded categorical feature. However it is mentioned in the data dictionary that the assignment of the numbers (1,2,3,4) is totally random and there is no inbuilt hierarchy / natural sequence it is essential to do an one hot encoding of this feature. Therefore it is necessary to convert this feature into one having ‘object’ datatype.

The encoding of the relevant features was done at a later stage. Hence the discussion related to the encoding of the categorical features will be taken up later in this report.

While checking the datatypes of the columns it was found that for some columns (‘Wife_age’ and ‘No_of_children_born’) the non – null count was different from the total number of rows.

Therefore there must be null / missing values in these two columns. The presence of null values was investigated upon and it was found that the above mentioned columns had 71 and 21 missing values respectively. The reason behind the presence of these null values may be the reluctance of disclosing the age and number of children on the end of the people who were surveyed upon.



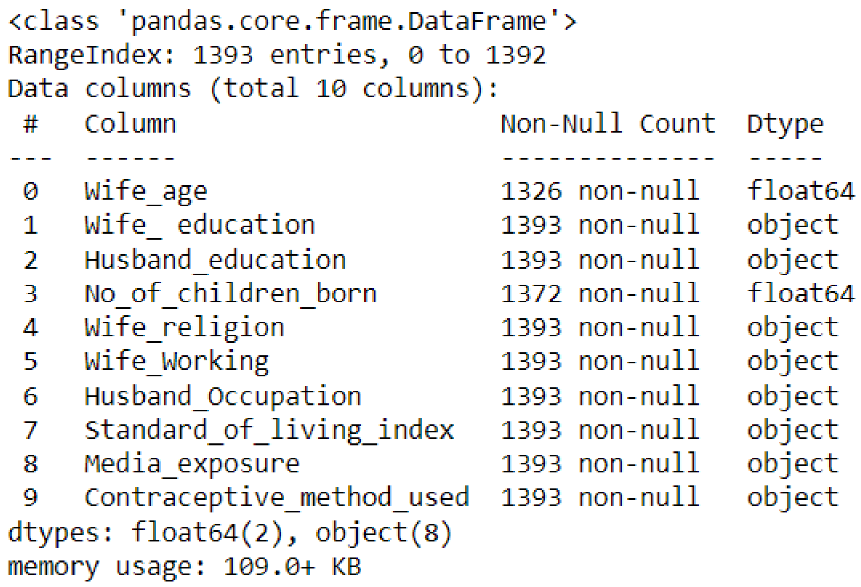
```
Wife_age                   71
Wife_ education             0
Husband_education           0
No_of_children_born        21
Wife_religion               0
Wife_Working                0
Husband_Occupation          0
Standard_of_living_index    0
Media_exposure              0
Contraceptive_method_used 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

TABLE 2.2 Column wise null count in the raw dataset

These features were subjected to null treatment later after the data description stage during the data preprocessing.

The dataset was found to consist of 80 duplicate records. They were dropped from the dataset and the indices of the remaining rows were adjusted accordingly after performing the drop. The dropping of duplicate records was carried out to prevent the classification algorithm from becoming biased towards these records. However this dataset had no feature which could be used as an unique identifier. Had it been there it would have been easier for us to judge whether two records are actually duplicates or are they referring to two different persons with same values for all other attributes except the unique identifier.

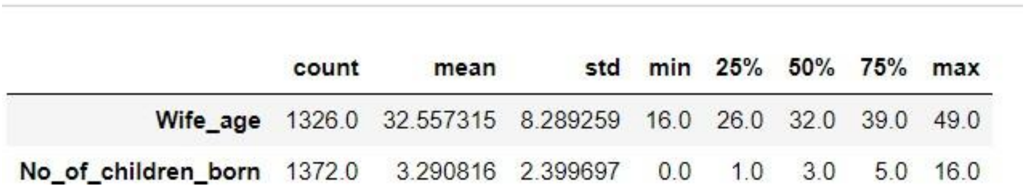
The datatype of the ‘Husband_Occupation’ feature was converted into ‘object’ due to reasons explained previously and the modification was found to have taken effect properly on rechecking the datatypes of all features.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1393 entries, 0 to 1392
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Wife_age                   1326 non-null   float64
 1   Wife_ education            1393 non-null   object 
 2   Husband_education          1393 non-null   object 
 3   No_of_children_born        1372 non-null   float64
 4   Wife_religion              1393 non-null   object 
 5   Wife_Working               1393 non-null   object 
 6   Husband_Occupation         1393 non-null   object 
 7   Standard_of_living_index   1393 non-null   object 
 8   Media_exposure             1393 non-null   object 
 9   Contraceptive_method_used  1393 non-null   object 
dtypes: float64(2), object(8)
memory usage: 109.0+ KB
```

TABLE 2.3 Datatype of the columns after changing the datatype of ‘Husband_Occupation

The descriptive statistical summary of the numerical features was obtained as follows.



	count	mean	std	min	25%	50%	75%	max
Wife_age	1326.0	32.557315	8.289259	16.0	26.0	32.0	39.0	49.0
No_of_children_born	1372.0	3.290816	2.399697	0.0	1.0	3.0	5.0	16.0

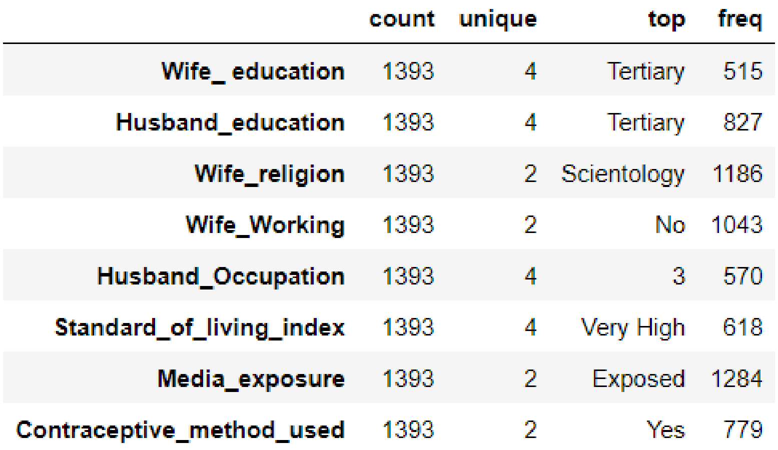
TABLE 2.4 : Statistical Summary of numerical columns

Both the numerical columns ‘Wife_age’ and ‘No_of_children_born’ are having a close to normal distribution due to the proximity of the mean and median values.

The survey was carried out on women with ages in the range 16 – 49 and having no child to a maximum of 16 children.

Visualisation of these results will provide clearer insights.

The descriptive statistical summary of the categorical columns was obtained as follows.



	count	unique	top	freq
Wife_ education	1393	4	Tertiary	515
Husband_education	1393	4	Tertiary	827
Wife_religion	1393	2	Scientology	1186
Wife_Working	1393	2	No	1043
Husband_Occupation	1393	4	3	570
Standard_of_living_index	1393	4	Very High	618
Media_exposure	1393	2	Exposed	1284
Contraceptive_method_used	1393	2	Yes	779

TABLE 2.5 : Statistical summary of the categorical columns

The above table nicely depicts the most frequently encountered categorical level or value of each of the categorical features in the dataset and also the number of occurences of that value (in other words, the mode of each categorical feature).

Most of the females surveyed were well educated (tertiary), the same can be said about their husbands also.

Most women belonged to the Scientology religion and were non-working.

Most of their husbands were occupied with occupation 3 out of the 4 existing husband occupations.

The standard of living was very high for most women and most of them were exposed to media.

Most of them were found to be using contraceptives.

This brings us to and end of data description at least through non visual methods.

After this data visualisation was performed but before that we need to do the data preprocessing.

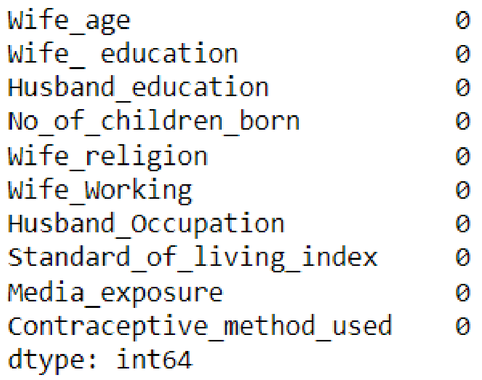
Bad or junk data is not really there in the dataset.

But we have already seen the existence of missing / null values.

So at this point it would be appropriate to do the missing value imputation as a part of the data cleanup or data preprocessing phase.

Since both the features containing missing values (‘Wife_age’ and ‘No_of_children_born’) were numeric features, the null values were treated using the medians of the respective features.

After the missing value imputation the missing value count in all features was found to be 0.



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

TABLE 2.6 : Column wise count of null values after missing value treatment

VISUAL ANALYSIS

UNIVARIATE ANALYSIS OF NUMERICAL FEATURES

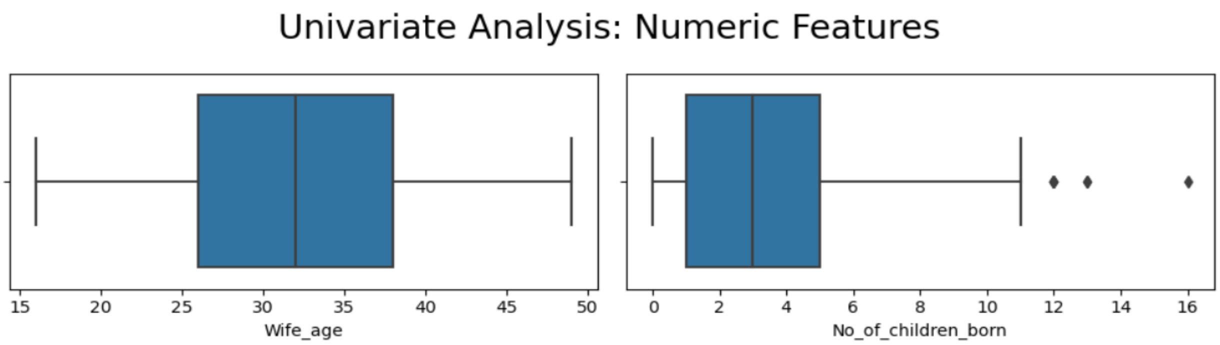


FIGURE 2.1 : Univariate analysis of numerical features (boxplots)

From the above boxplots we can see

- The ages of the females surveyed ranges between 16 – 49 years and the distribution is perfectly symmetric with median / mean around 32 years.
- 50% of the females surveyed had ages in the range 24 -38 years (Interquartile range).
- The number of children ranges from 0 to a maximum of 16.
- The distribution is slightly right skewed (mean = 3.2 > median = 3.0).

- So there is a trend of of high child birth among citizens of this country which may be the reason for this survey. The government may have done this survey as a part of population control drive and for spreading related awareness among the masses.
- There are a few outliers in the number of children feature.
- Detailed discussion will be done on these in the outlier treatment section as a part of the data preparation phase.

UNIVARIATE ANALYSIS OF CATEGORICAL FEATURES

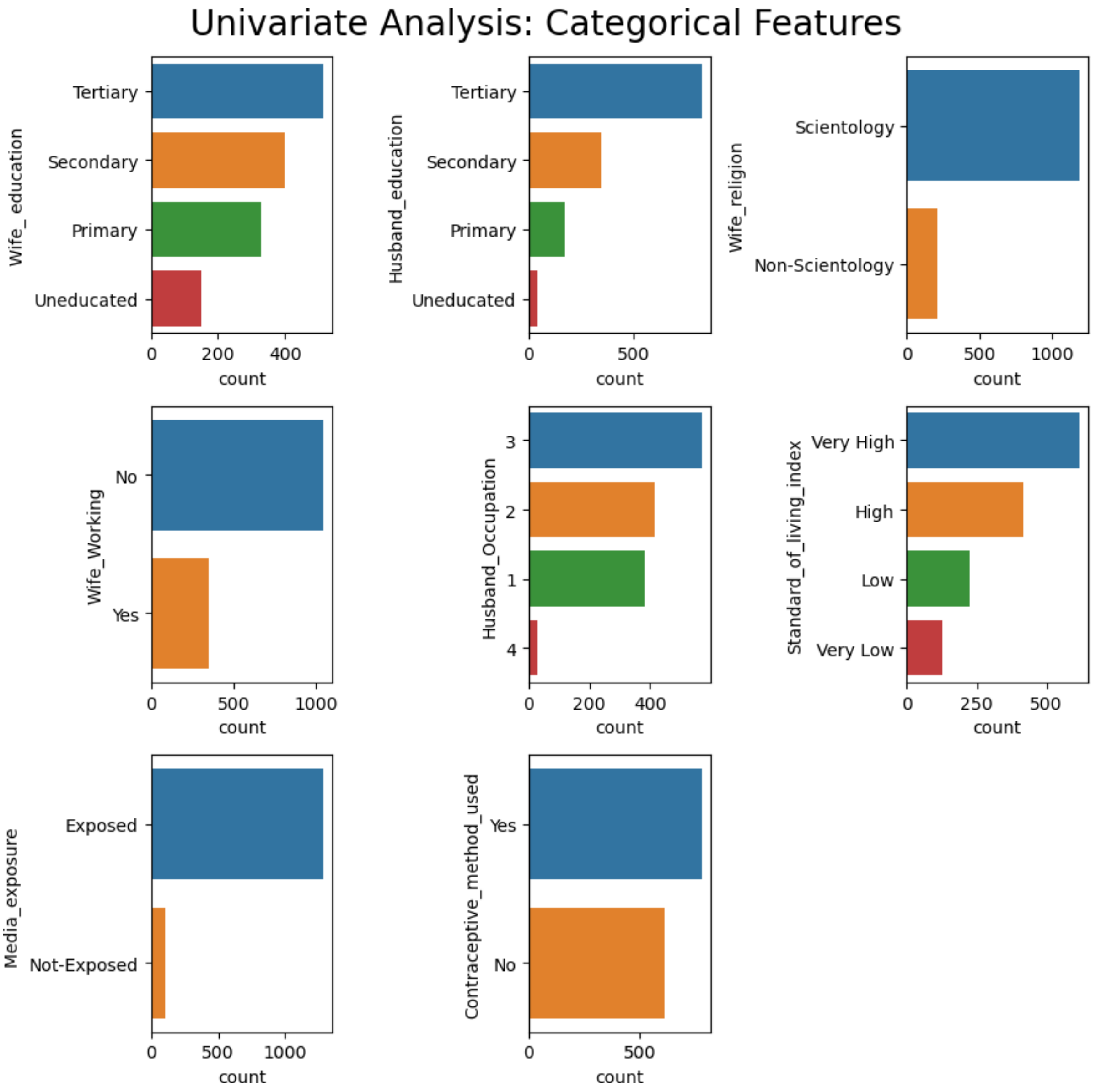


FIGURE 2.2 : Univariate analysis of categorical features (countplot)

The salient observations from this part of the visual analysis are as follows.

- The proportions of women is progressively higher for higher levels of education.
- The same thing is valid for their husbands also.
- Majority of the females belonged to the Scientology religion.
- Majority of the females were non working.
- Out of the four occupations listed, most of the husbands were under occupation 3 and occupation 4 was the least popular.
- As the standard of living increases from very low to very high, the number of females undergoing that standard of living also increases. So most people according to the data have a good life.
- A vast majority of females have access to various media.
- The females using contraceptives slightly outnumber those who do not.

BIVARIATE ANALYSIS OF NUMERIC FEATURES

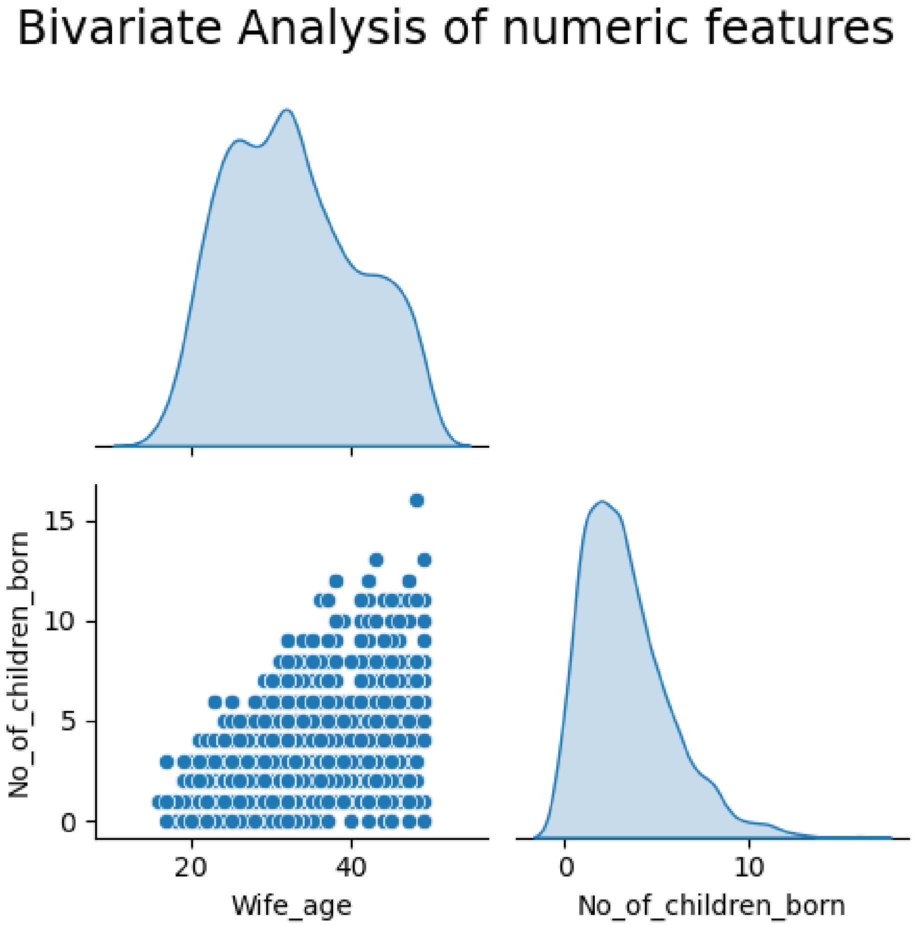


FIGURE 2.3 : Bivariate analysis of numerical features (pairplot)

- The two numeric columns are almost symmetrical.
- Wife age has a slight multimodal appearance which may be due to data collection across two different regions.
- The number of children feature is slightly right skewed with some outliers (genuine or bad data to be discussed later) as already seen from the boxplots.
- The scatterplot between these two features does not show any significant positive or negative correlation which indicates the absence of multicollinearity.
- Let us verify this once again using a correlation matrix or heatmap.

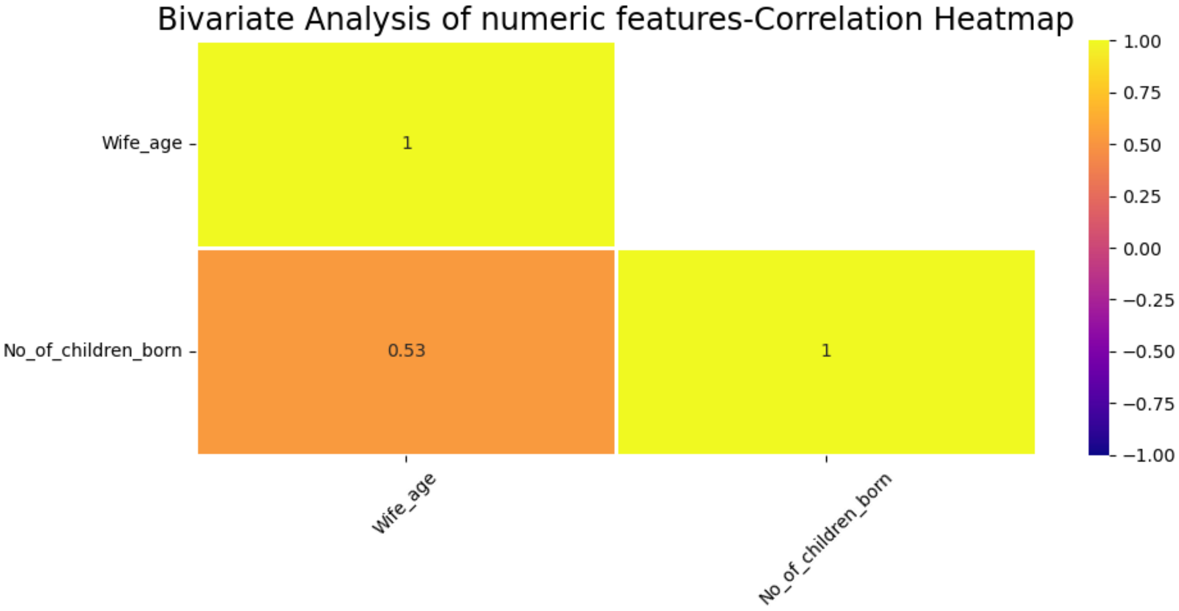


FIGURE 2.4 : Bivariate analysis of numerical features (heatmap)

According to the heatmap, there is some linear positive correlation between the two numeric features but that is not very strong so as to impact the outcomes. It is expected that someone having more number of children will be older than someone with fewer children.

BIVARIATE ANALYSIS [NUMERICAL AND CATEGORICAL FEATURES]

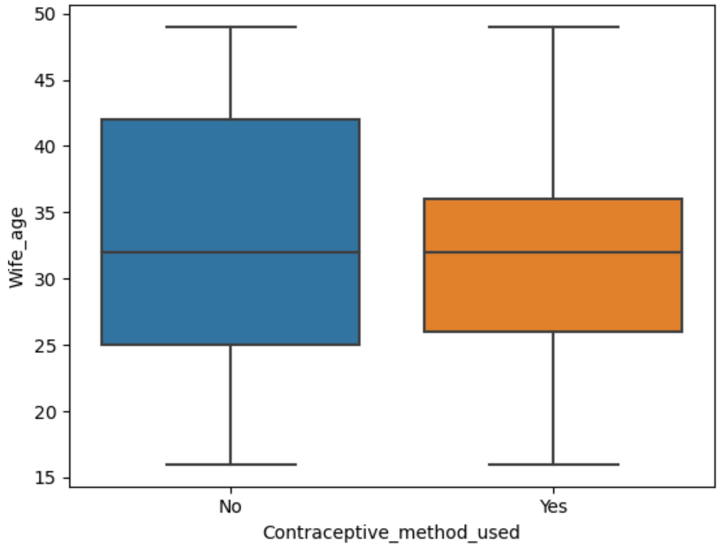


FIGURE 2.5 : Impact of self-age on usage of contraceptives

From the above plot it can be observed that the middle 50% of the people using contraceptives belong to the lower age group compared to the middle 50% of the people not using contraceptives. This may be due to the fact that some people after attaining a certain age feel that the reproductive period of their lives is over and stop using contraceptives. Moreover young people may be more accustomed to the different contraceptive methods than slightly aged people.

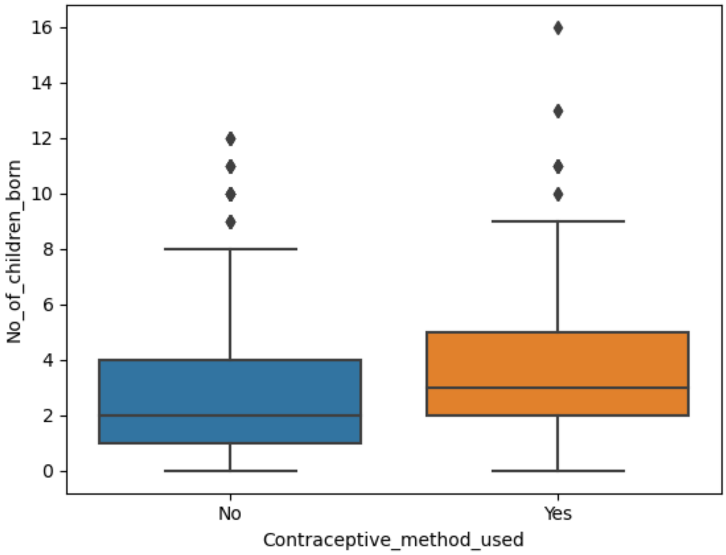


FIGURE 2.6 : Impact of contraceptive usage on child birth count

From the above plot we find a surprising thing that there is a tendancy of giving birth to greater number of children among those who use contraceptives. So people who use contraceptives use it intermittently and refrain from its use whenever they feel like giving birth to another child. This leads us to think that may be in Indonesia the use / lack of use of contraceptives is not the major reason behind the population explosion that the government is trying to address. May be the way of thinking of people is something that needs to be looked into for addressing the problem of over-population. People may also be motivated to use contraceptives on a consistent basis instead of using them intermittently.

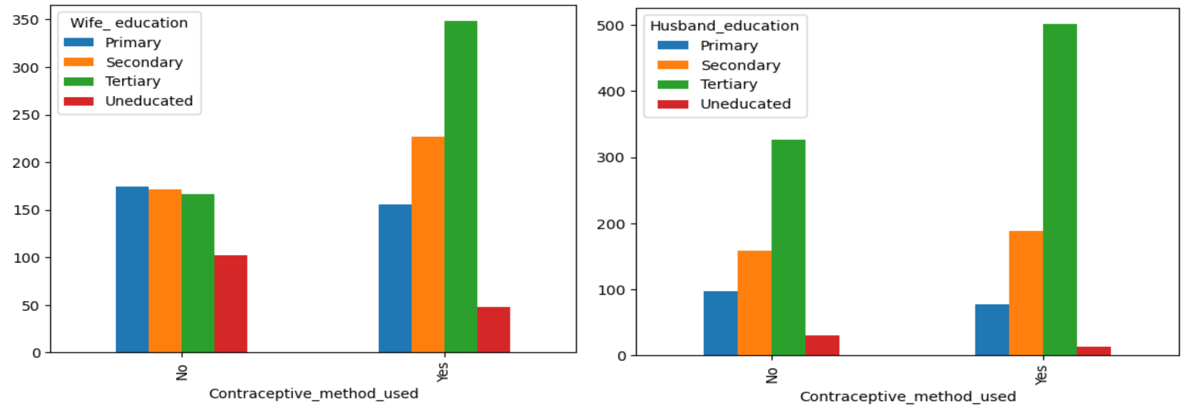


FIGURE 2.7 : Impact of education levels (self and husband) on contraceptive usage

From the above plot it can be observed that females and their husbands who are very highly educated (tertiary) are more accustomed to the use of contraceptives and those who are uneducated tend to refrain from its usage. Therefore this uneducated section of people must be targeted properly during any awareness campaign related to the promotion of use of contraceptives and the government must also look forward to bringing up the general education level of the citizens.

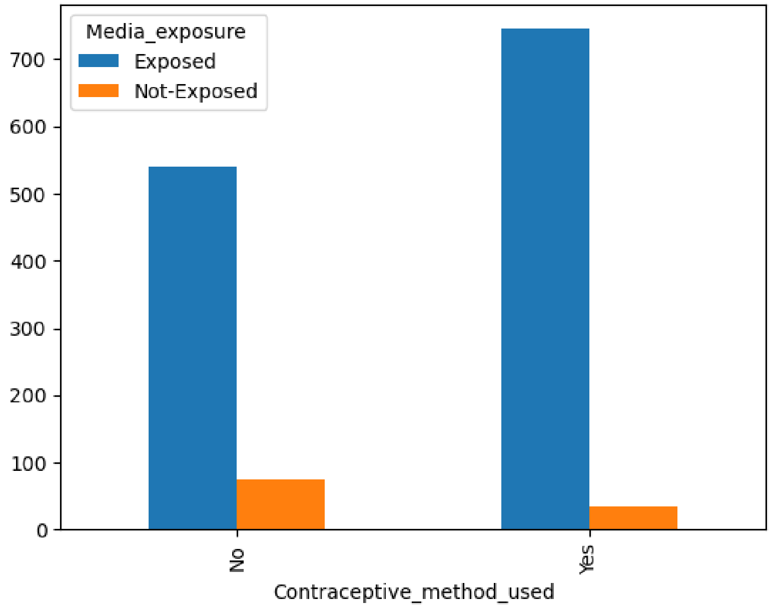


FIGURE 2.8 : Impact of media exposure on contraceptive usage

The media exposure impacts the use of contraceptives positively. Therefore the government must look forward to make the media easily accessible to more and more people.

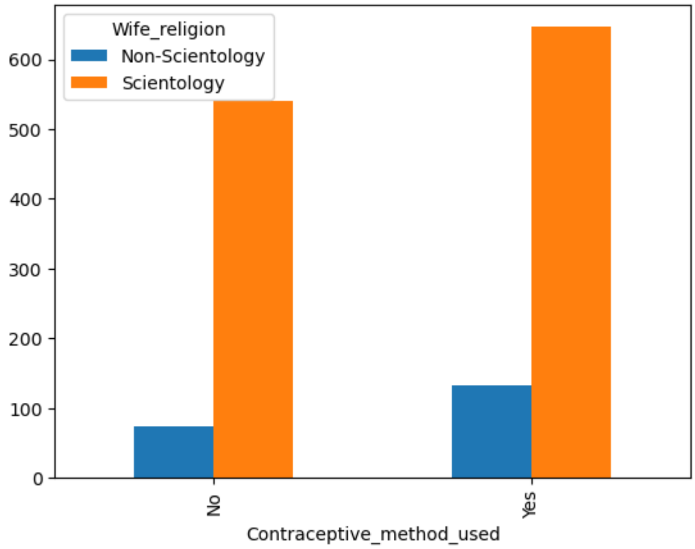


FIGURE 2.9 : Impact of religion on contraceptive usage

It can be observed that religion does not impact the use of contraceptives significantly. But there are indications that contraceptives are more popular among scientologists.

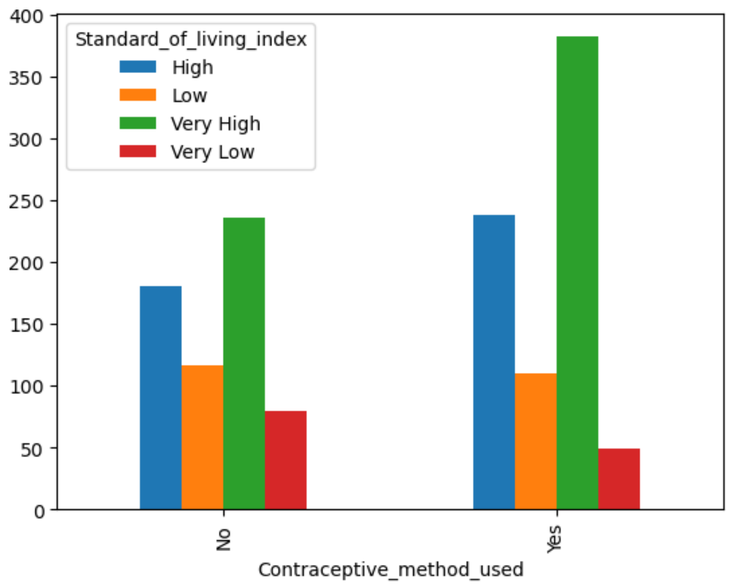
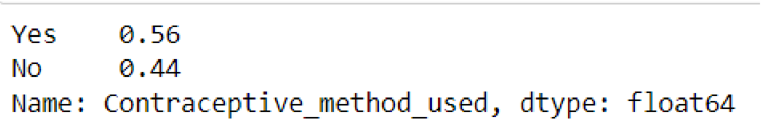


FIGURE 2.10 : Impact of standard of living on contraceptive usage

The people with very high and high standard of living are mostly the users of contraceptives whereas those with a very low standard of living are more prone to not using contraceptives. This may be the outcome of lack of awareness among the poor.

TARGET FEATURE CLASS PROPORTIONS

In this section we shall see how many records are there in each of the two classes and also decide the class label for each class and the appropriate metric for evaluating the classification models to which the data will be subjected.



```
Yes     0.56
No      0.44
Name: Contraceptive_method_used, dtype: float64
```

TABLE 2.7 : Target Feature class proportions

There are two classes. One consists of those women who use contraceptives and the other consists of those who don’t. Now it is expected that the objective of such a survey will be to identify the non-users of contraceptives among married women based on their demographic and socio-economic characteristics and once it is done, some targeted campaigning can be done to spread the awareness among these people and motivate them towards the usage of contraceptives and thus address the issue of population outburst. People belonging to the other class, i.e users of contraceptives may also be involved in such campaigns to share their views regarding the benefits of contraceptives.

Now the model is being prepared to identify the class which does not use contraceptives and therefore it is the class of interest. Moreover we can see from the above python output that it is the underrepresented class out of the two (although by a small margin). Keeping in mind these points, we assign the label of **positive class (class 1)** to the **non users of contraceptives** and the label of **negative class (class 0)** to the **users of contraceptives**.

Now this dataset is definitely not one where there is a huge class imbalance. So accuracy cannot be ruled out totally as a suitable metric for model evaluation. Now let us see out of the two possible misclassifications which one is costlier. According to this analysis the False Negative (Type II Error) is more costlier than False Positives (Type I Error). This is because if a non user of contraceptive is wrongly classified as user of contraceptive (False Negative) by the model then she will not be a part of the awareness campaign and may be will continue to stay away from contraceptives and contribute towards the exponentially growing population of Indonesia. On the other hand if a user of contraceptive is wrongly classified as non user of contraceptive (False Positive) by the model, the worst that can happen is that she may receive an invitation for participating in the awareness campaign which she can simply decline and continue with her conjugal life which is not detrimental for the population count.

So the objective will be to build a classification model which will minimise False Negatives and therefore maximise the Recall Score for the positive class across both training and testing data. This conclusion will be used while comparing different models and selecting the best for the given situation.

OUTLIER DETECTION AND TREATMENT

Let us start with the boxplots of the two numerical features in the given dataset.

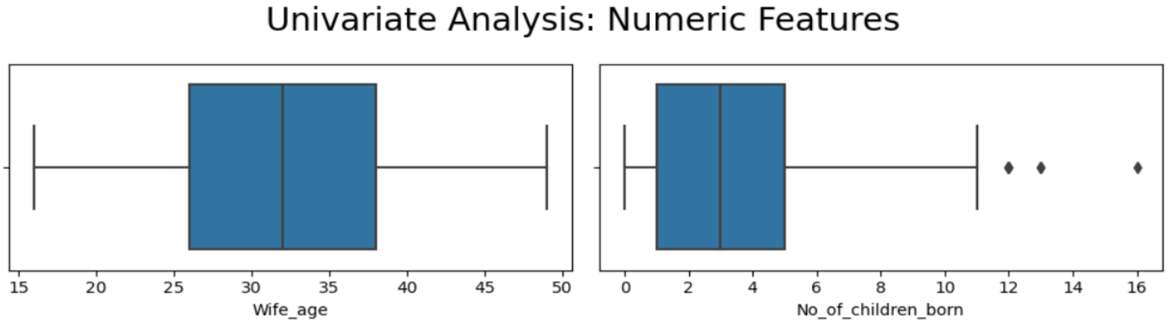


FIGURE 2.11 : Boxplots of numerical features before outlier treatment

Some outliers were spotted in the number of children feature.

Now these are the records of females with more than 11 children.

The number of such records were found to be only 0.5% (see attached code file) of the total number of records which could tempt us to leave them as they are thereby allowing the data to remain unchanged and allow the records with very high number of children to impact the model outcome.

But we decided to go ahead with the outlier treatment due to the following reasons.

Since some classification models use linear regression under the hood, and since linear regression is highly impacted by outliers, it is better to get rid of them since their total number is a very small fraction of total number of records in the entire dataset. We don’t want the best fit line (input of the sigmoid function in logistic regression) to change because of one or two extreme values.

When the records corresponding to these outliers were looked into, there were some indications that this data may not be genuine (e.g a person at the age of 38 is the mother of 12 children, someone with good media exposure and who is an user of contraceptives is mother of 13 children, the highest number of children belongs to someone who is an user of contraceptives etc.)

The outlier treatment was done by the traditional method of capping them to the upper bound. There was no outlier on the lower end.

The boxplots obtained after outlier treatment was used as an evidence of its correct execution.

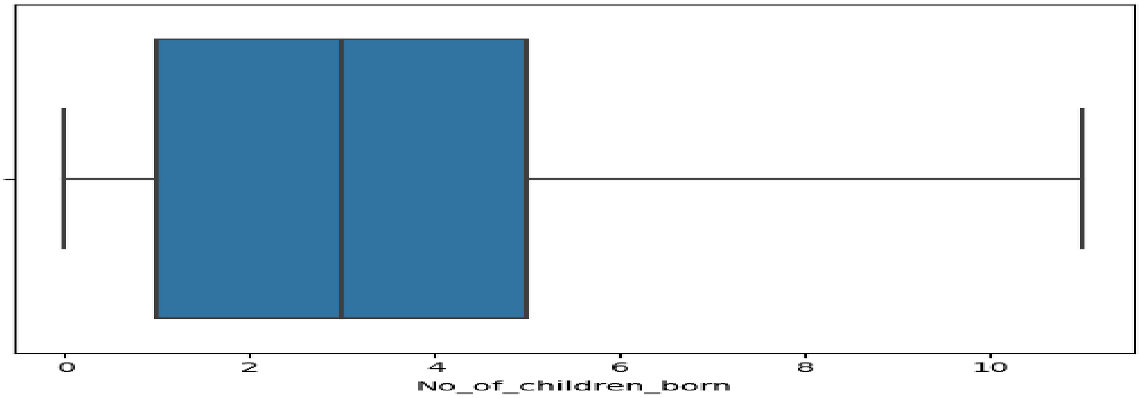


FIGURE 2.12 : Effect of outlier treatment on number of children born column

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.

ENCODING OF CATEGORICAL FEATURES

The values in some of the categorical features had a natural sequence / inbuilt hierarchy. To protect that the features 'Wife_education', 'Husband_education' and 'Standard_of_living_index' were encoded by using the ordinal encoding (done manually) and highest numerical index was provided to the value expected to occupy the highest position in the hierarchy.

The target feature was also manually encoded by assigning the label 1 to the positive class (non-users of contraceptives) and the label 0 to the negative class (users of contraceptives).

Rest of the categorical features were encoded using the one hot encoding and by setting the drop_first parameter to True. The increase in number of columns in the dataset due to one hot encoding was insignificant.

APPLYING LOGISTIC REGRESSION (CLASSIFIER 1)

The dataframe was divided into two subsets, one containing the independent features and the other containing the dependant feature.

The dataset (both independent and dependent features) was divided into two sets, viz the training set and the testing set in the ratio 70:30. The class proportions of the target variable (56% and 44% for positive and negative classes respectively) was maintained in the training and testing sets by using 'stratify' parameter and setting its value as the variable storing the target feature.

The logistic regression, linear discriminant analysis (LDA) and Decision Tree (CART) algorithm was applied on the dataset using the scikit learn library and the same training set was used to train all the models.

Two variants of the logistic regression classifier and decision tree were applied, one with the default values of the parameters and the other with the best possible values obtained by grid search cross validation technique.

Therefore the total number of classifiers to be compared is 5.

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.

LOGISTIC REGRESSION MODEL EVALUATION AND MODIFICATION

The predictions of the logistic regression model were obtained for the training dataset and the testing dataset.

Based on these predictions the confusion matrix of the model based on the training dataset is as follows.

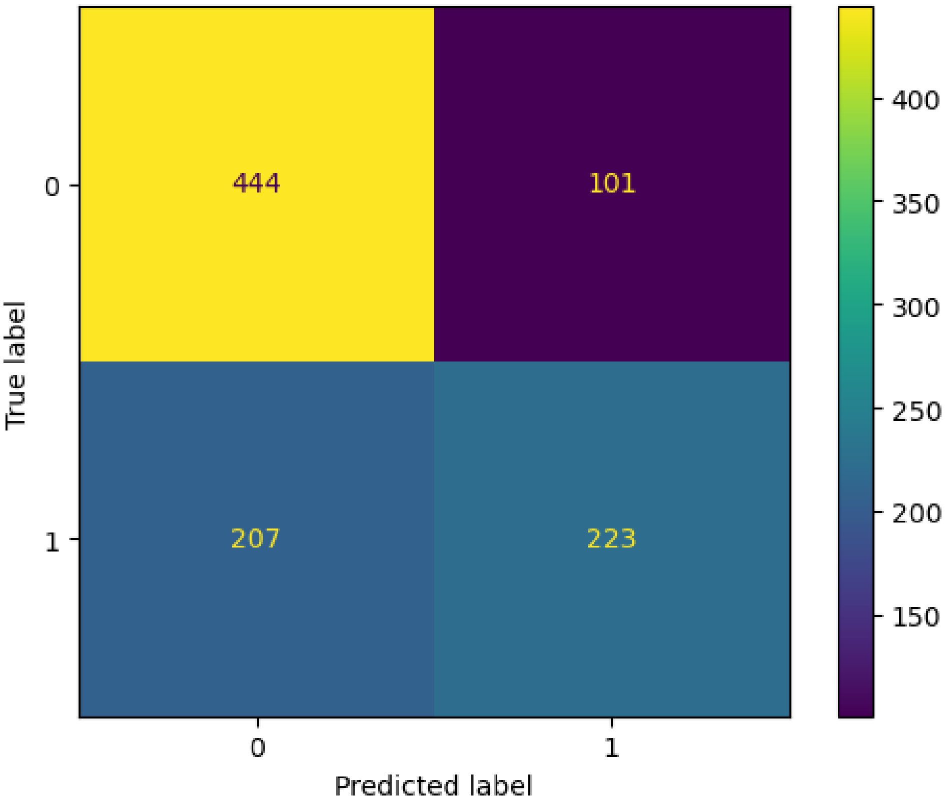
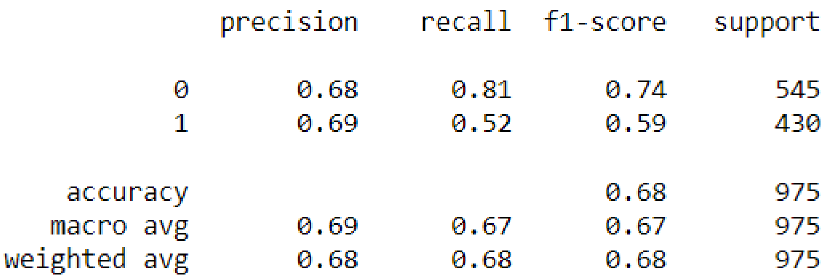


FIGURE 2.13 : Confusion matrix for default logistic regression model for training data

Based on the values in the confusion matrix, the classification report of the model for the training dataset was printed as follows.



```
              precision    recall  f1-score   support

           0       0.68      0.81      0.74       545
           1       0.69      0.52      0.59       430

    accuracy                           0.68       975
   macro avg       0.69      0.67      0.67       975
weighted avg       0.68      0.68      0.68       975
```

TABLE 2.8 : Classification report for default logistic regression model on training data

Let us also look at the confusion matrix and the classification report for the predictions of the logistic regression model for the testing data.

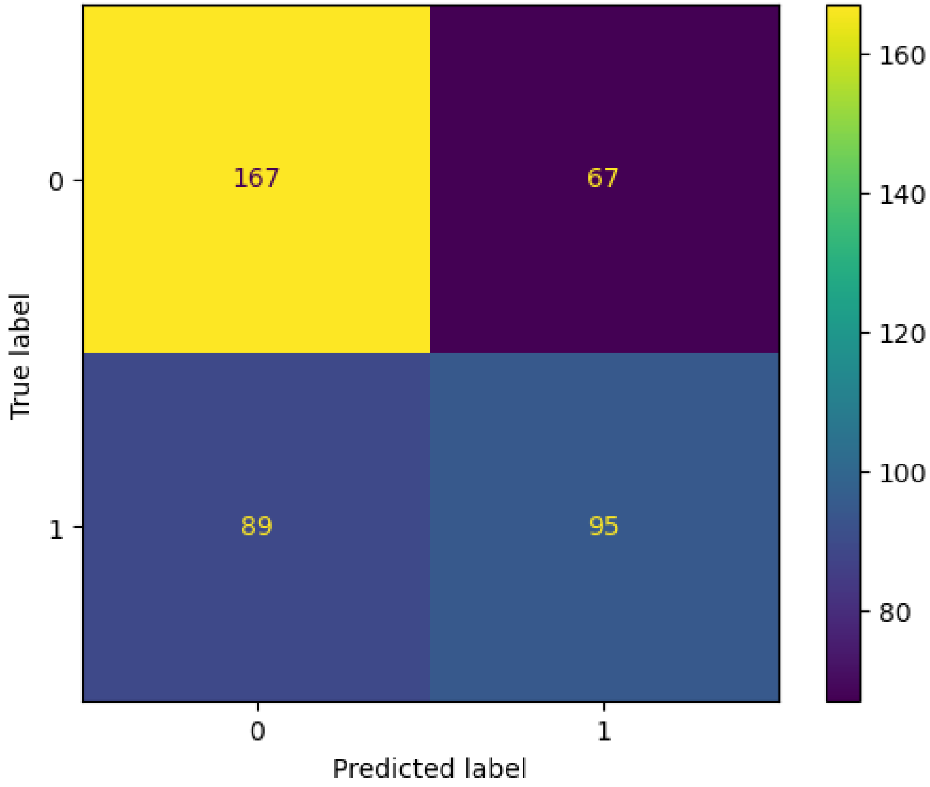
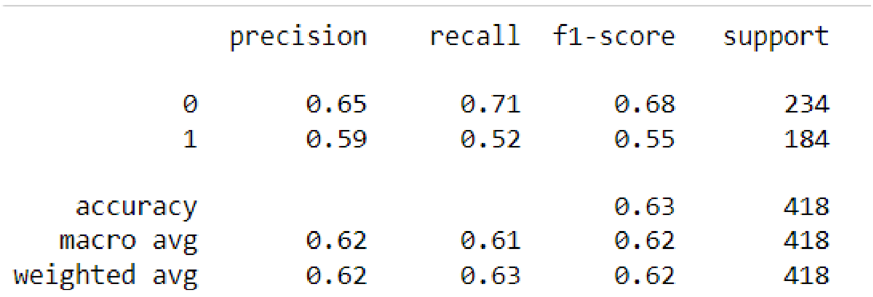


FIGURE 2.14 : Confusion matrix for default logistic regression model for testing data



```
              precision    recall  f1-score   support

           0       0.65      0.71      0.68       234
           1       0.59      0.52      0.55       184

    accuracy                           0.63       418
   macro avg       0.62      0.61      0.62       418
weighted avg       0.62      0.63      0.62       418
```

TABLE 2.9 : Classification report for default logistic regression model on testing data

We have already selected along with justification the ‘Recall’ score for the positive(1) class as the evaluation metric for the classification models applicable for the given data.

We can see that for both the training and the testing data, the Recall scores for the positive class are equal. This means that the model is not an overfit one but not a good predictor of the class because it can identify only 52 % of the non-users of contraceptive as non-users of contraceptive.

The AUC score for the model was obtained as 0.705 for the training dataset and 0.700 for the test dataset.

Both the values are close indicating absence of overfit and since the value lies in the range 0.7<AUC<0.8, the model can be referred to as an acceptable discriminator based on the AUC score.

The ROC Curves for the training dataset and testing dataset were as follows

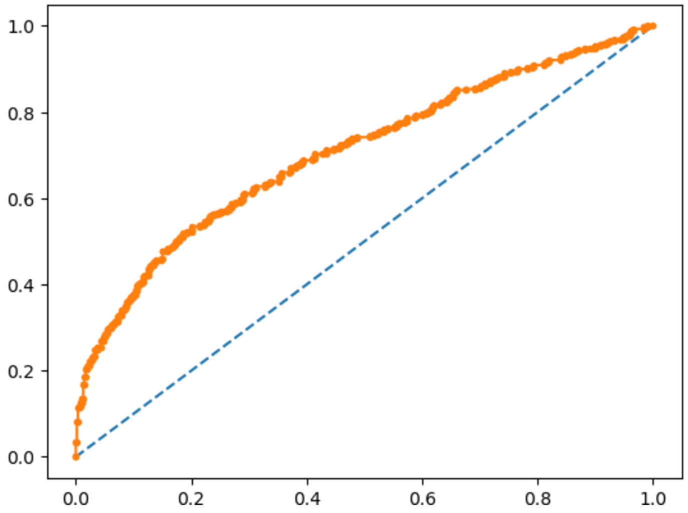


FIGURE 2.15 : ROC curve for default logistic regression model for training data

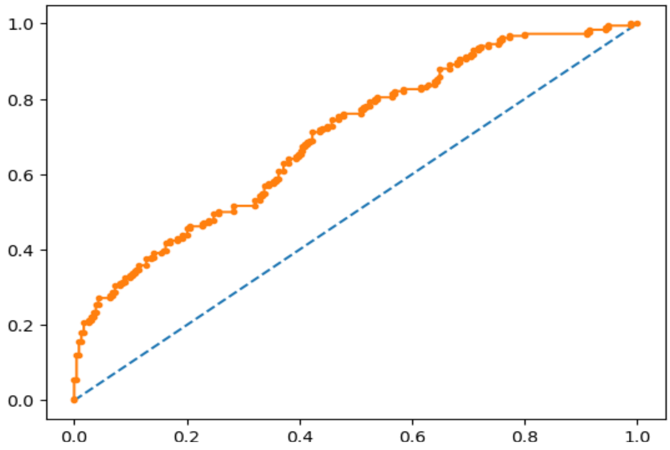


FIGURE 2.16 : ROC curve for default logistic regression model for testing data

The ROC curves for the model on both the training and testing data indicate that the model is better than a random classifier but not anywhere close to the best model.

LOGISTIC REGRESSION MODEL WITH HYPERPARAMETER TUNING USING GRID SEARCH CROSS VALIDATION

The previous logistic regression model was built using the default values of the different available parameters. Now we will evaluate another logistic regression model which was built using the best values of some of the parameters available and this was backed up by grid search with 5 fold cross validation. The parameters of the best model as determined by the grid search are as follows:

l1_ratio = 0.25

penalty = l2

solver = lbfgs

tol = 0.0001

The predictions of the best model were obtained using the training and the testing data and then the confusion matrix for the predictions on the training data was obtained as follows.

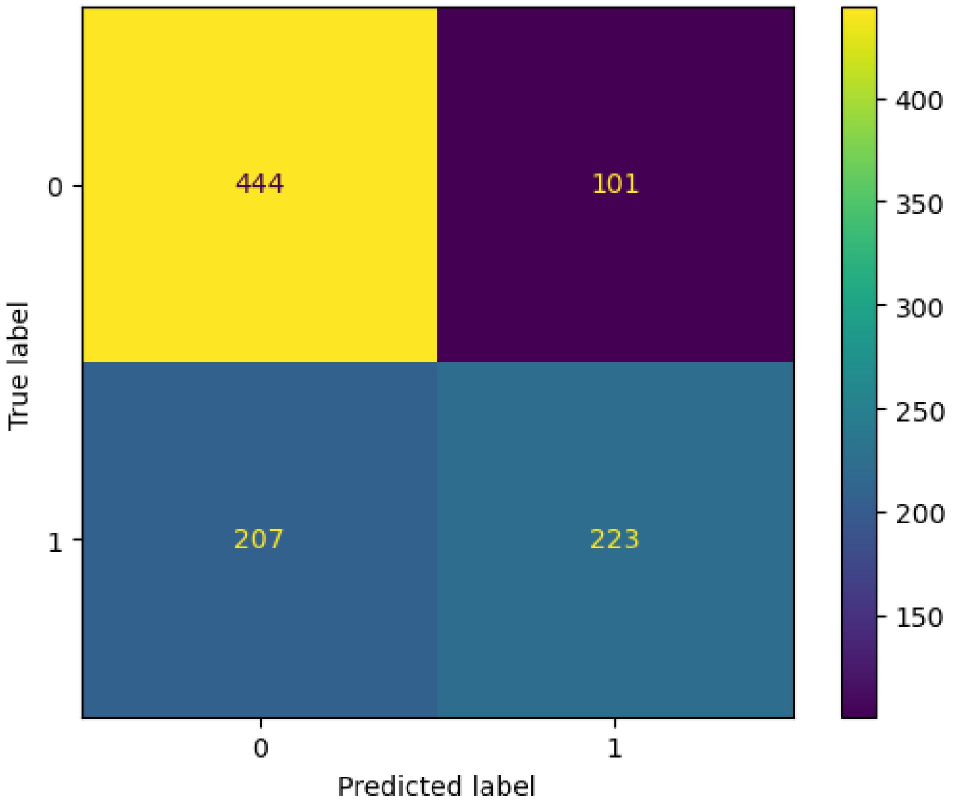
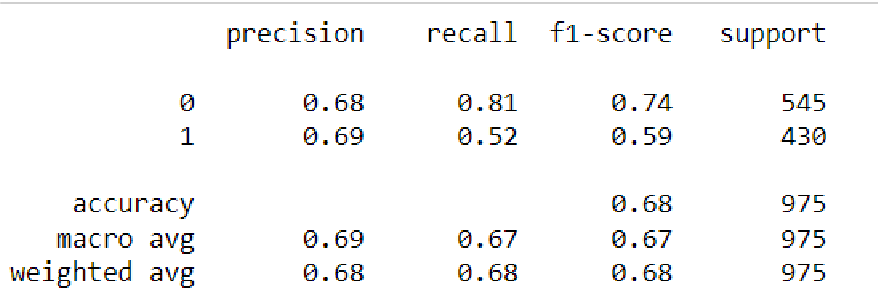


FIGURE 2.17 : Confusion matrix for hyperparameter tuned logistic regression model for training data

The classification report based on the training data for the best model was obtained as follows.



```
              precision    recall  f1-score   support

           0       0.68      0.81      0.74       545
           1       0.69      0.52      0.59       430

    accuracy                           0.68       975
   macro avg       0.69      0.67      0.67       975
weighted avg       0.68      0.68      0.68       975
```

TABLE 2.10 : Classification report for hyperparameter tuned logistic regression model on training data

The confusion matrix for the best model based on the testing data was obtained as follows.

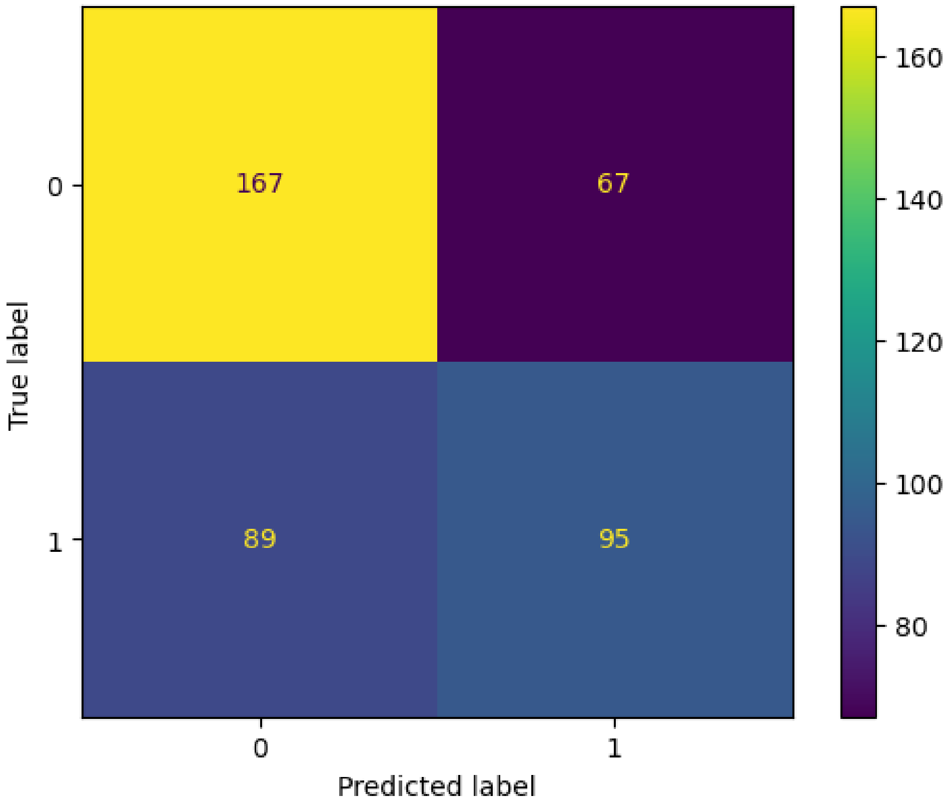
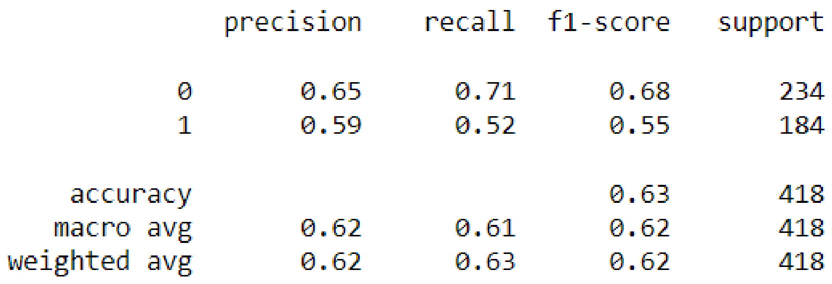


FIGURE 2.18 : Confusion matrix for hyperparameter tuned logistic regression model for testing data

The classification report based on the testing data was obtained as follows.



```
              precision    recall  f1-score   support

           0       0.65      0.71      0.68       234
           1       0.59      0.52      0.55       184

    accuracy                           0.63       418
   macro avg       0.62      0.61      0.62       418
weighted avg       0.62      0.63      0.62       418
```

TABLE 2.11 : Classification report for hyperparameter tuned logistic regression model on testing data

The classification reports, particularly the recall scores suggest that the model with tuned hyperparameters is not any improved version of the model with default parameters in terms of prediction performance. This shows that the default parameters are such that the model usually runs well on them and may be that is the reason why they are selected as the default parameters.

Let us also look at the ROC curve and the AUC scores for the best model.

The AUC scores of the best model on the training data and the testing data are 0.705 and 0.700 respectively which are same as that of the default model and therefore this model is also an acceptable classifier and not an overfit one as suggested by AUC score.

The ROC curve of the best model on the training data is as follows.

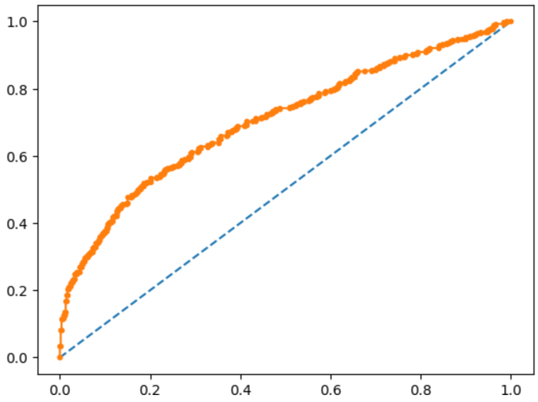


FIGURE 2.19 : ROC curve for hyperparameter tuned logistic regression model for training data

The ROC curve of the best model on the testing data is as follows:

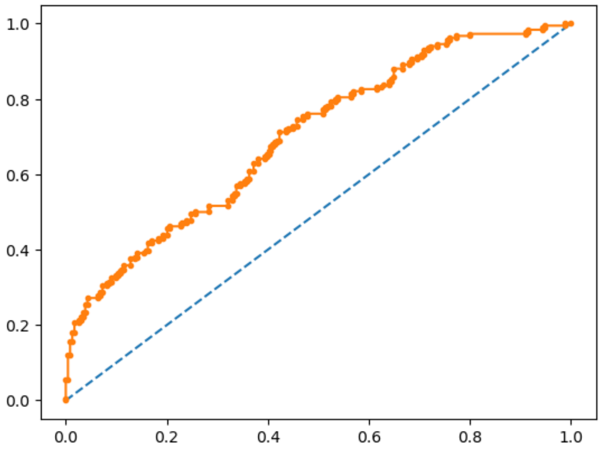


FIGURE 2.20 : ROC curve for hyperparameter tuned logistic regression model for testing data

The ROC curves for the model on both the training and testing data indicate that the model is better than a random classifier but not anywhere close to the best possible model.

LINEAR DISCRIMINANT ANALYSIS MODEL EVALUATION

After training the linear discriminant analysis model using the same training set as that used for the logistic regression classifier, the predictions of the model were obtained on training and testing datasets.

Based on these predictions the confusion matrix of the LDA model on the training dataset was obtained as follows.

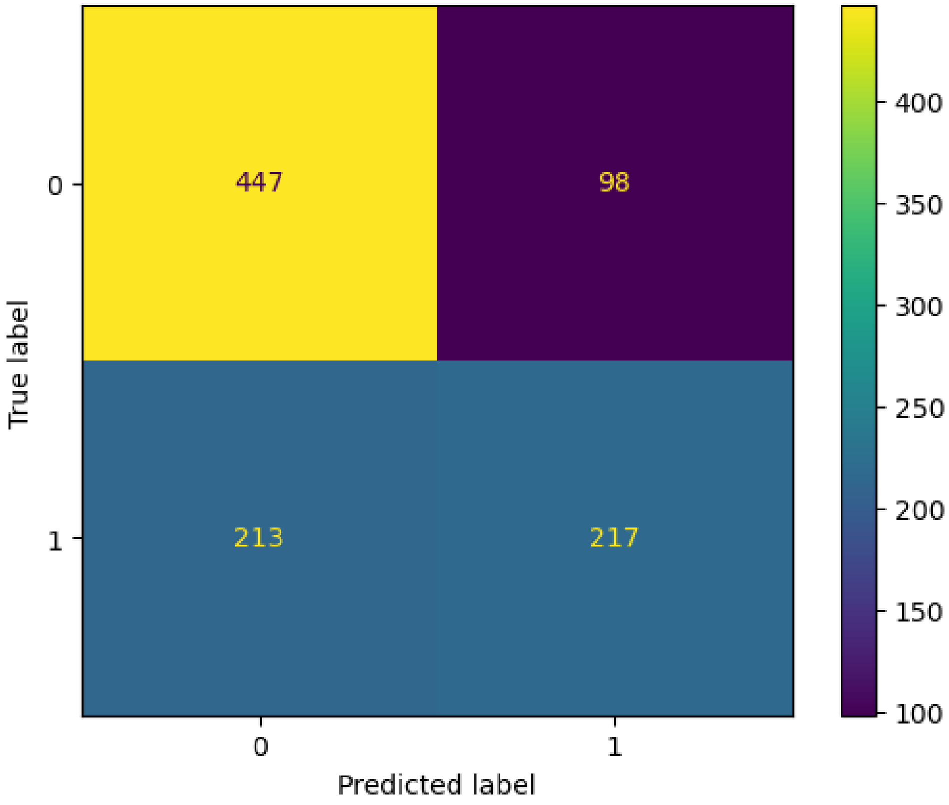


FIGURE 2.21 : Confusion matrix for LDA model for training data

The confusion matrix of the LDA model based on its predictions of the testing data was obtained as follows.

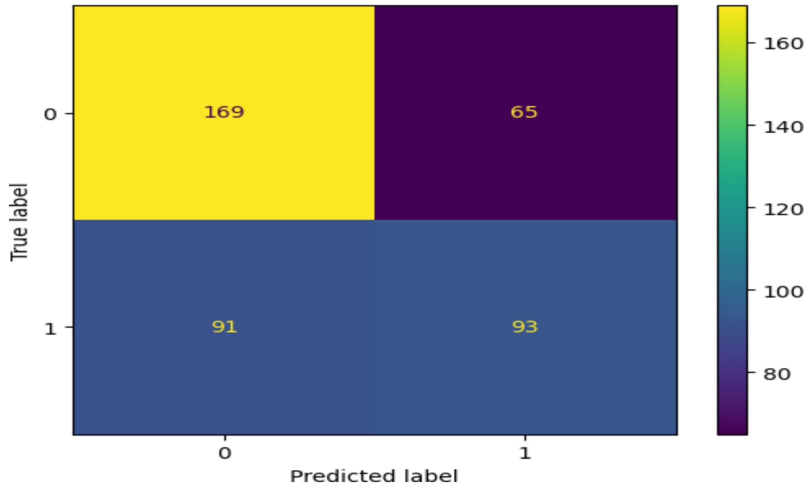
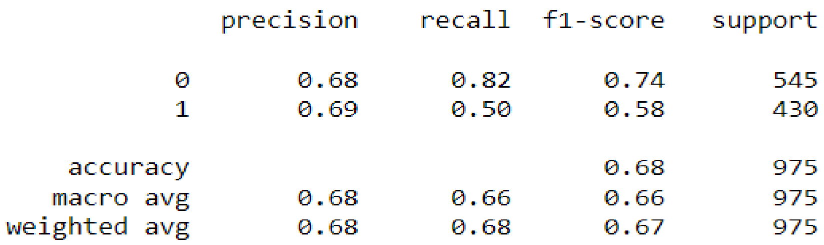


FIGURE 2.22 : Confusion matrix for LDA model for testing data

Based on the above confusion matrix, the classification report of the LDA model was obtained for the training dataset as follows.



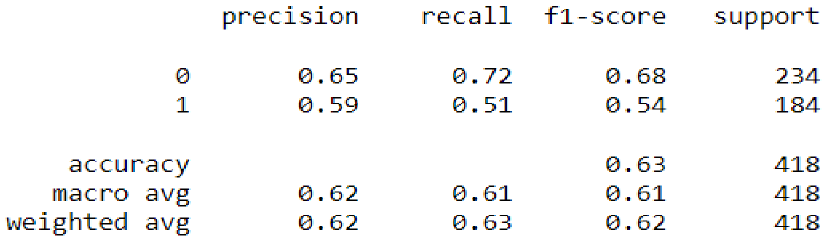
```
              precision    recall  f1-score   support

           0       0.68      0.82      0.74       545
           1       0.69      0.50      0.58       430

    accuracy                           0.68       975
   macro avg       0.68      0.66      0.66       975
weighted avg       0.68      0.68      0.67       975
```

TABLE 2.12 : Classification report for LDA model on training data

The classification matrix for the LDA model for the testing data was obtained as follows.



```
              precision    recall  f1-score   support

           0       0.65      0.72      0.68       234
           1       0.59      0.51      0.54       184

    accuracy                           0.63       418
   macro avg       0.62      0.61      0.61       418
weighted avg       0.62      0.63      0.62       418
```

TABLE 2.13 : Classification report for LDA model on testing data

The recall scores for the positive class suggest that the model is even worse than the logistic regression model but it is also not an overfit model.

The AUC scores of the LDA model on the training data and the testing data were 0.7056 and 0.699 which are slightly lower than those obtained for the logistic regression model which shows that this model will be definitely ruled out on being compared with any of the two logistic regression models.

The ROC curve for the LDA model on the training data is as follows.

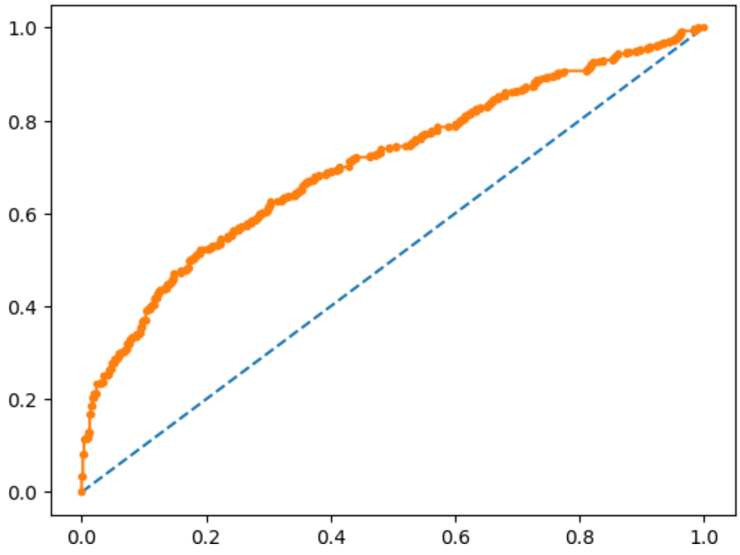


FIGURE 2.23 : ROC curve for LDA model for training data

The ROC curve of the LDA model based on the testing data is as follows.

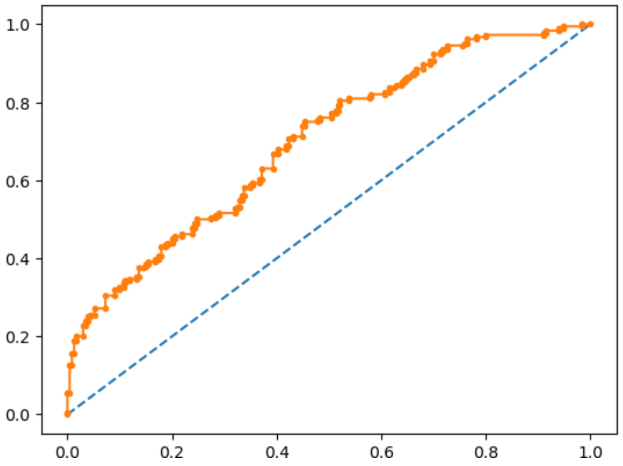


FIGURE 2.24 : ROC curve for LDA model for testing data

DECISION TREE CLASSIFIER MODEL EVALUATION

After training the linear discriminant analysis model with default values of all the parameters using the same training set as that used for the logistic regression classifier and LDA, the decision tree was visualized.

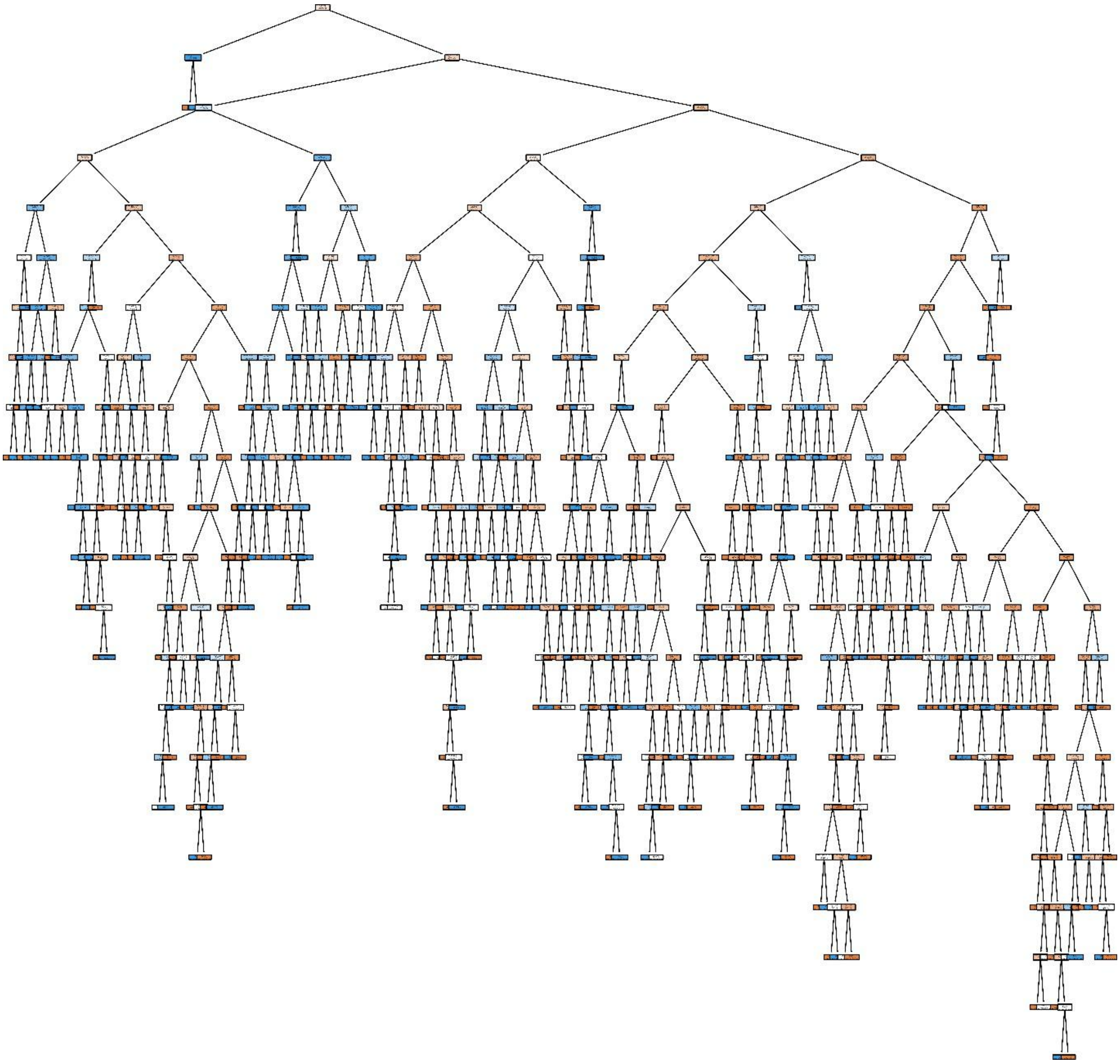


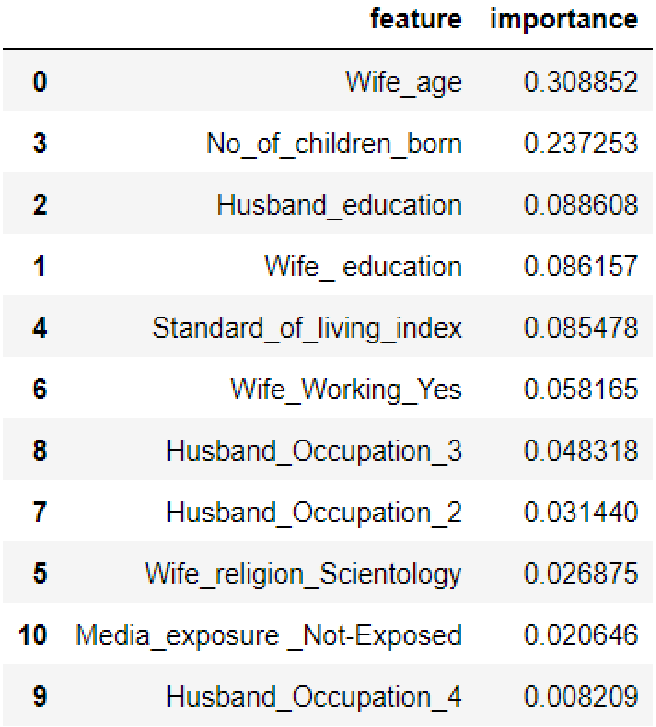
FIGURE 2.25 : Default (unrestricted) decision tree structure

Clearly it seems like an overfit overgrown tree which needs pruning / regularization.

However the predictions of this model were obtained on the training and testing data and based on those predictions the confusion matrices were obtained and the classification reports were printed.

FEATURE IMPORTANCES

Before having a look at the mentioned metrics let us also have a look at the relative importances of the different attributes in predicting the value / class of the target feature through the ‘feature importance’ facility which is an usp of the decision tree classifier.



	feature	importance
0	Wife_age	0.308852
3	No_of_children_born	0.237253
2	Husband_education	0.088608
1	Wife_ education	0.086157
4	Standard_of_living_index	0.085478
6	Wife_Working_Yes	0.058165
8	Husband_Occupation_3	0.048318
7	Husband_Occupation_2	0.031440
5	Wife_religion_Scientology	0.026875
10	Media_exposure _Not-Exposed	0.020646
9	Husband_Occupation_4	0.008209

TABLE 2.14 : Relative feature importances according to the decision tree algorithm

It can be observed that the features ‘Wife_age’ and ‘No_of_children_born’ together play the most crucial role in predicting whether the particular female is an user of contraceptive or not.

These two variables taken together has more than 50 % of the total predictive ability of all the independent features taken together.

The confusion matrix for the model based on the training dataset is as follows.

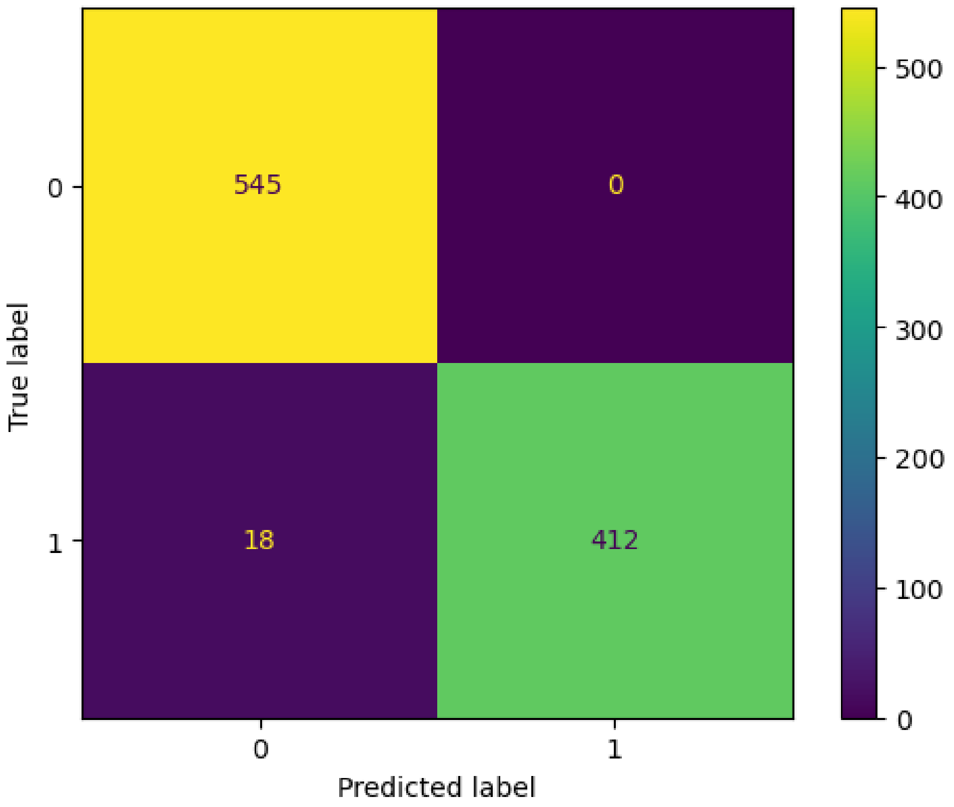
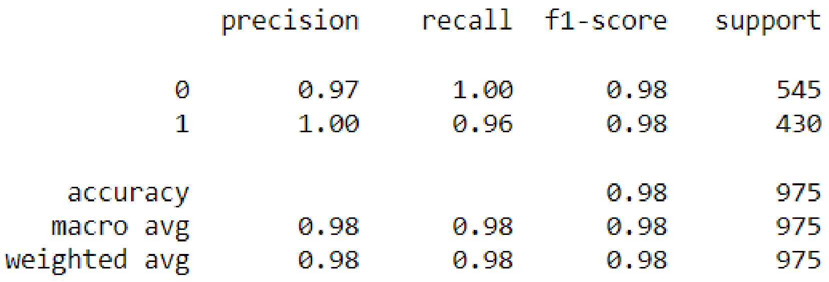


FIGURE 2.26 : Confusion matrix for default decision tree model for training data

The classification report for the model’s predictions based on the training dataset is as follows.



```
              precision    recall  f1-score   support

           0       0.97      1.00      0.98       545
           1       1.00      0.96      0.98       430

    accuracy                           0.98       975
   macro avg       0.98      0.98      0.98       975
weighted avg       0.98      0.98      0.98       975
```

TABLE 2.15 : Classification report for default decision tree classifier model on training data

The confusion matrix for the model based on the testing data is as follows.

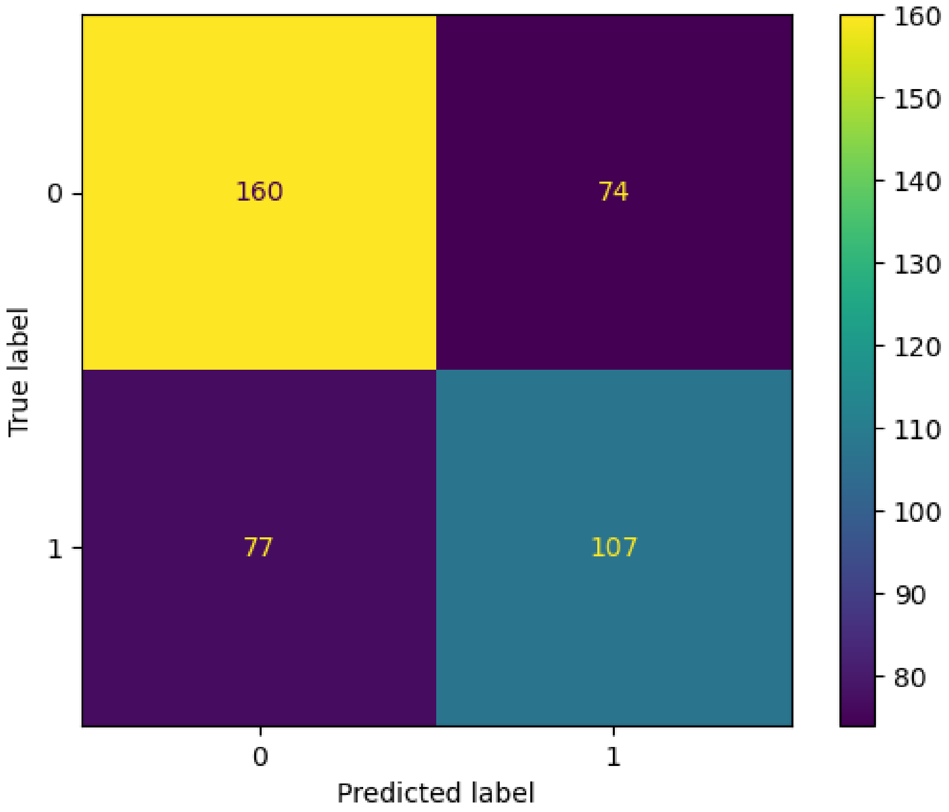
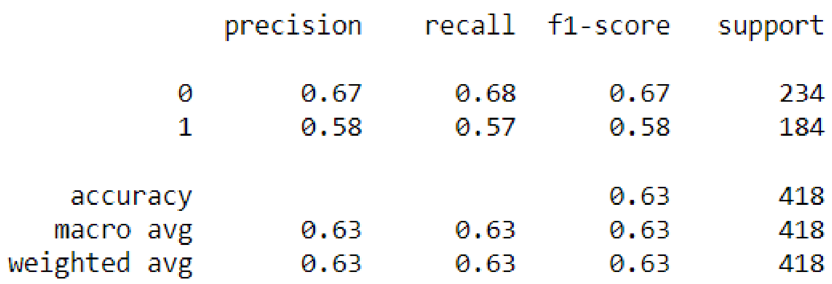


FIGURE 2.27 : Confusion matrix for default decision tree model for testing data

The classification report for the model's predictions based on the testing dataset is as follows.



```
              precision    recall  f1-score   support

           0       0.67      0.68      0.67       234
           1       0.58      0.57      0.58       184

    accuracy                           0.63       418
   macro avg       0.63      0.63      0.63       418
weighted avg       0.63      0.63      0.63       418
```

TABLE 2.16 : Classification report for default decision tree classifier model on testing data

Although the recall score for the positive class for the model is extremely high for the training data the model is an overfit model because it can be seen that there is a drastic drop in recall score for the positive class when the model is applied to data that was not exposed to it while it was being trained. So this overgrown decision tree model will not generalise well in production. However this decision tree model is giving the highest recall score (which itself is a low value) for the positive class on the testing dataset out of all the classification models built till now.

The AUC scores for this model based on the training and testing datasets are 0.99 and 0.63 which are characteristics of an extremely overfit model.

The ROC curve for this model based on the training data is as follows.

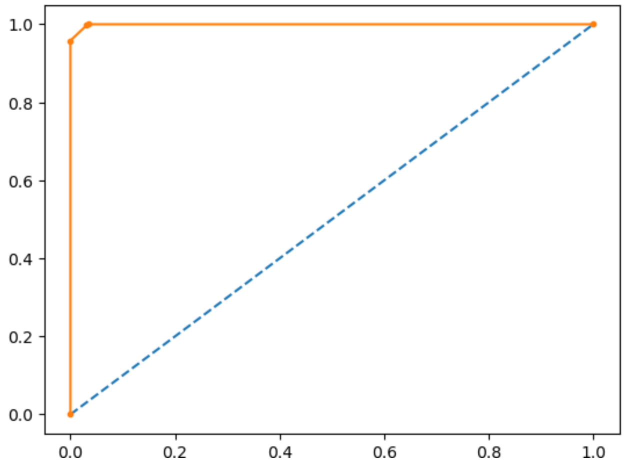


FIGURE 2.28 : ROC curve for default decision tree model for training data

This ROC curve is almost the ROC curve of the ideal classifier. This is due to the overgrown nature of the decision tree where each terminal node is a pure node.

The ROC curve for this model based on the testing data is as follows.

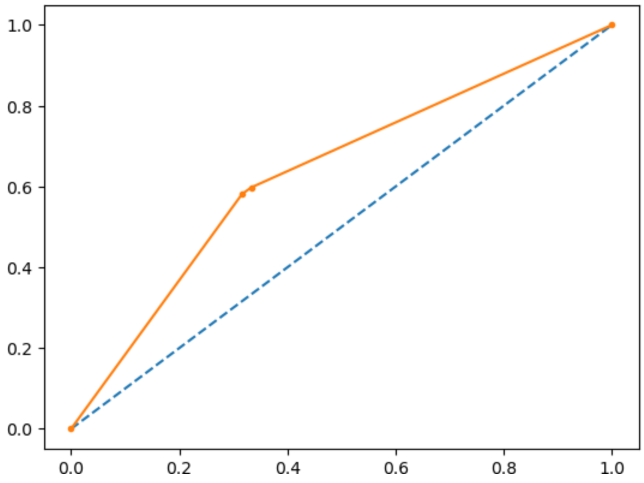


FIGURE 2.29 : ROC curve for default decision tree model for training data

PRUNED / REGULARIZED DECISION TREE MODEL (WITH HYPERPARAMETERS TUNED USING GRID SEARCH CROSS VALIDATION)

The previous decision tree was built using the default values of the available parameters which resulted in an overgrown and overfit tree which cannot be relied upon to do well in production. Therefore a revised version of the same was attempted with different combinations of values of the input parameters which is expected to put a limit to the level upto which the tree will grow.

The pruned decision tree with the best parameters [‘criterion’ = ‘entropy’, ‘max_depth’ = 14, ‘min_samples_leaf’ = 5 and ‘min_samples_split’ = 2] was visualised as follows.

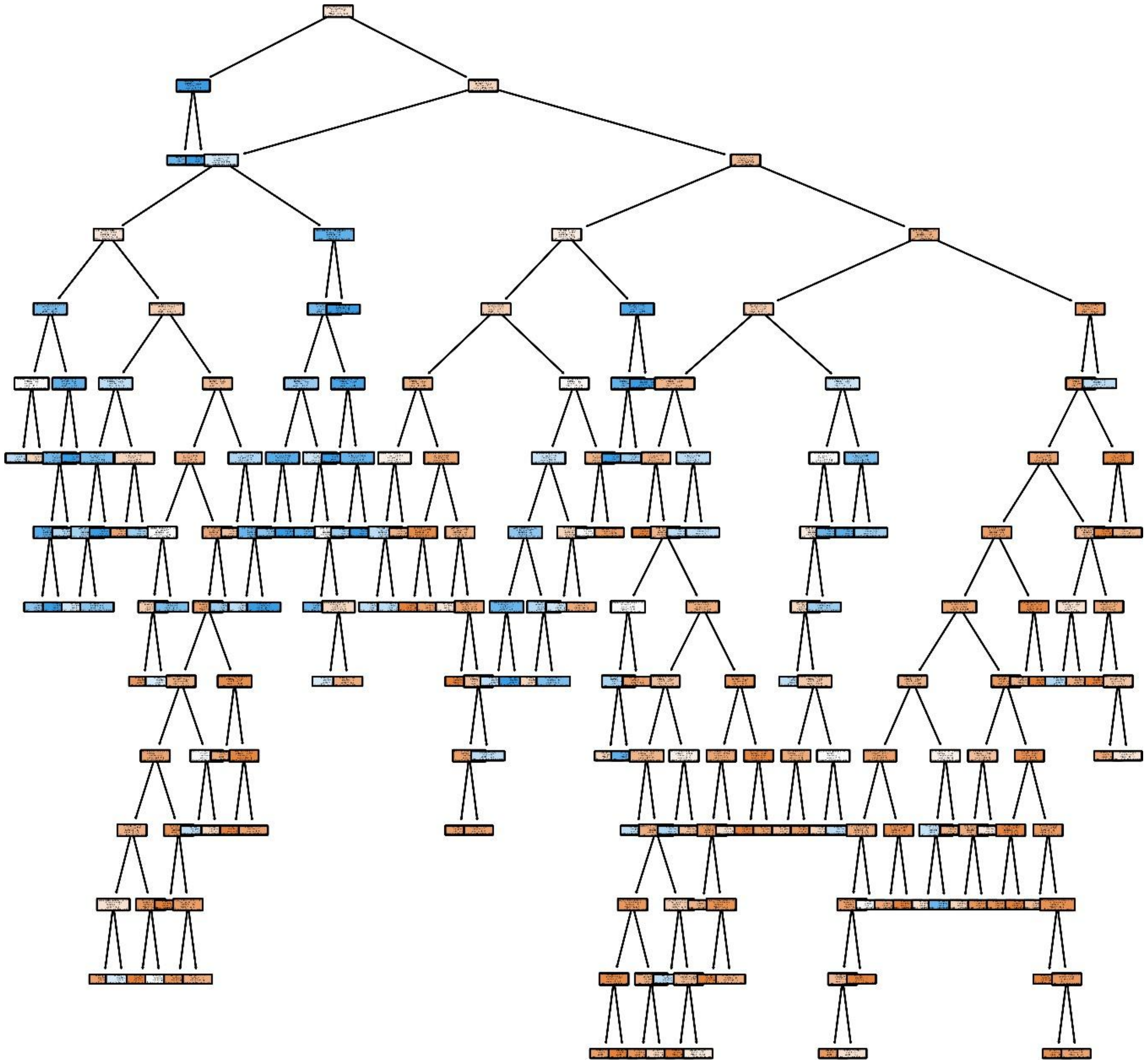


FIGURE 2.30 : Pruned Decision tree structure

The predictions of the pruned decision tree were obtained on the training and testing dataset and based on these predictions the confusion matrix for the training dataset was obtained as follows.

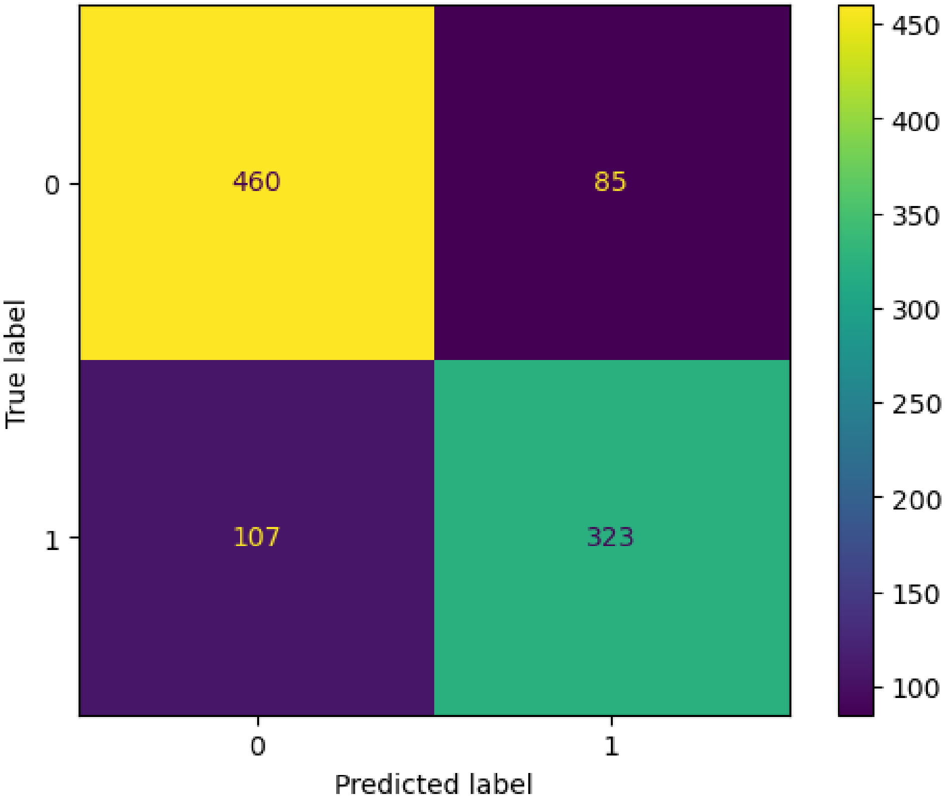
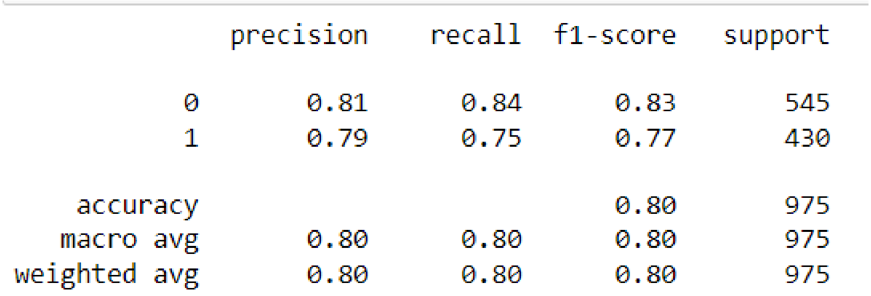


FIGURE 2.31 : Confusion matrix for pruned decision tree model for training data

The classification report for the pruned decision tree based on the training data is as follows.



```
              precision    recall  f1-score   support

           0       0.81      0.84      0.83       545
           1       0.79      0.75      0.77       430

    accuracy                           0.80       975
   macro avg       0.80      0.80      0.80       975
weighted avg       0.80      0.80      0.80       975
```

TABLE 2.17 : Classification report for pruned decision tree classifier model on training data

The confusion matrix for the pruned decision tree based on the training data is as follows.

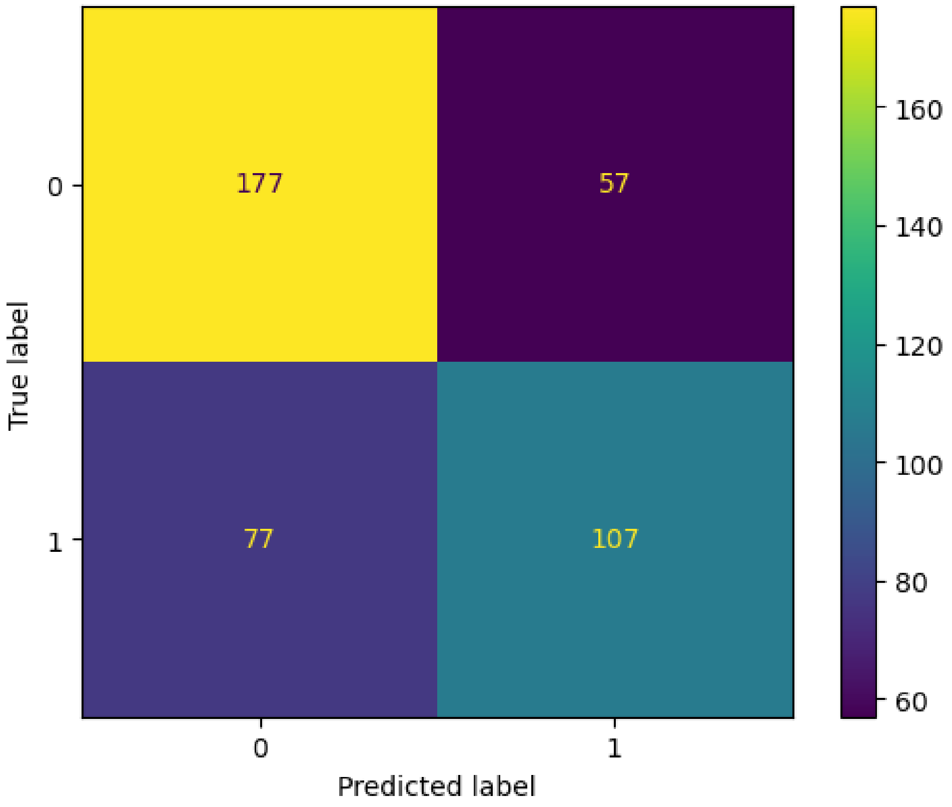
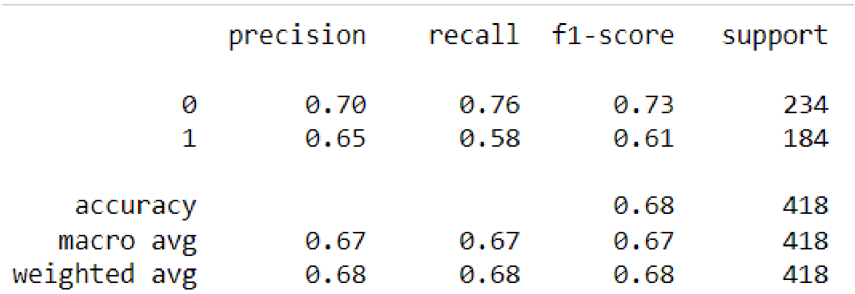


FIGURE 2.32 : Confusion matrix for pruned decision tree model for testing data

The classification report for the pruned decision tree based on the testing data is as follows.



```
              precision    recall  f1-score   support

           0       0.70      0.76      0.73       234
           1       0.65      0.58      0.61       184

    accuracy                           0.68       418
   macro avg       0.67      0.67      0.67       418
weighted avg       0.68      0.68      0.68       418
```

TABLE 2.18 : Classification report for pruned decision tree classifier model on testing data

Based on the recall scores for the positive class in the training and the testing datasets we can see that the performance has slightly improved on the testing data compared to the default decision tree. The amount of overfit has also gone down but there is still some amount of overfit.

The recall score for the positive class is also the highest among all the 5 classification models.

The AUC scores for the pruned decision tree model for the training set and the testing set are 0.896 and 0.716 respectively. The AUC score is the highest among all 5 models based on the testing set.

The ROC curve for the pruned decision tree based on the training data is as follows.

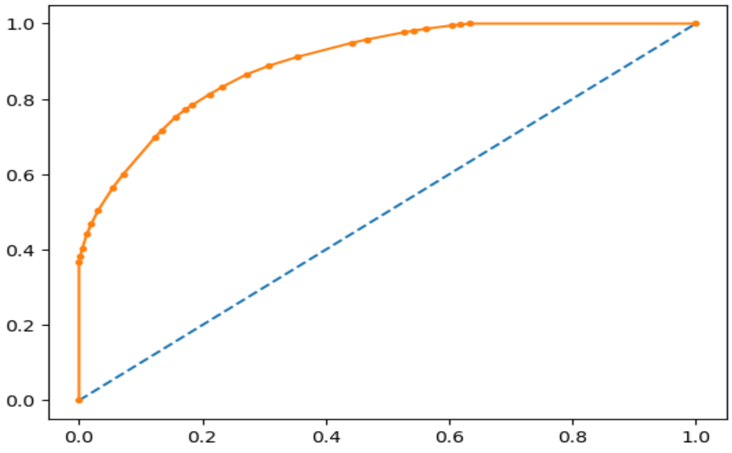


FIGURE 2.33 : ROC curve for pruned decision tree model for training data

The ROC curve is very close to the ideal classifier.

The ROC curve for the pruned decision tree for the testing data is as follows.

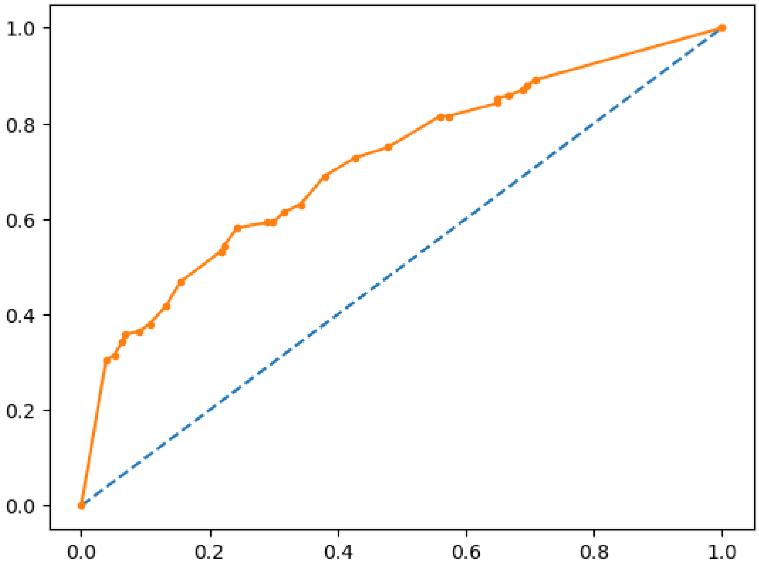


FIGURE 2.34 : ROC curve for pruned decision tree model for testing data

This ROC curve is the closest to the ideal classifier out of all the models for the testing data as also suggested by the respective AUC scores.

SUMMARY OF THE PERFORMANCES OF ALL CLASSIFICATION MODELS

Let us have a look at the important performance metrics for all the 5 classification models that we have used. As explained earlier, we will use the recall score for the positive class as the most important performance metric for each model. We will also refer to the AUC score. Moreover since the class imbalance is not too much (45-55% among the positive and negative class respectively) if we can neglect the difference in costliness of type I error and type II error for the time being we can also have a look at the accuracy for each model. However we will look at the value of each metric only for the testing data because the real worth of a model lies in how it performs on unseen data.

NAME OF CLASSIFIER	**PERFORMANCE METRIC (ON TESTING DATASET)**		
	RECALL SCORE ON POSITIVE CLASS	**AUC SCORE**	**ACCURACY**
Logistic Regression (Default)	0.52	0.70	0.63
Logistic Regression (Hyperparameter tuned)	0.52	0.70	0.63
Linear Discriminant Analayzer	0.51	0.70	0.63
Decision Tree (Default)	0.58	0.63	0.64
Decision Tree (Pruned)	0.58	0.72	0.68

TABLE 2.19 : Summary of evaluation metrics of all classification models

Based on the above table it will be logical to select the pruned decision tree as the best performing classification model. But since it is considered as a greedy algorithm, logistic regression model can also be an alternative. However it should also be noted that none of these classification models is giving a very good performance in terms of any of the metrics used.

Inference: Basis on these predictions, what are the insights and recommendations

Some of the key findings of this analysis are summarized below:

- The awareness about the use of contraceptives must be spread among women of slightly larger age group (40 – 55) who are still inside the reproductive period of their lives.
- The general education level of both females and their husbands must be looked into and as much as possible efforts should be made in order to cause an increase in the average education level as it was found that people with higher qualifications are more used to contraceptives. The awareness regarding contraceptives should spread out to the section with very low education level.
- The access to media must be enhanced because it was found that people with media exposure were more used to contraceptive usage. People with no media exposure must be targeted during the awareness campaign.
- People with poor standard of living are more common non users of contraceptives. They should be properly made aware of the benefits of contraceptives and limitations of unrestricted population increase.
- However it was a very surprising finding that the tendancy of child birth is greater among users of contraceptives compared to the non users. Therefore psychological factors must be considered and non usage of contraceptives should not be considered as the sole reason for explosion of population. Also the consistency in the usage of contraceptives should be stressed upon rather than intermittent use of contraceptives.
- Out of the classification models used on the data, the pruned decision tree gave the best results in terms of Recall, AUC score and accuracy on the testing data. It was followed by the default logistic regression model.
- However none of the models were too impressive in terms of their absolute performance.
- Therefore going forward, ensemble models may be applied on the data to extract better predictions (scope of future work).

NOTE: In the attached code file, some results may differ from those in this report since the randomness was mistakenly not seeded/ freezed and the notebook was rerun a few times after the preparation of the major part of the report. Kindly consider as much as possible.