IDTA COURSEWORK_1

ID-2293815

Overview

I am using the **Census** dataset for Machine Learning tasks, which has in total 569740 rows and 19 columns.

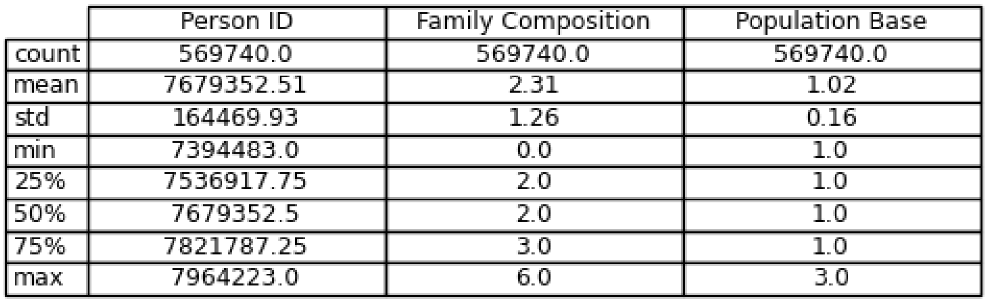
1. (a) Display and interpret basic statistics for all attributes.

For understanding and interpreting the statistics of the attributes, I removed the outlier which was the -9 values with 0. By doing that, I am neutralizing the effect of these outliers had on the data, they could have showed the misguided data on mean, median. By replacing it with 0, the statistics will better reflect on the central tendency of the majority of the data points.

- **Numerical Attributes:** For numerical attributes we get the following values for each attribute, and this is what they represent-
  - **Count:** It represents the total number of rows in the dataset.
  - **Mean:** The average value of that attribute.
  - **Standard Deviation(std):** It measures the spread of values around the mean, greater std value suggests, the data points are quite far from the mean value.
  - **Min:** It is the smallest value in the column.
  - **25% (1st quartile):** The value under which 25% of the data of that column falls.
  - **50% (Median):** The middle value of the column when the values of the column are sorted. 50% of the data points are above this and other 50% are below this median value.
  - **75% (3rd quartile):** The value under which 75% of the data of that column falls.
  - **Max:** The largest value of that column.

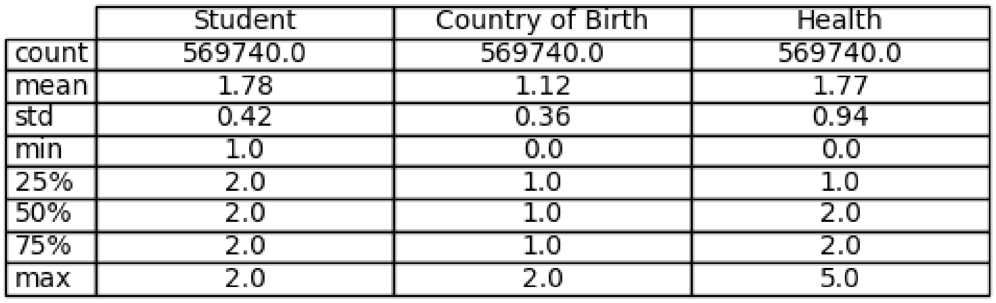
  Figure 1, Figure 2, Figure 3, Figure 4, Figure 5, Figure 6 describes the basic statistics for all the numerical attributes.

Figure 1 Basic Statistics for Person ID, Family Composition and Population Base



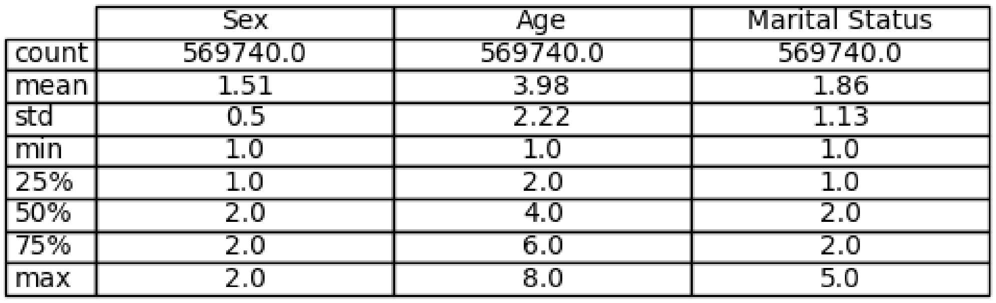
	Person ID	Family Composition	Population Base
count	569740.0	569740.0	569740.0
mean	7679352.51	2.31	1.02
std	164469.93	1.26	0.16
min	7394483.0	0.0	1.0
25%	7536917.75	2.0	1.0
50%	7679352.5	2.0	1.0
75%	7821787.25	3.0	1.0
max	7964223.0	6.0	3.0

Figure 2 Basic Statistics for Student, Country of Birth and Health



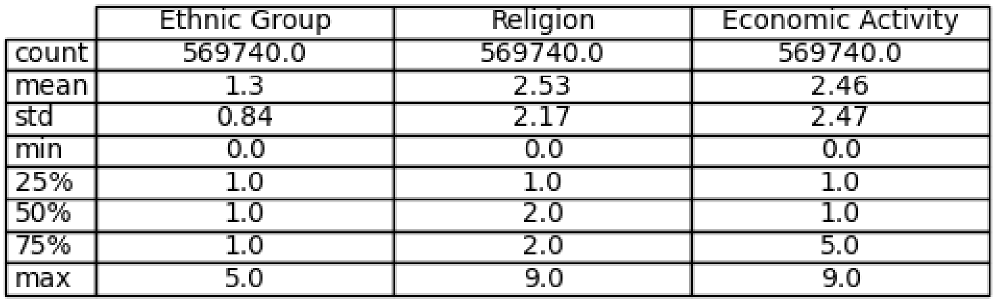
	Student	Country of Birth	Health
count	569740.0	569740.0	569740.0
mean	1.78	1.12	1.77
std	0.42	0.36	0.94
min	1.0	0.0	0.0
25%	2.0	1.0	1.0
50%	2.0	1.0	2.0
75%	2.0	1.0	2.0
max	2.0	2.0	5.0

Figure 3 Basic Statistics for Sex, Age and Marital Status



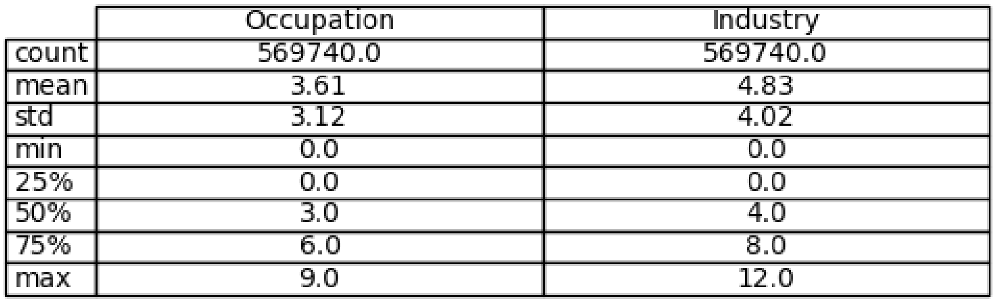
	Sex	Age	Marital Status
count	569740.0	569740.0	569740.0
mean	1.51	3.98	1.86
std	0.5	2.22	1.13
min	1.0	1.0	1.0
25%	1.0	2.0	1.0
50%	2.0	4.0	2.0
75%	2.0	6.0	2.0
max	2.0	8.0	5.0

Figure 4 Basic Statistics for Ethnic Group, Religion and Economic Activity



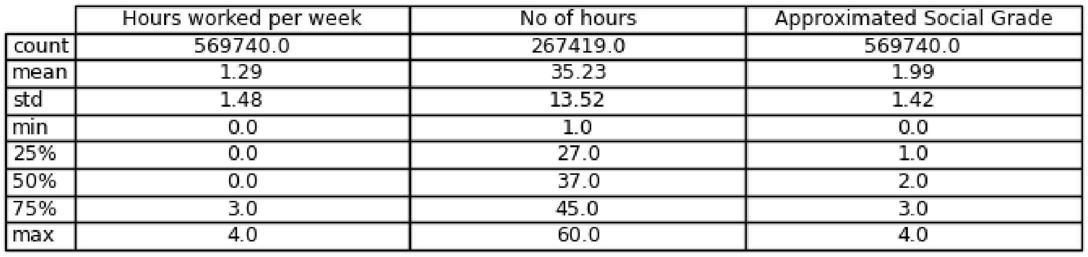
	Ethnic Group	Religion	Economic Activity
count	569740.0	569740.0	569740.0
mean	1.3	2.53	2.46
std	0.84	2.17	2.47
min	0.0	0.0	0.0
25%	1.0	1.0	1.0
50%	1.0	2.0	1.0
75%	1.0	2.0	5.0
max	5.0	9.0	9.0

Figure 5 Basic Statistics for Occupation and Industry



	Occupation	Industry
count	569740.0	569740.0
mean	3.61	4.83
std	3.12	4.02
min	0.0	0.0
25%	0.0	0.0
50%	3.0	4.0
75%	6.0	8.0
max	9.0	12.0

Figure 6 Basic Statistics for Hours worked per week, No of hours and Approximated social grade

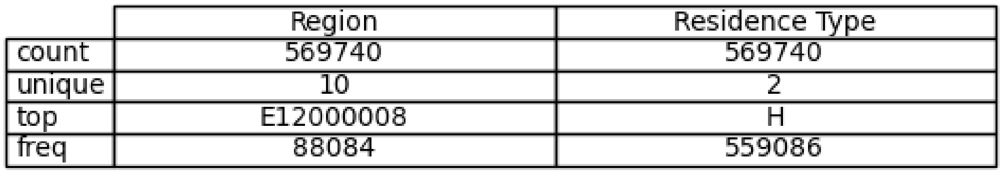


	Hours worked per week	No of hours	Approximated Social Grade
count	569740.0	267419.0	569740.0
mean	1.29	35.23	1.99
std	1.48	13.52	1.42
min	0.0	1.0	0.0
25%	0.0	27.0	1.0
50%	0.0	37.0	2.0
75%	3.0	45.0	3.0
max	4.0	60.0	4.0

- **Categorical Attributes:** For categorical attributes we get the following values for each attribute, and this is what they represent-
  - **Count:** It shows the total number of entries in that column.
  - **Unique:** Indicates the number of distinct values in that column.
  - **Top:** Specifies the value which most occurs in that column.
  - **freq(Frequency):** It specifies the number that how many time the top value occurs in that column.

  Figure 7 describes the basic statistics for the two categorical attributes in the dataset.

Figure 7 Basic Statistics for Region and Residence Type

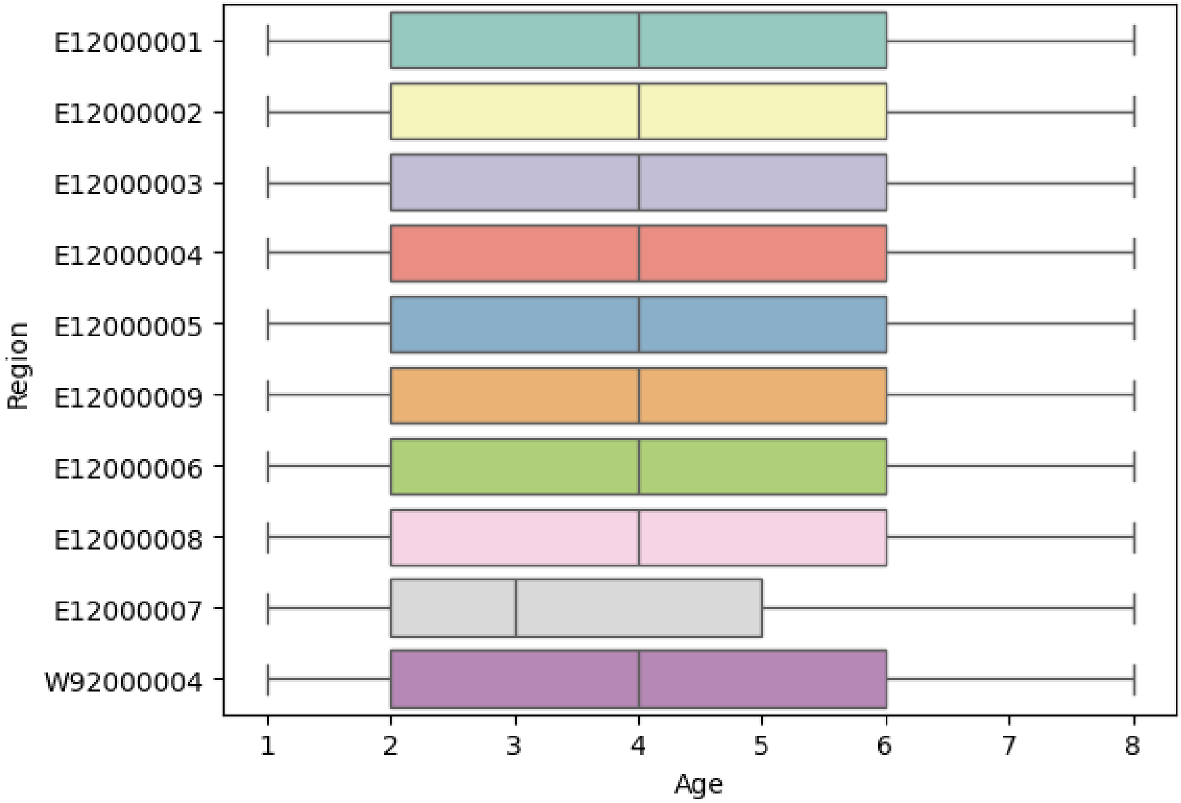


	Region	Residence Type
count	569740	569740
unique	10	2
top	E12000008	H
freq	88084	559086

1. (b) Display and interpret 5 visualization graphs; include one box plot and four visualizations that show relationships between 2 or more variables.

   1. **Box Plot (Numerical and Categorical attributes):**

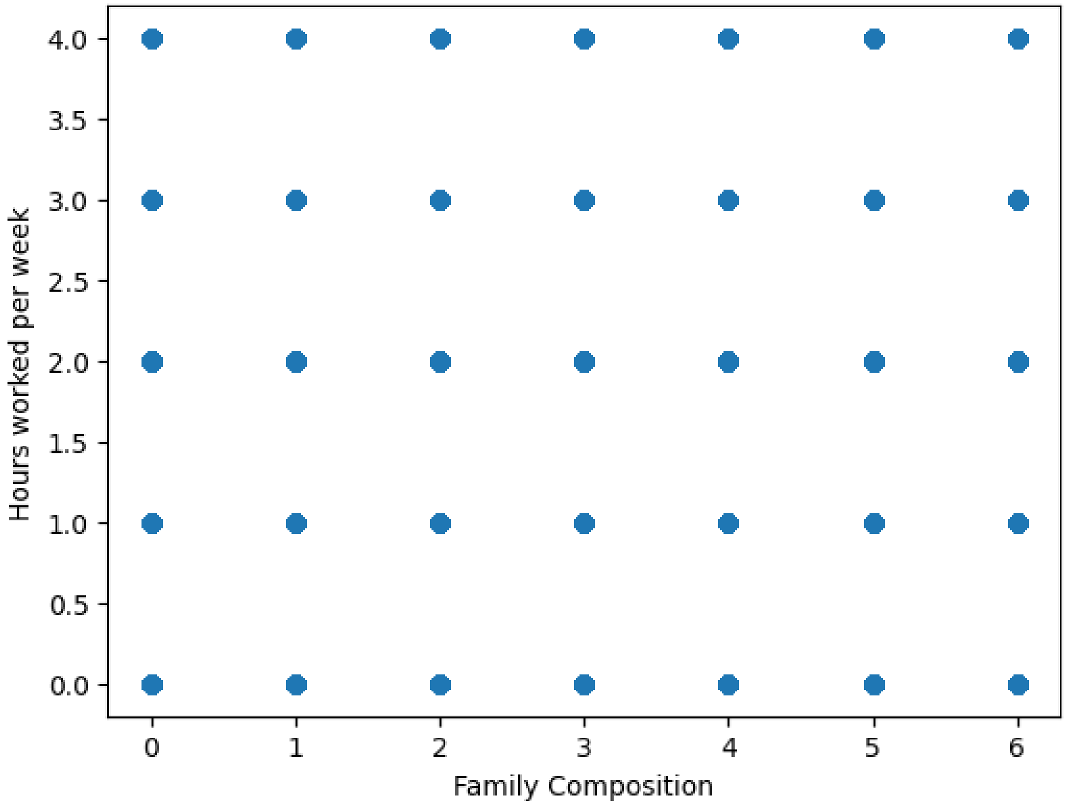
Figure 8 Box Plot for Numerical and Categorical Attributes



From Figure 8, Region E12000007 has the widest age range, with ages spanning from approximately 3 to 7. Region E12000001 has a median age of around 4.5, and the range is smaller compared to other regions. Region W92000004 shows a lower median age of around 3.5, with its distribution indicating younger populations overall.

2. Scatter Plot (Numerical attributes):

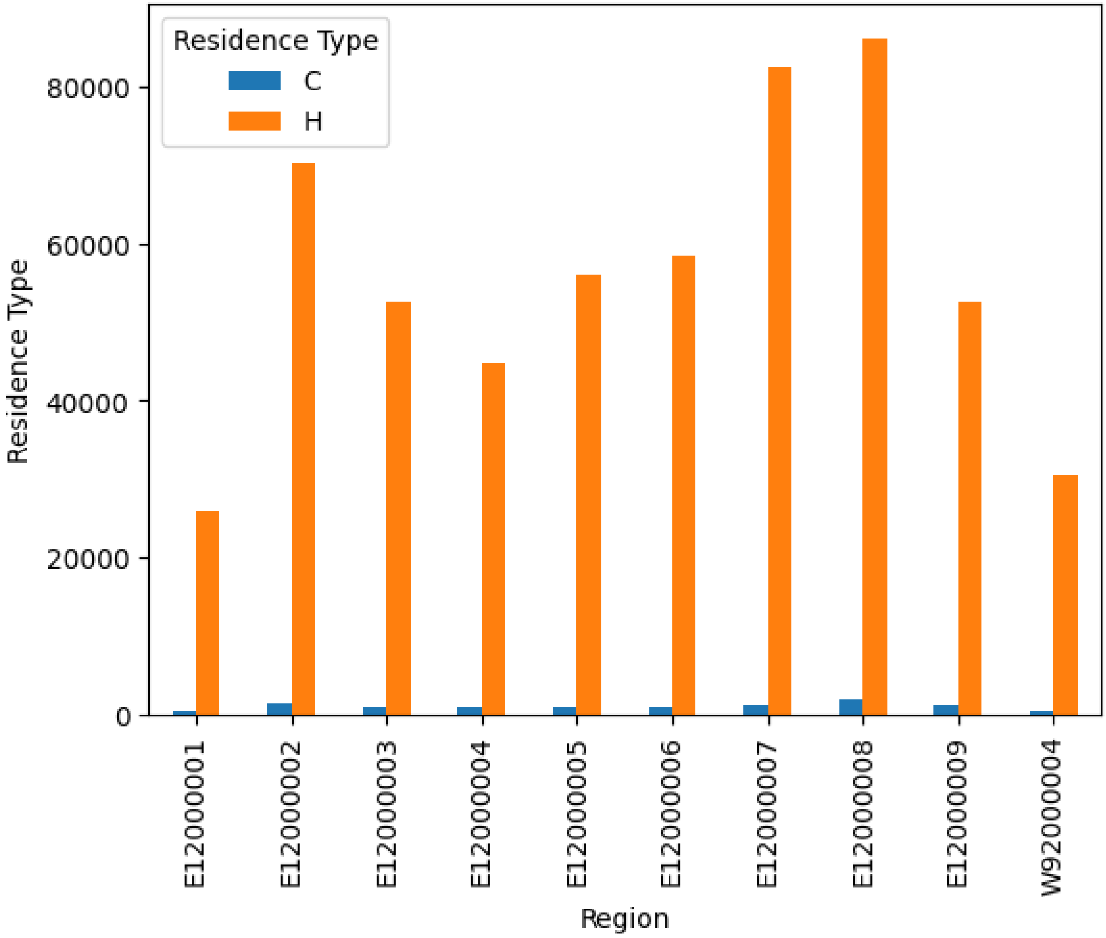
Figure 9 Scatter Plot for Numerical Attributes



In Figure 9, for each family composition category, all four hours-worked categories (0, 1, 2, 3, 4) are equally represented. There is no linear pattern or correlation between family composition and hours worked per week.

3. Bar Plot (Categorical attributes):

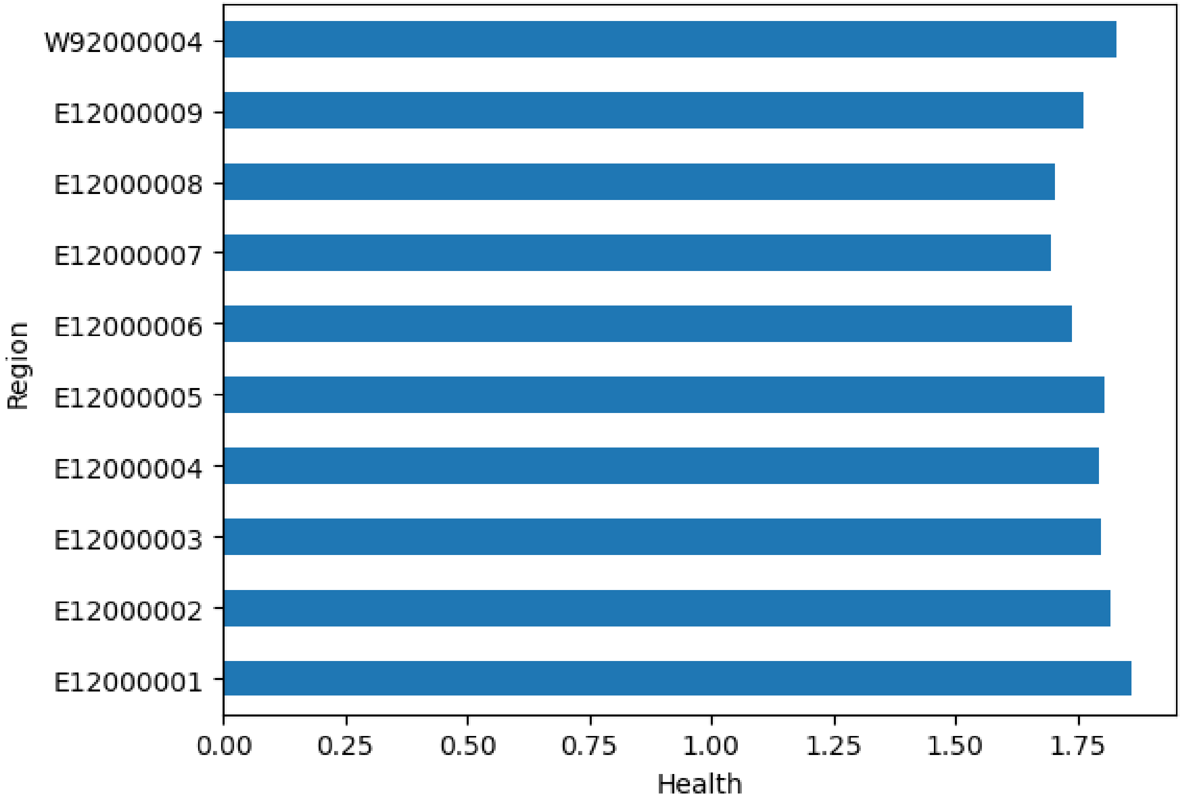
Figure 10(a) Bar Plot for Categorical Attributes



Here in Figure 10(a), Region E12000002 has the highest count, approximately 80,000. Regions E12000006, E12000007, and E12000008 also show high counts, each above 75,000. Region W92000004 has the lowest count for residence type H, around 20,000. Residence type C is consistently low across all regions, with counts below 5,000(approx.) for each region.

4. Crosstab Barh Plot (Numerical and Categorical attributes):

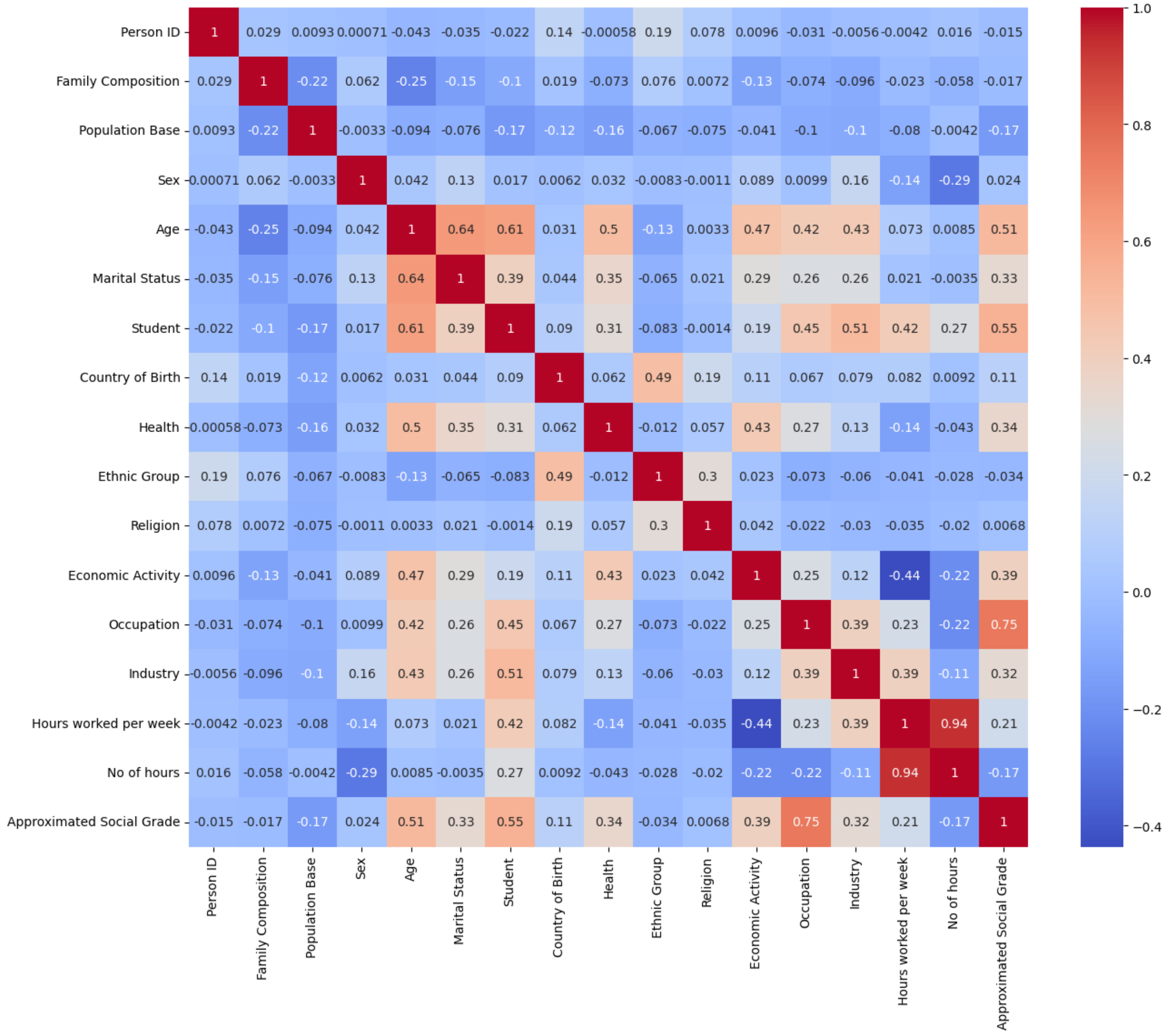
Figure 10(b) Crosstab Barh Plot for Numerical and Categorical Attributes



In Figure 10(b), the region W92000004 has the highest Health score, almost 1.9. The other regions (like E12000001, E12000002, etc.) have very similar scores, around 1.6 to 1.7. And the lowest value for health (which indicates most healthy people) lives in the E12000007 region.

5. Correlogram (All attributes):

Figure 11 Correlogram for all the attributes

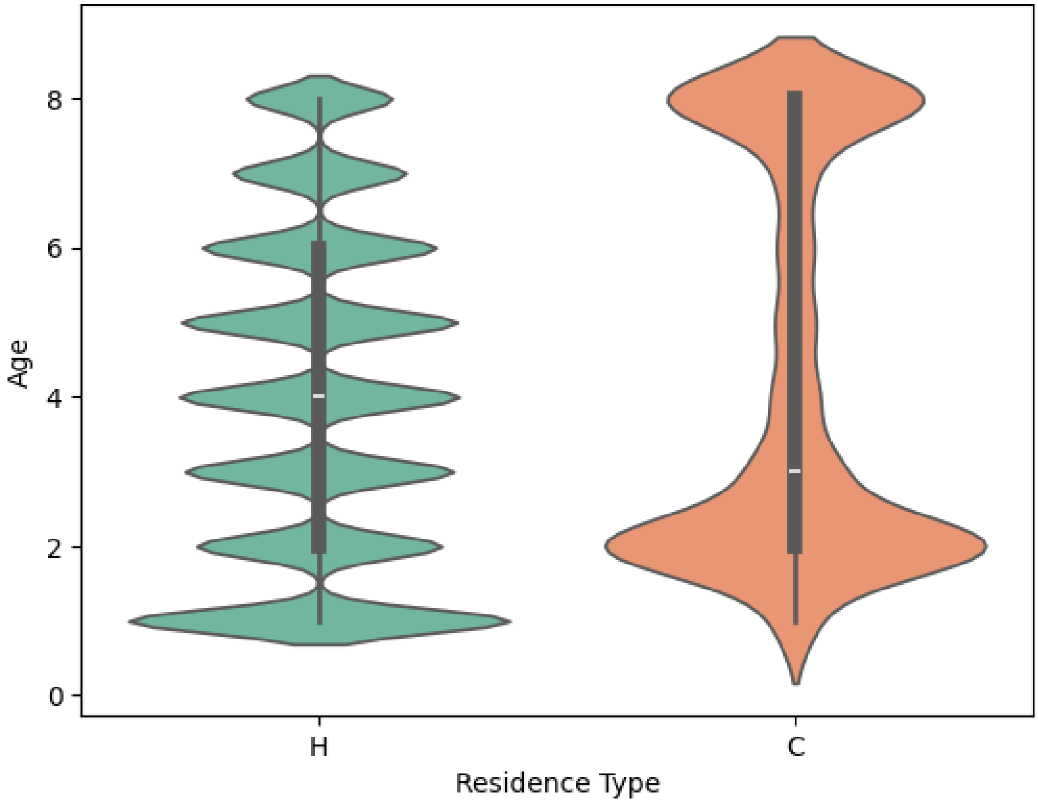


	Person ID	Family Composition	Population Base	Sex	Age	Marital Status	Student	Country of Birth	Health	Ethnic Group	Religion	Economic Activity	Occupation	Industry	Hours worked per week	No of hours	Approximated Social Grade
Person ID	1	0.029	0.0093	0.00071	-0.043	-0.035	-0.022	0.14	-0.00058	0.19	0.078	0.0096	-0.031	-0.0056	-0.0042	0.016	-0.015
Family Composition	0.029	1	-0.22	0.062	-0.25	-0.15	-0.1	0.019	-0.073	0.076	0.0072	-0.13	-0.074	-0.096	-0.023	-0.058	-0.017
Population Base	0.0093	-0.22	1	-0.0033	-0.094	-0.076	-0.17	-0.12	-0.16	-0.067	-0.075	-0.041	-0.1	-0.1	-0.08	-0.0042	-0.17
Sex	0.00071	0.062	-0.0033	1	0.042	0.13	0.017	0.0062	0.032	-0.0083	-0.0011	0.089	0.0099	0.16	-0.14	-0.29	0.024
Age	-0.043	-0.25	-0.094	0.042	1	0.64	0.61	0.031	0.5	-0.13	0.0033	0.47	0.42	0.43	0.073	0.0085	0.51
Marital Status	-0.035	-0.15	-0.076	0.13	0.64	1	0.39	0.044	0.35	-0.065	0.021	0.29	0.26	0.26	0.021	-0.0035	0.33
Student	-0.022	-0.1	-0.17	0.017	0.61	0.39	1	0.09	0.31	-0.083	-0.0014	0.19	0.45	0.51	0.42	0.27	0.55
Country of Birth	0.14	0.019	-0.12	0.0062	0.031	0.044	0.09	1	0.062	0.49	0.19	0.11	0.067	0.079	0.082	0.0092	0.11
Health	-0.00058	-0.073	-0.16	0.032	0.5	0.35	0.31	0.062	1	-0.012	0.057	0.43	0.27	0.13	-0.14	-0.043	0.34
Ethnic Group	0.19	0.076	-0.067	-0.0083	-0.13	-0.065	-0.083	0.49	-0.012	1	0.3	0.023	-0.073	-0.06	-0.041	-0.028	-0.034
Religion	0.078	0.0072	-0.075	-0.0011	0.0033	0.021	-0.0014	0.19	0.057	0.3	1	0.042	-0.022	-0.03	-0.035	-0.02	0.0068
Economic Activity	0.0096	-0.13	-0.041	0.089	0.47	0.29	0.19	0.11	0.43	0.023	0.042	1	0.25	0.12	-0.44	-0.22	0.39
Occupation	-0.031	-0.074	-0.1	0.0099	0.42	0.26	0.45	0.067	0.27	-0.073	-0.022	0.25	1	0.39	0.23	-0.22	0.75
Industry	-0.0056	-0.096	-0.1	0.16	0.43	0.26	0.51	0.079	0.13	-0.06	-0.03	0.12	0.39	1	0.39	-0.11	0.32
Hours worked per week	-0.0042	-0.023	-0.08	-0.14	0.073	0.021	0.42	0.082	-0.14	-0.041	-0.035	-0.44	0.23	0.39	1	0.94	0.21
No of hours	0.016	-0.058	-0.0042	-0.29	0.0085	-0.0035	0.27	0.0092	-0.043	-0.028	-0.02	-0.22	-0.22	-0.11	0.94	1	-0.17
Approximated Social Grade	-0.015	-0.017	-0.17	0.024	0.51	0.33	0.55	0.11	0.34	-0.034	0.0068	0.39	0.75	0.32	0.21	-0.17	1

In Figure 11, values less than -0.5 and greater than +0.5 indicates high correlation, with 0.75 score, Approximated Social grade and Occupation has the highest correlation.

6. Violin Plot (Numerical and Categorical attributes) *[Extra]:*

Figure 12 Violin Plot for Numerical and Categorical attributes



In Figure 12, this violin plot shows the spread of ages for two types of living areas. In the Residence Type H, there are all kinds of ages here, with some peaks at different points. Most of the ages are between 2 and 6 (which is age 16 to 24 and 55 to 64). In the C residence type, the type of ages live here are more balanced.

2. Perform classification for the "Approximate Social Grade" attribute using 3 algorithms; present and discuss the results; compare the results of the 3 algorithms.

- **Data Preparation:**

  For implementing the data preparation step, I first checked for the null values in the dataset. There were 302321 null values for "*No of hours*" attribute.

I filled them up with the mean value as it gives good measure of central tendency.

Now, we have to check the correlation or similarity between the attribute.

I dropped the *Person ID* attribute as it was not at all relevant for predictions, it was only random for numbering the entries.

For categorical attributes, we can check the correlation with *Chi Square* test. The “p” value determines the correlation between two categorical attributes and it ranges from 0 to 1. Closer to 0 means there is a relationship between the attributes. And for numerical attributes, I used *Pearsons Correlation matrix*. Closer to -1 or 1 suggests strong relationship between the attributes.

After doing the *Pearsons correlation matrix* for the numerical attributes, I didn’t find any value more than 0.8, so I didn’t drop any of the attributes.

For finding out the correlation of all the attributes, I turned the categorical attributes into numerical ones (which is needed for implementing classification tasks), even then, there was no correlation value more than 0.8. So, I kept them as it is.

After that, I divided the dataset into 2 parts, 1st part being the features meaning characteristics of the data, and the 2nd part being the target attribute, meaning we have to predict the target attribute by using the features of the data.

Then, for training our model with the dataset, I separated 75% of the data for training, and other 25% was for testing the data and making our model predict the results, so that I could evaluate how good or bad the model is performing.

After that, I applied Min-Max Normalization to the features of the dataset so that they are less deviated and have increased uniformity.

Now, I can apply the classifier models to my prepared data.

Common classification tasks which are done are

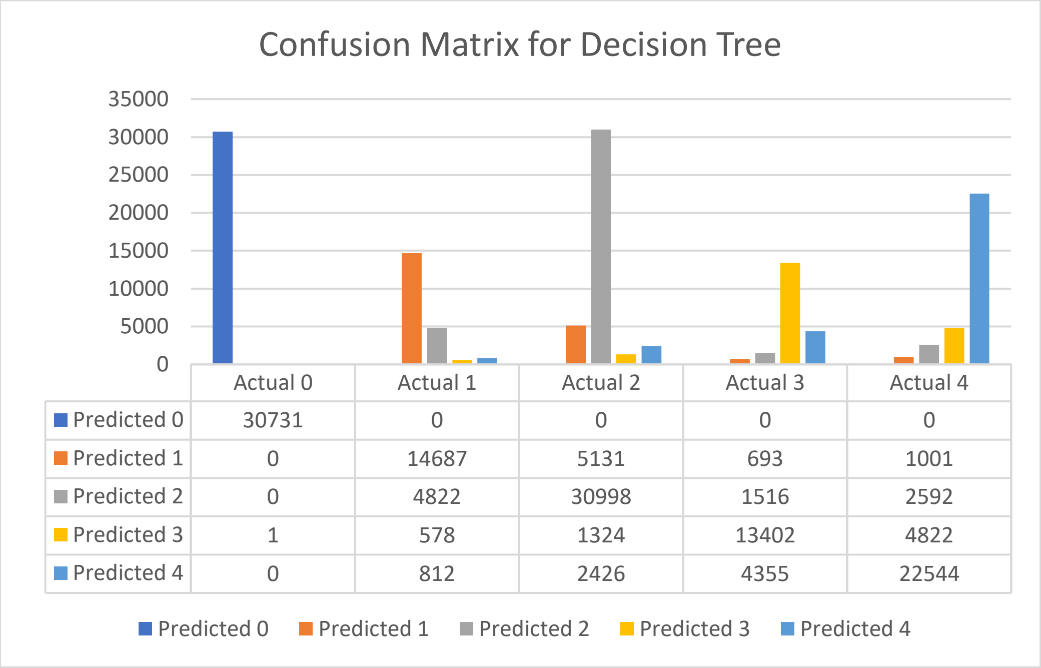
1. Import the classifier from scikit-learn;
2. Declare/Call the Classifier;
3. Fit the classifier to the training data;

4. Make predictions on the testing dataset;
5. Evaluate performance using Confusion Matrix and Classification report.

- **Decision Tree Classifier:**

  For this model, I am using *Entropy* as the metric. If I didn't specify any criterion, it would have used *Gini* as the default metric.

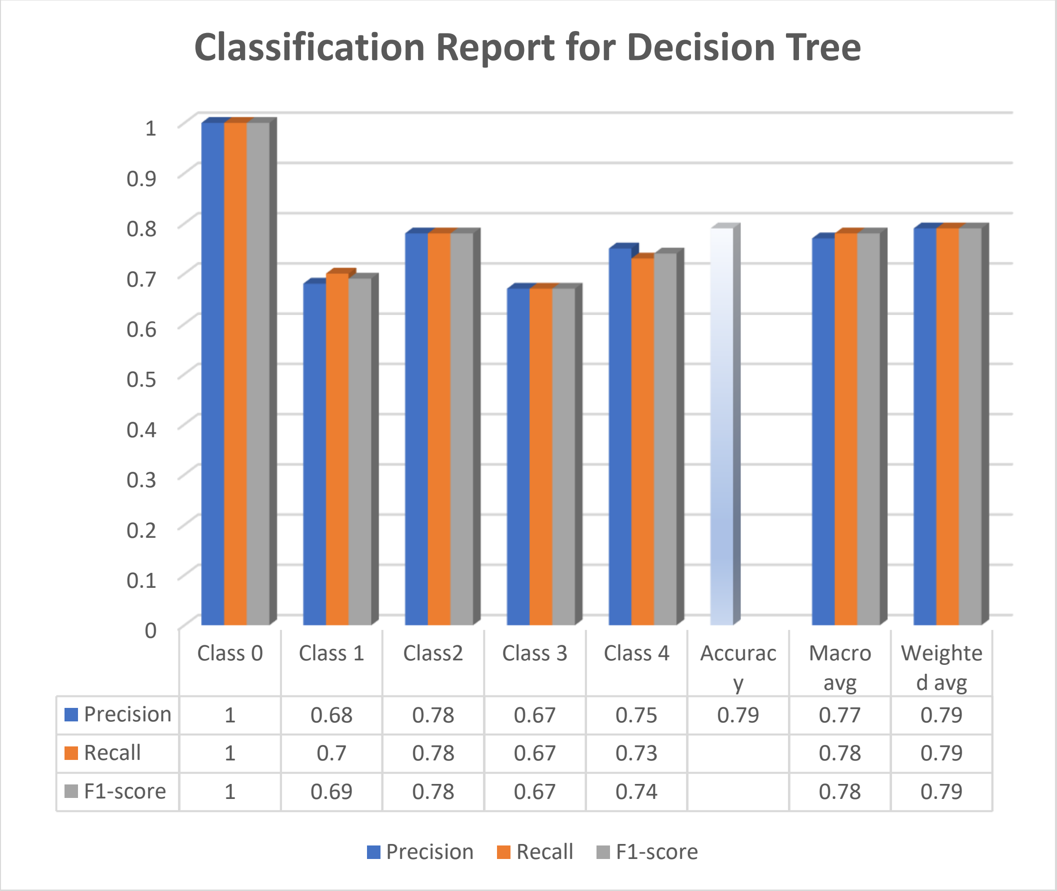
Figure 13 Confusion Matrix for Decision Tree



	Actual 0	Actual 1	Actual 2	Actual 3	Actual 4
Predicted 0	30731	0	0	0	0
Predicted 1	0	14687	5131	693	1001
Predicted 2	0	4822	30998	1516	2592
Predicted 3	1	578	1324	13402	4822
Predicted 4	0	812	2426	4355	22544

In Figure 13, from the confusion matrix we can see, class 0 has perfect classification whereas, class 3 and 4 have a higher number of misclassifications specially among themselves.

Figure 14 Classification Report for Decision Tree

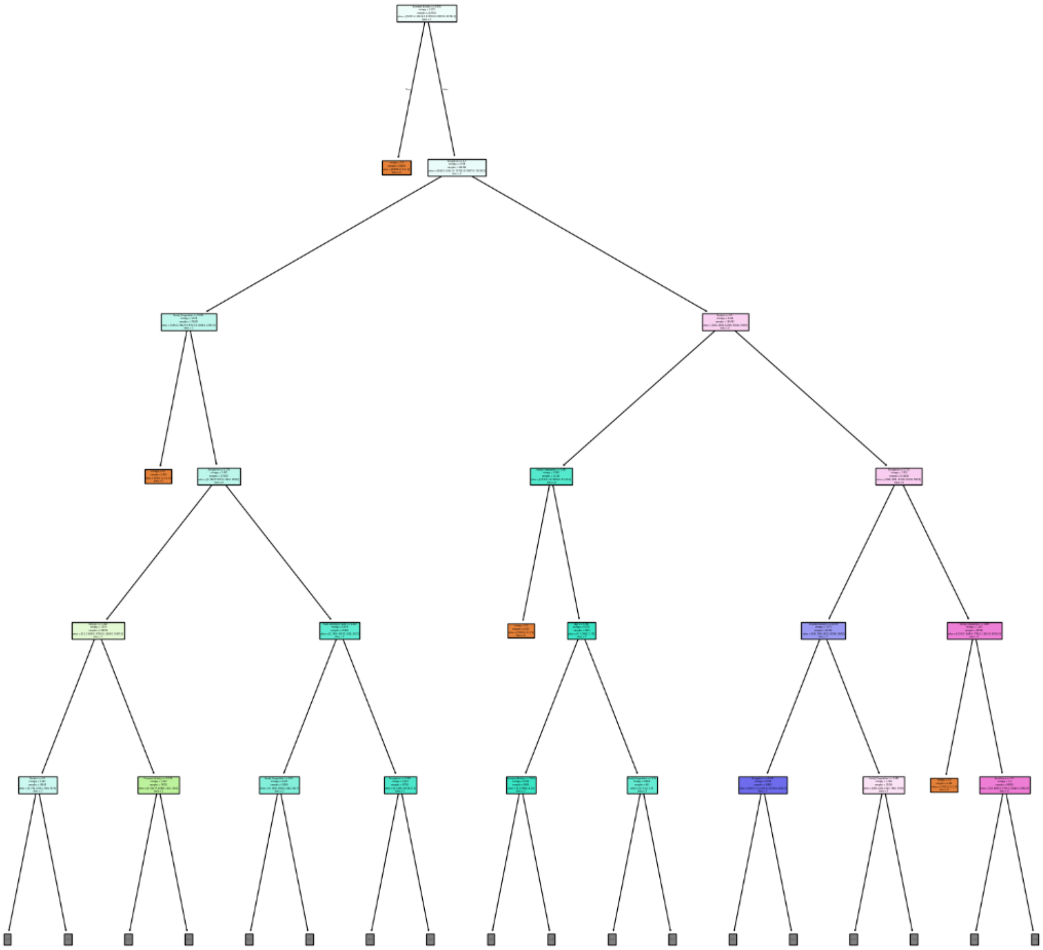


	Class 0	Class 1	Class2	Class 3	Class 4	Accuracy	Macro avg	Weighted avg
🟦 Precision	1	0.68	0.78	0.67	0.75	0.79	0.77	0.79
🟧 Recall	1	0.7	0.78	0.67	0.73		0.78	0.79
⬜ F1-score	1	0.69	0.78	0.67	0.74		0.78	0.79

In Figure 14, from the classification report, it can be concluded that class 0 has perfect classification with precision, recall, and F1-score all being 1. It can be assumed that, the high accuracy is likely due to a frequent pattern in the data making easier for the model to predict correctly.

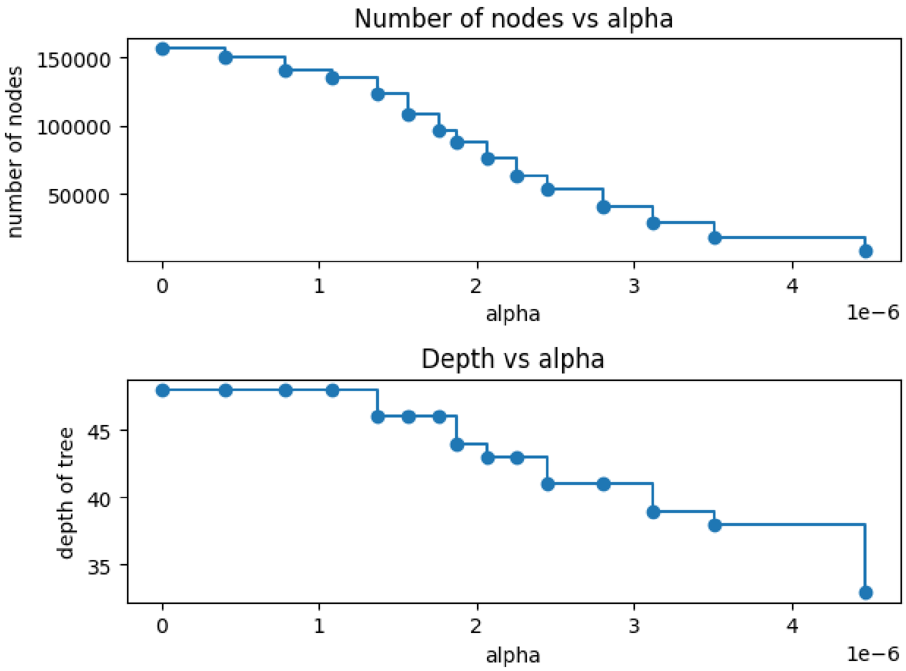
In Figure 15, I visualized the Decision tree.

Figure 15 Decision Tree before pruning



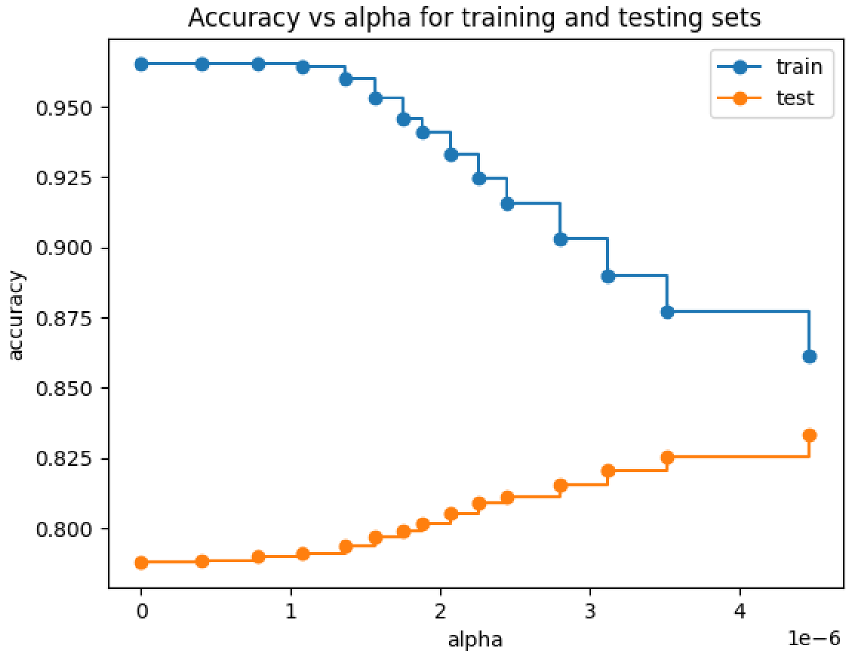
Then I did cost complexity pruning for my decision tree, it makes the decision tree less complex, and prevents the model from overfitting. The final pruned tree had 7 nodes, it reflects significant pruning and simplification of the tree. Then I generated plots to see in Figure 16, that as the number of alpha increases, the nodes of tree and depth decreases suggesting pruning the tree.

Figure 16 Alpha Effect on number of nodes and Tree depth



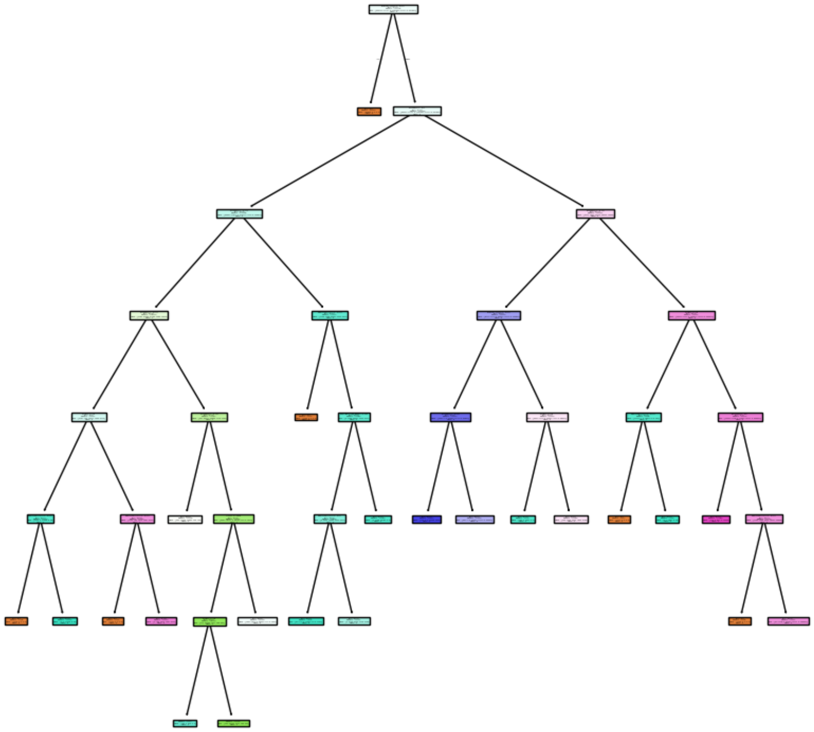
Also, as the alpha value increases, the training accuracy drops in Figure 17, which was at first at almost 96%, and the testing accuracy rises, suggesting the model is training better now and becoming generalized.

Figure 17 Alpha Effect on Training and Testing accuracy



Then I visualized the pruned tree, and noticed the smaller size of the tree after pruning in Figure 18.

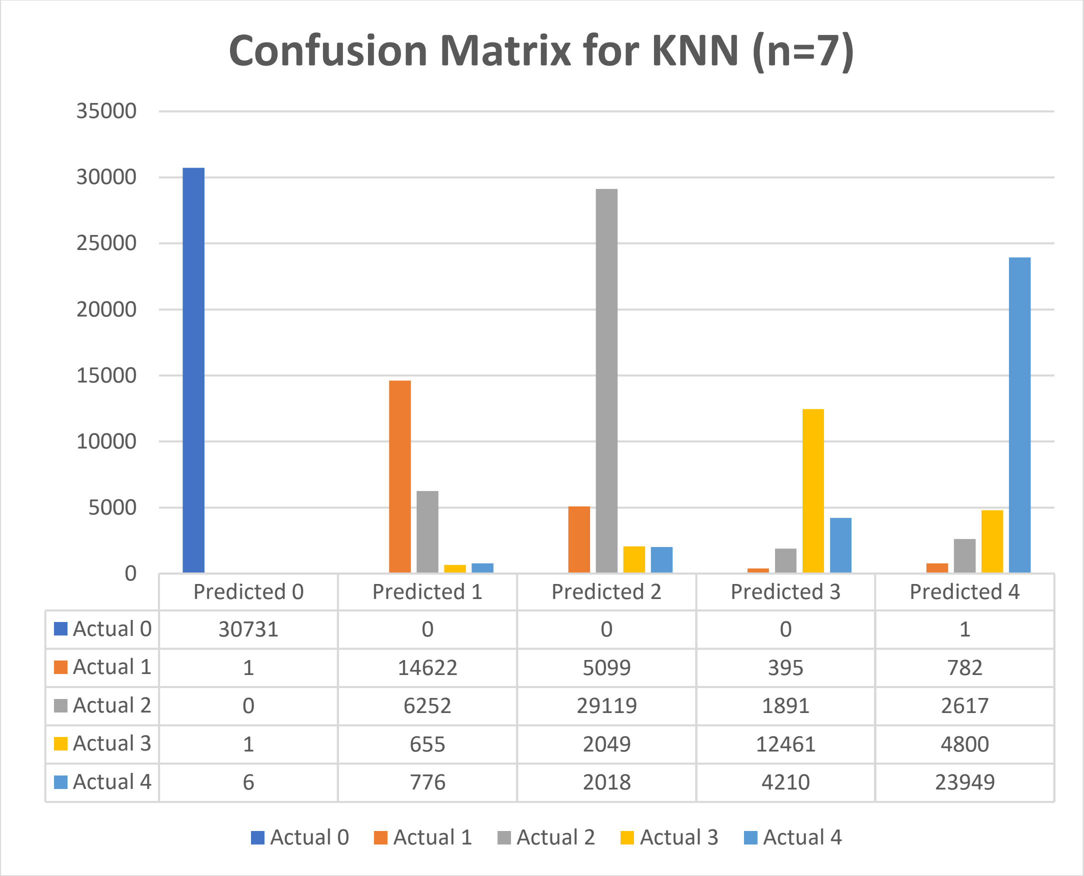
Figure 18 Pruned Decision Tree



- **K-Nearest Neighbor Classifier:**

  For calculating the distance of the neighbors, the model will use Euclidean distance.

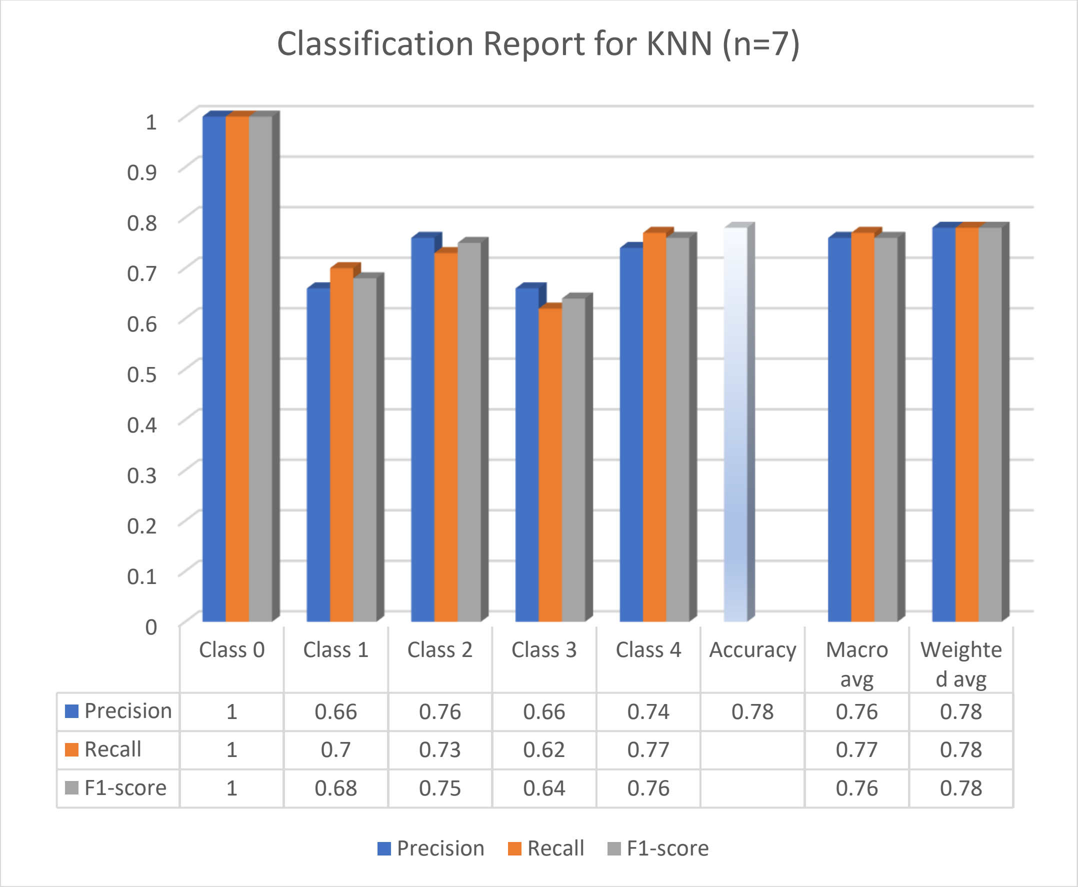
Figure 19 Confusion Matrix for KNN



	Predicted 0	Predicted 1	Predicted 2	Predicted 3	Predicted 4
Actual 0	30731	0	0	0	1
Actual 1	1	14622	5099	395	782
Actual 2	0	6252	29119	1891	2617
Actual 3	1	655	2049	12461	4800
Actual 4	6	776	2018	4210	23949

In Figure 19, we can see that Class 0 has perfect classification, whereas class 1 has the greatest number of misclassifications.

Figure 20 Classification Report for KNN

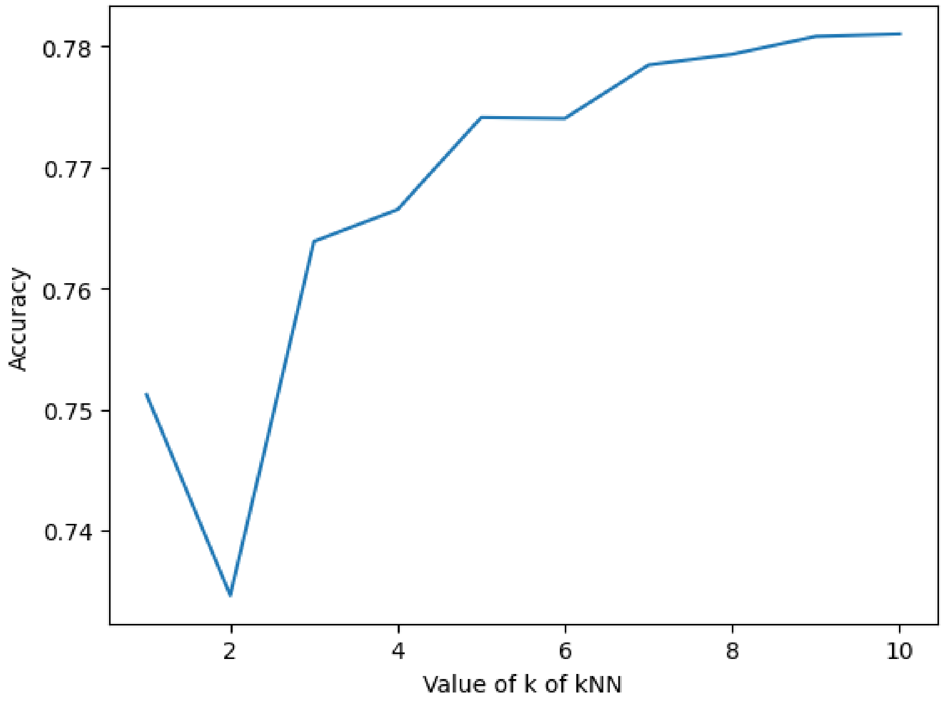


	Class 0	Class 1	Class 2	Class 3	Class 4	Accuracy	Macro avg	Weighted avg
🟦 Precision	1	0.66	0.76	0.66	0.74	0.78	0.76	0.78
🟧 Recall	1	0.7	0.73	0.62	0.77		0.77	0.78
⬜ F1-score	1	0.68	0.75	0.64	0.76		0.76	0.78

From Figure 20, the model struggles with class 3, maybe to fewer samples in this class. Rather than that, the performance is quite consistent among all the classes.

For figuring out the best K value I experimented values ranging from 2 to 10, among which, the best value for K is 10.

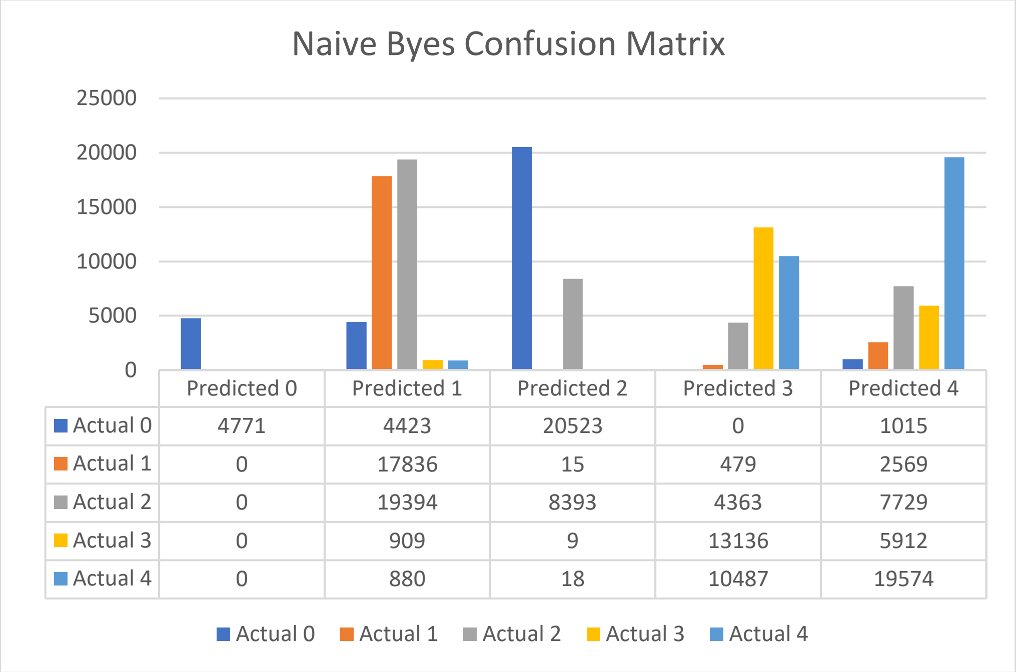
Figure 21 Finding optimal K



However, in Figure 21, the notable thing is, after 7, increasing the K improves the model with very less amount. And as KNN is computationally expensive, greater than 7 value is not optimum with the dataset.

- **Naïve Bayes Classifier:**

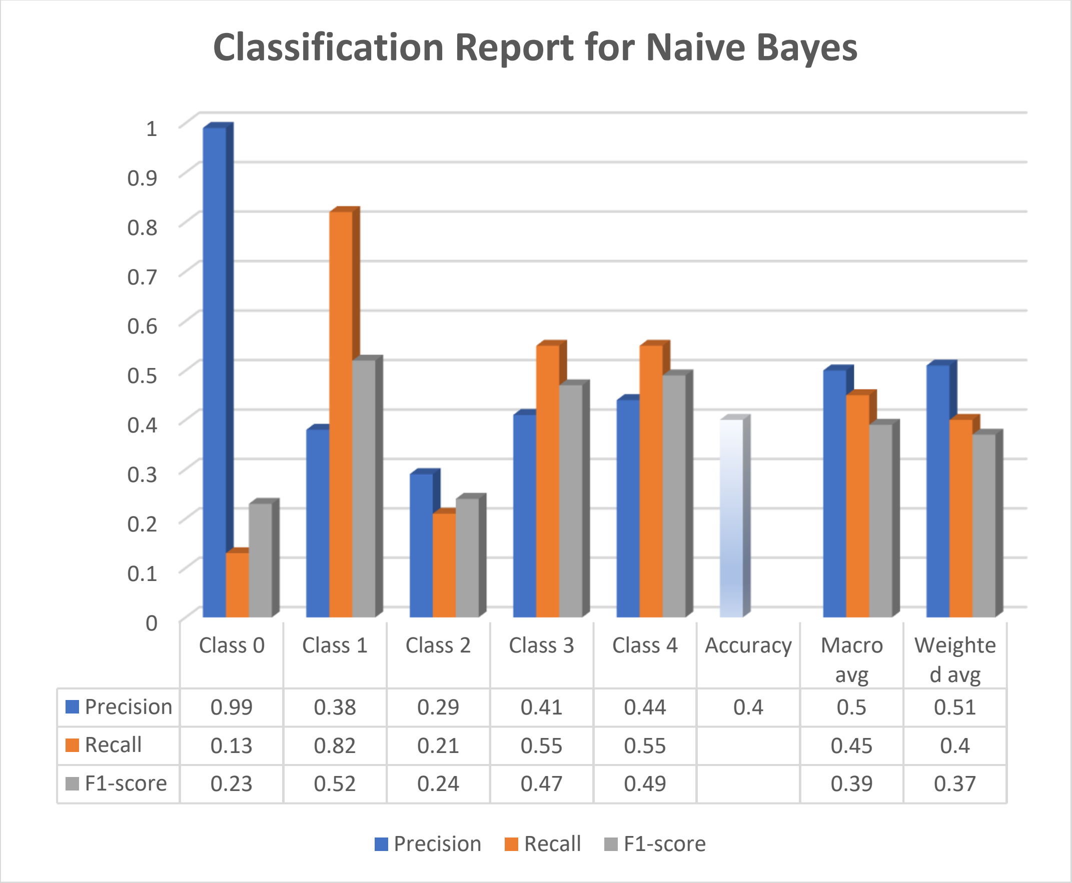
Figure 22 Confusion Matrix for Naïve Bayes



	Predicted 0	Predicted 1	Predicted 2	Predicted 3	Predicted 4
■ Actual 0	4771	4423	20523	0	1015
■ Actual 1	0	17836	15	479	2569
■ Actual 2	0	19394	8393	4363	7729
■ Actual 3	0	909	9	13136	5912
■ Actual 4	0	880	18	10487	19574

From Figure 22, it is evident that, this classifier has poor performance across all classes, the worst classification of class 2.

Figure 23 Classification Report for Naïve Bayes



	Class 0	Class 1	Class 2	Class 3	Class 4	Accuracy	Macro avg	Weighted avg
Precision	0.99	0.38	0.29	0.41	0.44	0.4	0.5	0.51
Recall	0.13	0.82	0.21	0.55	0.55		0.45	0.4
F1-score	0.23	0.52	0.24	0.47	0.49		0.39	0.37

From Figure 23, the model shows poor performance correctly classifying only 40% of all instances. It performs weak averaging all classes, its performance gets severely affected by larger classes like class 2.

- **XGBoost Classifier *(Extra)*:**

For using XGBoost classifier, hyperparameter tuning is important as it is very sensitive to the hyperparameters. 1st I started with the basic parameters-

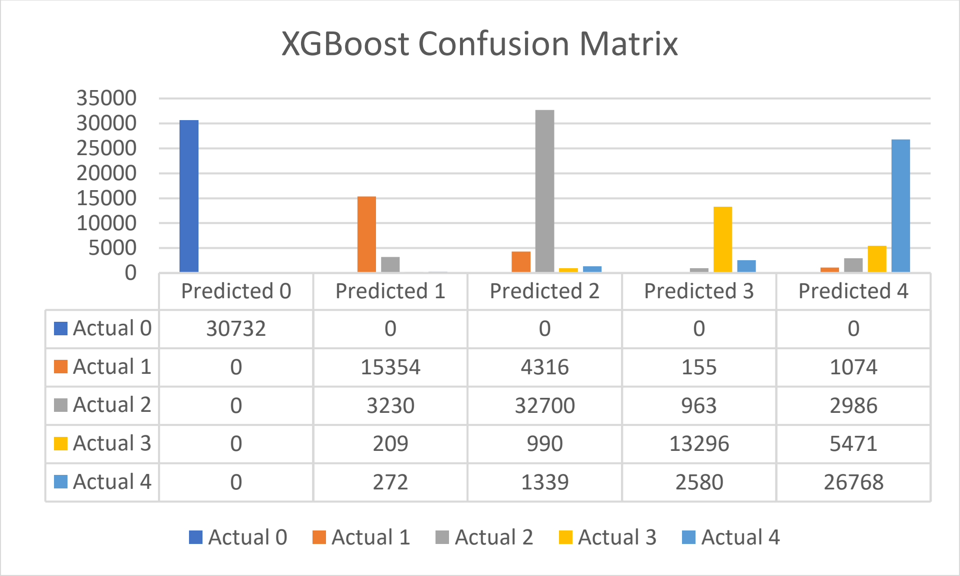
-n_estimators=100;

-max_depth=5;

Learning_rate=0.1;

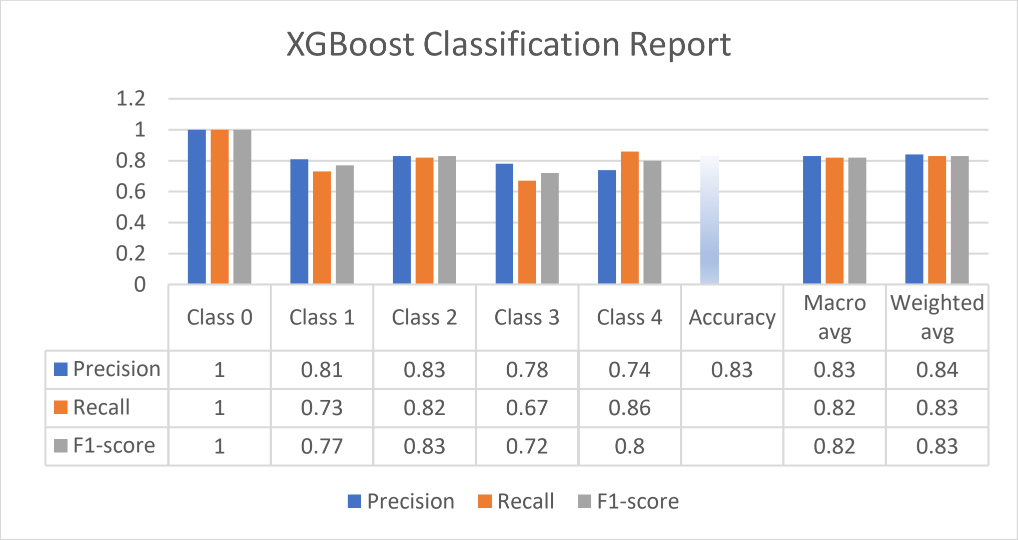
And achieved the confusion matrix and classification report as follows in Figure 24 and Figure 25 consecutively-

Figure 24 Confusion Matrix for XGBoost



	Predicted 0	Predicted 1	Predicted 2	Predicted 3	Predicted 4
Actual 0	30732	0	0	0	0
Actual 1	0	15354	4316	155	1074
Actual 2	0	3230	32700	963	2986
Actual 3	0	209	990	13296	5471
Actual 4	0	272	1339	2580	26768

Figure 25 Classification Report for XGBoost



	Class 0	Class 1	Class 2	Class 3	Class 4	Accuracy	Macro avg	Weighted avg
Precision	1	0.81	0.83	0.78	0.74	0.83	0.83	0.84
Recall	1	0.73	0.82	0.67	0.86		0.82	0.83
F1-score	1	0.77	0.83	0.72	0.8		0.82	0.83

From Figure 24 and Figure 25, it can be seen that, it has almost consistent results across all the classes and can classify 83% of the data correctly.

After that, I tried to find the best parameter among some given parameters, and the best parameters was-

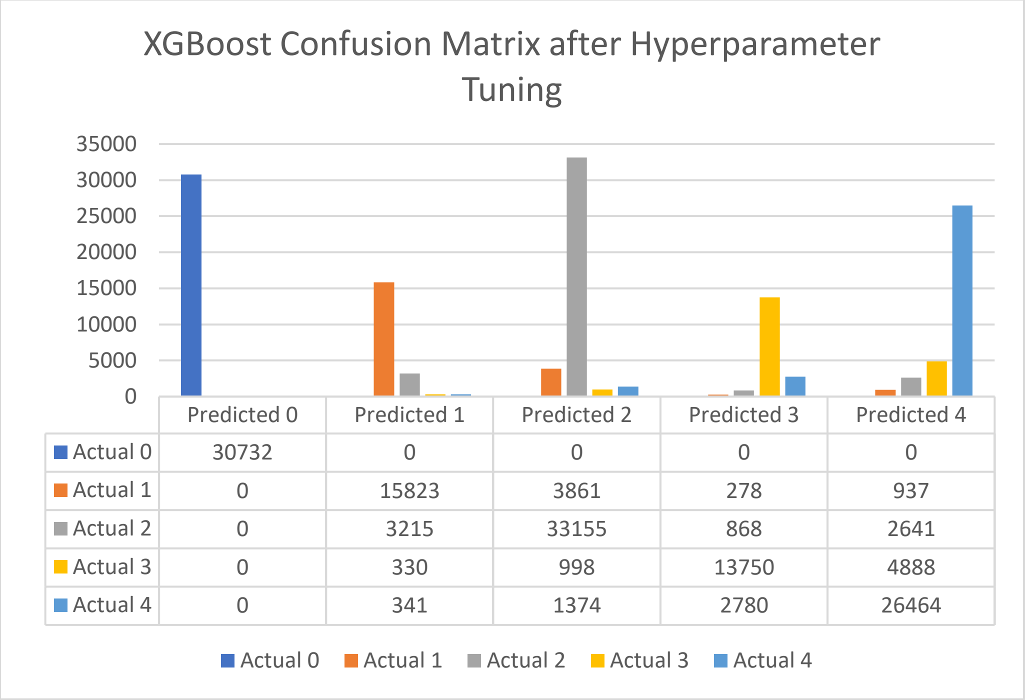
-n_estimators=200;

-max_depth=7;

Learning_rate=0.1;

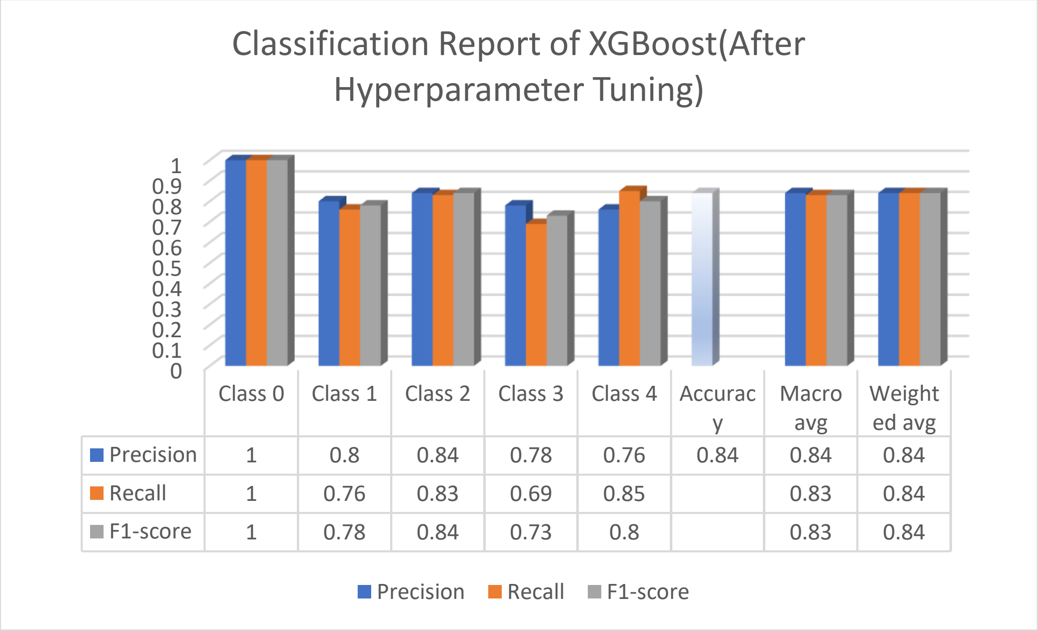
And I implemented the model after hyperparameter tuning and got these results showed in Figure 26 and Figure 27-

Figure 26 Confusion Matrix for XGBoost (Hyperparameter Tuning)



	Predicted 0	Predicted 1	Predicted 2	Predicted 3	Predicted 4
Actual 0	30732	0	0	0	0
Actual 1	0	15823	3861	278	937
Actual 2	0	3215	33155	868	2641
Actual 3	0	330	998	13750	4888
Actual 4	0	341	1374	2780	26464

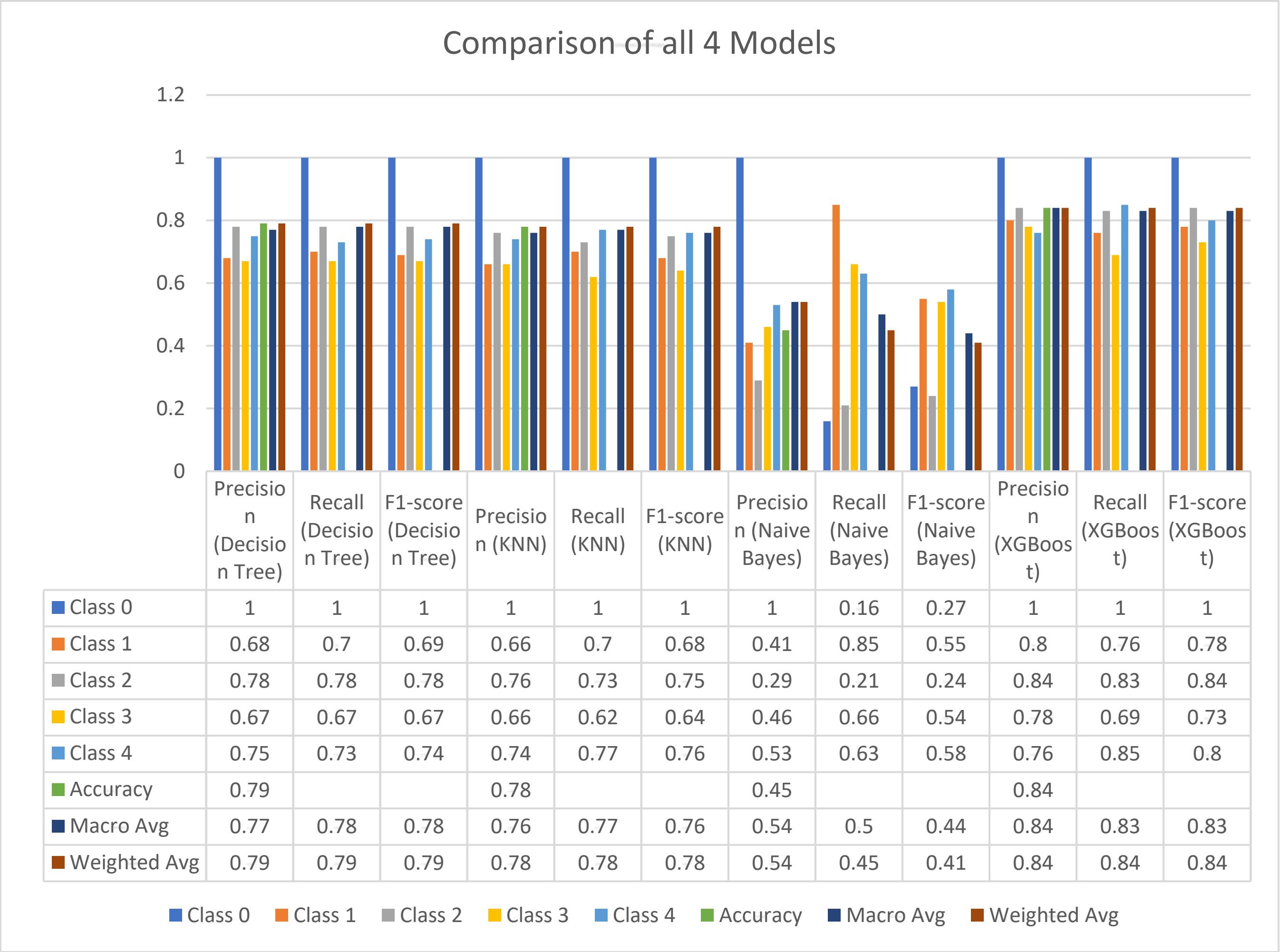
Figure 27 Classification Report for XGBoost (Hyperparameter Tuning)



	Class 0	Class 1	Class 2	Class 3	Class 4	Accuracy	Macro avg	Weighted avg
Precision	1	0.8	0.84	0.78	0.76	0.84	0.84	0.84
Recall	1	0.76	0.83	0.69	0.85		0.83	0.84
F1-score	1	0.78	0.84	0.73	0.8		0.83	0.84

The results improved by insignificant amount.

Figure 28 Comparison of all models across all the classes



	Precision (Decision Tree)	Recall (Decision Tree)	F1-score (Decision Tree)	Precision (KNN)	Recall (KNN)	F1-score (KNN)	Precision (Naive Bayes)	Recall (Naive Bayes)	F1-score (Naive Bayes)	Precision (XGBoost)	Recall (XGBoost)	F1-score (XGBoost)
Class 0	1	1	1	1	1	1	1	0.16	0.27	1	1	1
Class 1	0.68	0.7	0.69	0.66	0.7	0.68	0.41	0.85	0.55	0.8	0.76	0.78
Class 2	0.78	0.78	0.78	0.76	0.73	0.75	0.29	0.21	0.24	0.84	0.83	0.84
Class 3	0.67	0.67	0.67	0.66	0.62	0.64	0.46	0.66	0.54	0.78	0.69	0.73
Class 4	0.75	0.73	0.74	0.74	0.77	0.76	0.53	0.63	0.58	0.76	0.85	0.8
Accuracy	0.79			0.78			0.45			0.84		
Macro Avg	0.77	0.78	0.78	0.76	0.77	0.76	0.54	0.5	0.44	0.84	0.83	0.83
Weighted Avg	0.79	0.79	0.79	0.78	0.78	0.78	0.54	0.45	0.41	0.84	0.84	0.84

- **Comparison of all 4 Classifiers:**

  From Figure 28, we can conclude that XGBoost is the best classifier across al the classes and gives consistent result, whereas, Naïve bayes significantly struggles with the dataset suggesting it is not suitable for the dataset. Decision Tree and KNN gives good results, but struggles with consistency in class 3,4.

3. Perform regression on the “No of hours” attribute using 2 algorithms; present and discuss the results; compare the results of the 2 algorithms.

Data preparation:

For preparing the data, I followed the same steps as classification, as the regression tasks requires numerical values across all attributes. Then I separated the target attribute which is *“No of hours”* and other features and split the data into training and testing parts (75% for training, 25% for testing). Then I applied *min-max normalization* to all training data.

Common regression tasks which are done for every regression are as follows-

6. Import the regressor from scikit-learn;
7. Declare/Call the regressor;
8. Fit the regressor to the training data;
9. Make predictions on the testing dataset;
10. Evaluate performance using Mean Absolute error, mean squared error, Root mean squared error and adjusted R2 score.

Linear Regression Results in Table 1:

Table 1 Linear Regression Results

Metric	**Value**
Mean Absolute Error	5.60
Mean Squared Error	60.06
Root Mean squared error	7.75
R^2 Score	0.304
Adjusted R^2 Score	0.999995

Regression Tree Results in Table 2:

Table 2 Regression Tree Results

Metric	**Value**
Mean Absolute Error	1.93
Mean Squared Error	10.67

Root Mean squared error	3.27
R^2 Score	0.87
Adjusted R^2 Score	0.999999

Figure 29 Visualization of the Regressor Tree

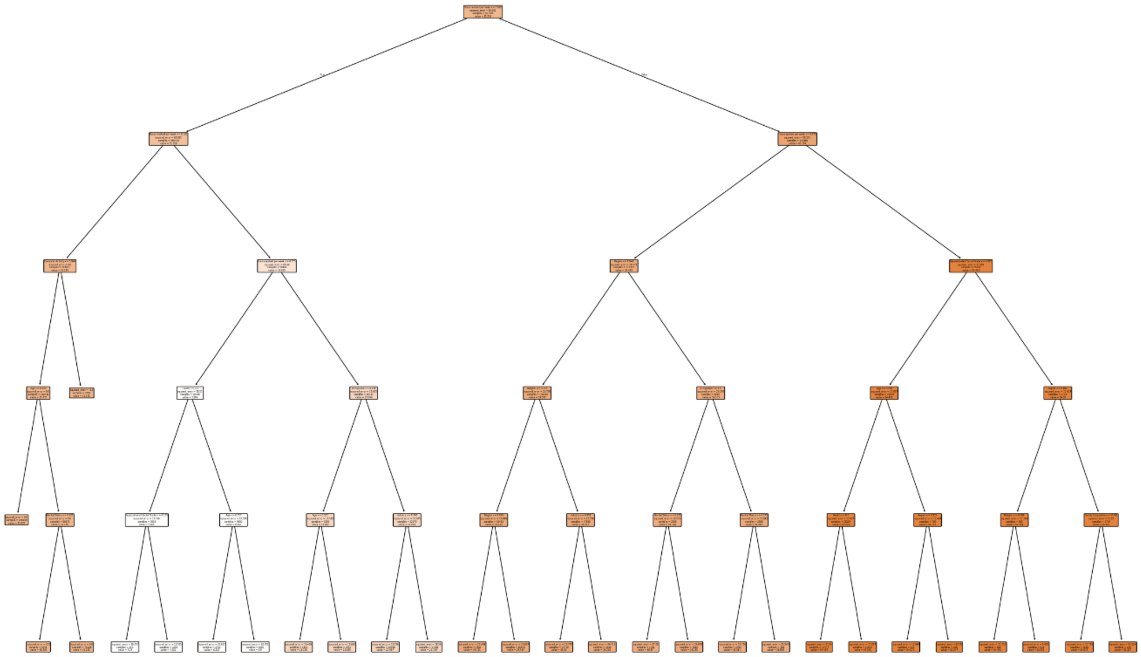
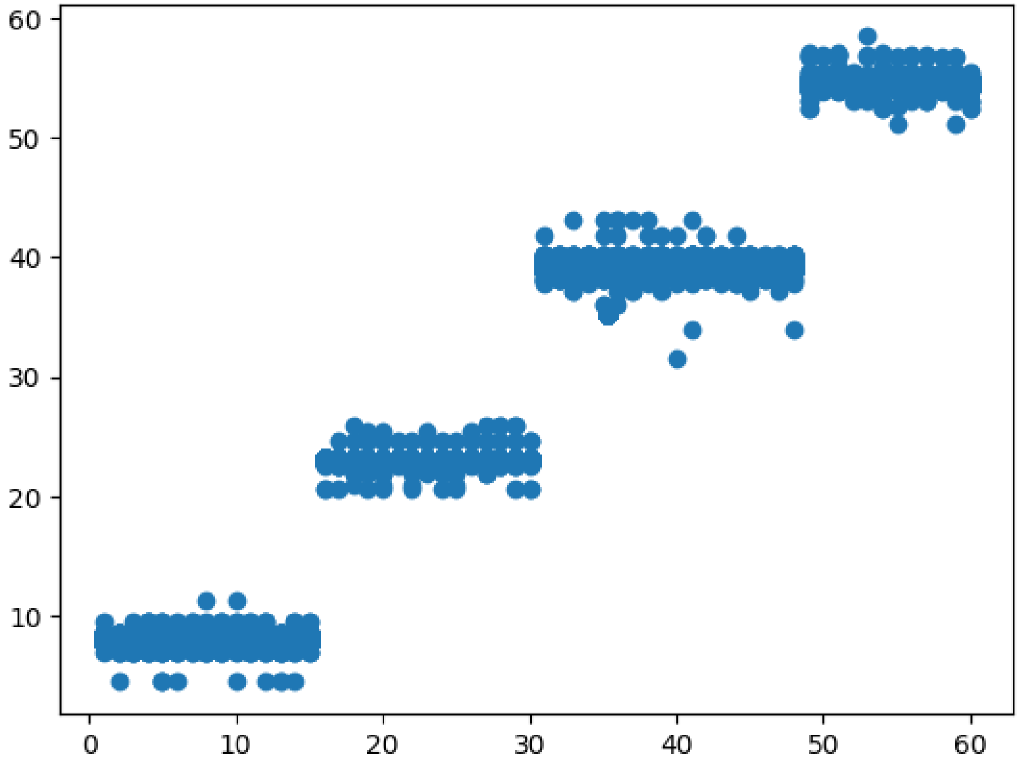


Figure 29 is the visualization of the Regressor Tree. In Figure 30, I generated a scatterplot to visualize any correlation, which shows positive correlation.

Figure 30 Scatterplot for visualizing correlation



Neural Network Regression Results in Table 3:

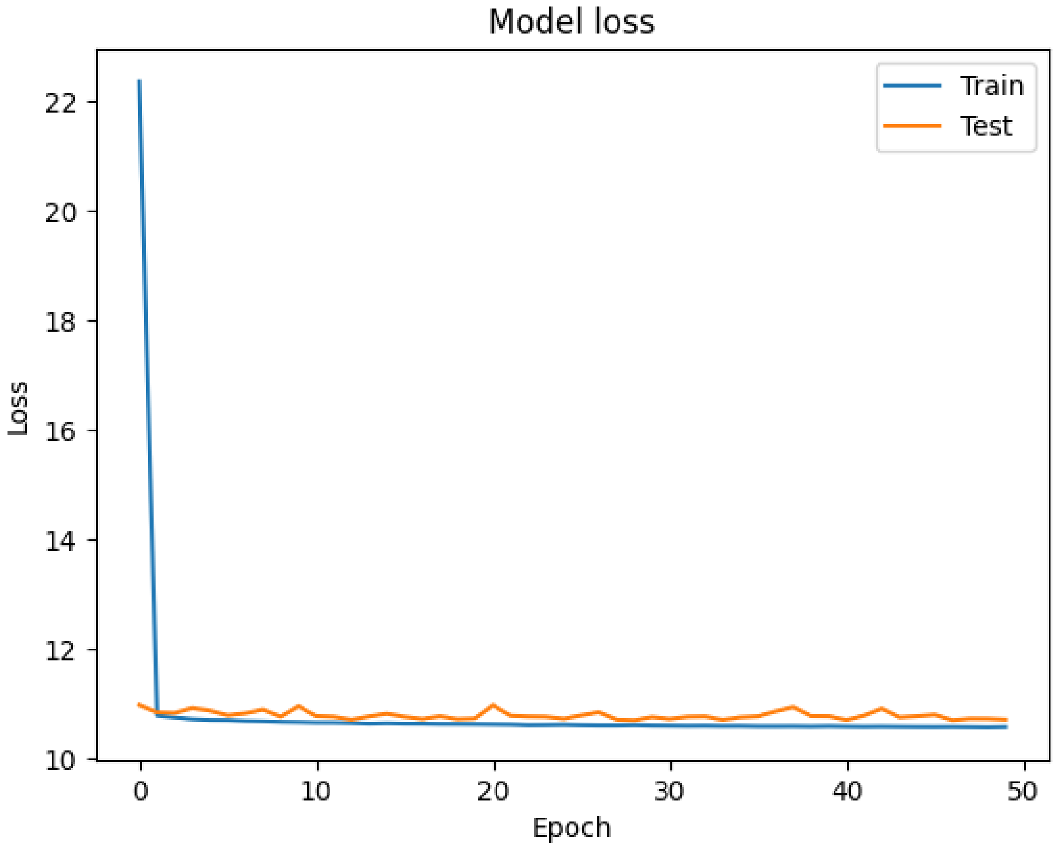
To work with Neural Network, I built my model with Sequential API. There is 2 hidden layers in my model with 64 and 32 units, and I used Relu activation function. I used the Adam optimizer for model training. Then I fitted the dataset to the model for training with 50 epochs. Below are the results.

Table 3 Neural Network Regression Results

Metric	**Value**
Mean Absolute Error	1.98
Mean Squared Error	10.71
Root Mean squared error	3.27
R^2 Score	0.87
Adjusted R^2 Score	0.999999

I plotted the Training vs Testing loss of the model across epoch in Figure 31 which suggests overfitting as the testing is not improving.

Figure 31 Training vs Testing Loss Curve



XGBoost Regression Results (Extra) in Table 4:

First, I went with the basic parameters-

-n_estimators=100;

-max_depth=6;

Learning_rate=0.1

for training the data and got these results.

Table 4 XGBoost Regression Results

Metric	**Value**
Mean Absolute Error	1.93

Mean Squared Error	10.67
Root Mean squared error	3.27
R^2 Score	0.87
Adjusted R^2 Score	0.999999

Then I tried to find the best parameters across few given parameters and found best parameters as-

-n_estimators=100;

-max_depth=3;

Learning_rate=0.1

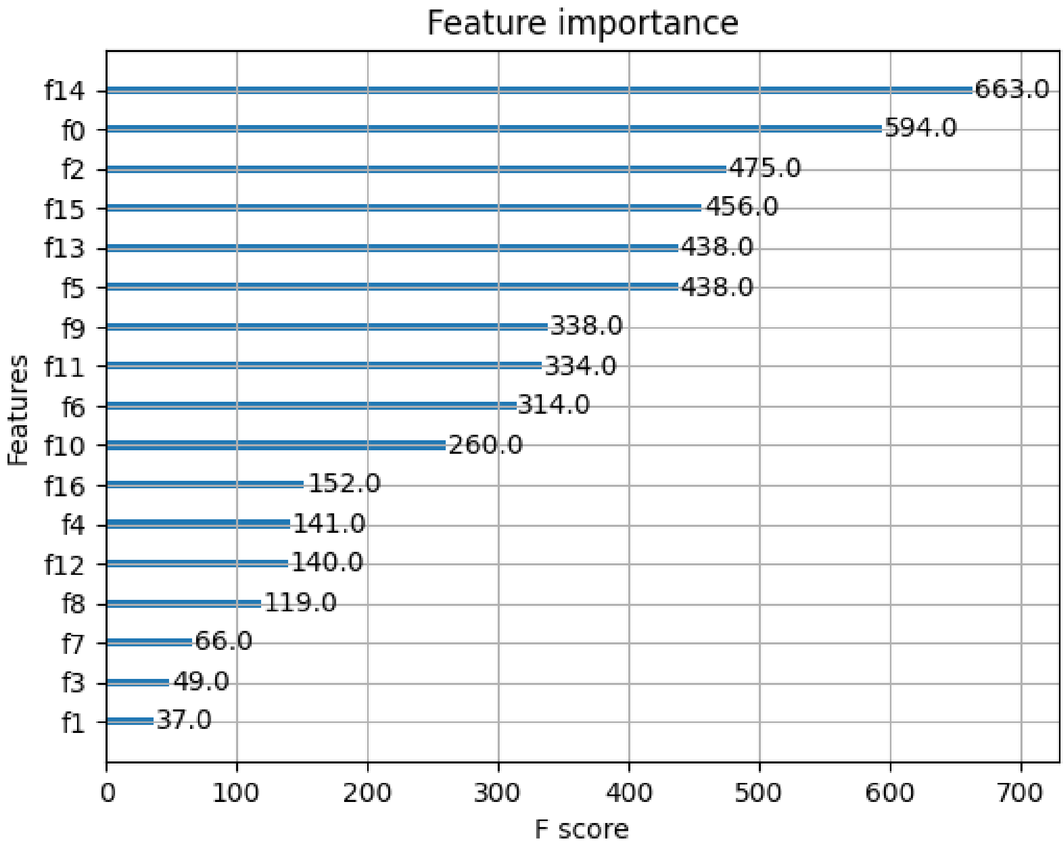
Then, I ran the model again with these parameters and got the results in Table 5 as-

Table 5 XGBoost Regression Results (After hyperparameter tune)

Metric	**Value**
Mean Absolute Error	1.93
Mean Squared Error	10.65
Root Mean squared error	3.26
R^2 Score	0.88
Adjusted R^2 Score	0.999999

I plotted the features in Figure 32 who had more contribution in training the dataset which shows Feature 14 or Occupation column is contributing the most.

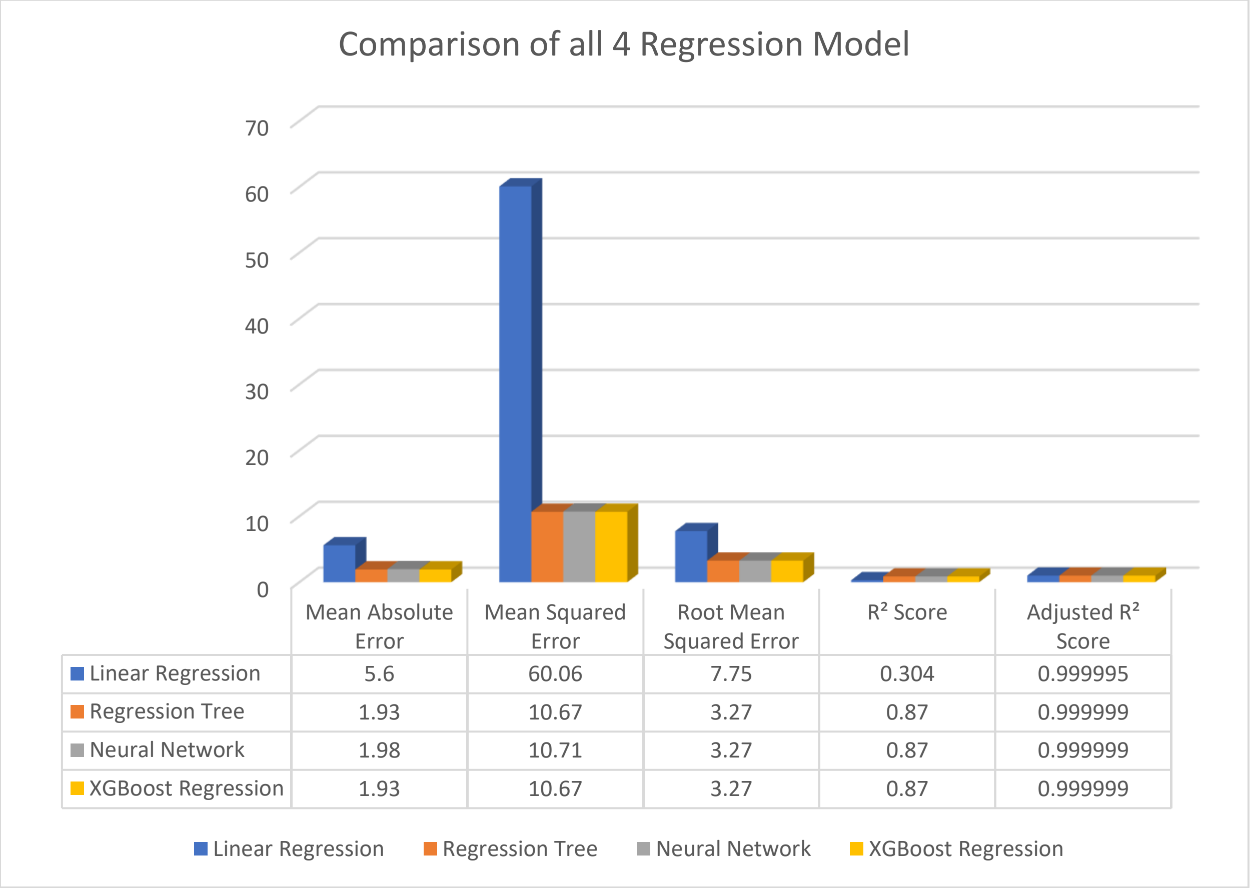
Figure 32 Feature Importance for XGBoost



Comparison of 4 Regression Results:

From Figure 33, it can be said that XGBoost and Decision Tree performed the best, on the otherhand Neural network has overfitted the data. Linear Regression performs poorly as given by the low R^2 score.

Figure 33 Comparison of all 4 models' performance



	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R^2 Score	Adjusted R^2 Score
🟦 Linear Regression	5.6	60.06	7.75	0.304	0.999995
🟧 Regression Tree	1.93	10.67	3.27	0.87	0.999999
◼ Neural Network	1.98	10.71	3.27	0.87	0.999999
🟨 XGBoost Regression	1.93	10.67	3.27	0.87	0.999999

4. Perform association rule mining using 1 algorithm and interpret the meaning of at least 5 rules.

Data Preparation:

For rule mining tasks, I did the mapping to transform numerical attribute to categorical attribute first, because we need categorical data for rule mining tasks.

Apriori Algorithm:

For Apriori, I used support threshold as 0.3 and confidence threshold as 0.5. Then, I generated the rules and selected the ones of which the lift was above 1.

Rules:

Table 6 Rules for Apriori Algorithm

Rule	Items	Antecedent	Consequent	Support	Confidence	Lift
1.	{Married/same-sex civil partnership couple family, Married or in a registered same-sex civil partnership, H, White, No}	{White, Married/same-sex civil partnership couple family, No}	{Married or in a registered same-sex civil partnership, H}	0.32	0.86	2.26
2.	{Usual resident, UK, Married/same-sex civil partnership couple family, Married or in a	{Usual resident, Married/same-sex civil partnership couple family, No}	{UK, Married or in a registered same-sex civil partnership}	0.30	0.72	2.27

	registered same-sex civil partnership, No}					
3.	{Married/same-sex civil partnership couple family, Married or in a registered same-sex civil partnership, H, White, No}	{Married or in a registered same-sex civil partnership}	{White, Married/same-sex civil partnership couple family, No, H}	0.32	0.84	2.25
4.	{Usual resident, Married/same-sex civil partnership couple family, Married or in a registered same-sex civil	{White, Married/same-sex civil partnership couple family, No}	{Usual resident, Married or in a registered same-sex civil partnership}	0.32	0.86	2.25

	partnership, White, No}					
5.	{Usual resident, UK, Married/same-sex civil partnership couple family, Married or in a registered same-sex civil partnership, H, No}	{Usual resident, UK, Married or in a registered same-sex civil partnership, H}	{Married/same-sex civil partnership couple family, No}	0.30	0.97	2.27

Interpretations of Table 6 rules:

Rule1: If an individual is White, part of a married/same-sex civil partnership couple family, and not a student, they are **2.26 times more likely** than random to be married or in a registered same-sex civil partnership and to live in residence type H. This has support of 32% of the dataset, and 86% of cases like this antecedent, meets the consequence.

Rule2: If an individual is a usual resident, part of a married/same-sex civil partnership couple family, and not a student, they are **2.27 times more likely** than random to be UK-born and married or in a registered same-sex civil partnership. This has support of 30% of the dataset, and 72% of cases like this antecedent, meets the consequence.

Rule3: If an individual is married or in a registered same-sex civil partnership, they are **2.25 times more likely** than random to be White, part of a married/same-sex civil partnership couple family, not a student, and living in residence type H. This has support of 32% of the dataset, and 84% of cases like this antecedent, meets the consequence.

Rule4: If an individual is White, part of a married/same-sex civil partnership couple family, and not a student, they are **2.25 times more likely** than random to be classified as a usual resident and be married or in a registered same-sex civil partnership. This has support of 32% of the dataset, and 86% of cases like this antecedent, meets the consequence.

Rule5: If an individual is a usual resident, UK-born, married or in a registered same-sex civil partnership, and living in residence type H, they are **2.27 times more likely** than random to belong to a married/same-sex civil partnership couple family and to not be a student. This has support of 30% of the dataset, and 97% of cases like this antecedent, meets the consequence.

FP-growth Algorithm (Extra):

For this algorithm, I used the same parameters as that of Apriori.

Table 7 Rules for FP growth Algorithm

Rule	Antecedent	Consequent	Support	Confidence	Lift
1.	(Marital Status_Married or in a registered same-sex civil partnership, Country of Birth_UK, Residence Type_H)	(Family Composition_Married/same-sex civil partnership couple family, Student_No, Population Base_Usual resident)	0.30	0.97	2.31

2.	(Family Composition_Married/same-sex civil partnership couple family, Student_No, Population Base_Usual resident)	(Marital Status_Married or in a registered same-sex civil partnership, Country of Birth_UK, Residence Type_H)	0.30	0.72	2.31
3.	(Marital Status_Married or in a registered same-sex civil partnership, Country of Birth_UK, Population Base_Usual resident, Residence Type_H)	(Family Composition_Married/same-sex civil partnership couple family, Student_No)	0.30	0.97	2.31
4.	(Family Composition_Married/same-sex civil partnership couple	(Marital Status_Married or in a registered same-sex civil partnership,	0.30	0.72	2.31

	family, Student_No)	Country of Birth_UK, Residence Type_H)			
5.	(Family Composition_Married/same-sex civil partnership couple family, Student_No)	(Marital Status_Married or in a registered same-sex civil partnership, Country of Birth_UK, Population Base_Usual resident, Residence Type_H)	0.30	0.72	2.31

Interpretations of rules in Table 7:

Rule1: If an individual is married or in a registered same-sex civil partnership, UK-born, and living in residence type H, they are **2.31 times more likely** than random to belong to a married/same-sex civil partnership couple family, be a usual resident, and not be a student. This rule covers 30% of the dataset, and 97% of cases with this antecedent meet the consequent.

Rule2: If an individual If an individual belongs to a married/same-sex civil partnership couple family, is a usual resident, and not a student, they are **2.31 times more likely** than random to be married or in a registered same-sex civil partnership, UK-born, and live-in residence type H. This rule covers to 30% of the dataset, and 72% of cases with this antecedent meet the consequent.

Rule3: If an individual is married or in a registered same-sex civil partnership, UK-born, a usual resident, and lives in residence type H, they are **2.31 times more likely** than random to belong to a married/same-sex civil partnership couple family and not be a student. This rule covers 30% of the dataset, and 97% of cases with this antecedent meet the consequent.

Rule4: If an individual belongs to a married/same-sex civil partnership couple family and is not a student, they are **2.31 times more likely** than random to be married or in a registered same-sex civil partnership, UK-born, and live in residence type H. This rule covers to 30% of the dataset, and 72% of cases with this antecedent meet the consequent.

Rule5: If an individual belongs to a married/same-sex civil partnership couple family and is not a student, they are **2.31 times more likely** than random to be married or in a registered same-sex civil partnership, UK-born, a usual resident, and live in residence type H. This rule covers 30% of the dataset, and 72% of cases with this antecedent meet the consequent.

5. Perform clustering using 2 algorithms; present and discuss the results; compare the results of the 2 algorithms.

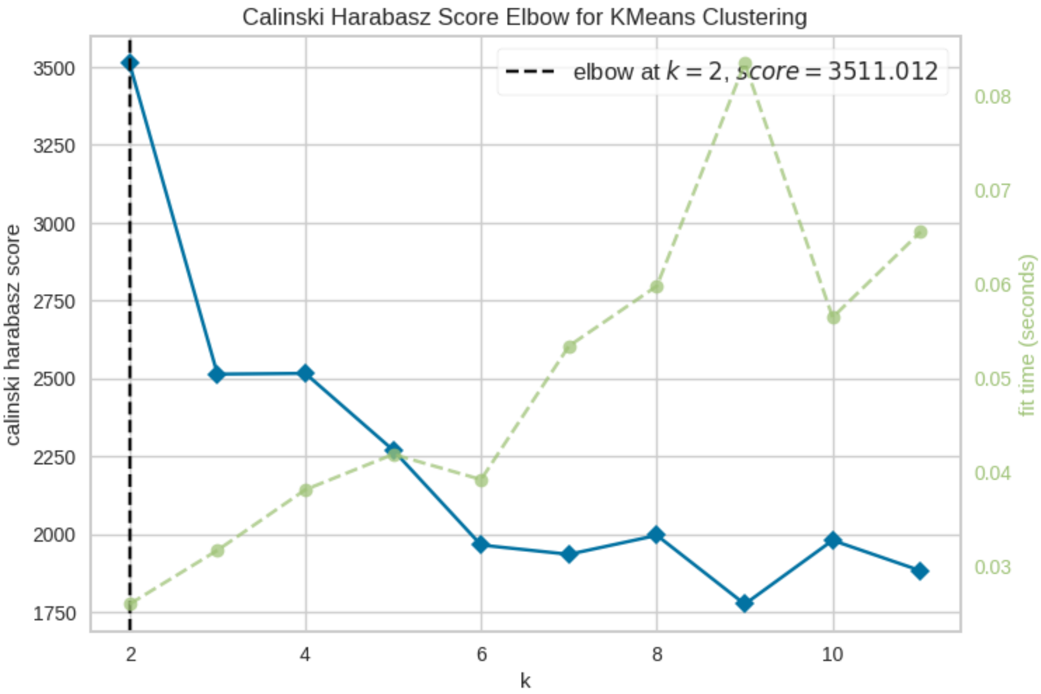
Data Preparation: Performed same data preparation steps as classification.

K-Means clustering:

For K-means, I achieved the silhouette score of 0.22 which suggests poor clustering.

For finding optimal K value, I used the elbow method and had optimal value as 2 in Figure 34.

Figure 34 Elbow Method for Finding optimum K



Then I generated silhouette plot for visualizing and found most narrow on k=2, in Figure 35. Then in Figure 36, I visualized the cluster for Age and Economic Activity and saw two meaningful clusters.

Figure 35 Silhouette Plot

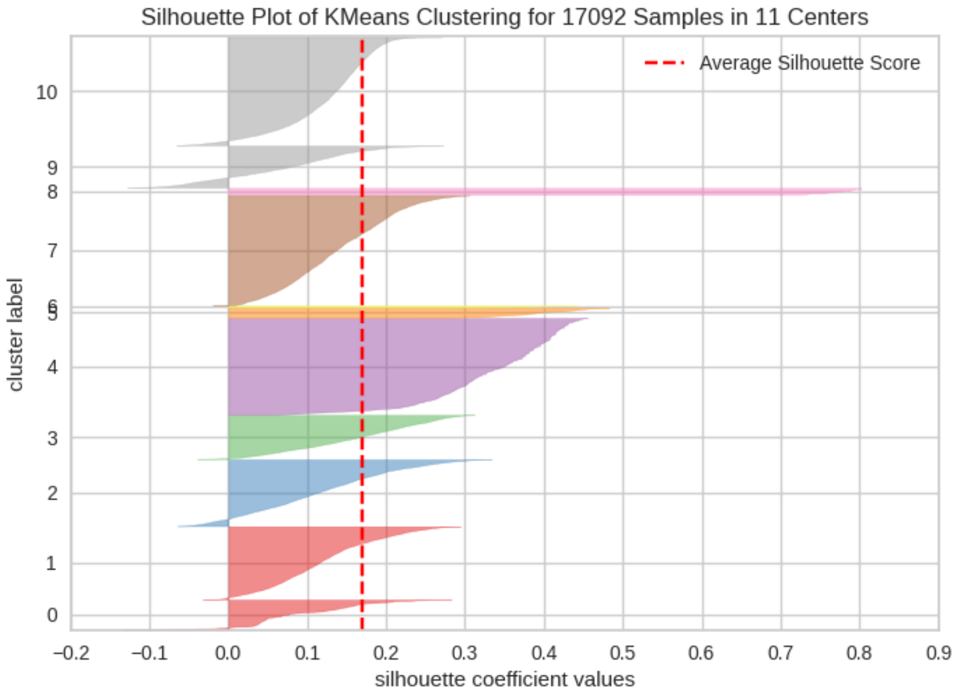


Figure 36 K Means Cluster Visualization (k=2)

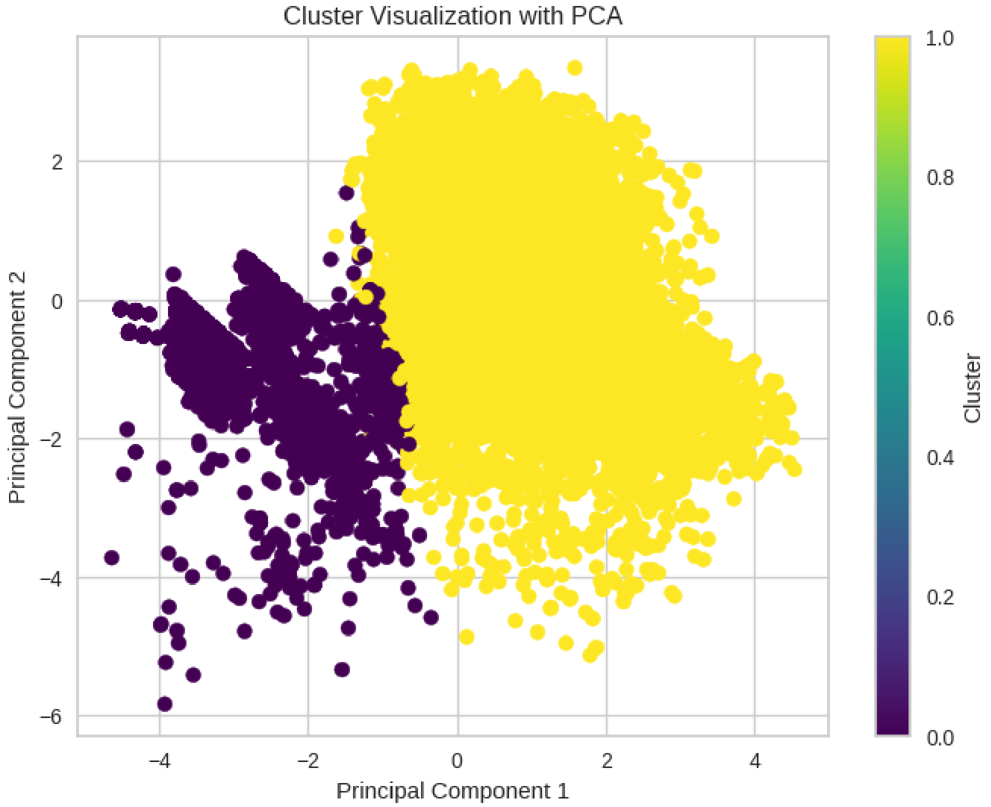


Figure 37 Contribution of each attribute

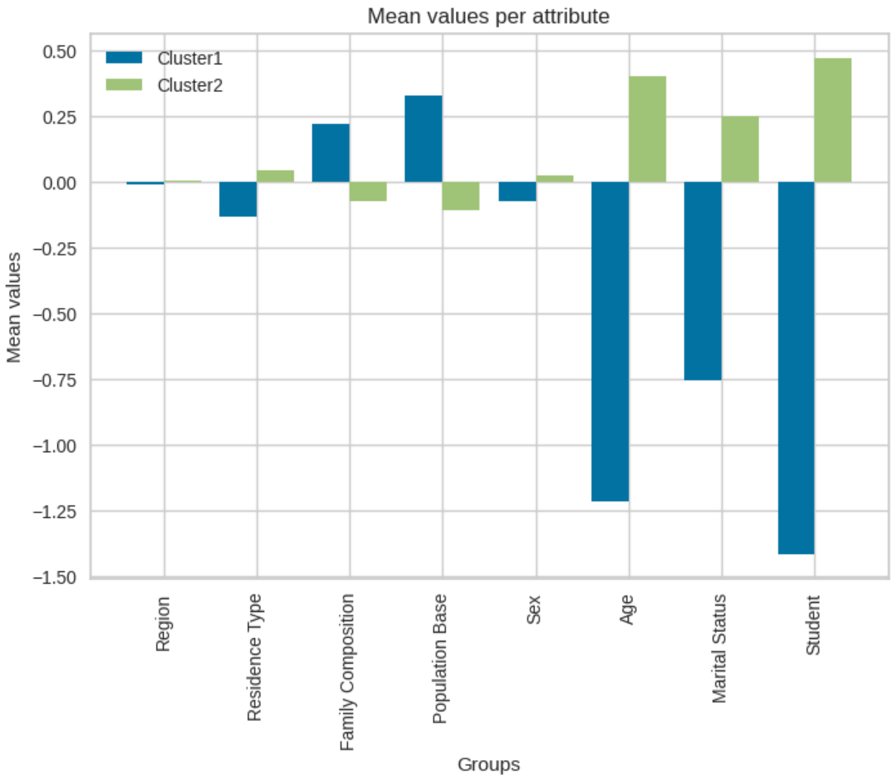


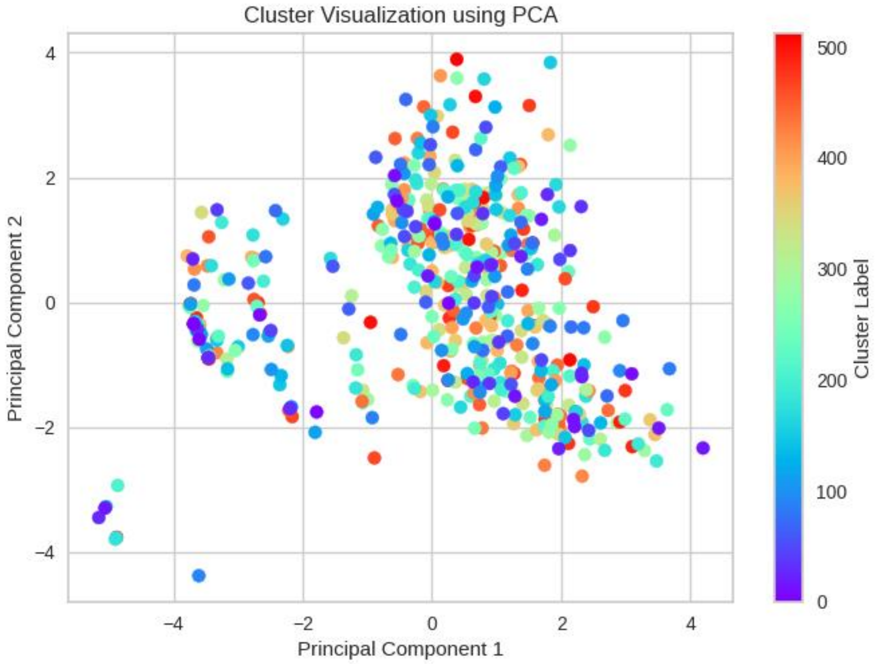
Figure 37 suggests "Age" and "Marital Status" are significant factors differentiating the clusters.

"Student" has a high positive mean in Cluster2.

Agglomerative Hierarchical clustering:

In Figure 38, the agglomerative hierarchical clustering is shown. Then I generated dendrogram in Figure 39, showing how all clusters come together as one.

Figure 38 Agglomerative Hierarchical Cluster visualization



For this clustering also, I got the silhouette score of 0.22 with 2 clusters.

In Figure 40, Age and Student has the most significance making the clusters, they have high positive and negative values in both clusters.

Figure 41, Figure 42, Figure 43, Figure 44 visualizes clusters on the values of n=2,3,4,5.

Figure 44, Figure 45, Figure 46, Figure 47 visualizes clusters with different linkage parameters (ward, average, complete, single) with 5 clusters and with *ward* as the linkage parameter, the clusters make more sense.

Figure 39 Hierarchical Cluster Dendrogram

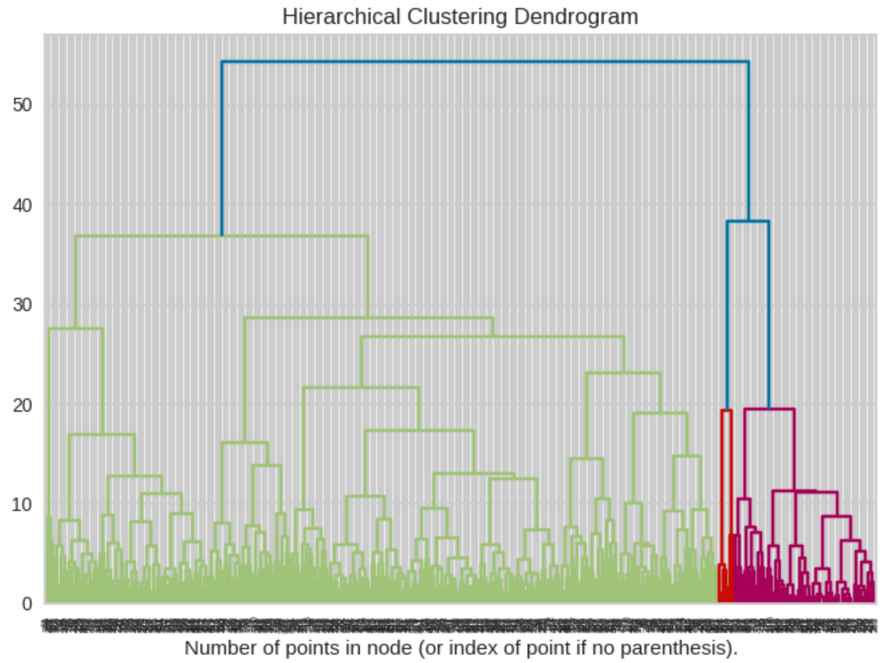


Figure 40 Mean values across the attribute

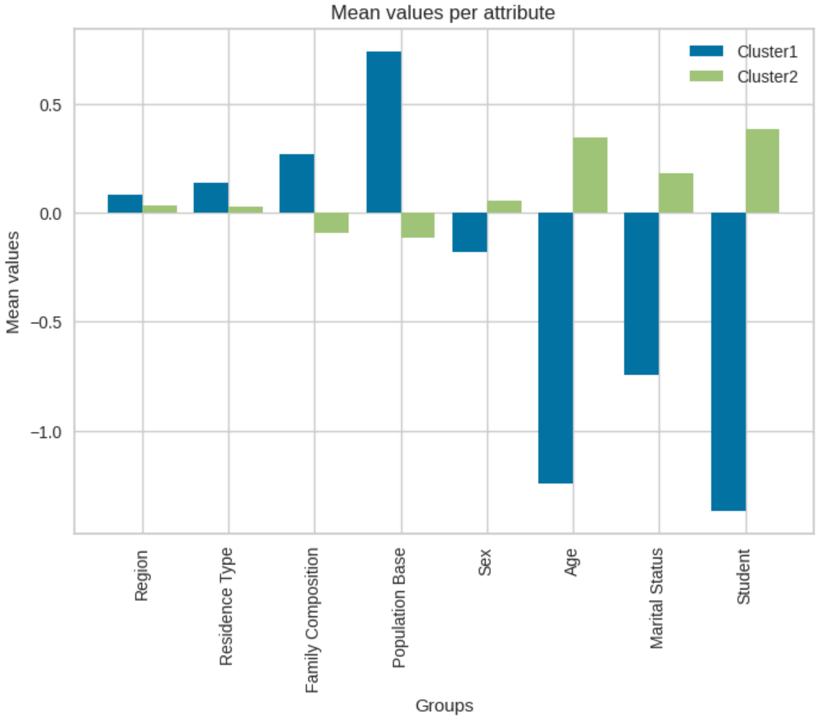


Figure 41 Hierarchical Cluster (n=2)

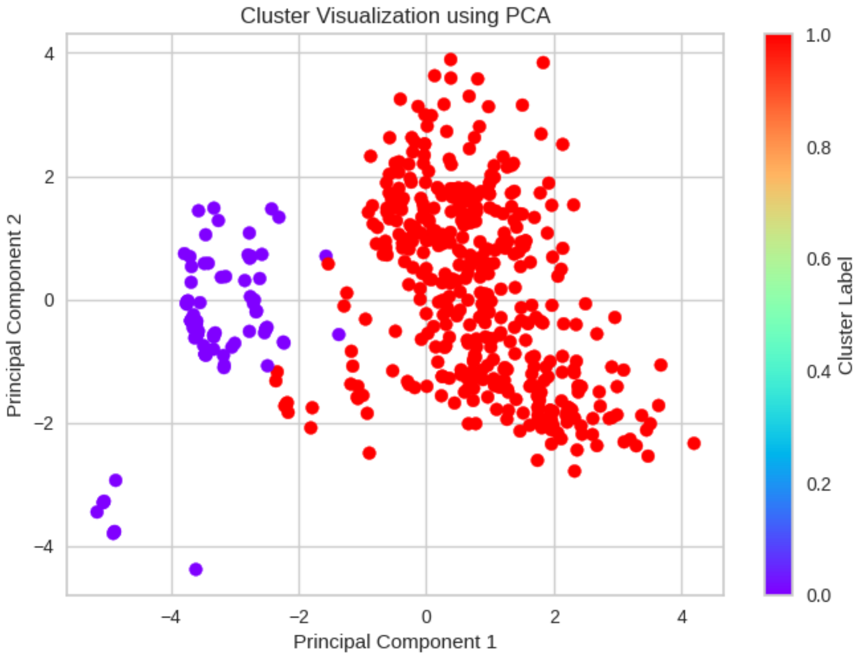


Figure 42 Hierarchical Cluster (n=3)

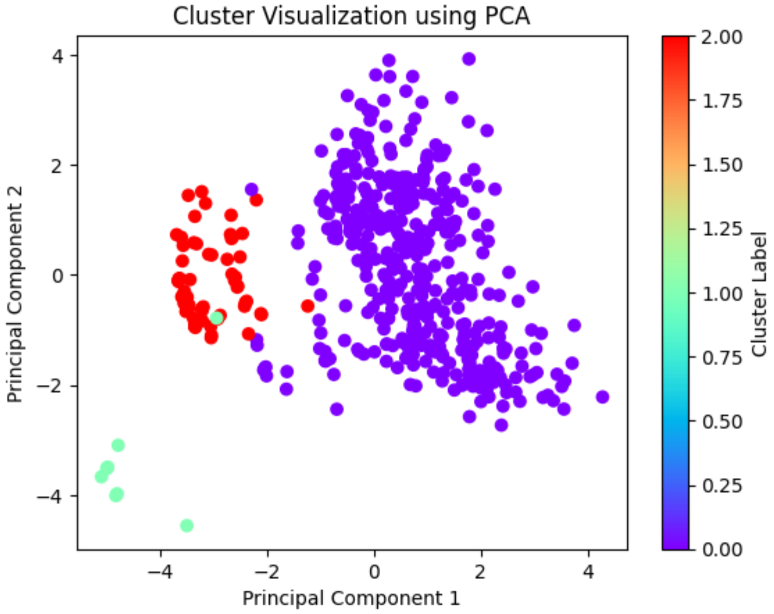


Figure 43 Hierarchical Cluster (n=4)

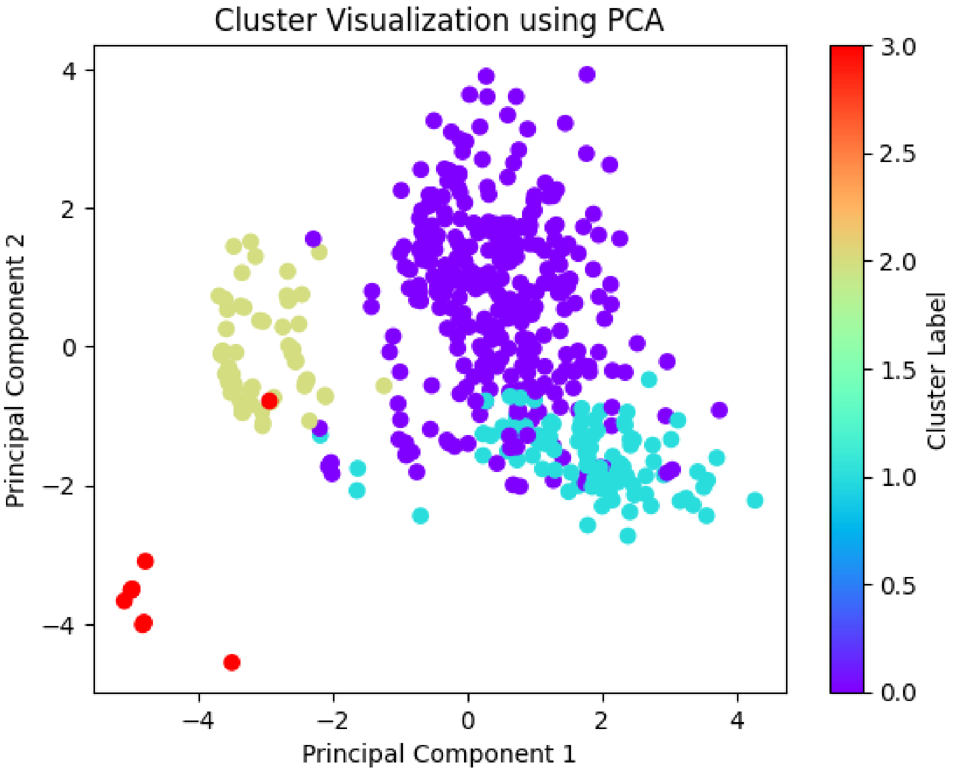


Figure 44 Hierarchical Cluster (n=5, linkage=ward)

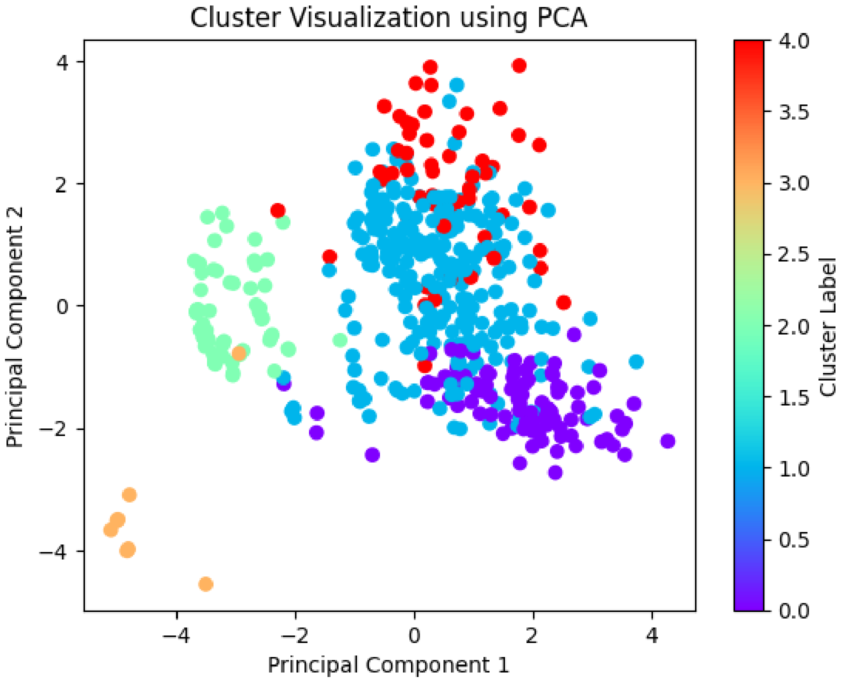


Figure 45 Hierarchical Cluster (n=5, linkage=average)

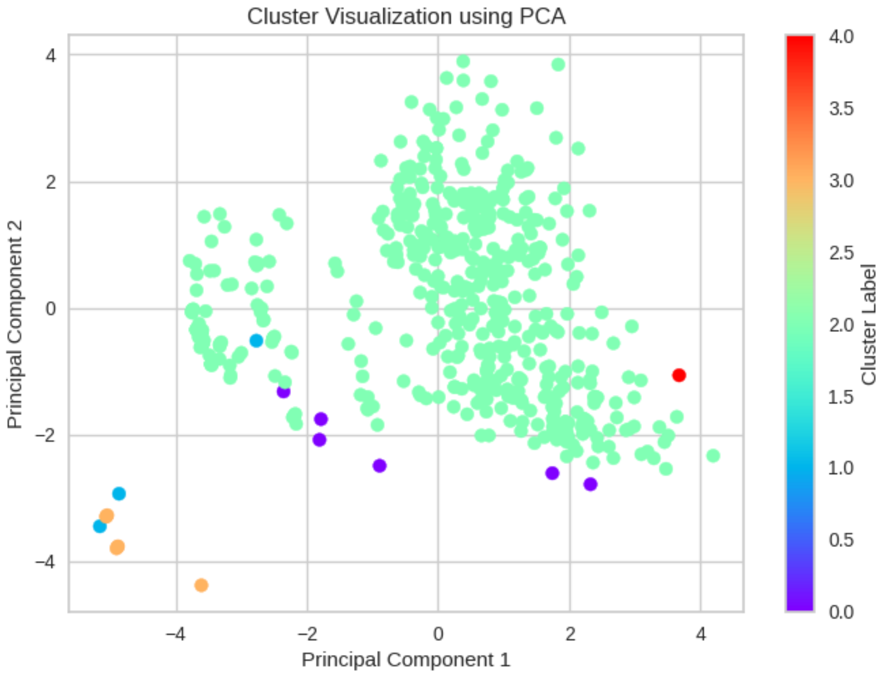


Figure 46 Hierarchical Cluster (n=5, linkage=complete)

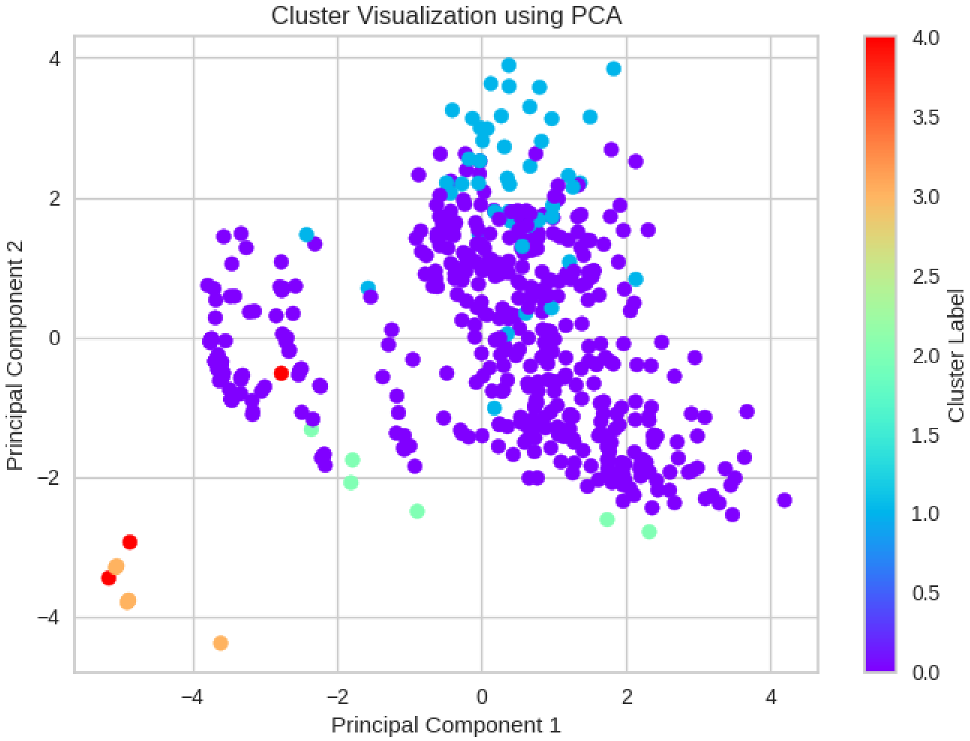
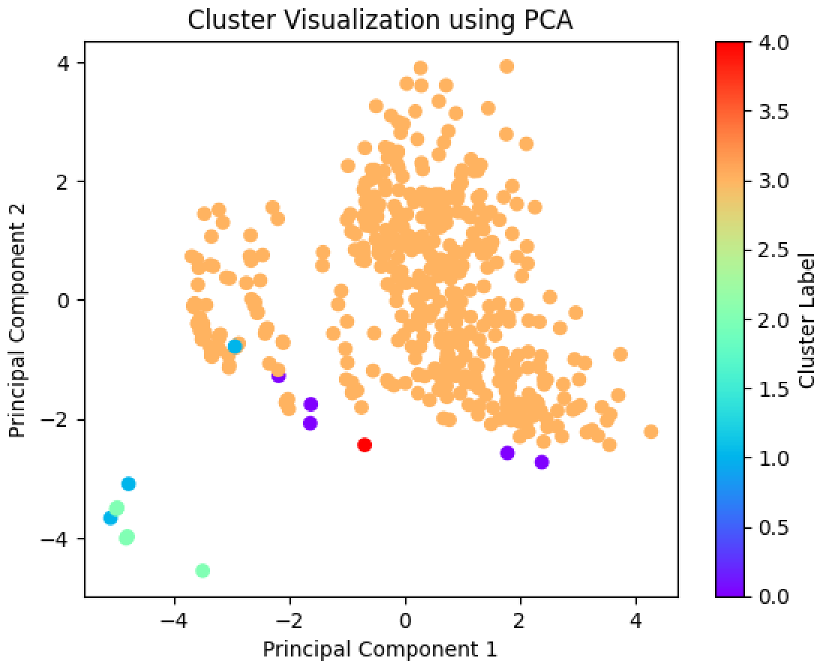


Figure 47 Hierarchical Cluster (n=5, linkage=single)



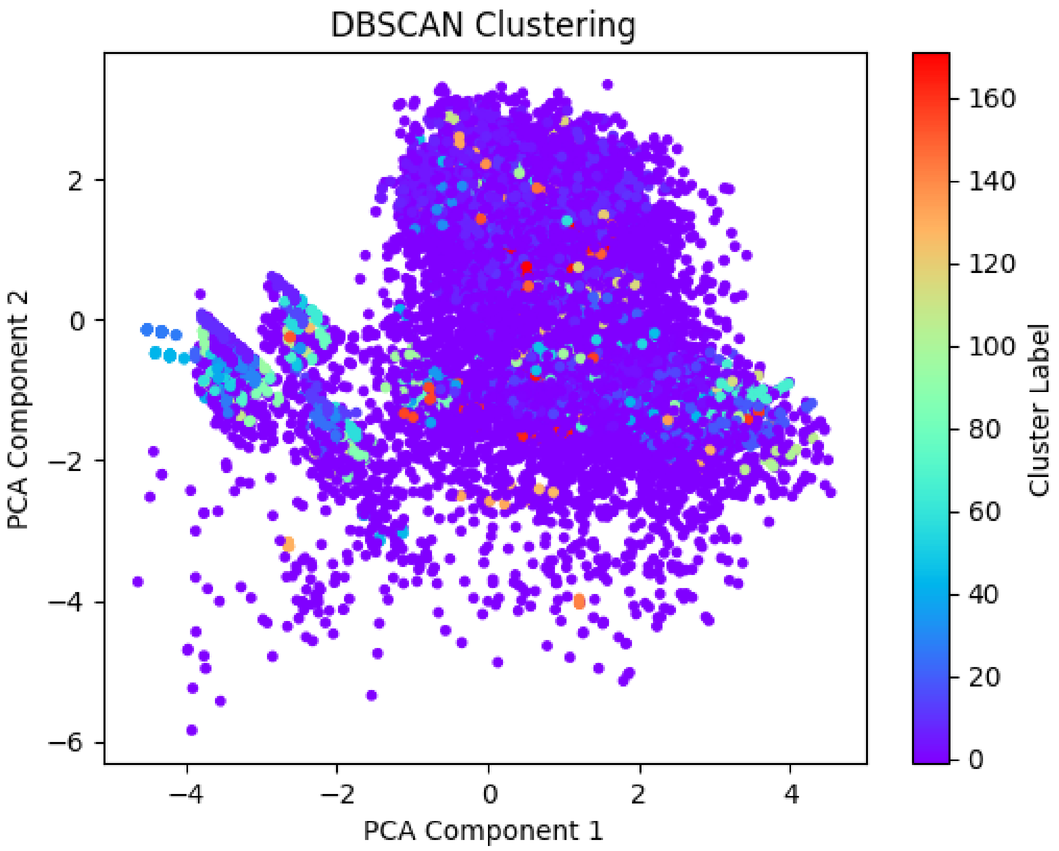
DBScan clustering (Extra):

DBScan is hyperparameter sensitive. It depends on the value of eps and min_samples. The best silhouette score I got from this algorithm is -0.27 which suggests, no clusters and not suitable to this dataset. The optimal hyperparameters I used to be-

- eps=0.3
- min_samples=5

Moreover, there were highest noise points than the number of datapoints in the clusters, and it created more than 167 clusters, which can be visualized in Figure 48.

Figure 48 DBSCAN Cluster



Comparison between all 3 clustering:

Among all three clustering algorithms, DBSCAN performed worst giving no clear or meaningful clusters. For K means and Agglomerative, they both did similar clustering as their silhouette score was also same for 2 clusters, their visualizations also look similar.

References

Jiawei Han, M. K. (2011). *Data Mining: Concepts and Techniques.* Elsevier.

OpenAI. (n.d.). *ChatGPT*. Retrieved from https://chat.openai.com/

[Open AI was used for sentence making, fixing grammatical errors, understand some concepts better.]