Business Report

Index-

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

This data set is of is a collection of a computer systems activity measures .Top 3 rows of Dataset are as below-

	lread	lwrite	scall	sread	swrite	fork	exec	rchar	wchar	pgout	...	pgscan	atch	pgin	ppgin	pflt	vflt	runqsz	freemem	freeswap	usr
0	1	0	2147	79	68	0.2	0.2	40671.0	53995.0	0.0	...	0.0	0.0	1.6	2.6	16.00	26.40	CPU_Bound	4670	1730946	95
1	0	0	170	18	21	0.2	0.2	448.0	8385.0	0.0	...	0.0	0.0	0.0	0.0	15.63	16.83	Not_CPU_Bound	7278	1869002	97
2	15	3	2162	159	119	2.0	2.4	NaN	31950.0	0.0	...	0.0	1.2	6.0	9.4	150.20	220.20	Not_CPU_Bound	702	1021237	87

We have to build a model a model which could predict Portion of time (%) that CPU's run in user mode, while using all the above attributes given in data set.

Let's check what all these attributes are-

lread - Reads (transfers per second) between system memory and user memory

lwrite - writes (transfers per second) between system memory and user memory

scall - Number of system calls of all types per second

sread - Number of system read calls per second .

swrite - Number of system write calls per second .

fork - Number of system fork calls per second.

exec - Number of system exec calls per second.

rchar - Number of characters transferred per second by system read calls

wchar - Number of characters transfreed per second by system write calls

pgout - Number of page out requests per second

ppgout - Number of pages, paged out per second

pgfree - Number of pages per second placed on the free list.

pgscan - Number of pages checked if they can be freed per second

atch - Number of page attaches (satisfying a page fault by reclaiming a page in memory) per second

pgin - Number of page-in requests per second

ppgin - Number of pages paged in per second

pflt - Number of page faults caused by protection errors (copy-on-writes).

vflt - Number of page faults caused by address translation .

runqsz - Process run queue size

freemem - Number of memory pages available to user processes

freeswap - Number of disk blocks available for page swapping.

usr - Portion of time (%) that cpus run in user mode

Let’s check what these variables contain and how large is our data set.

(8192, 22)

We have 8192 rows and 22 columns in our Dataset.

Let’s check the data types-

```
---  ------    --------------  -----
 0   lread     8192 non-null   int64
 1   lwrite    8192 non-null   int64
 2   scall     8192 non-null   int64
 3   sread     8192 non-null   int64
 4   swrite    8192 non-null   int64
 5   fork      8192 non-null   float64
 6   exec      8192 non-null   float64
 7   rchar     8088 non-null   float64
 8   wchar     8177 non-null   float64
 9   pgout     8192 non-null   float64
 10  ppgout    8192 non-null   float64
 11  pgfree    8192 non-null   float64
 12  pgscan    8192 non-null   float64
 13  atch      8192 non-null   float64
 14  pgin      8192 non-null   float64
 15  ppgin     8192 non-null   float64
 16  pflt      8192 non-null   float64
 17  vflt      8192 non-null   float64
 18  runqsz    8192 non-null   object
 19  freemem   8192 non-null   int64
 20  freeswap  8192 non-null   int64
 21  usr       8192 non-null   int64
dtypes: float64(13), int64(8), object(1)
```

There is one categorical variable rest all are either float or integer type

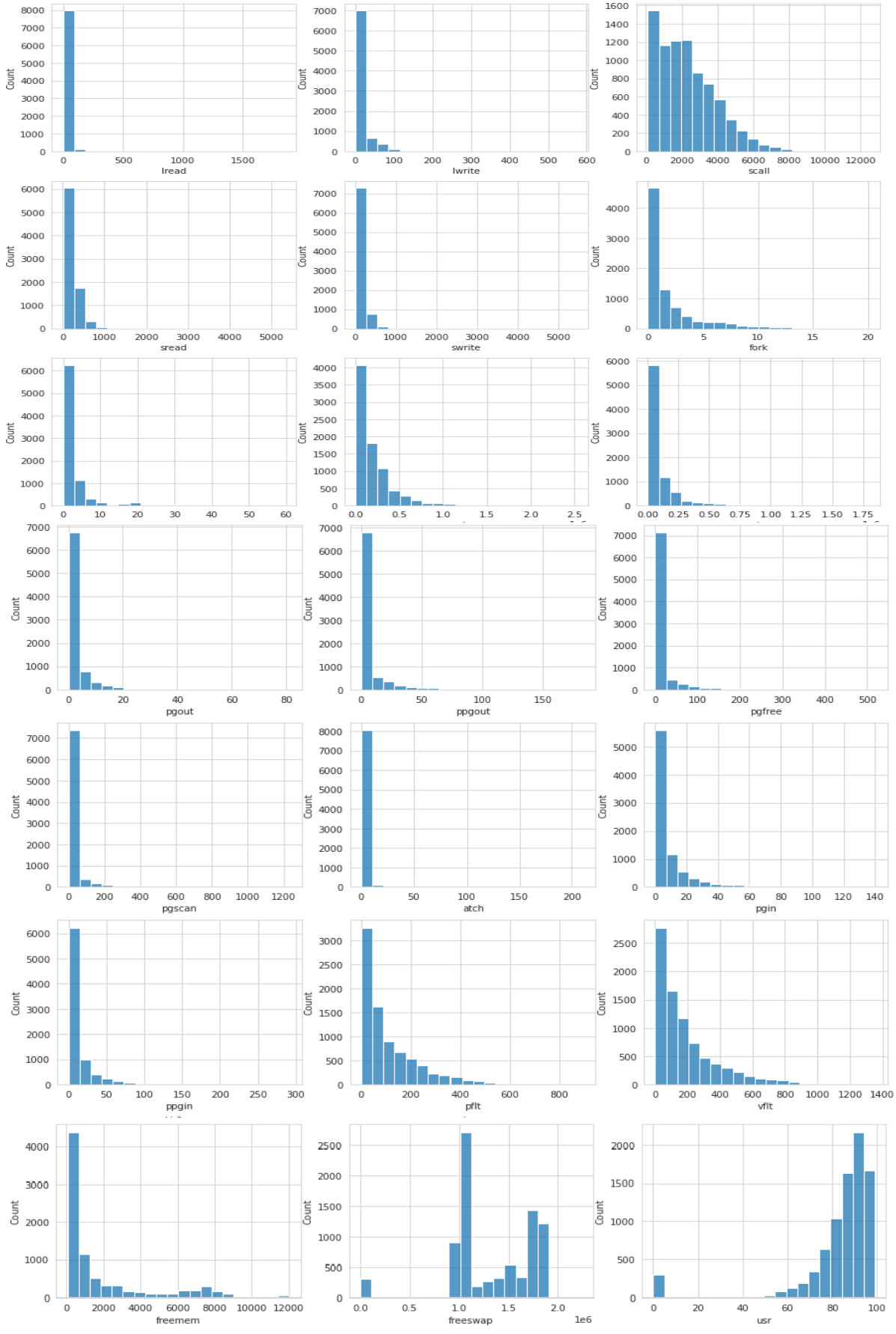
Let’s check the five point summary-

	count	mean	std	min	25%	50%	75%	max
lread	8192.0	1.955969e+01	53.353799	0.0	2.0	7.0	20.000	1845.00
lwrite	8192.0	1.310620e+01	29.891726	0.0	0.0	1.0	10.000	575.00
scall	8192.0	2.306318e+03	1633.617322	109.0	1012.0	2051.5	3317.250	12493.00
sread	8192.0	2.104800e+02	198.980146	6.0	86.0	166.0	279.000	5318.00
swrite	8192.0	1.500582e+02	160.478980	7.0	63.0	117.0	185.000	5456.00
fork	8192.0	1.884554e+00	2.479493	0.0	0.4	0.8	2.200	20.12
exec	8192.0	2.791998e+00	5.212456	0.0	0.2	1.2	2.800	59.56
rchar	8088.0	1.973857e+05	239837.493526	278.0	34091.5	125473.5	267828.750	2526649.00
wchar	8177.0	9.590299e+04	140841.707911	1498.0	22916.0	46619.0	106101.000	1801623.00
pgout	8192.0	2.285317e+00	5.307038	0.0	0.0	0.0	2.400	81.44
ppgout	8192.0	5.977229e+00	15.214590	0.0	0.0	0.0	4.200	184.20
pgfree	8192.0	1.191971e+01	32.363520	0.0	0.0	0.0	5.000	523.00
pgscan	8192.0	2.152685e+01	71.141340	0.0	0.0	0.0	0.000	1237.00
atch	8192.0	1.127505e+00	5.708347	0.0	0.0	0.0	0.600	211.58
pgin	8192.0	8.277960e+00	13.874978	0.0	0.6	2.8	9.765	141.20
ppgin	8192.0	1.238859e+01	22.281318	0.0	0.6	3.8	13.800	292.61
pflt	8192.0	1.097938e+02	114.419221	0.0	25.0	63.8	159.600	899.80
vflt	8192.0	1.853158e+02	191.000603	0.2	45.4	120.4	251.800	1365.00
freemem	8192.0	1.763456e+03	2482.104511	55.0	231.0	579.0	2002.250	12027.00
freeswap	8192.0	1.328126e+06	422019.426957	2.0	1042623.5	1289289.5	1730379.500	2243187.00
usr	8192.0	8.396887e+01	18.401905	0.0	81.0	89.0	94.000	99.00

We can check a lot of zero’s in most of the columns; almost 50% of the columns are having value equal to zero.

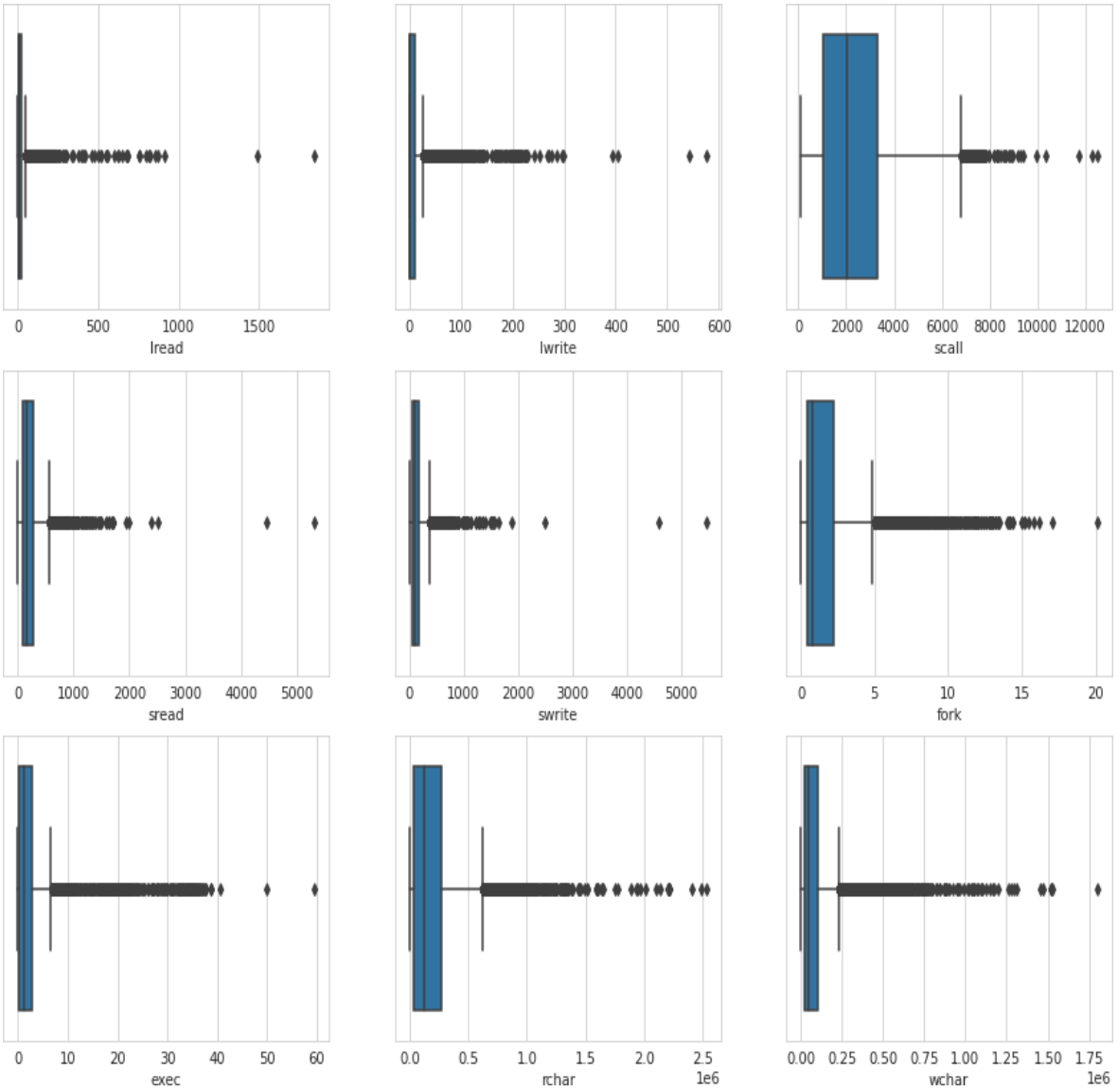
We can also check that there are few missing values too in the data set.

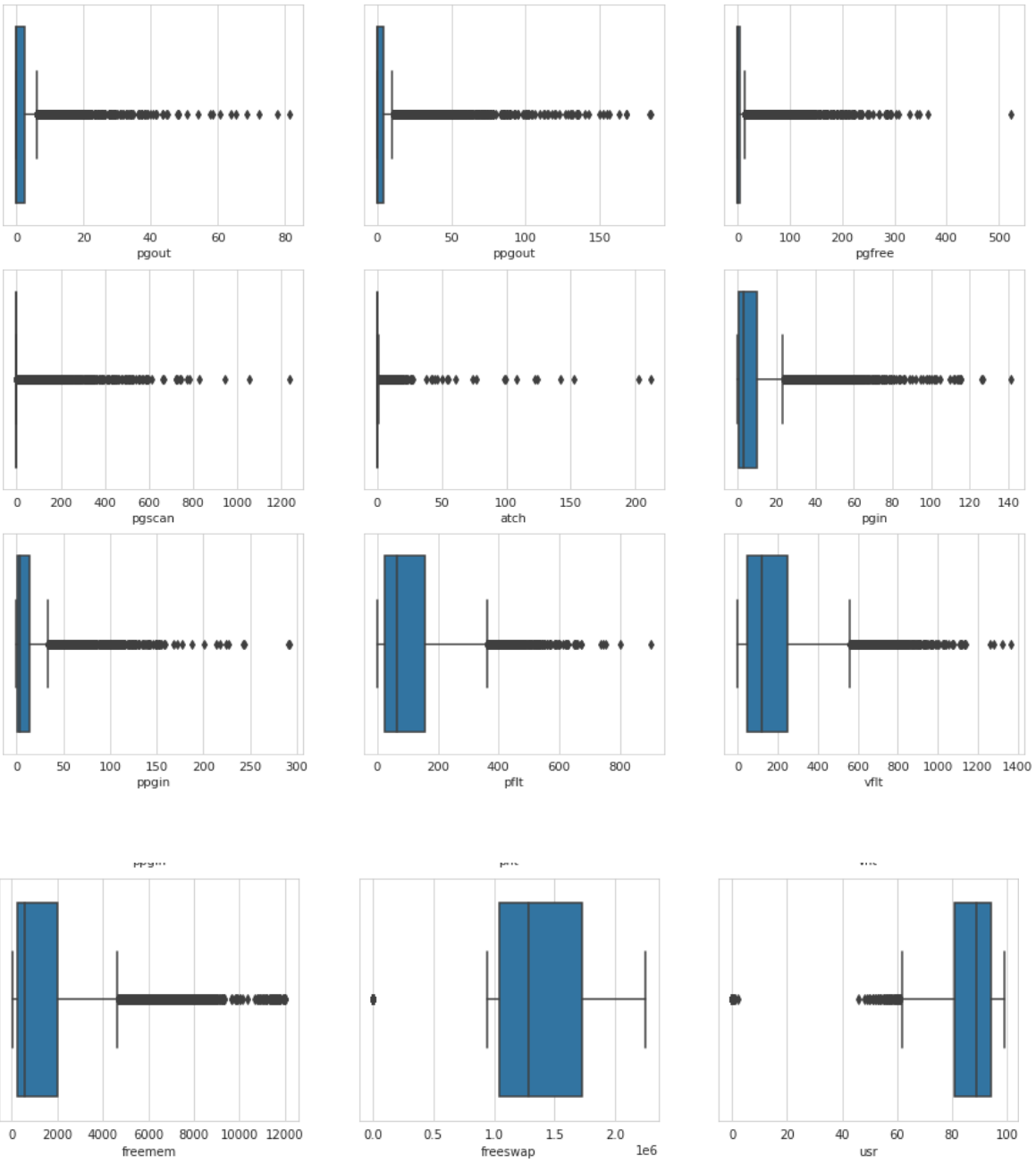
Let’s Perform Univariate, Bivariate Analysis, And Multivariate Analysis on this dataset first-


Count
lread
lwrite
scall
sread
swrite
fork
pgout
ppgout
pgfree
pgscan
atch
pgin
ppgin
pflt
vflt
freemem
freeswap
1e6
usr

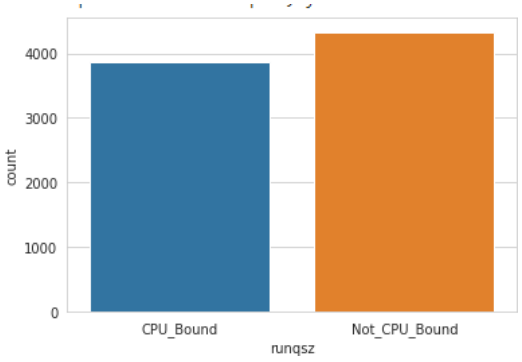
We can clearly see that most of the variables are right skewed; this again confirms that majority of values in all columns lie around zero. Apart from USR which is left skewed.

Let’s check out there box plots.

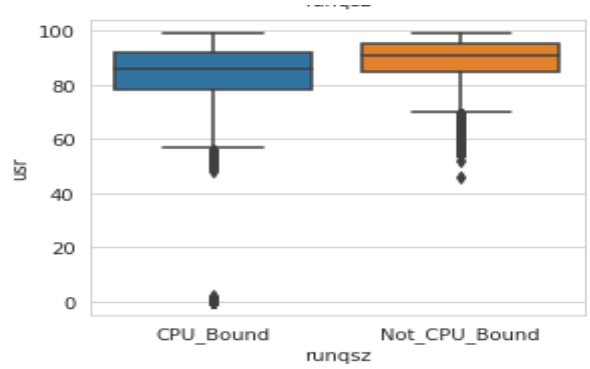




We can see a lot of outliers in each column of dataset.

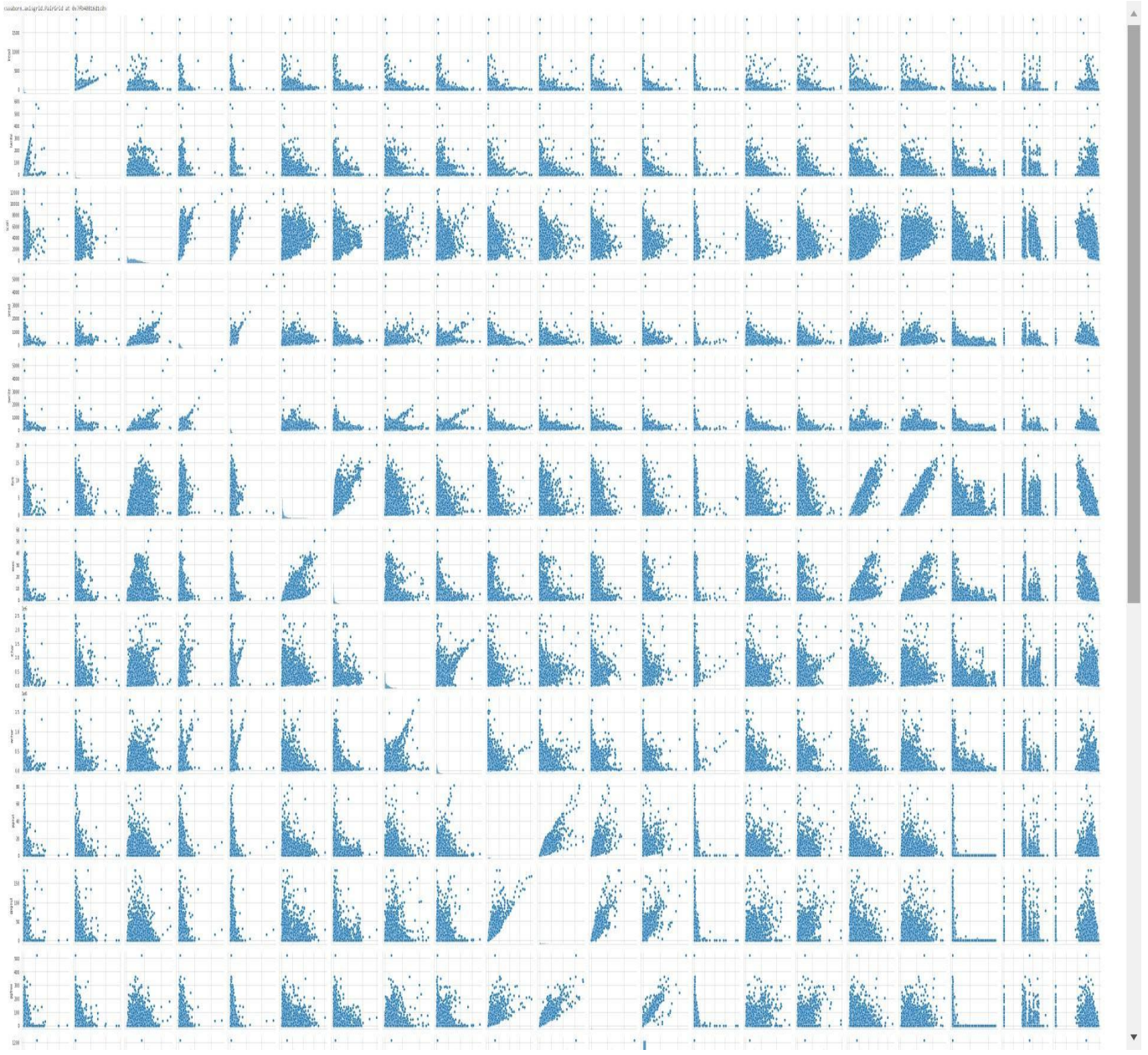


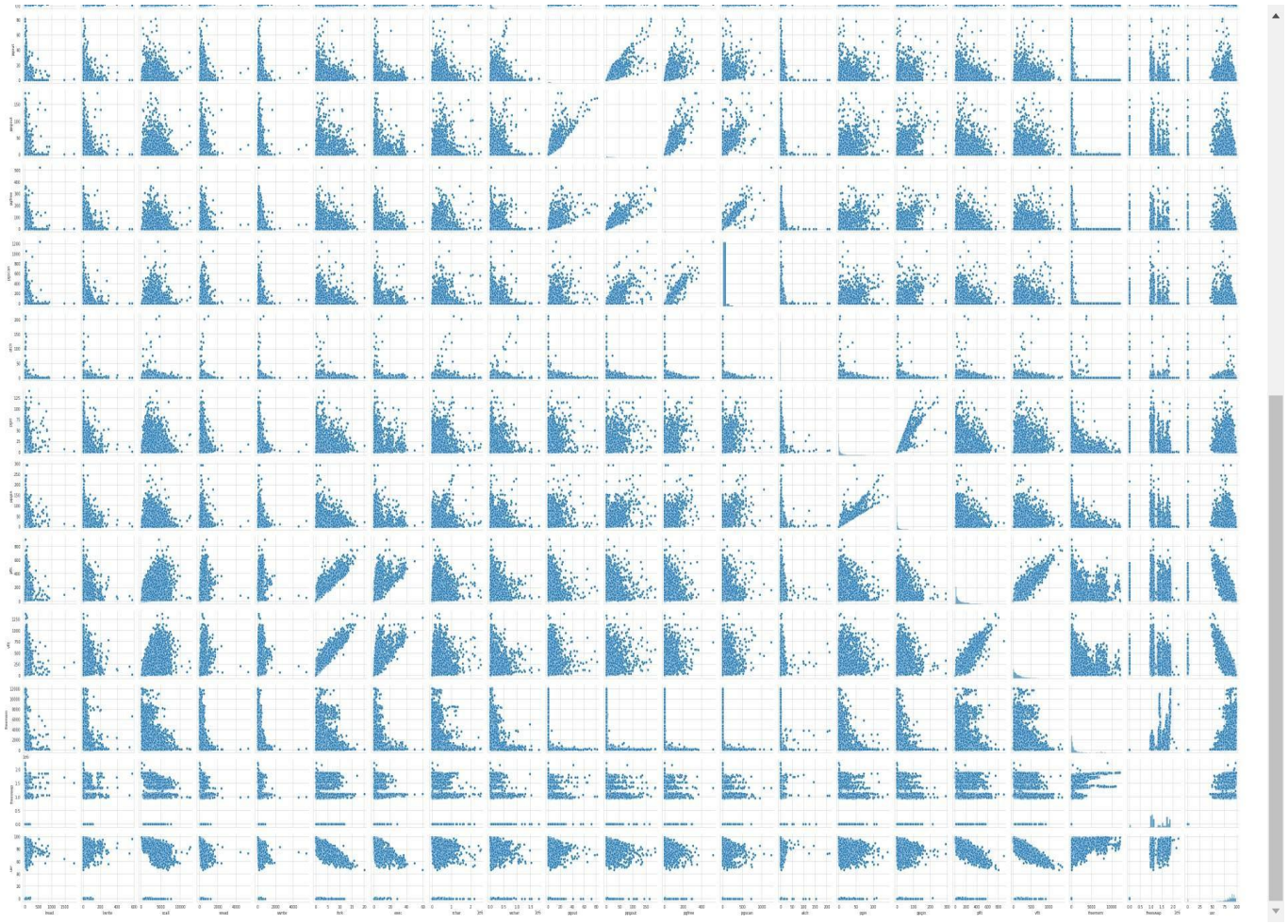
We can also see that Non CPU Bound users are slightly more than CPU Bound users.



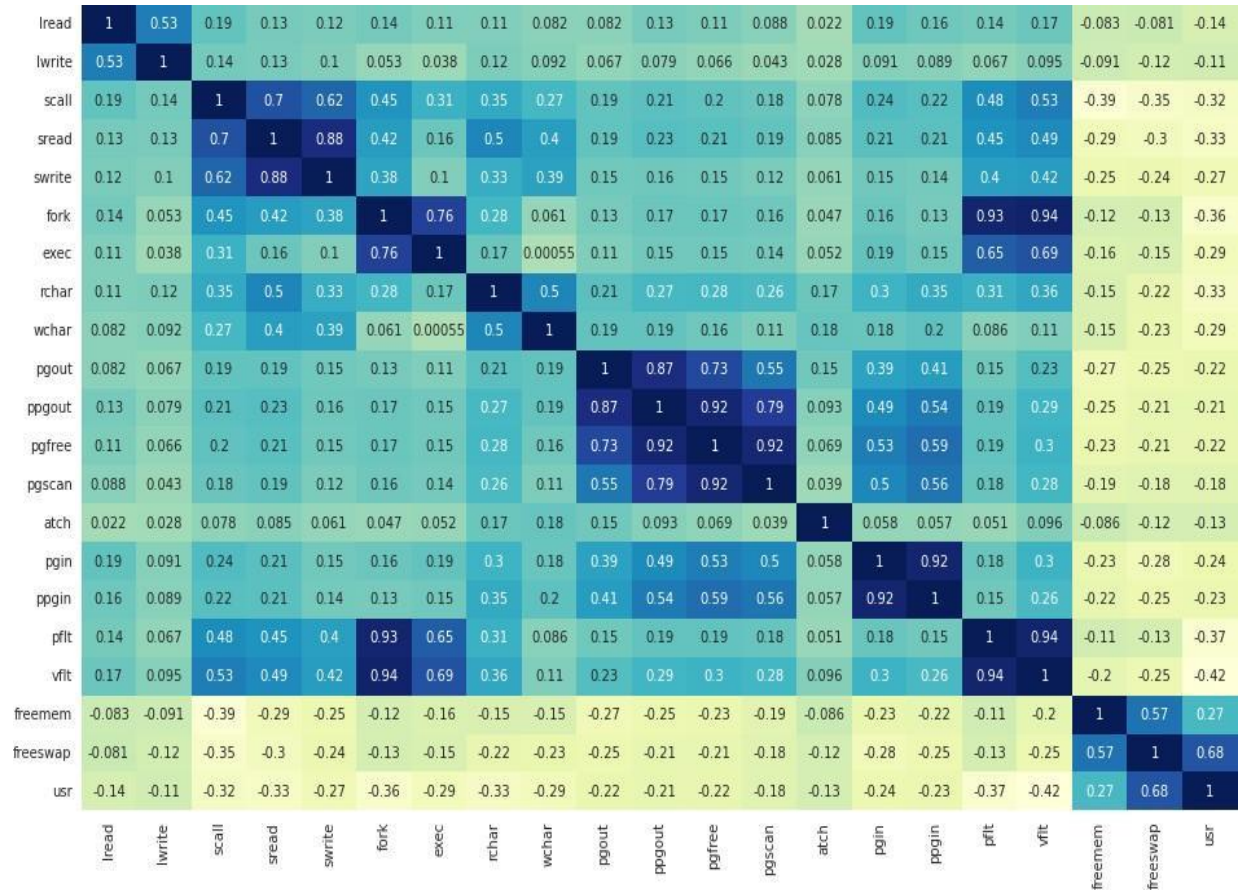
Doing CPU bound or Not-CPU Bound activities won't have much of an impact on percentage of time system is being used on User mode or not.

Let's check the pattern between all the numerical features-





A lot of linear spread could be seen among different variables, let’s check out the correlation between them to get a better clarity of this.



A lot of multicollinearity can be seen along columns. This makes us understand that VIF would be a key for us to work with, so that we can get rid of multicollinearity and thus there shall not be any issue while interpretation of final equation.

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

Let's check for null values if present-

```
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
```

There are missing values in rchar and wchar-

We have imputed these values by median of the column as there are a lot of outliers still present so that would impact mean. So imputing values with median is the right thing to do.

After imputation-

```
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
```

No null values any more-

Check for values that are equal to zero-

```
lread 675
lwrite 2684
scall 0
sread 0
swrite 0
fork 21
exec 21
rchar 0
wchar 0
pgout 4878
ppgout 4878
pgfree 4869
pgscan 6448
atch 4575
pgin 1220
ppgin 1220
pflt 3
vflt 0
runqsz 0
freemem 0
freeswap 0
usr 283
```

There are a lot of rows with zeros.

We have checked and made 2 models one with all the columns and other with these columns dropped ['pgout','ppgout','pgfree','pgscan','atch','lwrite'].

We have trained both the models using sklearn.

We found that there is no as such change in accuracy of the model.

For model with all the columns-

Training accuracy- 0.7917657945902528

Test accuracy - 0.7485953459607142

For model where we have dropped columns ['pgout','ppgout','pgfree','pgscan','atch','lwrite']

Training accuracy- 0.7877577709090766

Test accuracy - 0.744575406785511

Apart from that we have also checked for columns with major values zero.

Pgout – We checked the 5 point summary of USR with all rows with pgout as zero and then with all rows of dataset.

There is no significant difference between them means having zero’s in pgout doesn’t have significant impact on USR

```
mean       87.489750
std        15.389183
min         0.000000
25%        86.000000
50%        91.000000
75%        95.000000
max        99.000000
Name: usr, dtype: flo
mean       83.968872
std        18.401905
min         0.000000
25%        81.000000
50%        89.000000
75%        94.000000
max        99.000000
```

Similar is the case with other columns too.

So we decide to drop the columns with a lot of zero's in them. And go with the remaining columns of dataset to predict USR.

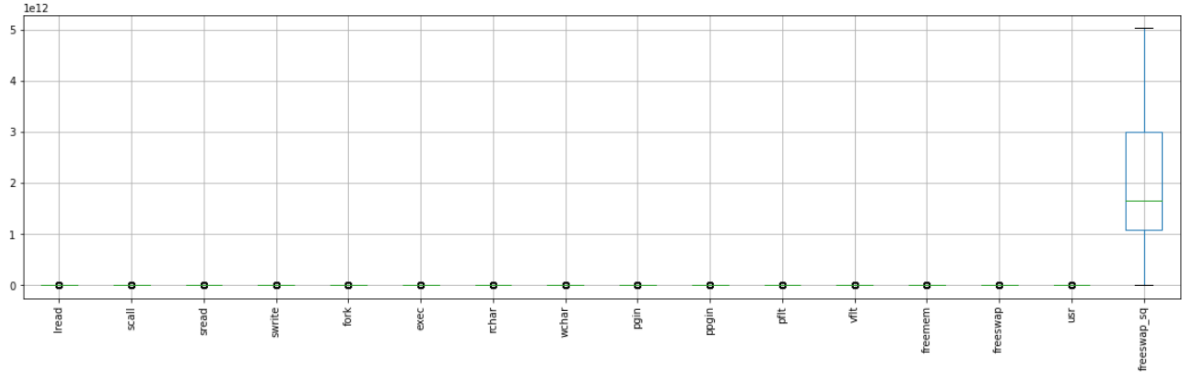
Squaring up column freeswap shows a significant increment in train and test accuracy which is now up to –

Train Accuracy - 0.85

Test Accuracy- 0.84

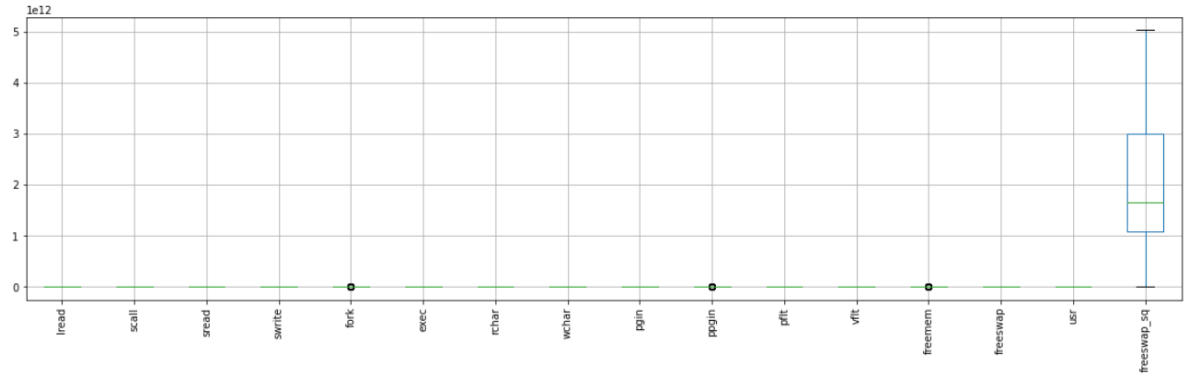
No duplicates are found in dataset-

We have treated the outliers-



Let's cap these outliers at 10% and 90% , as linear regression is highly sensitive to outliers-

After capping-



Outliers removed--------

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

Let's encode the Data –

```
0   lread                8192 non-null   float64
1   scall                8192 non-null   float64
2   sread                8192 non-null   float64
3   swrite               8192 non-null   float64
4   fork                 8192 non-null   float64
5   exec                 8192 non-null   float64
6   rchar                8192 non-null   float64
7   wchar                8192 non-null   float64
8   pgin                 8192 non-null   float64
9   ppgin                8192 non-null   float64
10  pflt                 8192 non-null   float64
11  vflt                 8192 non-null   float64
12  freemem              8192 non-null   float64
13  freeswap             8192 non-null   float64
14  usr                  8192 non-null   float64
15  freeswap_sq          8192 non-null   int64
16  runqsz_Not_CPU_Bound 8192 non-null   uint8
```

After creating dummy variables we get column - runqsz_Not_CPU_bound.

After splitting the data (70:30)

And appling liner regression using sklearn we get

Train and test accuracy as-

Train Accuracy - 0.85

Test Accuracy- 0.84

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.852
Model:	OLS	Adj. R-squared:	0.852
Method:	Least Squares	F-statistic:	2059.
Date:	Sun, 05 Mar 2023	Prob (F-statistic):	0.00
Time:	08:03:02	Log-Likelihood:	-15010.
No. Observations:	5734	AIC:	3.005e+04
Df Residuals:	5717	BIC:	3.017e+04
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	123.9933	0.670	185.134	0.000	122.680	125.306
lread	-0.0245	0.003	-7.171	0.000	-0.031	-0.018
scall	-0.0012	4.95e-05	-23.727	0.000	-0.001	-0.001
sread	-0.0017	0.001	-2.090	0.037	-0.003	-0.000
swrite	-0.0045	0.001	-3.841	0.000	-0.007	-0.002
fork	-0.6207	0.090	-6.880	0.000	-0.798	-0.444
exec	-0.2703	0.042	-6.511	0.000	-0.352	-0.189
rchar	-3.643e-06	4.13e-07	-8.812	0.000	-4.45e-06	-2.83e-06
wchar	-4.587e-06	7.75e-07	-5.919	0.000	-6.11e-06	-3.07e-06
pgin	-0.0630	0.022	-2.898	0.004	-0.106	-0.020
ppgin	-0.0663	0.015	-4.539	0.000	-0.095	-0.038
pflt	-0.0184	0.002	-12.058	0.000	-0.021	-0.015
vflt	-0.0101	0.001	-9.360	0.000	-0.012	-0.008
freemem	0.0001	2.76e-05	4.343	0.000	6.58e-05	0.000
freeswap	-3.864e-05	8.29e-07	-46.622	0.000	-4.03e-05	-3.7e-05
freeswap_sq	1.38e-11	2.77e-13	49.887	0.000	1.33e-11	1.43e-11
runqsz_Not_CPU_Bound	-1.0145	0.103	-9.893	0.000	-1.216	-0.813

Omnibus:	686.732	Durbin-Watson:	1.963
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1577.586
Skew:	-0.713	Prob(JB):	0.00
Kurtosis:	5.137	Cond. No.	3.38e+13

Let’s check for VIF to check multicollinearity and try to get rid of it.

const	0.000000
lread	1.305995
scall	3.008115
sread	6.005291
swrite	5.450387
fork	12.642756
exec	3.109923
rchar	2.156012
wchar	1.577577
pgin	13.829330
ppgin	13.691326
pflt	10.298732
vflt	13.757375
freemem	1.898367
freeswap	41.947880
freeswap_sq	41.227046
runqsz_Not_CPU_Bound	1.361609

There is a lot if multi collinearity between variable which would impact our interpretability of the model so we will try to reduce this.

After dropping few columns one by one we get down to-

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.838
Model:	OLS	Adj. R-squared:	0.838
Method:	Least Squares	F-statistic:	3300.
Date:	Sun, 05 Mar 2023	Prob (F-statistic):	0.00
Time:	08:03:15	Log-Likelihood:	-15265.
No. Observations:	5734	AIC:	3.055e+04
Df Residuals:	5724	BIC:	3.062e+04
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	123.0225	0.692	177.771	0.000	121.666	124.379
lread	-0.0324	0.004	-9.170	0.000	-0.039	-0.026
scall	-0.0015	4.03e-05	-37.692	0.000	-0.002	-0.001
rchar	-5.095e-06	3.84e-07	-13.275	0.000	-5.85e-06	-4.34e-06
wchar	-4.524e-06	7.64e-07	-5.920	0.000	-6.02e-06	-3.03e-06
pgin	-0.1826	0.007	-25.393	0.000	-0.197	-0.168
pflt	-0.0470	0.001	-76.044	0.000	-0.048	-0.046
freeswap	-3.793e-05	8.64e-07	-43.925	0.000	-3.96e-05	-3.62e-05
freeswap_sq	1.389e-11	2.87e-13	48.402	0.000	1.33e-11	1.45e-11
runqsz_Not_CPU_Bound	-0.9411	0.106	-8.895	0.000	-1.149	-0.734

Omnibus:	543.017	Durbin-Watson:	1.992
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1268.852
Skew:	-0.575	Prob(JB):	2.97e-276
Kurtosis:	4.997	Cond. No.	3.35e+13

Let’s check for VIF-

const	0.000000
lread	1.282522
scall	1.825194
rchar	1.701870
wchar	1.405113
pgin	1.384528
pflt	1.549548
freeswap	41.729879
freeswap_sq	40.662933
runqsz_Not_CPU_Bound	1.327941

dtype: float64

Multicollinearity has decreased significantly for a loss of around 1.4

Let’s try to reduce few more columns to make our model simpler –

OLS Regression Results

Dep. Variable:	usr	R-squared:	0.833
Model:	OLS	Adj. R-squared:	0.833
Method:	Least Squares	F-statistic:	4766.
Date:	Sun, 05 Mar 2023	Prob (F-statistic):	0.00
Time:	08:03:26	Log-Likelihood:	-15357.
No. Observations:	5734	AIC:	3.073e+04
Df Residuals:	5727	BIC:	3.077e+04
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	120.4534	0.614	196.220	0.000	119.250	121.657
scall	-0.0015	3.91e-05	-38.568	0.000	-0.002	-0.001
rchar	-5.678e-06	3.56e-07	-15.941	0.000	-6.38e-06	-4.98e-06
pgin	-0.1909	0.007	-26.341	0.000	-0.205	-0.177
pflt	-0.0479	0.001	-79.858	0.000	-0.049	-0.047
freeswap	-3.614e-05	8.07e-07	-44.767	0.000	-3.77e-05	-3.46e-05
freeswap_sq	1.342e-11	2.71e-13	49.556	0.000	1.29e-11	1.39e-11

Omnibus:	530.046	Durbin-Watson:	1.999
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1188.356
Skew:	-0.575	Prob(JB):	8.95e-259
Kurtosis:	4.911	Cond. No.	2.91e+13

For a sacrifice of 0.5 of accuracy-

Let’s check VIF-

const	0.000000
scall	1.662116
rchar	1.420429
pgin	1.363418
pflt	1.418468
freeswap	35.344028
freeswap_sq	35.075101

dtype: float64

There is still some multicollinearity between freeswap and freeswap_sq but removing one of them is decreasing the accuracy of model drastically-

So we are keeping them for now-

Let’s check out the RMSE, MAE, and accuracy of this model on train and test data-

We are getting

RMSE_train- 3.5227019304440943

RMSE_test- 3.5621867328993004

They are pretty close no overfitting is there-

MAE for train data is - 2.567840125333411

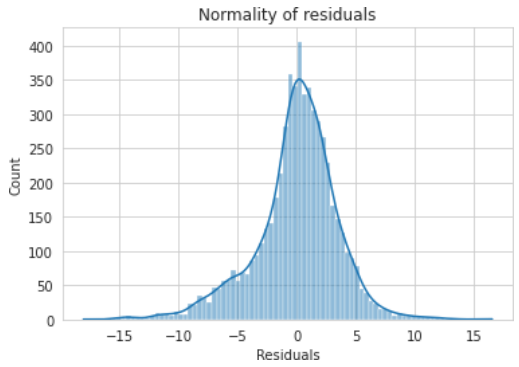
Let’s check the accuracy of model-

Train accuracy - 0.8331447690683988

Test accuracy- 0.8201117417756697

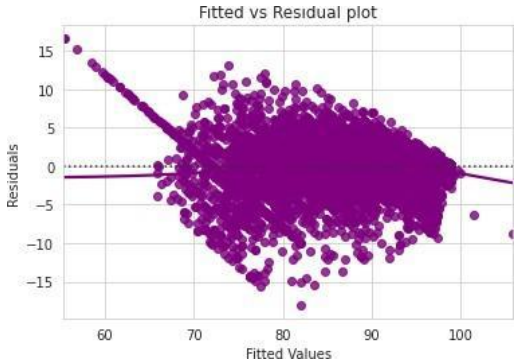
Let’s check out the assumptions of LR-

1. Normality-



Residuals are quite normally distributed visually.

2. Independence and Homoscedasticity-



There is no as such pattern found, the line at top is very much covered at bottom. Variance is also not changing drastically. No fanning effect found visually.

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

Various steps followed in this model are-

After going through all the features and checking there histograms and boxplots we found that data is mostly right skewed and there are a lot of outliers in data. There were a lot zeros present in a lot of columns. So we initially didn’t remove those columns and checked for missing values. After treating missing values for 2 columns with median of individual columns we then checked the correlation of various columns using heat map.

After that we capped outliers at 95% and 5%, and then split the data into train and test data. After that we checked the accuracy of model using sklearn model of LR.
Then we again capped the outliers at 10% and 90% , and again run the model and we got a bit of increase in accuracy.
Although there are a lot of columns with almost 50% of the values as zero so we removed those columns and checked the accuracy. And we found that accuracy is not getting affected by these columns.

Then we tried to add squares of different columns to increase the accuracy, we found that freeswap is one column and the square of this column almost shoot the train accuracy to almost 85%.

After that we applied the stats model and checked for VIF to check if there is multicollinearity. We removed columns one by one and checked for a minimal drop in accuracy and continued till we reach to below equation. Although there is a bit of multicollinearity in the equation but for accuracy's sake we have kept it as it is. Then we checked for the assumptions of LR, which we are kind of okay with for now.

Final Equation is –

usr = 120.45338257913316 + -0.0015064082403879401 * (scall) +

-5.677930414405032e-06 * (rchar) + -0.19089803638275435 * (pgin) +

-0.04793787181620117 * (pflt) + -3.614227317591907e-05 * (freeswap) +

1.341888870757357e-11 * (freeswap_sq)

USR- Portion of time (%) that cpus run in user mode
Scall – scall - Number of system calls of all types per second
Rchar- Number of characters transferred per second by system read calls
Pgin- Number of page-in requests per second
Pflt- Number of page faults caused by protection errors (copy-on-writes).
Freeswap- Number of disk blocks available for page swapping.
Freeswap_sq- (Number of disk blocks available for page swapping.)^2

Conclusion and Business insight-

When scall , Rchar , pgin and pflt increases then Portion of time CPU'S run on user mode decreases. So keep a track of

Number of system calls/ sec, No. of characters transferred / sec by system read calls, No. of page in request/sec and No. of page faults caused by protection errors (copy-on-writes). If any of these increases then portion of time cpu's run in user mode decreases.

Although- for Freeswap weather it increases or decreases we have to check for the magnitude of

– 3.614227317591907e-05 * (freeswap) +

1.341888870757357e-11 * (freeswap_sq) if its increasing or decreasing.

We could also have dropped freeswap to make the interpretation better but then accuracy is getting down drastically so we kept it.