Predictive Modelling Graded Project

By Giridharan Velmurugan

Contents

Problem I Statement :-

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

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

DATA DICTIONARY:

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 sec
pgin - Number of page-in requests per second
ppgin - Number of pages paged in per second
pflt - Number of page faults caused by protection errors (copy-on-writes).
vflt - Number of page faults caused by address translation .
runqsz - Process run queue size (The number of kernel threads in memory that are waiting for a CPU to run. Typically, this value should be less than 2. Consistently higher values mean that the system might be CPU-bound.)
freemem - Number of memory pages available to user processes
freeswap - Number of disk blocks available for page swapping.
usr - Portion of time (%) that cpus run in user mode.

Part I

About the Data Frame

The Data Frame has three types of data – Integer, Float and Object. Along we also have the shape and number of features/columns.

- **Data Types:**
  - **Integer columns**: lread, lwrite, scall, sread, swrite, freemem, freeswap, and usr
  - **Float columns:** fork, exec, rchar, wchar, pgout, ppgout, pgfree, pgscan, atch, pgin, ppgin, pflt, and vflt
  - **Object columns:** runqsz

- **Shape of the Dataset:**
  - **Number of Rows/Instances:** 8192
  - **Number of Columns:** 22

We do have some outliers and null values within the data, but we do not have any duplicated instances which is a boon.

Data Statistics

The column 'usr', which is the target variable, ranges from 0 to 99 with a mean of approximately 83.97.

Some features like 'lread', 'lwrite', 'fork', etc., show a wide range of values, suggesting varying scales.

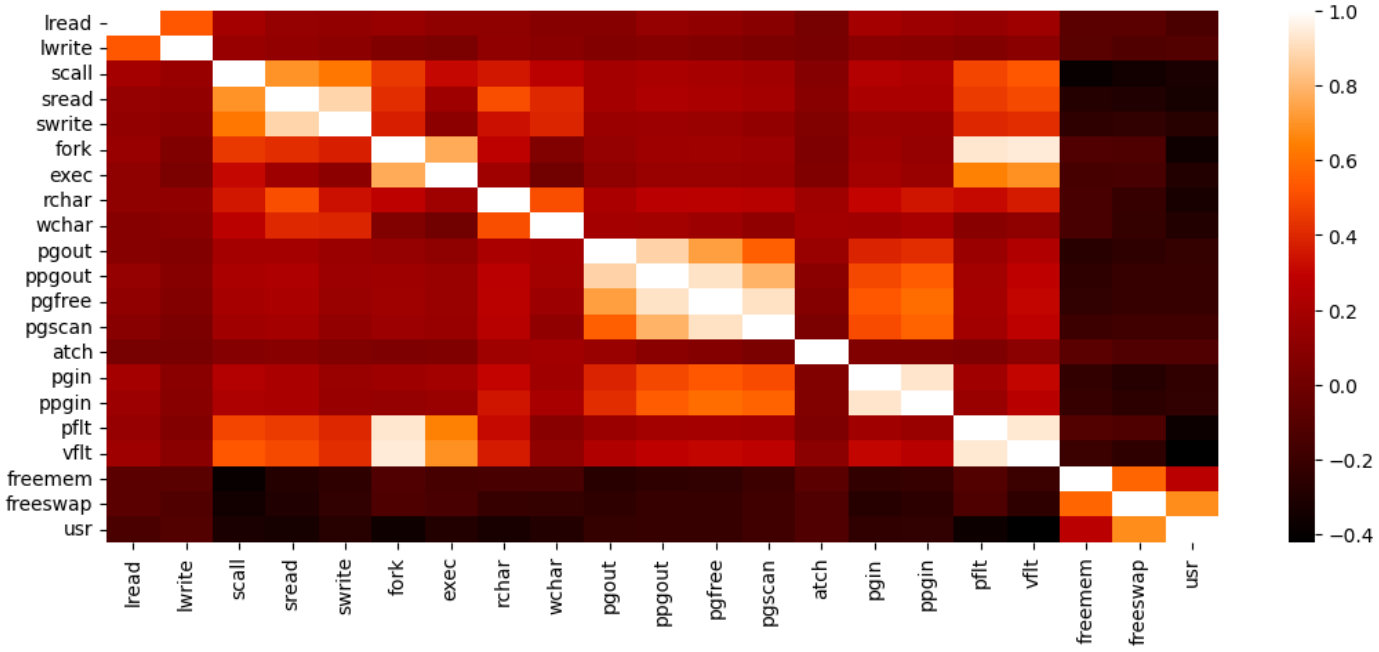
Features like 'rchar' and 'wchar' have large numbers, which may need scaling or transformation, moreover, 'rchar' and 'wchar' are the ones that have null values while the rest 20 columns are populated fully. This will be dealt and discussed in the coming topics.

	Count	mean	std	min	25%	50%	75%	max
lread	8192	19.56	53.35	0.00	2.00	7.00	20.00	1845.00
lwrite	8192	13.11	29.89	0.00	0.00	1.00	10.00	575.00
scall	8192	2306.32	1633.62	109.00	1012.00	2051.50	3317.25	12493.00
sread	8192	210.48	198.98	6.00	86.00	166.00	279.00	5318.00
swrite	8192	150.06	160.48	7.00	63.00	117.00	185.00	5456.00
fork	8192	1.88	2.48	0.00	0.40	0.80	2.20	20.12
exec	8192	2.79	5.21	0.00	0.20	1.20	2.80	59.56
rchar	8088	197385.70	239837.49	278.00	34091.50	125473.50	267828.75	2526649.00
wchar	8177	95902.99	140841.71	1498.00	22916.00	46619.00	106101.00	1801623.00
pgout	8192	2.29	5.31	0.00	0.00	0.00	2.40	81.44
ppgout	8192	5.98	15.21	0.00	0.00	0.00	4.20	184.20
pgfree	8192	11.92	32.36	0.00	0.00	0.00	5.00	523.00
pgscan	8192	21.53	71.14	0.00	0.00	0.00	0.00	1237.00
atch	8192	1.13	5.71	0.00	0.00	0.00	0.60	211.58
pgin	8192	8.28	13.87	0.00	0.60	2.80	9.77	141.20
ppgin	8192	12.39	22.28	0.00	0.60	3.80	13.80	292.61
pflt	8192	109.79	114.42	0.00	25.00	63.80	159.60	899.80
vflt	8192	185.32	191.00	0.20	45.40	120.40	251.80	1365.00
freemem	8192	1763.46	2482.10	55.00	231.00	579.00	2002.25	12027.00
freeswap	8192	1328126.00	422019.43	2.00	1042623.50	1289289.50	1730379.50	2243187.00
usr	8192	83.97	18.40	0.00	81.00	89.00	94.00	99.00

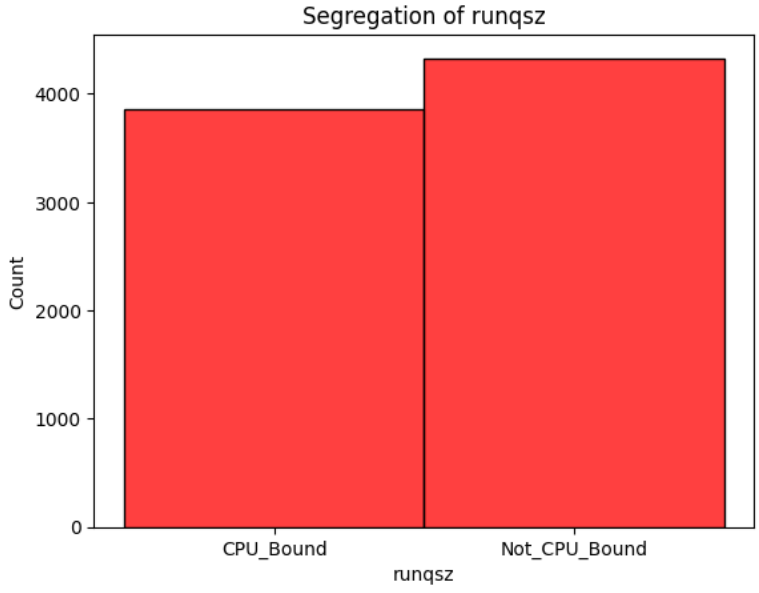
The above table misses only on the 'runqsz' feature, which is an object type, here it denotes that is a categorical feature. 'CPU_Bound' and 'Not_CPU_Bound" are the two states it can be set at.

Variables and Relationships

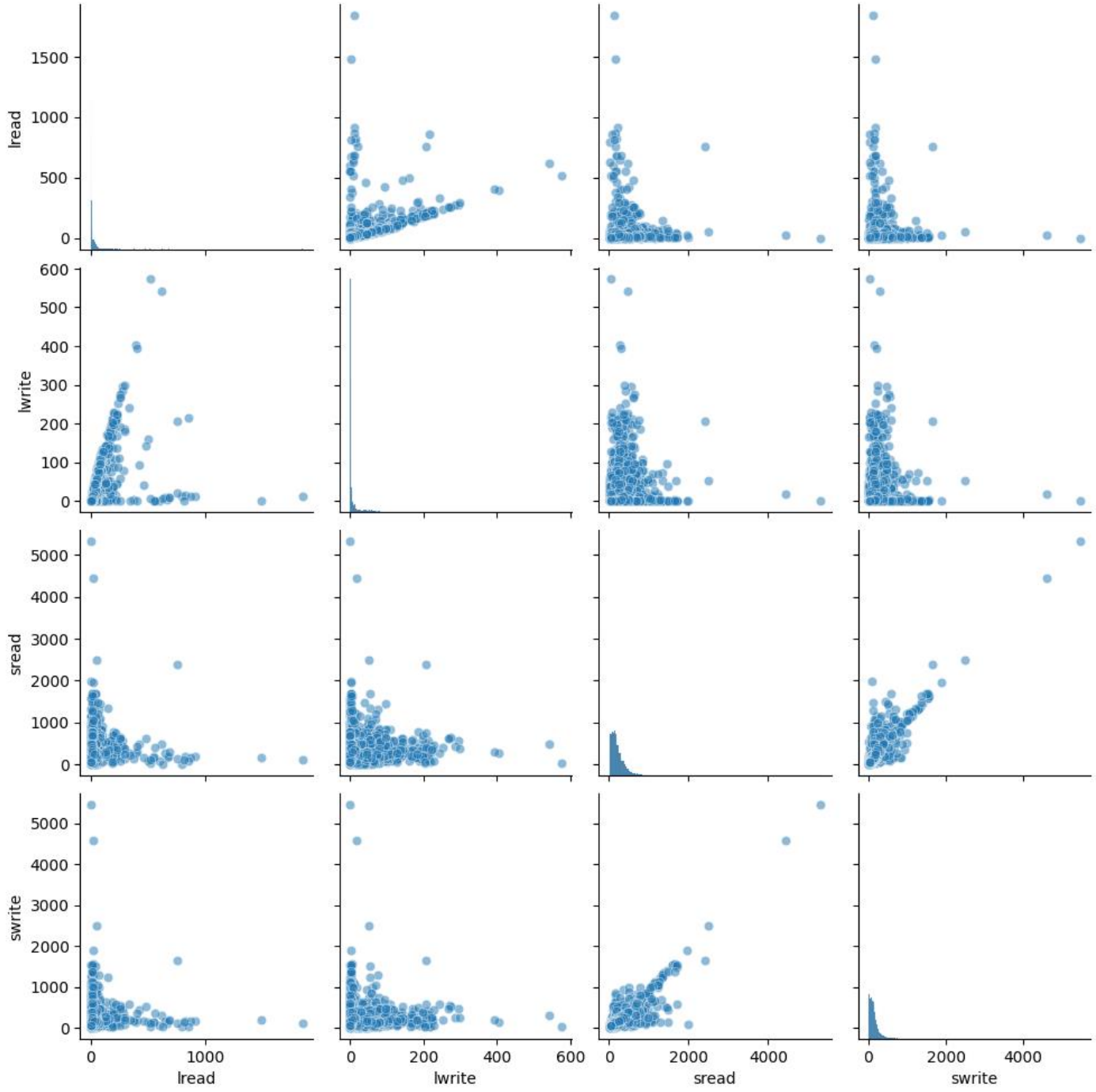
Please find below a correlation matrix of the whole set of features, we can see that the data is largely concentrated between 0.4 to -0.4 of correlation. Meaning we have moderately correlated data that spans on both ends, we do have some exceptions like 'fork' and 'exec'. In a sense this also shows that there is less collinearity.



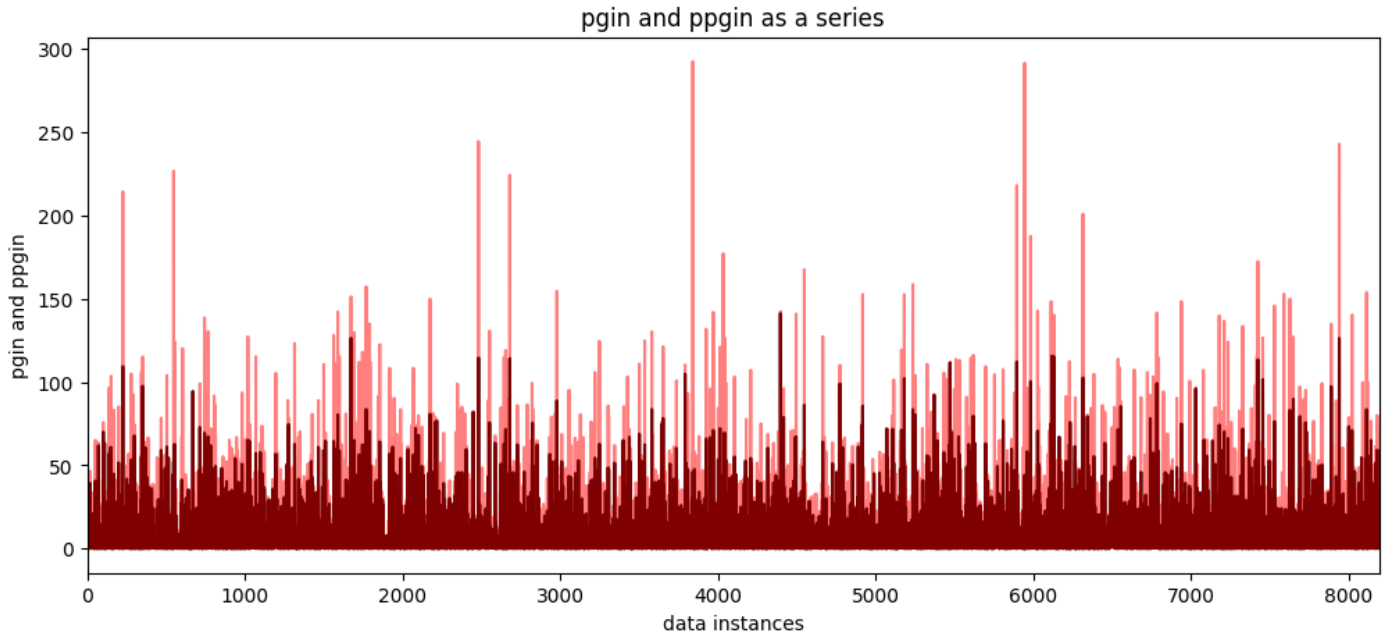
Now let's look at another image that shows the segregation of values of the only object type within this dataset. The split is close but 'Not_CPU_Bound' is ahead of 'CPU_Bound'.



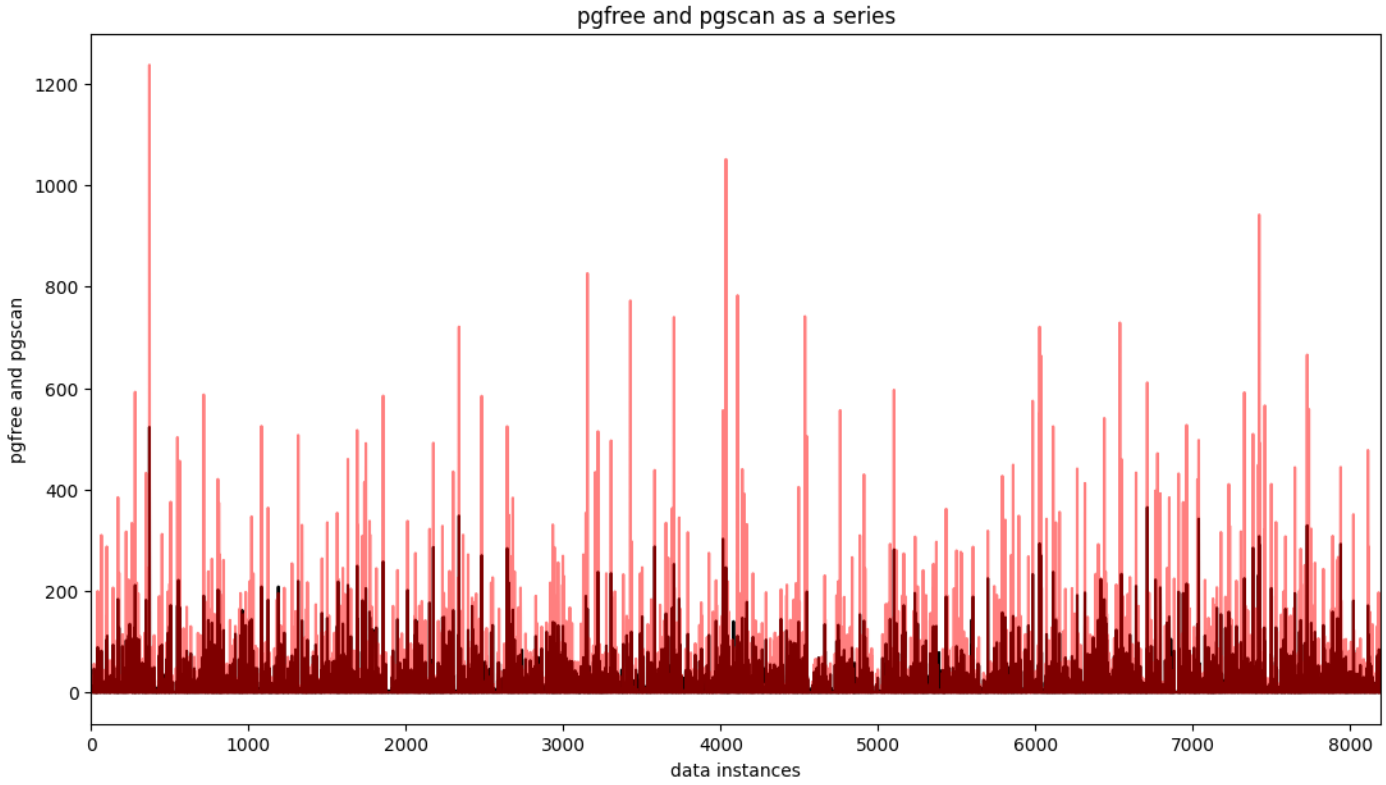
Let's, see a plot between that features 'lread', 'lwrite', 'sread', 'swrite'. This is called as a pairplot and it shows the relationships between them. As seen in the correlation plot the data is moderately correlated and thus the traffic at the lower levels.



Some of the variables can be plotted over a period. Please find the below images

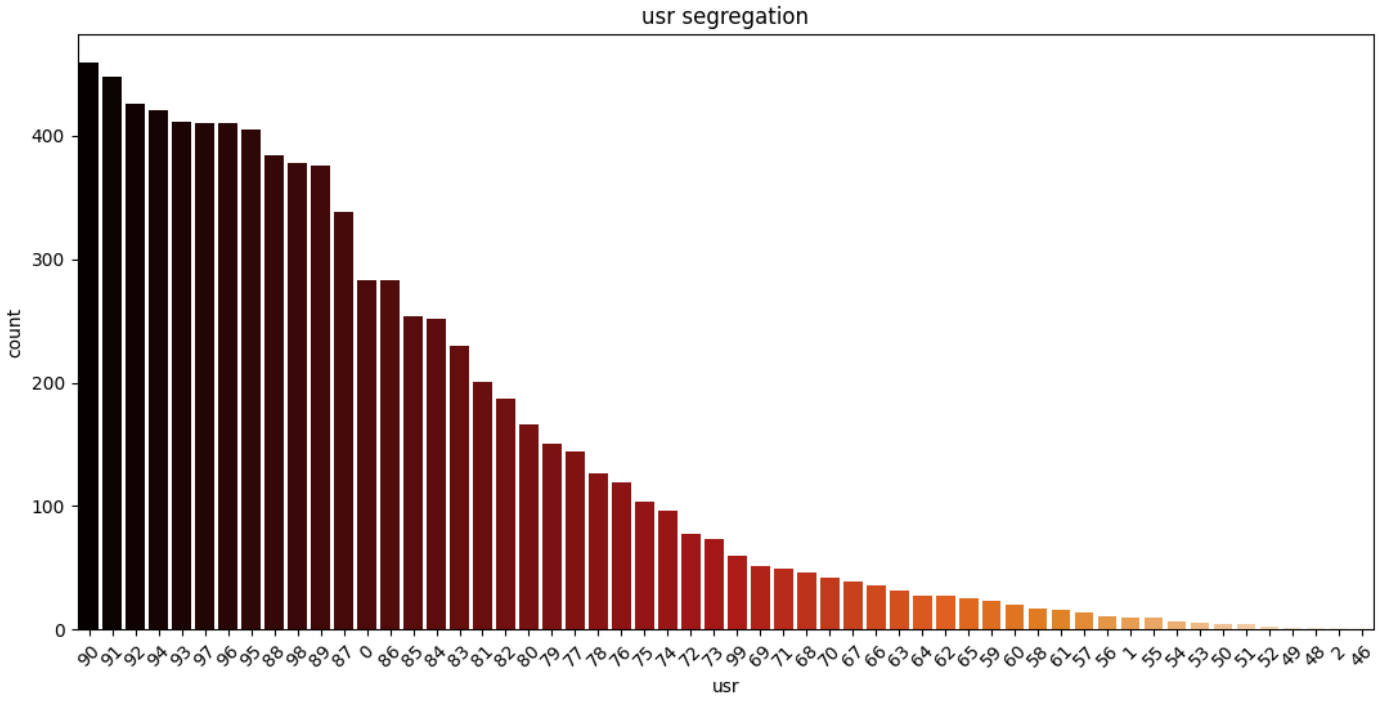


The darker shade points to 'ppgin' and the lighter one denotes 'pgin', most of the values are concentrated below the level of 50. Both pgin and ppgin do have considerable number of values that are zero. This shows that the page-in requests is not met and there is considerable portion of request's waiting in que at any point of time.



Comparing 'pgfree' which shows the number of pages that are free against the 'pgscan' which denotes the number of pages that can be freed. The darker shade corresponds to the former and the lighter shade corresponds to the latter. At any given moment we can see that the freed pages never are almost always greater.

Finally, let's have a look at the 'usr' feature.

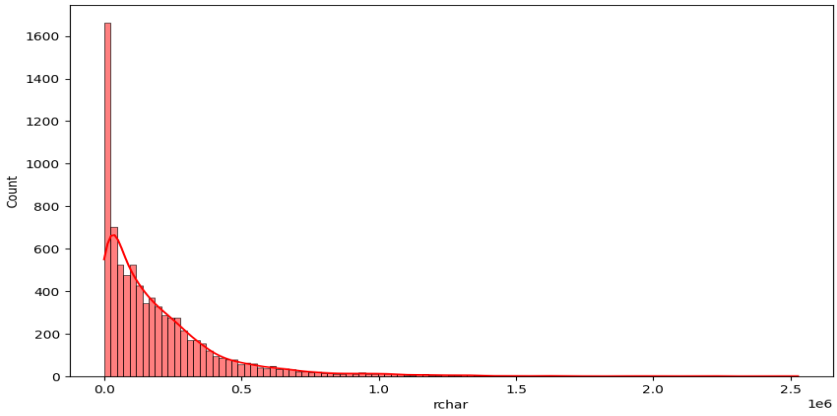


The image above, shows the value distribution, although this is a continuous variable in a mild sense the values are always between 0 and 99. We can also see that this feature has a lot of values as zero.

Part 2

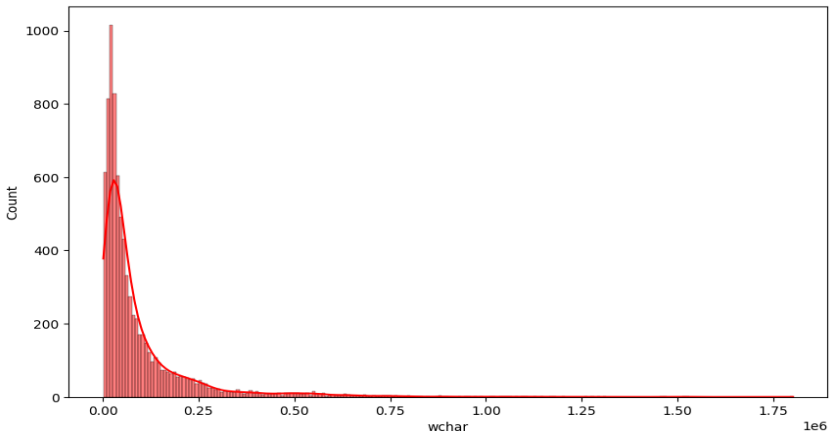
Null's, Zero's, Outliers, Duplicates

Firstly, the data does not have any duplicates when each instance is taken as a whole, this is a very good sign as this makes good use of the whole dataset.



Secondly, we do have some null values, in features 'rchar' and 'wchar'.

We can see that the data is heavily one sided and the data also extends to a considerable degree. Hence, we are going to fill the null values with median(rchar – 125473, wchar – 46619).



After imputing the median values, the decision did not hamper the original relationship and it barely moved.

The count of null values was considerably less, but not to the extent of removing those rows. Hence the decision to impute.

	rchar before	rchar after	wchar before	wchar after
mean	197386	196472.80	95902.99	95812.75
std	239837.49	238446.00	140841.71	140728.5

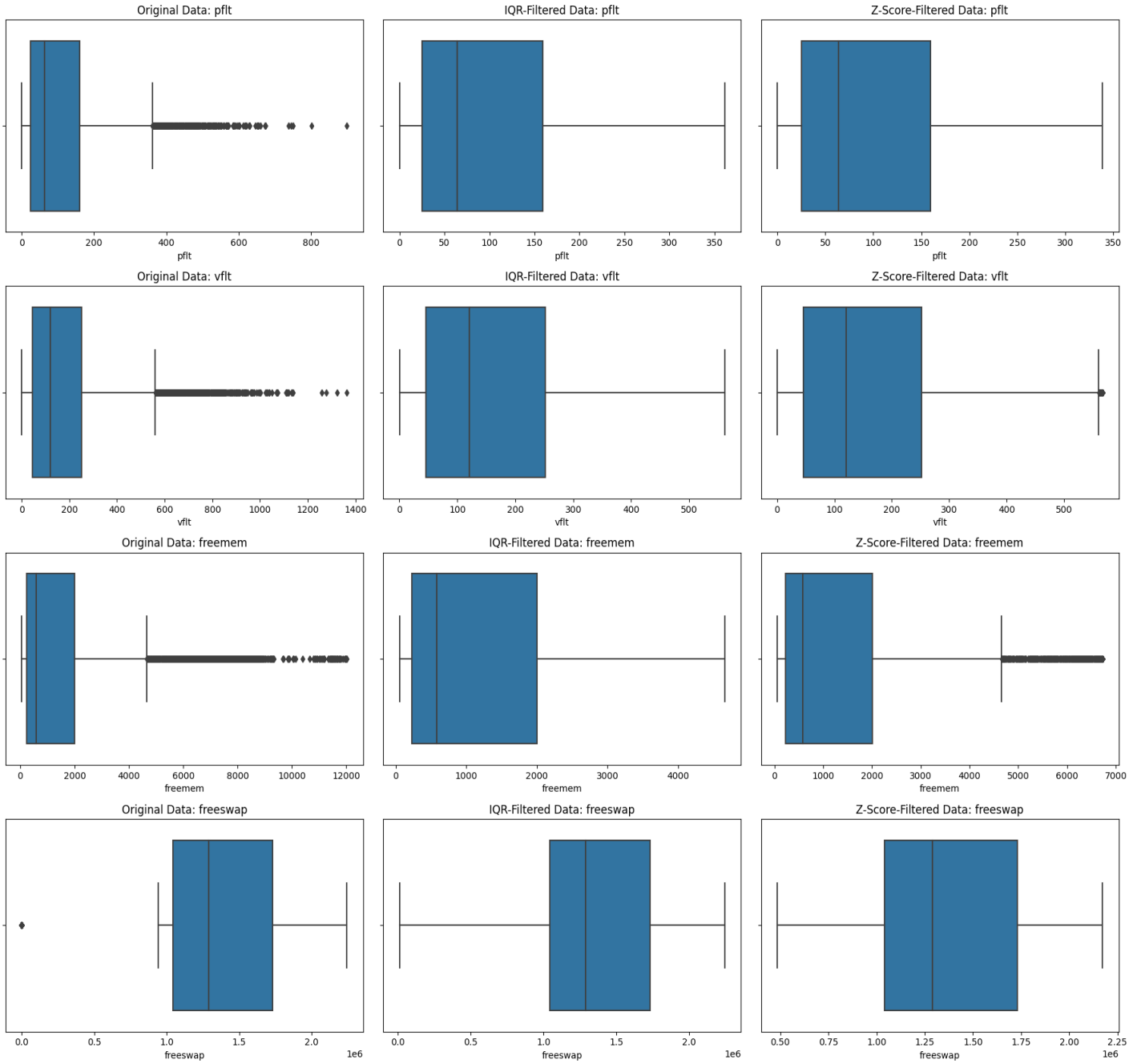
The table below shows the percentage of zero values present for a feature column.

Feature	Percentage of zero values
'lread'	8.24
'lwrite'	32.76
'scall'	0.00
'sread'	0.00
'swrite'	0.00
'fork'	0.26
'exec'	0.26
'rchar'	0.00
'wchar'	0.00
'pgout'	59.55
'ppgout'	59.55
'pgfree'	59.44
'pgscan'	78.71
'atch'	55.85
'pgin'	14.89
'ppgin'	14.89
'pflt'	0.04
'vflt'	0.00
'runqsz'	0.00
'freemem'	0.00
'freeswap'	0.00
'usr'	3.45

Features 'pgout', 'ppgout', 'pgfree', 'pgscan' and 'atch' all have 'zero' values at more than 55%, pgscan tops the list at 78%. All mentioned features can have zero values but the presence of this much, will not provide us with a helpful figure when we develop the models.

This will become insignificant when compared to those which are populated variably. Hence it is better to discard this. We do have features that are moderately populated with zero values in comparison, like lwrite, pgin, ppgin.

Now lets discuss outliers, below is an multi-plot image that shows the outliers(only a few features are provided her as sample). The boxplots on the left shows the data as it is, we can see that we have quite a number of outliers. This is expected as we had data in various scales and at extremes. All the features considered are skewed in some sense and hence using z-score method of outliers treatment is not proving to be helpful.



Inter-Quartile-Range technique has been chosen as the method to handle outliers; the skewness is the primary push.

The sample features given in the above image 'pflt', 'vflt', 'freemem', 'freeswap' all are comparatively better presented than the rest(the vertical middle order boxplots denote IQR treated features).

Feature engineering

Finally, coming to topic of engineering new features, the dataset is a kind where we can have a lot of cell values as 'zero'. This is not inherently wrong or out of order as all columns can have zeros. but we are left with models that do not capture the relationship effectively and hence will ultimately struggle to make meaningful predictions.

To get the best out of the features and to increase the performance we have chosen to square all the independent variables and then use them along with the original ones to determine the performance. Why squaring ?, squared features fundamentally enhance the variability of the dataset that we are working on. The model will be done in two progressions, one without engineered features and one with.

Along with the creation of new features, the existing ones that has most of their cell values as zero are also removed(pgout, ppgout, pgfree, pgscan, atch)

The performance metrics are discussed in the coming topics.

Part 3

Encoding of 'runqsz'

The dataset has only one object feature called as 'runqsz', we will be using the get_dummies method to change them from categorical into numerical binary versions. This basically creates two new variables(since we have two categories) one for 'CPU_Bound' and another for 'Not_CPU_Bound'. If the value is of either, then that instance will be populated with 1 and the other variable will be populated with zero. Please find a sample below.

runqsz before	runqsz after	
	runqsz_CPU_Bound	runqsz_Not_CPU_Bound
CPU_Bound	1	0
Not_CPU_Bound	0	1
Not_CPU_Bound	0	1
Not_CPU_Bound	0	1
Not_CPU_Bound	0	1

Data split for training and testing

The data is split using the train_test_split method in the ratio of 70 : 30. The training dataset has 70% of the volume and the test has the rest 30 percent. This is a random split and not under any influence.

Data setup

This is what has been done with the data till now.

- Imputed null values with median
- Removed outliers using IQR method.
- Removed features with more than 50% zeros.
- Engineered new values by squaring all independent variables.
- Encoded 'runqsz' using get_dummies function(one hot encoding).
- Training and test data split in the ratio of 70 : 30 using train_test_split.

No of features without adding any engineered columns – 17

No of Features adding engineered columns – 34

Generating the model - 1

Applying linear regression on the dataset without any engineered features. We get the following features as significant.

The significance is like a hypothesis test that is done, we check the p-value for each of the feature and if the said p-value is less than a reference value(generally it is taken as 0.05) then that feature is significant, if the value is over then if it not significant.

Feature	P-value	Significance
const	0.000	Significant
lread	0.000	Significant
lwrite	0.038	Significant
scall	0.798	Insignificant
sread	0.781	Insignificant
swrite	0.000	Significant
fork	0.001	Significant
exec	0.000	Significant
rchar	0.000	Significant
wchar	0.000	Significant
pgin	0.053	Borderline
ppgin	0.000	Significant
pflt	0.000	Significant
vflt	0.631	Insignificant
freemem	0.000	Significant
freeswap	0.000	Significant
runqsz_CPU_Bound	0.000	Significant
runqsz_Not_CPU_Bound	0.000	Significant

The cell colours in the shades of orange are insignificant, while the cell shaded in blue is right on the value. This means that the addition of the value is at our discretion.

The below table shows the performance characteristics of the model.

Data/Data	Statsmodel		Scikit learn	
	Test	Train	Test	Train
R-squared	0.7890	0.7890	0.7740	0.7894
Mean squared error	0.2242	0.2110	0.2243	0.2110
Adjusted R-square	0.7890	0.7890	0.7739	0.7887
Root Mean squared error	0.4735	0.4593	0.4736	0.4594

From the above stats we can see that R-squared is close to 0.78 to 0.79 in both the models, this shows that a good number of relationships are generated, and we have a decent model in place.

The Root Mean Square Error is also close to zero(we must mind that we have standardized the data and it looks different to the ones that we have been provided or the original). This fundamentally means that we have with us a model that is predicting values to a considerable degree of trustworthiness.

The R-squared and RMSE are slightly better for the training set than for the test set, this is a very mild indicator of the data being overfitted. The difference is tiny. Hence, we are good.

Generating the model – 2

Applying linear regression on the dataset without any engineered features. We get the following features significance values.

Feature	P-Value	Significance
lread	0.000	Significant
lwrite	0.019	Significant
scall	0.000	Significant
sread	0.046	Significant
swrite	0.000	Significant
fork	0.317	Not Significant
exec	0.273	Not Significant
rchar	0.000	Significant
wchar	0.000	Significant
pgin	0.007	Significant
ppgin	0.083	Not Significant
pflt	0.000	Significant
vflt	0.000	Significant
freemem	0.000	Significant
freeswap	0.000	Significant
runqsz_CPU_Bound	0.000	Significant
runqsz_Not_CPU_Bound	0.000	Significant
lread_squared	0.077	Not Significant
lwrite_squared	0.299	Not Significant

Feature	P-Value	Significance
scall_squared	0.007	Significant
sread_squared	0.339	Not Significant
swrite_squared	0.782	Not Significant
fork_squared	0.000	Significant
exec_squared	0.000	Significant
rchar_squared	0.422	Not Significant
wchar_squared	0.096	Not Significant
pgin_squared	0.229	Not Significant
ppgin_squared	0.612	Not Significant
pflt_squared	0.089	Not Significant
vflt_squared	0.000	Significant
freemem_squared	0.000	Significant
freeswap_squared	0.000	Significant
runqsz_CPU_Bound_squared	0.000	Significant
runqsz_Not_CPU_Bound_squared	0.000	Significant

Engineering of values does seem to be useful to some extent, we can see that we have some more variables that are proving to be significant which can help in extending the models performance. The same colour schemes apply here too, as previously seen.

Now lets us see this model that has been trained with engineered data performs, please find the table below discussing the performance characteristics.

Data/Data	Statsmodel		Scikit learn	
	Test	Train	Test	Train
R-squared	0.9030	0.9030	0.8958	0.9031
Mean squared error	0.1036	0.0970	0.1036	0.0970
Adjusted R-squared	0.9030	0.9030	0.8943	0.9025
Root Mean squared error	0.3219	0.3115	0.3219	0.3115

From all aspects we can see a major jump in the performance of the model, there is roughly 14% increase in the R-squared section for both test and training parts. Indicating a better fit than model – 1.

Mean Square Error (MSE) and Root Mean Square error (RMSE) values has decreased in the engineered model for both the test and training sets, indicating the model's predictions are closer to the actual values.

This model, like the previous one is also plagued with a mild overfitting stance. The values are again very less, but it indicates a presence.

With this we can clearly go with the model – 2. This does come at the cost of pre-processing the data to a great extent but the performance pay's off in the end.

Part 4

Summary

This project involved several key steps to go from raw data to obtaining information. These steps include:

Exploratory Data Analysis: The initial step was to understand the dataset's structure, features, and summary statistics. This helped in laying the groundwork for the subsequent phases.

Data Preprocessing: Missing values were imputed using the median, outliers were capped, and new features were engineered. This ensured that the data was prepared for the modelling phase.

Model Building: Linear Regression models were built and evaluated based on R-squared, RMSE, and Adjusted R-squared. The best model (Model 2 – Engineered features) had a good fit and predictive power.

Variable Significance: Significant variables were identified to understand their impact on the target variable 'usr'.

Inference

- Most influential features are – Freeswap, pflt, scall, and the engineered feature Freeswap_squared. These features should be closely monitored for optimal performance.
- We had some features like 'pgfree' and 'pgscan' which had majority of values as 'zeros', the significance of these features will always be a hit or miss and they are also not compliant for any further processing.
- 'usr' is always focussed on values between 99 and 75 or 0 for most instances. Assuming '0' to be powered down or sleep mode, we can see that majority of the time the system runs in user mode. Moreover, the top ten values were 90 or above.
- The dataset provided did not have any duplicates, hence the current recording process is successful in weeding out duplicates.
- Since the features are off varied ranges, we had to scale the data hence pre-processing is a must.
- Linear regression is not the best model is decent, engineering features enhanced the performance to a great extent but at the cost of time and effort(manual and compute units). Hence other models should be explored before concluding.

Problem 2 Statement :-

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

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

DATA DICTIONARY:

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

Part I

Data Frame Briefing

The data dictionary gives an outlook on what we are the data types within. There are only three numerical columns while the rest are categorical features. There are no key's or identification columns of any kind.

SOME STATISTICS:

Wife_age: Varies from 16 to 49 years, with an average age of around 32.6 years.

Wife_ education: Most commonly 'Tertiary' education.

Husband_education: Also, most commonly 'Tertiary' education.

No_of_children_born: Ranges from 0 to 16, with an average of about 3.25 children.

Wife_religion: Mostly 'Scientology'.

Wife_Working: Majority come under as ‘not working’.

Husband_Occupation: Average occupation level is around 2.14, making it between 2 or 3. This is a categorical column but one that is captured in numerical values.

Standard_of_living_index: Most common is 'Very High'.

Media_exposure: Majority are 'Exposed'.

Contraceptive_method_used: More instances of 'Yes' than 'No'.

++

Total instances of data – 1473 and total number of columns is 10.

++

NULLS –

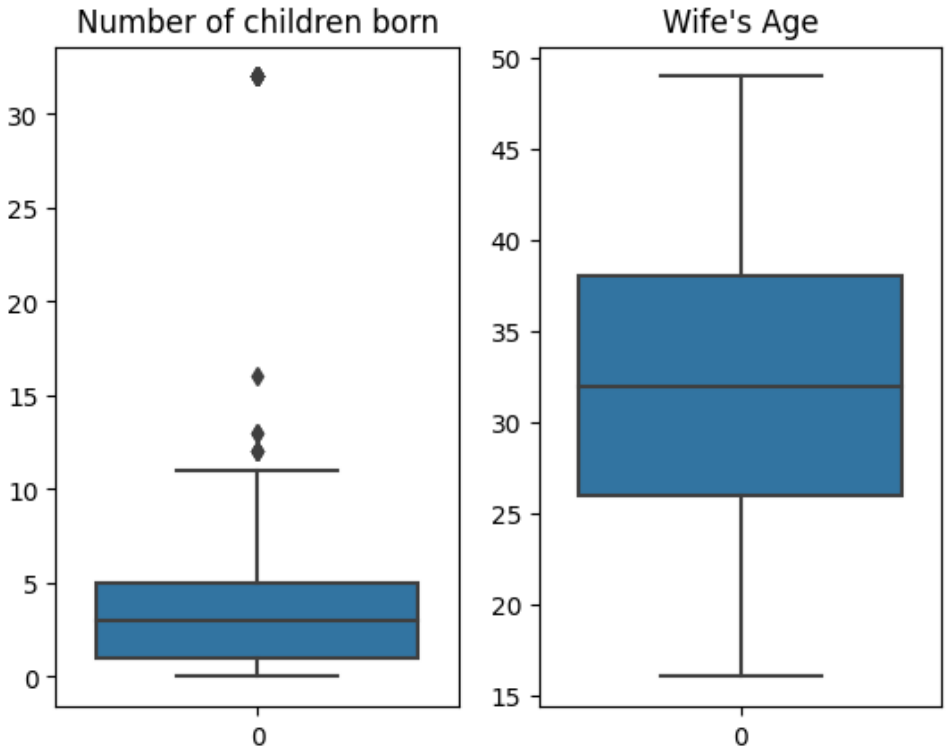
We do have Null values in the features 'Wife_age'(71 instances) and 'No_of_children_born'(21 instances). All of them will be filled with corresponding median values .

DUPLICATES –

This dataset has a total of 80 duplicated instances, this is when we consider the data frame, hence we need to get this treated before we attempt any of the model generation or testing. If there was a key or unique identifier column, we could assume these duplicates as legitimate entries.

OUTLIERS –

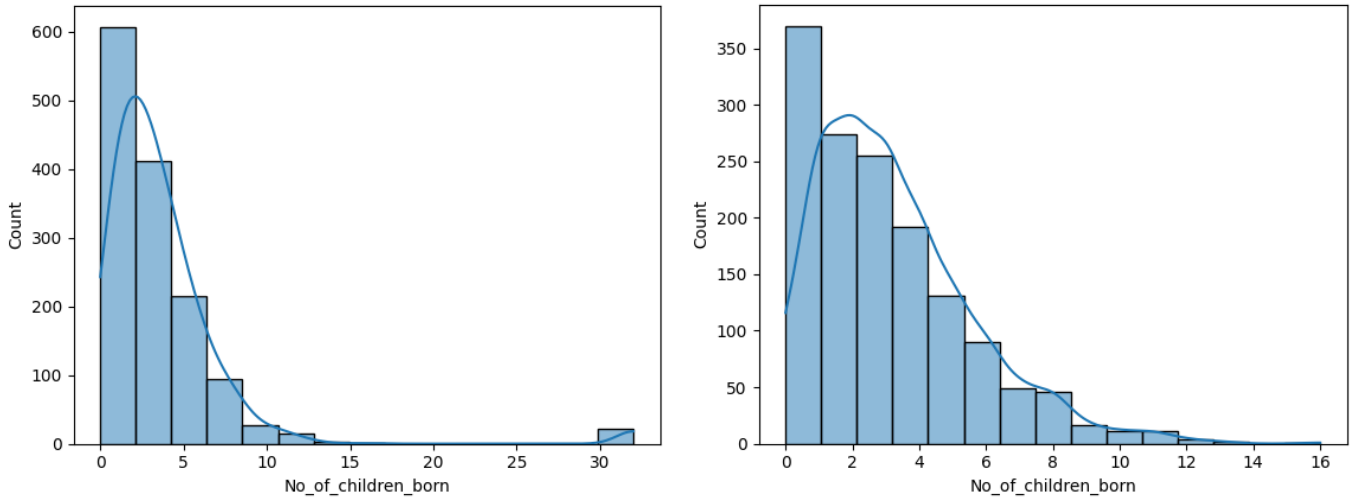
We have only two continuous numerical features, the boxplots of the same are given below.



Upon further analysing, there are a few values that are 32.0 in the feature 'No_of_children_born'. Hence it has popped in the boxplot above.

We do know of societies of the past, even now in certain parts of India, of families having 8 to 10 children. So having large families is not new. Though 16 is on the higher side and rare it did occur in our near past. Hence under this pre-text we are only going to deal with the values that are 32.0.

We will be imputing the outlier values with median as the dataset is skewed.



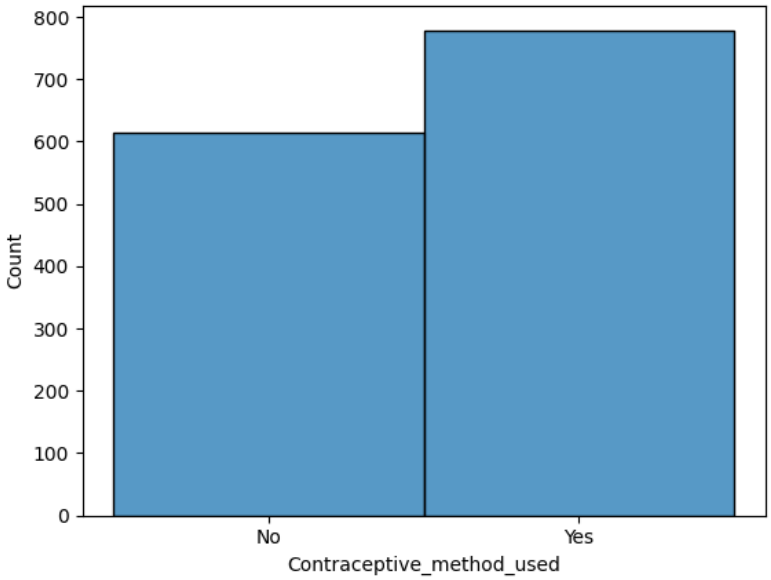
The left image shows the number of children before outlier treatment, we can see a small hill at 30+ bar, the right side is the same feature after removing those by imputing the median in place of them.

We have not completely removed the outliers in the column, just the ones that are impractical.

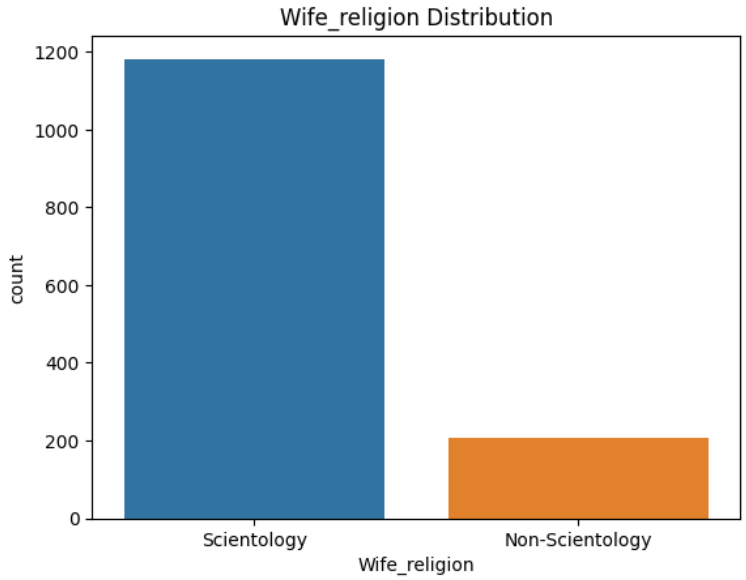
All the other features are categorical in nature; hence we will not be having outliers, but we can have very one-sided distribution of values.

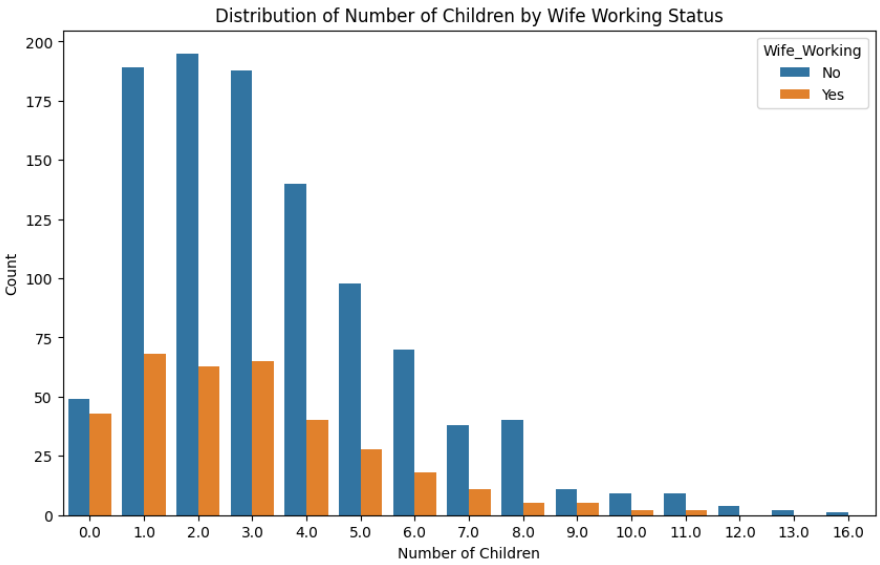
Variables and Relationships

Let's have a look at the target variable and its values count. The count of 'Yes' is high in number, but it is not overwhelmingly high over the other category.



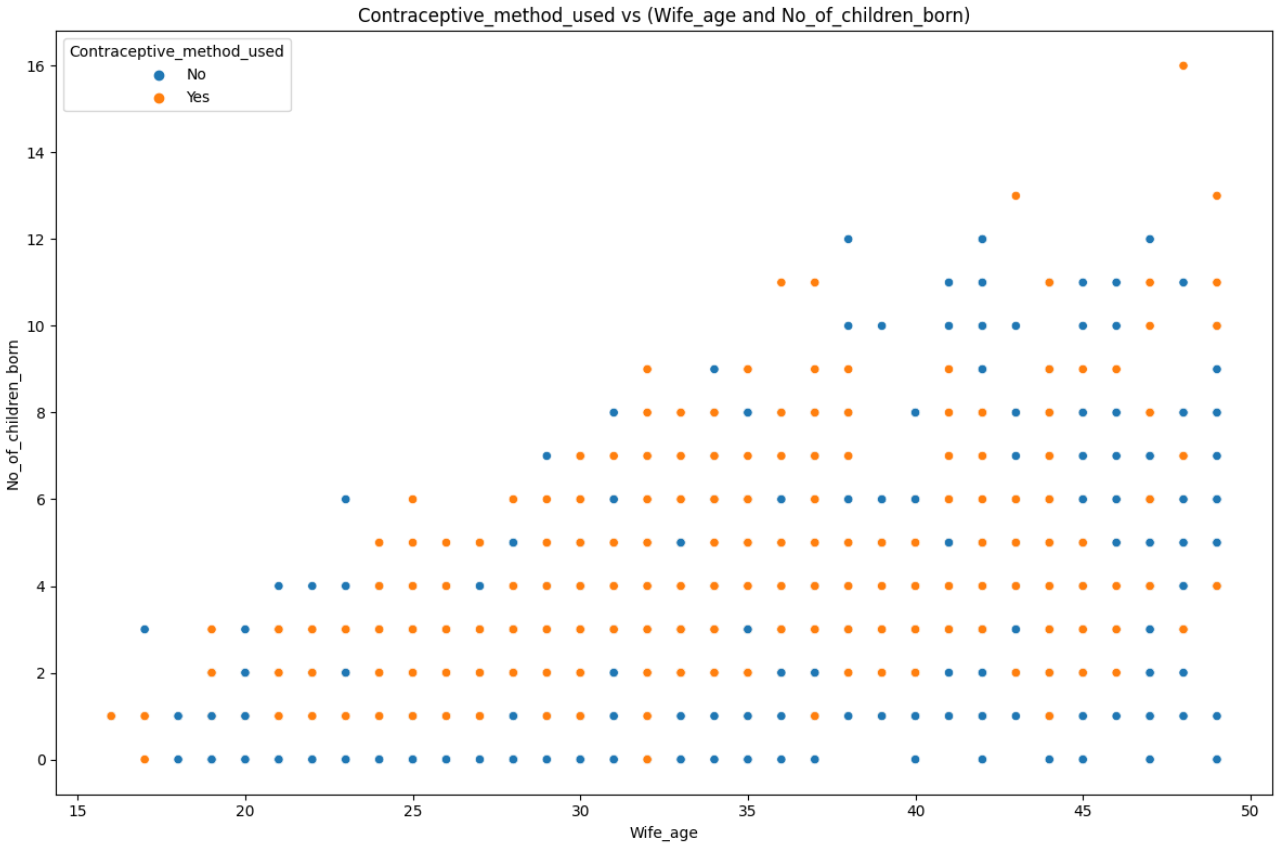
The below image shows the distribution of a wife's faith. Unlike the target variable above we can see that the scientology is overwhelmingly high, this might have an affect on the model as there is not much to support category 'Non-Scientology'.





Wife working and the number of children born shows reveals a very surprising information. Women who are not working have high count of children and top the list in all ages. Women who do work, though not matching the scale, are present in all the categories except in values of 12 to 16. Having 1 through 3 children takes the high spots.

The below scatter plot shows the distribution of the number of children born when paired against age, basing on contraceptive method. Younger women with fewer children seem to be more inclined towards using contraceptives. Older women with more children are a mixed bag. Middle-Aged Women are also mixed but leans slightly more towards using contraceptives.



Part 2

Encoding / Splitting

All the categorical values will be changed to numerical values as this helps the models to perform better.

We will combine the columns 'Wife_ education' and 'Husband_education' into one, we will encode them with unique odd values and add them together, thereby engineering a new feature.

Scribing used – 'Uneducated': 0, 'Primary': 3, 'Secondary': 5, 'Tertiary': 7

Value counts to show the results,

Category	Count
14	477.00
10	255.00
12	248.00
8	153.00
6	87
3	71
5	48
0	26
7	23

14 here represents both husband and wife having till tertiary education. Which takes the top spot.

We will also be using train_test_split method to split the data into 70 : 30 ratios. 70 being the train data and 30 being the test data.

Note: One thing about introducing numerical counterparts for categorical variables is that it might involuntarily bring in order, like higher and lower.

Applying Logistic Regression

Applying logistic regression keeping max iterations as 10000 and no penalty, we get the accuracy score to be 67.3 for training and 60.9 for the test dataset. The model successfully converged at 80 iterations.

Applying Linear Discriminant Analysis

Applying Linear Discriminant Analysis, we get the accuracy score of the train and test dataset to be 67.4 and 61.8.

Applying CART – Decision Tree Classifier

Applying CART – Decision Tree Classifier, we get the accuracy of the train and test dataset to be 67.4 and 61.8 respectively.The below table shows the order of most influential features, in descending order. The engineered feature 'Wife_Husband_education' takes the third spot, it has been moderately helpful. The two mildly continuous features take the top spots over the other categorical features.

Feature Name	Importance's
Wife_age	0.328
No_of_children_born	0.231
Wife_Husband_education	0.154
Standard_of_living_index	0.108
Husband_Occupation	0.089
Wife_Working	0.049
Wife_religion	0.028
Media_exposure	0.013

Part 3

Performance Metrics

The below table discusses the key values to determine the performance of the models we have used.

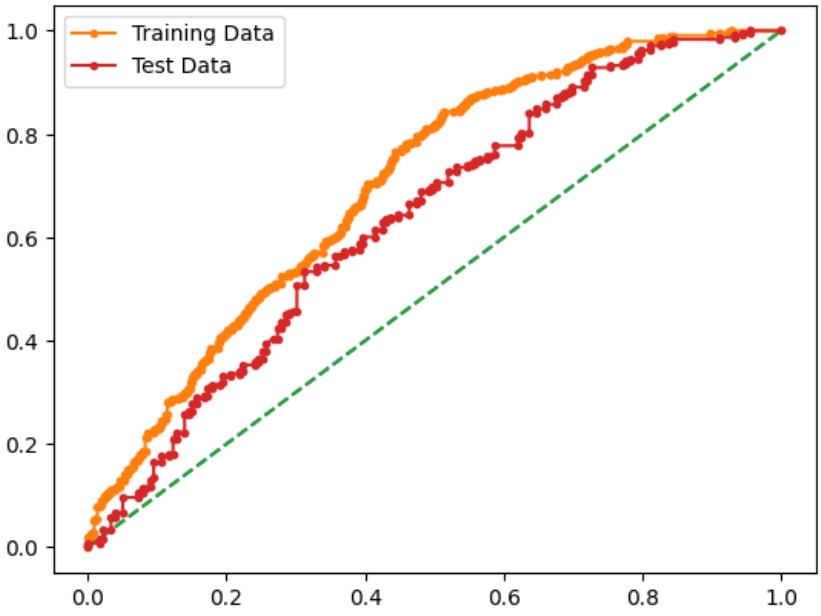
Data/Data		Accuracy	Precision(0/1)	Recall(0/1)	AUC score
Logistic regression	Train	0.667	0.66/0.66	0.53/0.79	0.702
	test	0.609	0.57/0.63	0.39/0.78	0.638
Linear discriminant Analysis	train	0.674	0.69/0.67	0.50/0.81	0.701
	test	0.618	0.59/0.63	0.37/0.80	0.636
Cart – Decision tree classifier	train	0.674	0.97/1.00	1.00/0.97	0.701
	test	0.618	0.52/0.64	0.55/0.62	0.636

The below table discusses the Confusion matrix and its sections.

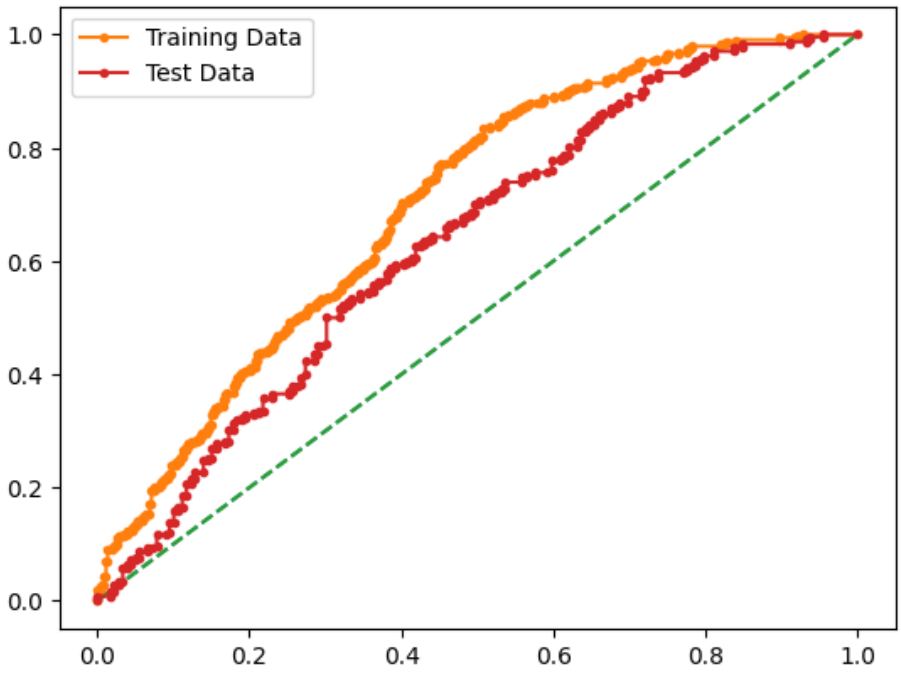
Data/Data		True Negative	False Positive	False negative	True positive
Logistic regression	Train	229	206	111	425
	test	68	110	53	185
Linear discriminant Analysis	train	219	216	100	436
	test	67	112	47	191
Cart – Decision tree classifier	train	434	1	14	522
	test	98	81	91	147

ROC Curve's

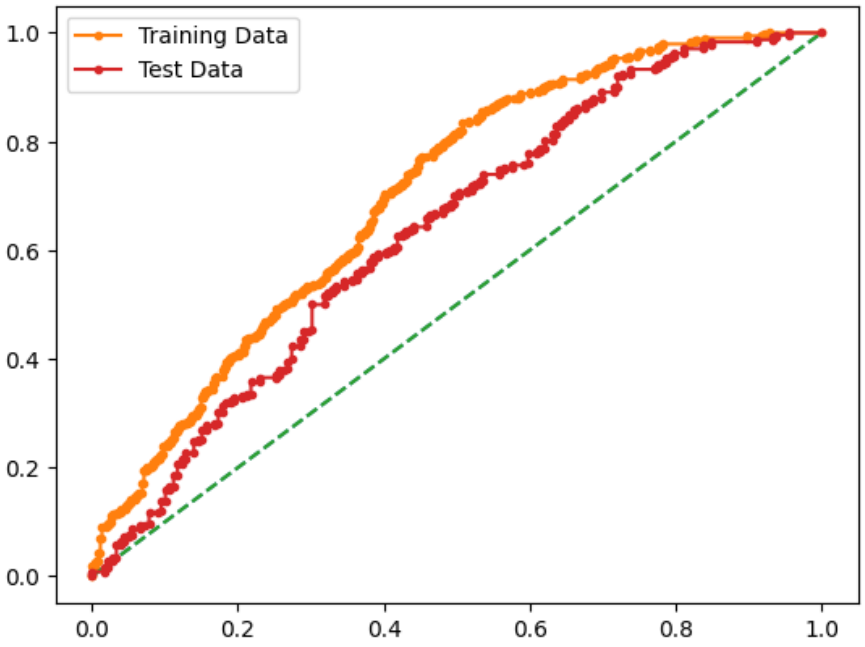
ROC Curve for Logistic regression is provided below.



The ROC curve for Linear discriminant Analysis is provided below,



The ROC Curve for CART – Decision Tree Classifier is provided below.



Final Model Comparison:

Logistic Regression and **LDA**: Similar performance on both train and test sets. Both models are stable but perform only moderately well.

CART: Performs exceptionally well on the train set but only moderately well on the test set. Indicates possible overfitting.

AUC Scores: All three models have AUC scores around 0.6 for the test and 0.7 on the training data, indicating that they are slightly better than random guessing but not great classifiers.

Thereby concluding that while all the models are prone to overfitting **CART** takes it the extreme. But this is also the one that could perform the best on the test set but is prone to overfitting and might require heavy processing and setup.

Logistic Regression and **LDA** are more stable but offer moderate performance, based on the minimal data processing that we have done.

Given the metrics, **Logistic Regression** or **LDA** could be a more reliable choice for this problem, considering they are less prone to overfitting than **CART**.

Part 4

Summary

Data Preparation and Analysis: The dataset was ingested and explored, revealing demographic and lifestyle factors that could influence contraceptive usage. We also treated missing values, outliers, and duplicate entries to clean the dataset.

Model Building and Performance: Logistic Regression, LDA, and CART were the chosen algorithms. All three models showed similar performance, with slight nuances in their reliability and stability.

Inference

- Most influential features – Wife's education, Number of children born, Standard of living and the engineered feature wife and husband education.
- Younger women, especially those with fewer children, are more likely to use contraceptives.
- Both men and women in the dataset are varied between well-educated and no education, suggesting that education does not necessarily correlate with contraceptive use in this case.
- Religious beliefs and the working status of the wife also did not show a strong correlation with contraceptive use.
- Data provided had duplicates close to 5.5% of the total volume.
- Data provided did also have some impractical outliers, indicating a possible error in data collection.
- CART was overfitting while training and hence it did not prove to be a very useful model. Linear Discriminant Analysis and Logistic Regression had similar results and did perform moderately high.
- Since we had duplicates and the data is a one, such that the possibility of duplicates occurring is high, it is better to run them in models that are robust to handling duplicates.