UNIVERSITY OF PORTSMOUTH

COURSEWORK 1

MODULE TITLE- INTELLIGENT DATA AND TEXT ANALYTICS

MODULE CODE- M33147

TOPIC- Multi-Method Data Analysis On The 2011 UK Census Data: Descriptive Analytics, Classification, Regression, Association Rule Mining, And Clustering

TABLE OF CONTENTS

Task 1: Descriptive Analytics

1. Basic Statistics of all Attributes

Before starting with the basic statistics of each attribute, it's important to mention that the values present inside each column are mapped to the actual name of the values present in the data dictionary (microdatateachingvariables.pdf) provided to us before performing statistical and visual analysis of each attribute for the sake of easy interpretation and correct analysis of results. Moreover, the default data types of 'Family Composition', 'Population Base', 'Sex', 'Age', 'Marital Status', 'Student', 'Country of Birth', 'Health', 'Ethnic Group', 'Religion', 'Economic Activity', 'Occupation', 'Industry', 'Hours worked per week', and 'Approximated Social Grade' columns are given as numeric data type (int64). However, the actual nature of the above-mentioned columns is categorical. As a result, these columns are first converted from numerical (int64) to categorical (object). This conversion is done because our analysis will only be correct and logical when these categorical attributes are treated as categorical while describing using the describe() function.

The other columns that do not need conversion of data type and appears to be correct by default are 'Region', 'Residence Type', and 'No of hours', which are supposed to be categorical, categorical, and numerical, respectively, and are provided in the correct data type by default.

Now, we proceed to describe each attribute present in our dataset.

1.1 Basic Statistics of 'Region' Attribute

```
census['Region'].describe()
```

```
count         569740
unique            10
top       South East
freq           88084
Name: Region, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Region' column
unique	10	There are 10 unique values in the 'Region' column
top	South East	'South East' is the most frequently occurring value in the 'Region' column
frequency	88084	The number of times South East appears in the 'Region' column

		is 88084
Name	Region	This is the name of the column which is ‘Region’
dtype	object	The datatype of this column is object, which indicates categorical data

1.2 Basic Statistics of ‘Residence Type’ Attribute

```
census['Residence Type'].describe()
```

```
count                                          569740
unique                                              2
top       Not resident in a communal establishment
freq                                           559086
Name: Residence Type, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the ‘Residence Type’ column
unique	2	There are 2 unique values in the ‘Residence Type’ column
top	Not resident in a communal establishment	‘Not resident in a communal establishment’ is the most frequently occurring value in the ‘Residence Type’ column.
frequency	559086	The number of times ‘Not resident in a communal establishment’ appears in the ‘Residence Type’ column is 559086
Name	Residence Type	This is the name of the column which is ‘Residence Type’
dtype	object	The datatype of this column is object, which indicates categorical data

1.3 Basic Statistics of 'Family Composition' Attribute

```
census['Family Composition'].describe()
```

```
count                                                569740
unique                                                    7
top       Married/same-sex civil partnership couple family
freq                                                 300961
Name: Family Composition, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Family Composition' column
unique	7	There are 7 unique values in the 'Family Composition' column
top	Married/same-sex civil partnership couple family	'Married/same-sex civil partnership couple family' is the most frequently occurring value in the 'Family Composition' column
frequency	300961	The number of times 'Married/same-sex civil partnership couple family' appears in the 'Family Composition' column is 300961
Name	Family Composition	This is the name of the column which is 'Family Composition'
dtype	object	The data type of this column is object, which indicates categorical data

1.4 Basic Statistics of 'Population Base' Attribute

```
census['Population Base'].describe()
```

```
count             569740
unique                 3
top       Usual resident
freq              561039
Name: Population Base, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the

		‘Population Base’ column
unique	3	There are 3 unique values in the ‘Population Base’ column
top	Usual resident	‘Usual resident’ is the most frequently occurring value in the ‘Population Base’ column
frequency	561039	The number of times ‘Usual resident’ appears in the ‘Population Base’ column is 561039
Name	Population Base	This is the name of the column which is ‘Population Base’
dtype	object	The data type of this column is object, which indicates categorical data

1.5 Basic Statistics of ‘Sex’ Attribute

```
census['Sex'].describe()
```

```
count     569740
unique         2
top       Female
freq      289172
Name: Sex, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the ‘Sex’ column
unique	2	There are 2 unique values in the ‘Sex’ column
top	Female	‘Female’ is the most frequently occurring value in the ‘Sex’ column
frequency	289172	The number of times ‘Female’ appears in the ‘Sex’ column is 289172
Name	Sex	This is the name of the column which is ‘Sex’
dtype	object	The data type of this column is object, which indicates categorical data

1.6 Basic Statistics of 'Age' Attribute

```
census['Age'].describe()
```

```
count      569740
unique          8
top       0 to 15
freq       106832
Name: Age, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Age' column
unique	8	There are 8 unique values in the 'Age' column
top	0 to 15	'0 to 15' is the most frequently occurring age group in the 'Age' column
frequency	106832	The number of times '0 to 15' appears in the 'Age' column is 106832
Name	Age	This is the name of the column which is 'Age'
dtype	object	The data type of this column is object, which indicates categorical data

1.7 Basic Statistics of 'Marital Status' Attribute

```
census['Marital Status'].describe()
```

```
count                                                                  569740
unique                                                                      5
top       Single (never married or never registered a same-sex civil partnership)
freq                                                                   270999
Name: Marital Status, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Marital Status' column
unique	5	There are 5 unique values in the 'Marital Status' column
top	Single (never married or never	'Single (never married or never

	registered a same-sex civil partnership)	registered a same-sex civil partnership)' is the most frequently occurring value in the 'Marital Status' column
frequency	270999	The number of times 'Single (never married or never registered a same-sex civil partnership)' appears in the 'Marital Status' column is 270999
Name	Marital Status	This is the name of the column which is 'Marital Status'
dtype	object	The data type of this column is object, which indicates categorical data

1.8 Basic Statistics for 'Student' Attribute

```
census['Student'].describe()
```

```
count     569740
unique         2
top           No
freq      443203
Name: Student, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Student' column
unique	2	There are 2 unique values in the 'Student' column
top	No	'No' is the most frequently occurring value in the 'Student' column
frequency	443203	The number of times 'No' appears in the 'Student' column is 443203
Name	Student	This is the name of the column which is 'Student'
dtype	object	The data type of this column is object, which indicates categorical data

1.9 Basic Statistics of 'Country of Birth' Attribute

```
census['Country of Birth'].describe()
```

```
count     569740
unique         3
top           UK
freq      485645
Name: Country of Birth, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Country of Birth' column
unique	3	There are 3 unique values in the 'Country of Birth' column
top	UK	'UK' is the most frequently occurring value in the 'Country of Birth' column
frequency	485645	The number of times 'UK' appears in the 'Country of Birth' column is 485645
Name	Country of Birth	This is the name of the column which is 'Country of Birth'
dtype	object	The data type of this column is object, which indicates categorical data

1.10 Basic Statistics of 'Health' Attribute

```
census['Health'].describe()
```

```
count                569740
unique                    6
top       Very good health
freq                 264971
Name: Health, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Health' column
unique	6	There are 6 unique values in the 'Health' column
top	Very good health	'Very good health' is the most frequently occurring value in

		the ‘Health’ column
frequency	264971	The number of times ‘Very good health’ appears in the ‘Health’ column is 264971
Name	Health	This is the name of the column which is ‘Health’
dtype	object	The data type of this column is object, which indicates categorical data

1.11 Basic Statistics of ‘Ethnic Group’ Attribute

```
census['Ethnic Group'].describe()
```

```
count     569740
unique         6
top        White
freq      483477
Name: Ethnic Group, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the ‘Ethnic Group’ column
unique	6	There are 6 unique values in the ‘Ethnic Group’ column
top	White	‘White’ is the most frequently occurring value in the ‘Ethnic Group’ column
frequency	483477	The number of times ‘White’ appears in the ‘Ethnic Group’ column is 483477
Name	Ethnic Group	This is the name of the column which is ‘Ethnic Group’
dtype	object	The data type of this column is object, which indicates categorical data

1.12 Basic Statistics of 'Religion' Attribute

```
census['Religion'].describe()
```

```
count        569740
unique           10
top       Christian
freq         333481
Name: Religion, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Religion' column
unique	10	There are 10 unique values in the 'Religion' column
top	Christian	'Christian' is the most frequently occurring value in the 'Religion' column
frequency	333481	The number of times 'Christian' appears in the 'Religion' column is 333481
Name	Religion	This is the name of the column which is 'Religion'
dtype	object	The data type of this column is object, which indicates categorical data

1.13 Basic Statistics of 'Economic Activity' Attribute

```
census['Economic Activity'].describe()
```

```
count                              569740
unique                                 10
top       Economically active: Employee
freq                               216024
Name: Economic Activity, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Economic Activity' column
unique	10	There are 10 unique values in the 'Economic Activity' column
top	Economically active: Employee	'Economically active:

		Employee' is the most frequently occurring value in the 'Economic Activity' column
frequency	216024	The number of times 'Economically active: Employee' appears in the 'Economic Activity' column is 216024
Name	Economic Activity	This is the name of the column which is 'Economic Activity'
dtype	object	The data type of this column is object, which indicates categorical data

1.14 Basis Statistics of 'Occupation' Attribute

```
census['Occupation'].describe()
```

```
count
569740
unique
10
top       No code required (People aged under 16, people who have never worked and students or schoolchildren living away durin
g term-time)
freq
149984
Name: Occupation, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Occupation' column
unique	10	There are 10 unique values in the 'Occupation' column
top	No code required (People aged under 16, people who have never worked and students or schoolchildren living away during term-time)	'No code required (People aged under 16, people who have never worked and students or schoolchildren living away during term-time)' is the most frequently occurring value in the 'Occupation' column
frequency	149984	The number of times 'No code required (People aged under 16, people who have never worked and students or schoolchildren living away during term-time)' appears in the 'Occupation'

		column is 149984
Name	Occupation	This is the name of the column which is ‘Occupation’
dtype	object	The data type of this column is object, which indicates categorical data

1.15 Basic Statistics of ‘Industry’ Attribute

```
census['Industry'].describe()
```

```
count
569740
unique
13
top       No code required (People aged under 16, people who have never worked, and students or schoolchildren living away duri
ng term-time)
freq
149984
Name: Industry, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the ‘Industry’ column
unique	13	There are 13 unique values in the ‘Industry’ column
top	No code required (People aged under 16, people who have never worked and students or schoolchildren living away during term-time)	‘No code required (People aged under 16, people who have never worked and students or schoolchildren living away during term-time)’ is the most frequently occurring value in the ‘Industry’ column
frequency	149984	The number of times ‘No code required (People aged under 16, people who have never worked and students or schoolchildren living away during term-time)’ appears in the ‘Industry’ column is 149984
Name	Industry	This is the name of the column which is ‘Industry’
dtype	object	The data type of this column is object, which indicates categorical data

1.16 Basic Statistics of 'Hours worked per week' Attribute

```
census['Hours worked per week'].describe()
```

```
count
569740
unique
5
top       No code required (People aged under 16, people not working, and stude
nts or schoolchildren living away during term-time)
freq
302321
Name: Hours worked per week, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Hours worked per week' column
unique	5	There are 5 unique values in the 'Hours worked per week' column
top	No code required (People aged under 16, people not working, and students or schoolchildren living away during term-time)	'No code required (People aged under 16, people not working, and students or schoolchildren living away during term-time)' is the most frequently occurring value in the 'Hours worked per week' column
frequency	302321	The number of times 'No code required (People aged under 16, people not working, and students or schoolchildren living away during term-time)' appears in the 'Hours worked per week' column is 302321
Name	Hours worked per week	This is the name of the column which is 'Hours worked per week'
dtype	object	The data type of this column is object, which indicates categorical data

1.17 Basic Statistics of 'No of hours' Attribute

```
census['No of hours'].describe()
```

```
count    267419.000000
mean         35.234789
std          13.520881
min           1.000000
25%          27.000000
50%          37.000000
75%          45.000000
max          60.000000
Name: No of hours, dtype: float64
```

STATISTIC	VALUE	DESCRIPTION
count	267419	There are 267419 entries in the 'No of hours' column
mean	35.234789	The average no of working hours is 35.234789
std	13.520881	The standard deviation shows that the no of hours worked varies by around 13.520881 hours around the mean which basically means that it varies below or above average (35.234789) by 13.520881 hours
min	1	The minimum no of hours is 1 hour
25%	27	It means that 25% of the people worked 27 hours and less. It refers to the first quartile, which indicates that around 25% of the dataset has values less than or equal to 27 hours as no of hours and 75% of the dataset has values greater than 27 hours.
50%	37	It means that 50% of the people worked 37 hours or less. It refers to the median, which indicates that around 50% of the dataset has values less than or equal to 37 hours as the number of hours and 50% of the dataset has values greater than 37 hours
75%	45	It means that 75% of the people

		worked 45 hours or less. It refers to the third quartile, which indicates that around 75% of the dataset has values less than or equal to 45 hours as the number of hours, and 25% of the dataset has values greater than 45 hours.
max	60	The maximum no of hours is 60 hours
Name	No of hours	This is the name of the column which is 'No of hours'
dtype	float64	The data type of this column is float, which indicates numerical data

1.18 Basic Statistics of 'Approximated Social Grade' Attribute

```
census['Approximated Social Grade'].describe()
```

```
count      569740
unique          5
top            C1
freq       159642
Name: Approximated Social Grade, dtype: object
```

STATISTIC	VALUE	DESCRIPTION
count	569740	There are 569740 entries in the 'Approximated Social Grade' column
unique	5	There are 5 unique values in the 'Approximated Social Grade' column
top	C1	'C1' is the most frequently occurring value in the 'Approximated Social Grade' column
frequency	159642	The number of times 'C1' appears in the 'Approximated Social Grade' column is 159642
Name	Approximated Social Grade	This is the name of the column

		which is ‘Approximated Social Grade’
dtype	object	The data type of this column is object, which indicates categorical data

Reason for Excluding the Person ID Column from Descriptive Analysis with describe() function
Descriptive analysis is not performed on the ‘Person ID’ column because it is simply an identifier and it does not have any statistical meaning on its own. This column will be deleted during the preprocessing phase before performing classification in Task 2.

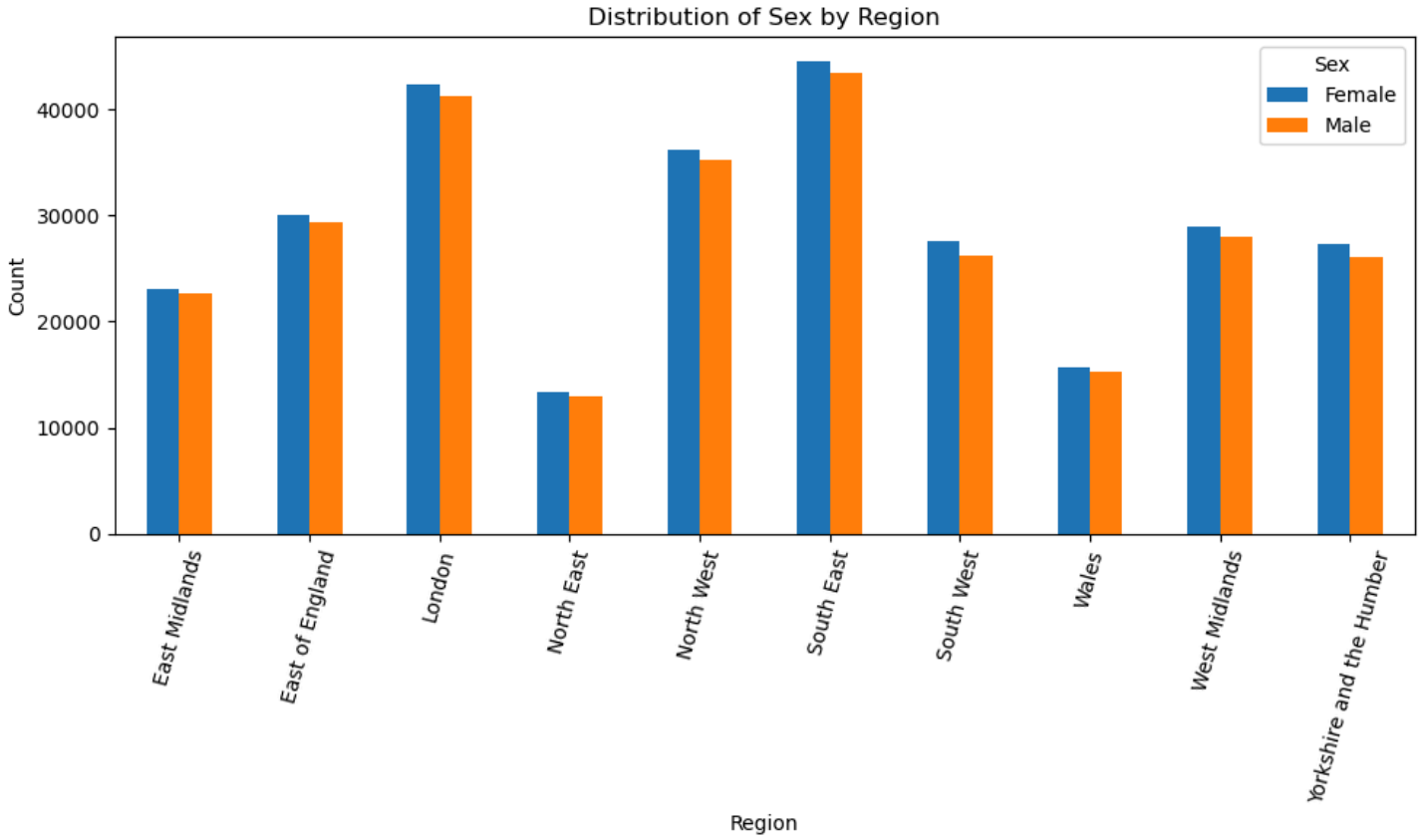
2. Visualisations to Analyse Relationships Between Two or More Important Attributes

2.1 Distribution of 'Sex' by 'Region'

2.1.1 Using ‘Contingency Table’ to show the distribution of ‘Sex’ by ‘Region’.

Sex	Female	Male
Region		
East Midlands	23095	22687
East of England	30060	29351
London	42382	41200
North East	13404	12945
North West	36215	35221
South East	44582	43502
South West	27541	26233
Wales	15670	15306
West Midlands	28895	27980
Yorkshire and the Humber	27328	26143

2.1.2 Using ‘Grouped Bar Chart’ to show the distribution of ‘Sex’ by ‘Region’



Key Observations and Interpretations from the Graph

Region	**Observations on the Distribution of ‘Sex’**
East Midlands	The counts of males and females are almost equal with females being slightly higher than males.
East of England	Females have a slightly higher count as compared to males.
London	Both genders have very high counts with females higher than males.
North East	Both genders have the lowest count compared to other regions with females being slightly higher than males. This also indicates the region with the lowest population density.
North West	Both sexes have considerably high counts with females slightly higher than males.
South East	This region has the highest counts for both sexes as compared to other regions with females slightly higher than males. This also indicates the region with the highest population density.

South West	Both sexes have moderate counts with females being higher than males.
Wales	The counts of both sexes are lower with slightly more than the 'North East' region. The distribution of males and females is almost equal with females having a slightly higher count than males.
West Midlands	Both genders have considerably high counts with females a bit higher than males.
Yorkshire and the Humber	Females have slightly higher counts than males.

Final Interpretation

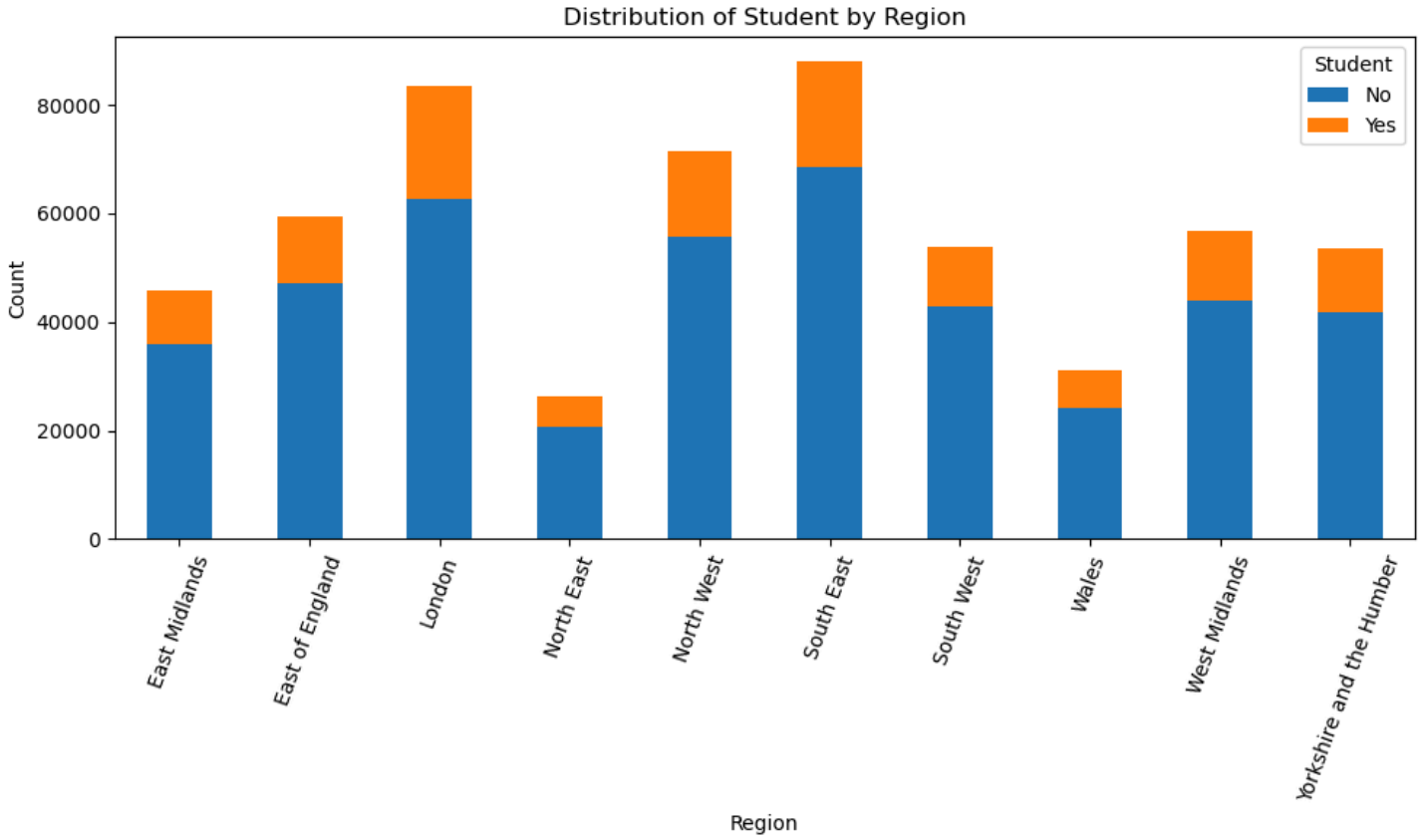
- Females showed higher counts as compared to males across all the regions.
- South East region has the highest population density.
- North East region has the lowest population density.

2.2 Distribution of 'Student' across different 'Region'

2.2.1 Using 'Contingency Table' to show the distribution of 'Student' across different 'Region'

Student	No	Yes
Region		
East Midlands	35817	9965
East of England	47048	12363
London	62748	20834
North East	20628	5721
North West	55781	15655
South East	68644	19440
South West	42773	11001
Wales	24053	6923
West Midlands	44036	12839
Yorkshire and the Humber	41675	11796

2.2.2 Using ‘Stacked Bar Chart’ to show the distribution of ‘Student’ across different ‘Region’



Key Observations and Interpretations from the Graph

Region	**Observations on the Distribution of ‘Student’**
East Midlands	This region has a moderate concentration of students.
East of England	Moderate concentration of students with more count than East of Midlands.
London	This region has the highest student population as compared to other regions.
North East	This region has the lowest student population among all the regions.
North West	The highest concentration of students after London and South East.
South East	This region has a very high student population, it is the second most popular region for students after London.
South West	Moderate concentration of students.
Wales	This region has a very low concentration of students with a slightly higher count than North

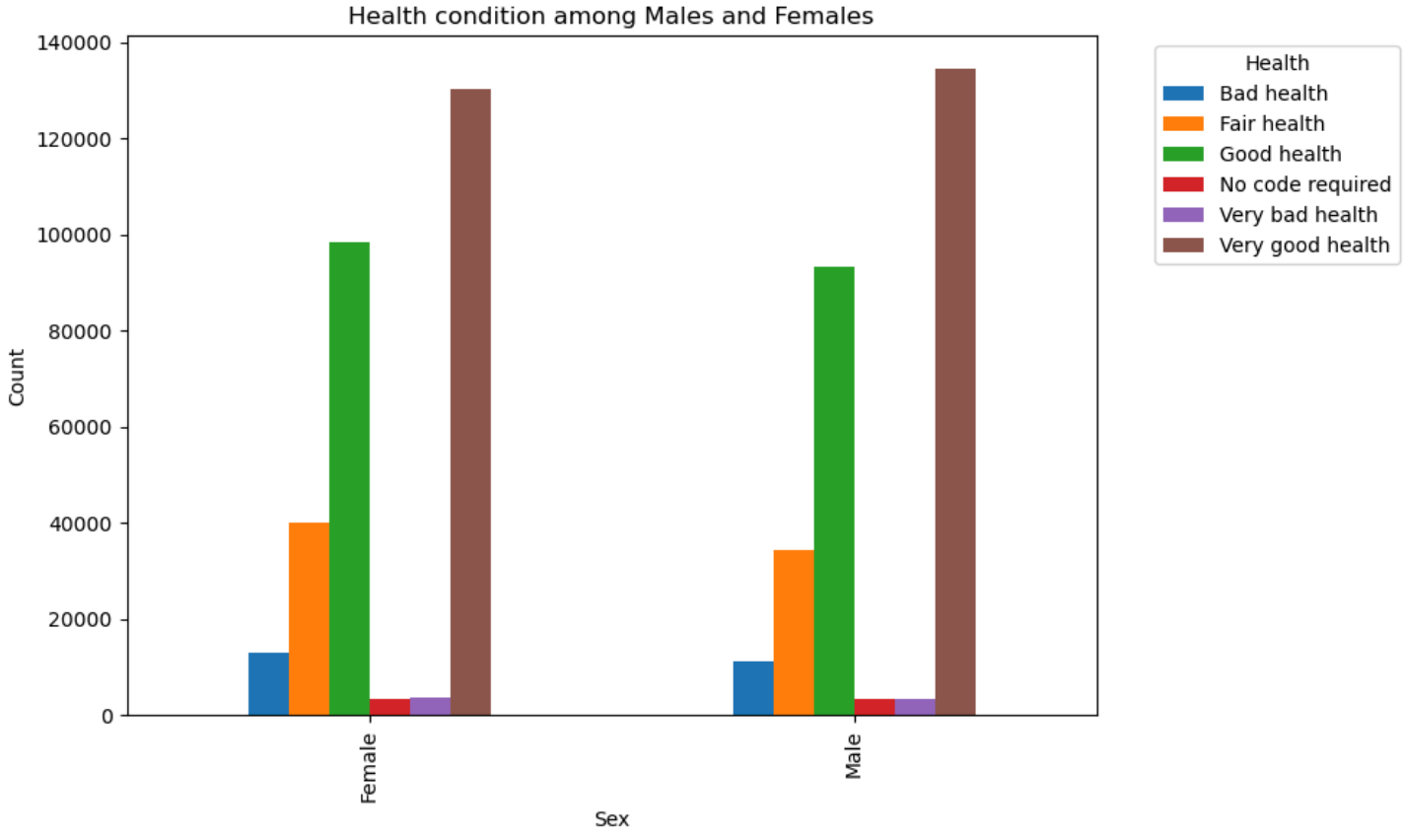
	East.
West Midlands	Moderate to high concentration of students having count slightly more than South West.
Yorkshire and the Humber	Moderate concentration of students having a slightly higher count than East Midlands.
Final Interpretation • London, South East and North West have the highest population of students which basically indicates a good hub for education and academic opportunities. • North East and Wales have the lowest population of students which indicates a lack of good educational and academic infrastructure to attract students.	

2.3 'Health' conditions among the two 'Sex'

2.3.1 Using 'Contingency Table' to analyse the condition of 'Health' among the two 'Sex'

Health Sex	Bad health	Fair health	Good health	No code required	Very bad health	Very good health
Female	13191	40121	98383	3371	3795	130311
Male	11367	34359	93360	3433	3389	134660

2.3.2 Using 'Grouped Bar Chart' to analyse the condition of 'Health' among the two 'Sex'



Key Observations and Interpretations from the Graph

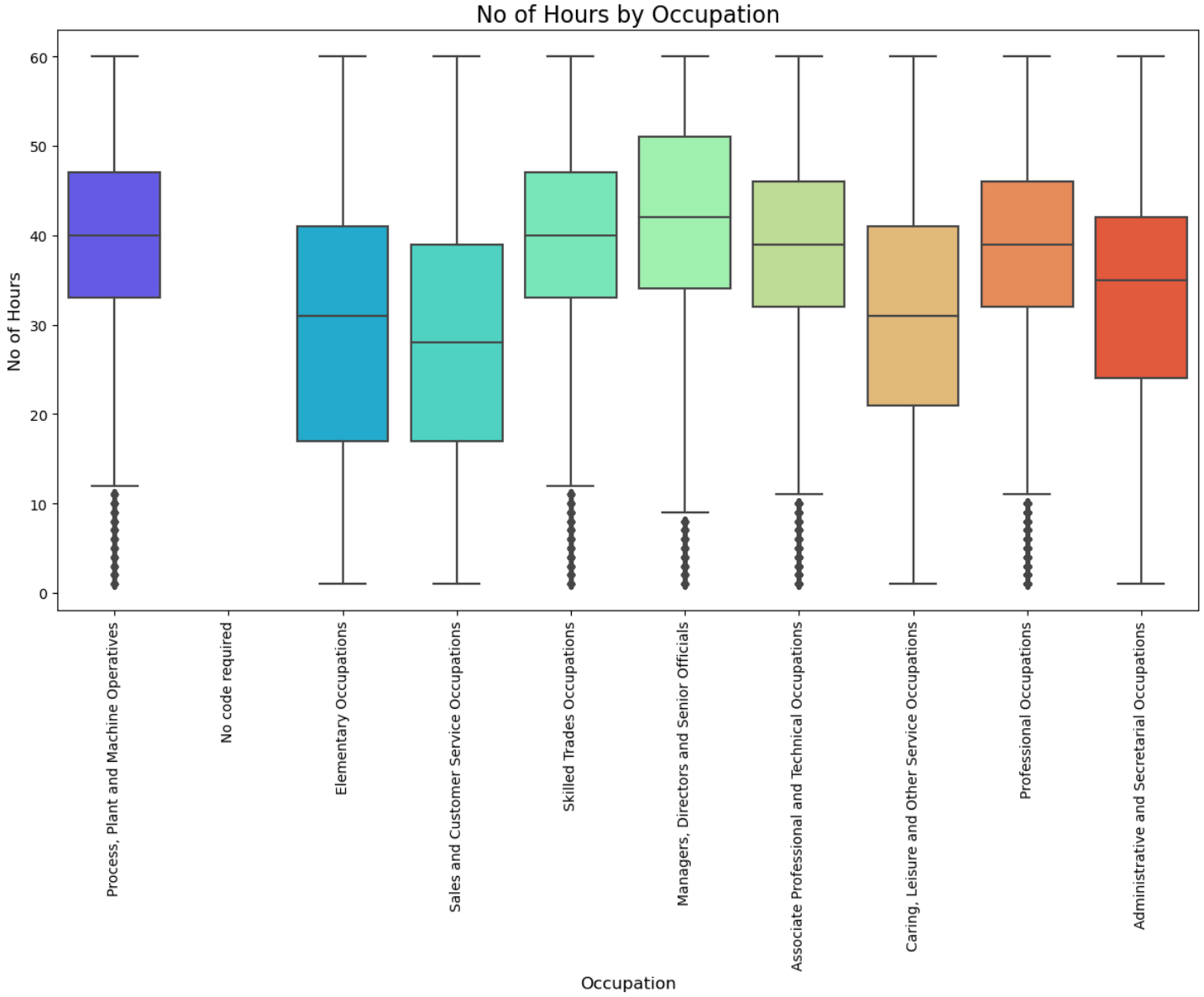
Sex	**Observation on the distribution of 'Health' Status**
Female	More females report 'Very Good Health' and 'Good Health' among other health statuses, and the count of 'Very Good Health' status is more than 'Good Health' status. But, as compared to male, slightly more females report 'Very Bad Health'.
Male	More number of males also report 'Very Good Health' and 'Good Health' among other health statuses, and the count of 'Very Good Health' status is more than 'Good Health' status.

Final Interpretation

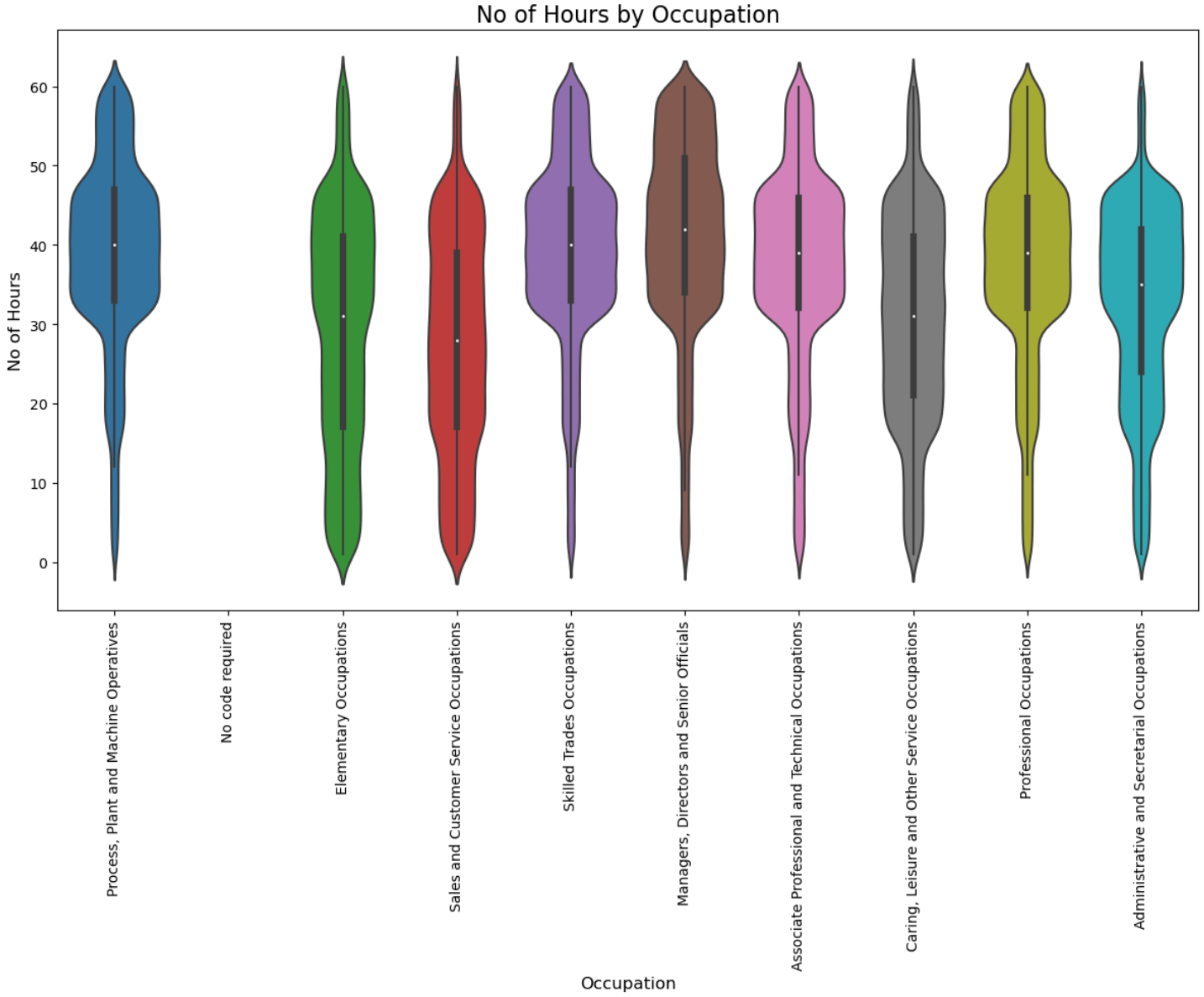
- Both genders commonly report "Very Good Health' and 'Good Health' which indicates a very good general health among the population.
- Although the count of 'Very Good Health' and 'Good Health' are high for both the genders but females report more cases of 'Very Bad Health' as compared to males

2.4 Distribution of ‘No of hours’ based on ‘Occupation’

2.4.1 Using ‘Boxplot’ to show the distribution of ‘No of hours’ based on ‘Occupation’



2.4.2 Using ‘Violinplot’ to show the distribution of ‘No of hours’ based on ‘Occupation’



Key Observations and Interpretations from the Graphs

Occupation	**No of hours**
Process, Plant and Machine Operatives	The median working hours is around 40 which means that 50% of the people pursuing this occupation work for 40 hours and less and the other 50 % work for more than around 40 hours. Most workers in this field work between 30 and 50 hours. The maximum working hour is 60 and the minimum is around 10, some people are working less than around 10 hours suggesting outliers.
No code required	No box plot shown because this represents the non-working class of people who are aged under 16, people who have never worked and students or schoolchildren living away during term-time.

Elementary Occupations	The median working hours is slightly more than 30. Most workers in this field work between 15 to 40 hours approximately.
Sales and Customer Service Occupations	The median working hours is slightly less than 30. Most workers in this field work between 15 to slightly less than 40 hours approximately.
Skilled Trades Occupations	The median working hours is around 40. Most workers in this field work between 30 and 50 hours approximately. The maximum working hour is 60 and the minimum is around 10, some people are working less than around 10 hours suggesting outliers.
Managers, Directors and Senior Officials	The median working hours is slightly above 40 hours. Most workers in this field work between 35 to 50 hours approximately. The maximum working hour is 60 and the minimum is around 10, some people are working less than around 10 hours suggesting outliers.
Associate Professional and Technical Occupations	The median working hours is slightly less than 40. Most workers in this field work between 30 to 50 hours approximately. The maximum working hour is 60 and the minimum is around 10, some people are working less than around 10 hours suggesting outliers.
Caring, Leisure and Other Service Occupations	The median working hours is around 30. Most workers in this field work between 20 to 40 hours approximately.
Professional Occupations	The median working hours is around 40. Most workers in this field work between 30 and 50 hours approximately. The maximum working hour is 60 and the minimum is around 10, some people are working less than around 10 hours suggesting outliers.
Administrative and Secretarial Occupations	The median working hours is slightly less than 40. Most workers in this field work between 25 to 40 hours approximately.

Final Interpretation

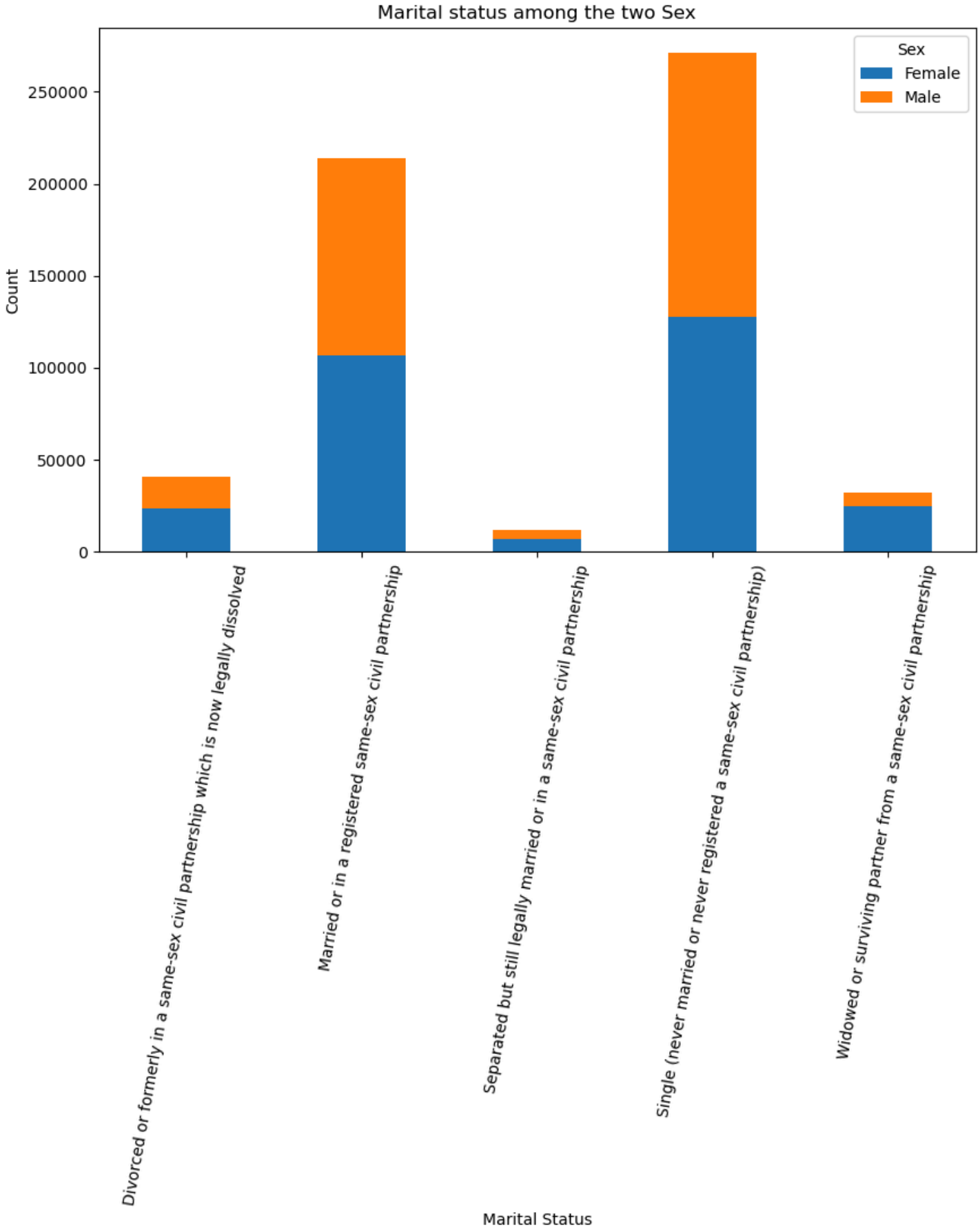
- Most occupations have median working hours of around 35 to 40 hours which basically means that 50% of the population work for 35 to 40 hours and less and the other 50% of the population work for more than 35 to 40 hours.
- Some occupations have outliers with fewer than 10 hours per week which indicates irregular working patterns.

2.5 Analysis of ‘Marital Status’ between the two ‘Sex’

2.5.1 Using ‘Contingency Table’ to analyse the ‘Marital Status’ between the two ‘Sex’

Sex	Female	Male
Marital Status		
Divorced or formerly in a same-sex civil partnership which is now legally dissolved	23318	17395
Married or in a registered same-sex civil partnership	106665	107514
Separated but still legally married or in a same-sex civil partnership	6864	5087
Single (never married or never registered a same-sex civil partnership)	127679	143320
Widowed or surviving partner from a same-sex civil partnership	24646	7252

2.5.2 Using ‘Stacked Bar Chart’ to analyse the ‘Marital Status’ between the two ‘Sex’



Key Observations and Interpretations from the Graph

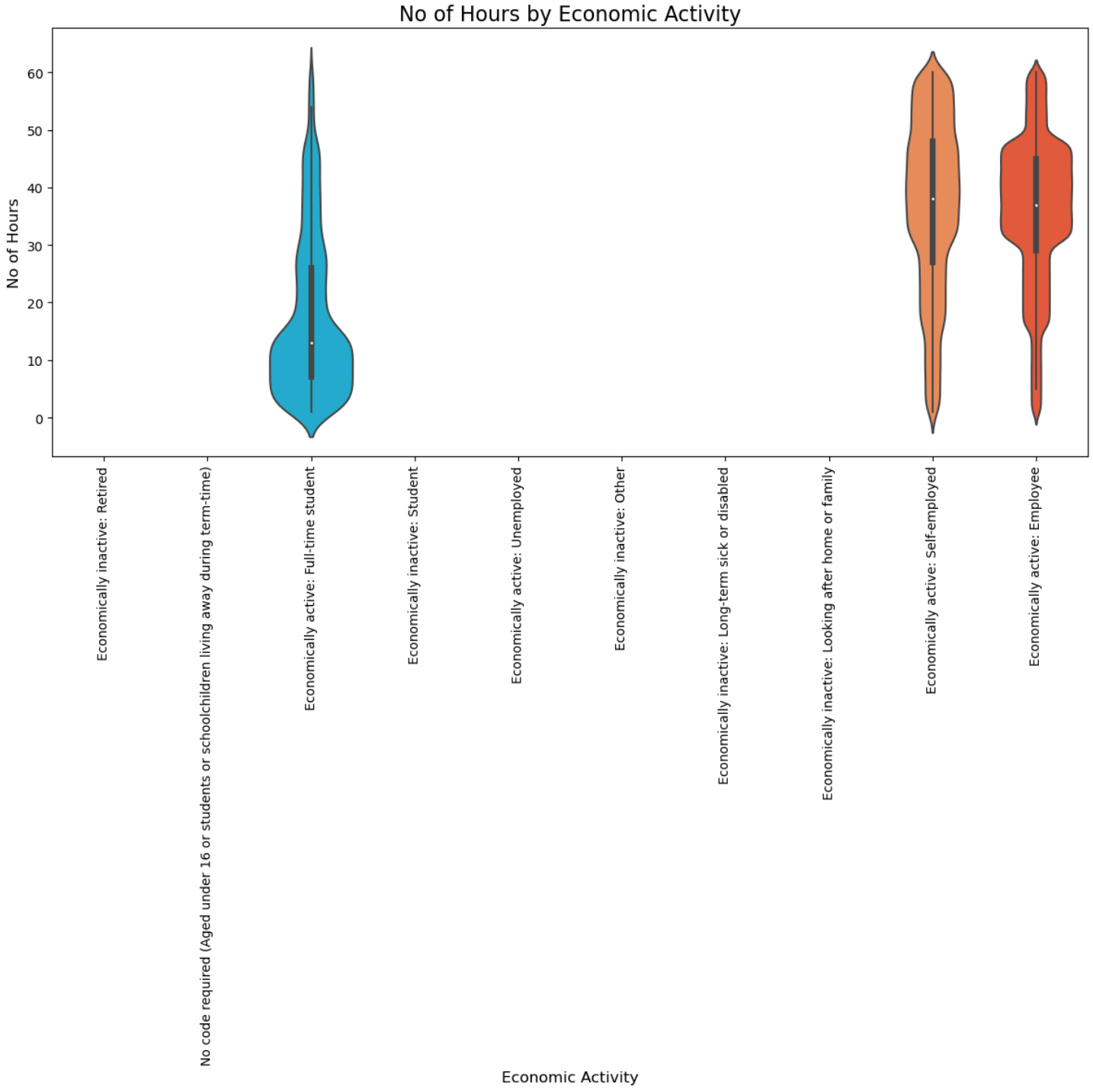
Marital Status	**Analysis (Male vs Female)**
Divorced or formerly in a same-sex civil partnership (legally dissolved)	More females are divorced than men which indicates a higher divorce rate among women. But, the number of this marital status is very less as compared to other marital statuses.
Married or in a registered same-sex civil partnership	The count of females who are married or in a registered same-sex civil partnership is almost the same as that of males and the count of males is slightly more than females.
Separated but still legally married or in a same-sex civil partnership	This status has the least count as compared to other marital statuses and females have higher count than males.
Single (never married or never registered a same-sex civil partnership)	This marital status has the highest count as compared to other marital statuses and males have higher count than females.
Widowed or surviving partner from a same-sex civil partnership	A very high number of females are widowed as compared to males.

Final Interpretation

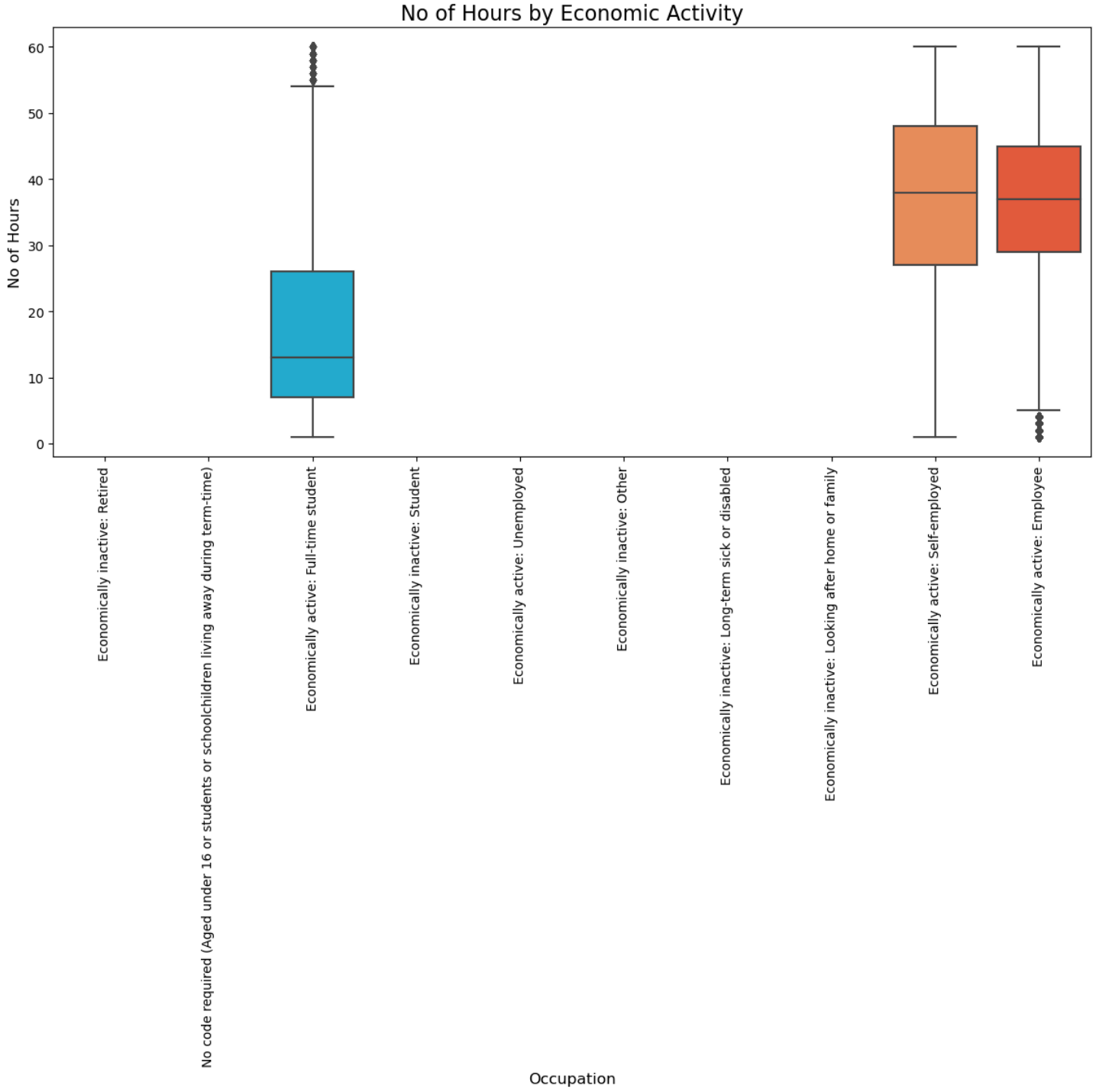
- Both males and females show similar marriage rates but males tend to remain more single as compared to females.
- More number of women are widowed, which might indicate a longer life expectancy of females as compared to males, or it could also mean that women tend to marry males older than their age.

2.6 Distribution of ‘No of hours’ based on ‘Economic Activity’

2.6.1 Using ‘Violinplot’ to show the distribution of ‘No of hours’ based on ‘Economic Activity’



2.6.2 Using ‘Boxplot’ to show the distribution of ‘No of hours’ based on ‘Economic Activity’



Key Observations and Interpretations from the Graphs

Economic Activity	**No of hours**
Economically active: Full-time student	The median working hours for this economic activity is around 10 hours. Most students work between 5 to 25 hours approximately. The maximum working hour is around 55 hours, some students are working more than 55 hours, suggesting outliers and the minimum working hour is near to 0.
Economically active: Self-employed	The median working hours in this economic activity is around 35. Most workers in this category work between 25 to 50 hours approximately.
Economically active: Employee	The median working hours is around 35 hours. Most people in this category work between 28 hours to 45 hours approximately. The minimum working hours is around 5, some people are working less than 5 hours suggesting outliers.

Final Interpretation

- The economic activities with no graph plotted are- Economically inactive: Retired, No code required (People aged under 16, people who have never worked and students or schoolchildren living away during term-time), Economically inactive: Student, Economically active: Unemployed, Economically inactive: Other, Economically inactive: Long-term sick or disabled, and Economically inactive: Looking after home or family. These categories represent the non-working population which does not contribute to the economy.
- The median working hours for the working population is around 35 hours, except for students who have 10 hours as the median working hours.

Task 2: Classification on ‘Approximated Social Grade’

1. Preprocessing for Classification

The following preprocessing steps are undertaken before performing classification-

SL No.	Preprcosessing
1.1. Importing the Dataset	Importing the provided dataset again to perform preprocessing from scratch and is stored as **‘census_num’**
1.2. Deleting the ‘Person ID’ column:	This column is deleted because it is simply a unique identifier that identifies each row in the dataset and has no numerical significance or meaning on its own.
1.3. Converting the datatypes of columns to numerical:	All columns by default are in numeric data type except ‘Region’ and ‘Residence Type’ which are in ‘object’ datatype, so these columns are converted into numerical datatype using LabelEncoder.
1.4. Treating the outlier:	The value ‘-9’ is an outlier which is being replaced by 0 in the entire dataset.
1.5. Treating the missing values	There are 302321 missing values in the ‘No of hours’ column. The following steps are performed- ● Assigning 'No of hours' with 0 for individuals who belongs to the non-working class in the ‘Economic Activity’ column. The codes belonging to the non-working class are- 0(before -9)- No code required (Aged under 16 or students or schoolchildren living away during term-time) 3- Economically active: Unemployed 5- Economically inactive: Retired 6- Economically inactive: Student 7- Economically inactive: Looking after home or family 8- Economically inactive: Long-term sick or disabled 9- Economically inactive: Other

	The identification of the above non-working class is done using the visualisation **2.6.1** and **2.6.2** in **Task 1**. ● Even after assigning 0 to the non-working class in the economic activity column, there still remain 3354 numbers of missing values, which suggests that there might be missing values in the working class of the 'Economic Activity' column as well. These missing values are imputed using the median occupation hours from the 'Occupation' column for each working class having null values. And finally, we have a dataset (census_num) with no missing values.
1.6. Segregating Features and Target Variable:	The features (represented by census_X) include all the columns except 'Approximated Social Grade' which is our target variable (represented by y) for classification.
1.7. Feature Scaling:	census_X which contains all the features are normalised using MinMaxScaler and stored as variable X. The target variable (y) is not scaled.
1.8. Partitioning into Train and Test data:	The scaled features(X) and target variable (y) are split into 75% for training and 25% for testing where **X_train** contains **75%** of the features for training, **X_test** contains **25%** of the features for testing, **y_train** contains **75%** of the target data for training, and **y_test** contains **25%** of the target data for testing.

2. Classification Models

2.1 Classification using "Random Forest Classifier"

Model Training and Evaluation:

- The RandomForestClassifier is first imported from the ensemble module within the sklearn library.
- The RandomForestClassifier is then initialised as RF with random_state=0 for reproducibility and max_depth=20 to avoid further growth of the tree and avoid overfitting.
- The initialised model is trained on X_train (which contains 75% of the feature data for training) and y_train (which contains 75% of the target data for training)
- After training, the result is predicted using X_test (which contains 25% of the feature data for testing)
- The performance of the trained model is analysed using y_test (actual values) and y_pred (predicted values)
- Cross validation is performed to evaluate the average accuracy of the model using 6 folds

Result:

```
Confusion Matrix:
 [[30732     0     0     0     0]
 [    0 15865  3796   259   979]
 [    0  3384 32860   829  2806]
 [    0   362   942 13766  4896]
 [    0   374  1344  2741 26500]]

Mean Absolute Error: 0.2119493102116755
Mean Squared Error: 0.33593568996384315
Root Mean Squared Error: 0.5795995945166311

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     30732
           1       0.79      0.76      0.78     20899
           2       0.84      0.82      0.83     39879
           3       0.78      0.69      0.73     19966
           4       0.75      0.86      0.80     30959

    accuracy                           0.84    142435
   macro avg       0.83      0.83      0.83    142435
weighted avg       0.84      0.84      0.84    142435

Accuracy: 0.8405448099132938
AUC (One-vs-Rest): 0.9738638884968814
```

Accuracy Evaluation of Random Forest Classifier using 6-Fold Cross-Validation

```
Accuracy scores for each fold: [0.73110987 0.68891182 0.66885011 0.76027044 0.79373605 0.65106997]
Accuracy: 0.72 (+/- 0.10)
```

Interpretation of the Confusion Matrix

Actual/Predicted	Predicted Label 0 (No code)	Predicted Label 1 (AB)	Predicted Label 2 (C1)	Predicted Label 3 (C2)	Predicted Label 4 (DE)
Actual Label 0 (No code)	All 30732 instances of label 0 are correctly classified	No instances of label 0 misclassified as label 1	No instances of label 0 misclassified as label 2	No instances of label 0 misclassified as label 3	No instances of label 0 misclassified as label 4
Actual Label 1 (AB)	No instances of label 1 misclassified as label 0	15865 instances of label 1 correctly classified	3796 instances of label 1 misclassified as label 2	259 instances of label 1 misclassified as label 3	979 instances of label 1 misclassified as label 4
Actual Label 2 (C1)	No instances of label 2 misclassified as label 0	3384 instances of label 2 misclassified as label 1	32860 instances of label 2 correctly classified	829 instances of label 2 misclassified as label 3	2806 instances of label 2 misclassified as label 4
Actual Label 3 (C2)	No instances of label 3 misclassified as label 0	362 instances of label 3 misclassified as label 1	942 instances of label 3 misclassified as label 2	13766 instances of label 3 correctly classified	4896 instances of label 3 misclassified as label 4
Actual Label 4 (DE)	No instances of label 4 misclassified as label 0	374 instances of label 4 misclassified as label 1	1344 instances of label 4 misclassified as label 2	2741 instances of label 4 misclassified as label 3	26500 instances of label 4 correctly classified

Error Metrics	
Mean Absolute Error	Predictions made by our model vary by around 0.211 units on average from the actual values.
Mean Squared Error	The value suggests that on average the squared difference between the actual and predicted value is 0.3359. This measure has more impact on the larger errors made.
Root Mean Squared Error	This value suggests that the square root of the mean square error is 0.579. This value gives

	more weight to larger errors making it more sensitive to large prediction mistakes

Analysis of the Classification Report:

Target Labels (Approximated Social Grade)	**Analysis**
0 (No code)	The model perfectly classifies label 0 with 100% precision, recall, and F1 Score. There are 30732 actual instances of label 0 in total.
1 (AB)	A precision of 0.79 suggests that when the model classifies label 1 (AB), it is correct 79% of the time. A recall of 0.76 suggests that the model is able to capture 76% of actual label 1 (AB) instances. An f-1 score of 0.78 suggests the overall balance between precision and recall. There are 20899 actual instances of label 1 in total.
2 (C1)	A precision of 0.84 suggests that when the model classifies label 2 (C1), it is correct 84% of the time. A recall of 0.82 suggests that the model is able to capture 82% of actual label 2(C1) instances. An f-1 score of 0.83 suggests the overall efficiency in classifying label 2 (C1) and a tradeoff between precision and recall. There are a total of 39879 instances of label 2 (C1)
3 (C2)	A precision of 0.78 suggests that when the model classifies label 3 (C2), it correctly classifies 78% of the time. A recall of 0.69 suggests that the model captures 69% of total actual cases of label 3(C2). The F-1 score of 0.73 suggests a lower ability of the model to classify label 3 (C2) as compared to other labels. There are 19966 actual overall cases of label 3(C2)
4 (DE)	A precision of 0.75 suggests that when the trained model classifies label 4 (DE), it correctly classifies 75% of the time. A recall of 0.86 suggests that the model captures 86% of the total actual instances of label 4 (DE). There are a total 30959 actual instances of label 4(DE)

Overall Model Performance:

Accuracy	The accuracy before performing cross-validation was **84.05%**, and after performing cross-validation with 6 folds, the accuracy dropped to **72%** (average accuracy of 6 folds). This figure gives us more reliable information on the overall percentage of correct predictions.
Macro Average	This is the average of precision, recall, and F1-score across all classes, it is calculated without considering the support of each class.

	The macro average of **83 %** suggests that on average, each class performs at a similar level.
Weighted Average	This is the average of precision, recall, and F1-score and it takes into account the support of each class. The weighted average in this case is **84%**, reflecting the overall performance weighted by class size.
AUC (One-vs-Rest)	A score of **0.973** indicates the ability of our model to distinguish each class from the other classes, suggesting a very high predictive accuracy across the different classes

2.2 Classification using "Bagging Classifier"

Model Training and Evaluation:

- The BaggingClassifier is first imported from the ensemble module from the sklearn library.
- The BaggingClassifier is then initialised as BC with DecisionTreeClassifier as the estimator. The maximum depth of each decision tree is set as 20 and the number of decision trees used in the model is 30. The random state is set as 0 for reproducibility.
- The initialised model is trained on X_train (which contains 75% of the feature data for training) and y_train (which contains 75% of the target data for training).
- After training, the result is predicted using X_test (which contains 25% of the feature data for testing).
- The performance of the trained Bagging Classifier model is analysed using y_test (actual values) and y_pred (predicted values).
- Cross validation is performed to evaluate the average accuracy of the model using 6 folds

Result:

```
[[30724     0     4     2     2]
 [    0 15672  3982   367   878]
 [    0  3398 33018  1009  2454]
 [    0   397  1054 13858  4657]
 [    0   410  1617  3254 25678]]

Mean Absolute Error: 0.21701126829782005
Mean Squared Error: 0.3394671253554253
Root Mean Squared Error: 0.5826380740695078

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     30732
           1       0.79      0.75      0.77     20899
           2       0.83      0.83      0.83     39879
           3       0.75      0.69      0.72     19966
           4       0.76      0.83      0.79     30959

    accuracy                           0.84    142435
   macro avg       0.83      0.82      0.82    142435
weighted avg       0.84      0.84      0.83    142435

Accuracy: 0.8351177730192719
AUC (One-vs-Rest): 0.9723555005334588
```

Accuracy Evaluation of Bagging Classifier using 6-Fold Cross-Validation

```
Accuracy scores for each fold: [0.71377571 0.66291058 0.67848605 0.80620702 0.73640423 0.6851805 ]
Mean Accuracy: 0.71 (+/- 0.10)
```

Interpretation of the Confusion Matrix

Actual/Predicted	Predicted Label 0 (No code)	Predicted Label 1 (AB)	Predicted Label 2 (C1)	Predicted Label 3 (C2)	Predicted Label 4 (DE)
Actual Label 0 (No code)	All 30724 instances of label 0 are correctly classified	No instances of label 0 misclassified as label 1	4 instances of label 0 misclassified as label 2	2 instances of label 0 misclassified as label 3	2 instances of label 0 misclassified as label 4
Actual Label 1 (AB)	No instances of label 1 misclassified as label 0	15672 instances of label 1 correctly classified	3982 instances of label 1 misclassified as label 2	367 instances of label 1 misclassified as label 3	878 instances of label 1 misclassified as label 4
Actual Label	No instances	3398 instances	33018	1009 instances	2454 instances

2 (C1)	of label 2 misclassified as label 0	of label 2 misclassified as label 1	instances of label 2 correctly classified	of label 2 misclassified as label 3	of label 2 misclassified as label 4
Actual Label 3 (C2)	No instances of label 3 misclassified as label 0	397 instances of label 3 misclassified as label 1	1054 instances of label 3 misclassified as label 2	13858 instances of label 3 correctly classified	4657 instances of label 3 misclassified as label 4
Actual Label 4 (DE)	No instances of label 4 misclassified as label 0	410 instances of label 4 misclassified as label 1	1617 instances of label 4 misclassified as label 2	3254 instances of label 4 misclassified as label 3	25678 instances of label 4 correctly classified

Error Metrics	
Mean Absolute Error	Predictions made by our model vary by around 0.217 units on average from the actual values.
Mean Squared Error	The metric suggests that on average the squared difference between the actual value and predicted value predicted by our model is 0.339. This measure has more impact on the larger errors made.
Root Mean Squared Error	This value suggests that the square root of the mean square error is 0.582. This value gives more weight to larger errors making it more sensitive to large prediction mistakes.

Analysis of the Classification Report:

Target Labels (Approximated Social Grade)	**Analysis**
0 (No code)	The model perfectly classifies label 0 with 100% precision, recall, and F1 Score. There are 30732 actual instances of label 0 in total.
1 (AB)	A precision of 0.79 suggests that when the model classifies label 1 (AB), it is correct 79% of the time. A recall of 0.75 suggests that the model is able to capture 75% of actual label 1 (AB) instances. An f-1 score of 0.77 suggests the overall balance between precision and recall. There are 20899 actual instances of label 1

	in total.
2 (C1)	A precision of 0.83 suggests that when the model classifies label 2 (C1), it is correct 83% of the time. A recall of 0.83 suggests that the model is able to capture 83% of actual label 2(C1) instances. An f-1 score of 0.83 suggests the overall efficiency in classifying label 2 (C1) and a tradeoff between precision and recall. There are a total of 39879 instances of label 2 (C1)
3 (C2)	A precision of 0.75 suggests that when the model classifies label 3 (C2), it correctly classifies 75% of the time. A recall of 0.69 suggests that the model captures 69% of total actual cases of label 3(C2). The F-1 score of 0.72 suggests a lower ability of the model to classify label 3 (C2) as compared to other labels. There are 19966 actual overall cases of label 3(C2)
4 (DE)	A precision of 0.76 suggests that when the trained model classifies label 4 (DE), it correctly classifies 76% of the time. A recall of 0.83 suggests that the model captures 83% of the total actual instances of label 4 (DE). There are a total 30959 actual instances of label 4(DE)

Overall Model Performance:

Accuracy	The accuracy before performing cross-validation was **83.51%**, and after performing cross-validation with 6 folds, the accuracy dropped to **71%** (average accuracy of 6 folds). This figure gives us more reliable information on the overall percentage of correct predictions.
Macro Average	This is the average of precision, recall, and F1-score across all classes, it is calculated without considering the support of each class. The macro average of **83 %**, for precision and **82 %** for recall and f1 score suggests that on average, each class performs at a similar level.
Weighted Average	This is the average of precision, recall, and F1-score and it takes into account the support of each class. The weighted average in this case is **84 %** for precision and recall and **83%** for f1 score, reflecting the overall performance weighted by class size.
AUC (One-vs-Rest)	A score of **0.972** indicates the ability of our model to distinguish each class from the other classes, suggesting a very high predictive accuracy across the different classes

2.3 Classification using “MLP Classifier”

Model Evaluation and Training

- The MLPClassifier is first imported from the neural_network module from the sklearn library.
- The MLPClassifier is then initialised as MLP by setting the random sate to 0 for reproducibility, maximum iterations (max_iter) is set to 1000, learning rate (learning_rate_init) is set to 0.001, the solver used is 'adam', and the activation function used is 'relu'.
- The initialised model is trained on X_train (which contains 75% of the features for training) and y_train (which contains 75% of the target data for training).
- After training, the result is predicted using X_test (which contains 25% of the features for testing).
- The performance of the trained MLP Classifier model is analysed using y_test (actual values) and y_pred_mlp (predicted values)
- Cross validation is performed to evaluate the average accuracy of the model using 6 folds

Result:

```
Confusion Matrix:
 [[30732     0     0     0     0]
 [    0 15542  4095   237  1025]
 [    0  3682 32287  1191  2719]
 [    0   418   786 13872  4890]
 [    0   404  1207  3461 25887]]

Mean Absolute Error: 0.22153262891845404
Mean Squared Error: 0.3460525853898269
Root Mean Squared Error: 0.5882623440182339

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     30732
           1       0.78      0.74      0.76     20899
           2       0.84      0.81      0.83     39879
           3       0.74      0.69      0.72     19966
           4       0.75      0.84      0.79     30959

    accuracy                           0.83    142435
   macro avg       0.82      0.82      0.82    142435
weighted avg       0.83      0.83      0.83    142435

Accuracy: 0.8306947028469126
AUC: 0.9710952131155282
```

Accuracy Evaluation of MLP Classifier using 6-Fold Cross-Validation

```
Accuracy scores for each fold: [0.72620239 0.68836421 0.66807081 0.80883979 0.82257045 0.61638022]
Mean Accuracy: 0.72 (+/- 0.15)
```

Interpretation of the Confusion Matrix

Actual/Predicted	**Predicted Label 0 (No code)**	**Predicted Label 1 (AB)**	**Predicted Label 2 (C1)**	**Predicted Label 3 (C2)**	**Predicted Label 4 (DE)**
Actual Label 0 (No code)	All 30732 instances of label 0 are correctly classified	No instances of label 0 misclassified as label 1	No instances of label 0 misclassified as label 2	No instances of label 0 misclassified as label 3	No instances of label 0 misclassified as label 4
Actual Label 1 (AB)	No instances of label 1 misclassified as label 0	15542 instances of label 1 correctly classified	4095 instances of label 1 misclassified as label 2	237 instances of label 1 misclassified as label 3	1025 instances of label 1 misclassified as label 4
Actual Label 2 (C1)	No instances of label 2 misclassified as label 0	3682 instances of label 2 misclassified as label 1	32287 instances of label 2 correctly classified	1191 instances of label 2 misclassified as label 3	2719 instances of label 2 misclassified as label 4
Actual Label 3 (C2)	No instances of label 3 misclassified as label 0	418 instances of label 3 misclassified as label 1	786 instances of label 3 misclassified as label 2	13872 instances of label 3 correctly classified	4890 instances of label 3 misclassified as label 4
Actual Label 4 (DE)	No instances of label 4 misclassified as label 0	404 instances of label 4 misclassified as label 1	1207 instances of label 4 misclassified as label 2	3461 instances of label 4 misclassified as label 3	25887 instances of label 4 correctly classified

Error Metrics	
Mean Absolute Error	Predictions made by our model vary by around 0.221 units on average from the actual values.
Mean Squared Error	The metric suggests that on average the squared difference between the actual value and predicted

	value predicted by our model is 0.346. This measure has more impact on the larger errors made.
Root Mean Squared Error	This value suggests that the square root of the mean square error is 0.588. This value gives more weight to larger errors making it more sensitive to large prediction mistakes.

Analysis of the Classification Report:

Target Labels (Approximated Social Grade)	**Analysis**
0 (No code)	The model perfectly classifies label 0 with 100% precision, recall, and F1 Score. There are 30732 actual instances of label 0 in total.
1 (AB)	A precision of 0.78 suggests that when the model classifies label 1 (AB), it is correct 78% of the time. A recall of 0.74 suggests that the model is able to capture 74% of actual label 1 (AB) instances. An f-1 score of 0.76 suggests the overall balance between precision and recall. There are 20899 actual instances of label 1 in total.
2 (C1)	A precision of 0.84 suggests that when the model classifies label 2 (C1), it is correct 84% of the time. A recall of 0.81 suggests that the model is able to capture 81% of actual label 2(C1) instances. An f-1 score of 0.83 suggests the overall efficiency in classifying label 2 (C1) and a tradeoff between precision and recall. There are a total of 39879 instances of label 2 (C1)
3 (C2)	A precision of 0.74 suggests that when the model classifies label 3 (C2), it correctly classifies 74% of the time. A recall of 0.69 suggests that the model captures 69% of total actual cases of label 3(C2). The F-1 score of 0.72 suggests a lower ability of the model to classify label 3 (C2) as compared to other labels. There are 19966 actual overall cases of label 3(C2)
4 (DE)	A precision of 0.75 suggests that when the trained model classifies label 4 (DE), it correctly classifies 75% of the time. A recall of 0.84 suggests that the model captures 84% of the total actual instances of label 4 (DE). There are a total 30959 actual instances of label 4(DE)

Overall Model Performance:

Accuracy	The accuracy before performing cross-validation was **83.06%**, and after performing cross-validation with 6 folds, the accuracy dropped to **72%** (average accuracy of 6 folds). This figure gives us more reliable information on the overall percentage of correct predictions.
Macro Average	This is the average of precision, recall, and F1-score across all classes, it is calculated without considering the support of each class. The macro average of **82 %**, for precision, recall and f1 score suggests that on average, each class performs at a similar level.
Weighted Average	This is the average of precision, recall, and F1-score and it takes into account the support of each class. The weighted average in this case is **83 %** for precision, recall and f1 score, reflecting the overall performance weighted by class size.
AUC (One-vs-Rest)	A score of **0.971** indicates the ability of our model to distinguish each class from the other classes, suggesting a very high predictive accuracy across the different classes

2.4 Classification using "Cat Boost" Classifier

Model Training and Evaluation:

- The CatBoostClassifier is first imported from the catboost module.
- The CatBoostClassifier is initialised as catboost_model and the random seed is set to 0 for reproducibility, iterations (iterations) is set to 1000, learning rate (learning_rate) is set to 0.1, the maximum depth of each tree is set to 8, and logging frequency (verbose) is set to 100.
- The initialised model is trained on X_train (which contains 75% of the features for training) and y_train (which contains 75% of the target data for training).
- After training, the result is predicted using X_test (which contains 25% of the features for testing).
- The performance of the trained CatBoostClassifier model is analysed using y_test (actual values) and y_pred_catboost (predicted values)
- Cross validation is performed to evaluate the average accuracy of the model using 6 folds

Result:

```
Confusion Matrix:
 [[30732     0     0     0     0]
 [    0 15835  3849   323   892]
 [    0  3226 33198   904  2551]
 [    0   364  1002 13896  4704]
 [    0   369  1429  2969 26192]]

Mean Absolute Error: 0.20901463825604663
Mean Squared Error: 0.32766525081616177
Root Mean Squared Error: 0.572420519213071

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     30732
           1       0.80      0.76      0.78     20899
           2       0.84      0.83      0.84     39879
           3       0.77      0.70      0.73     19966
           4       0.76      0.85      0.80     30959

    accuracy                           0.84    142435
   macro avg       0.83      0.83      0.83    142435
weighted avg       0.84      0.84      0.84    142435

Accuracy: 0.8414575069329869
AUC (One-vs-Rest): 0.9746723298338715
```

Accuracy Evaluation of CatBoost Classifier using 6-Fold Cross-Validation

```
Accuracy scores for each fold: [0.71299641 0.65964595 0.65056815 0.8027107  0.73105438 0.63562071]
Mean Accuracy: 0.70 (+/- 0.12)
```

Interpretation of the Confusion Matrix:

Actual/Predicted	**Predicted Label 0 (No code)**	**Predicted Label 1 (AB)**	**Predicted Label 2 (C1)**	**Predicted Label 3 (C2)**	**Predicted Label 4 (DE)**
Actual Label 0 (No code)	All 30732 instances of label 0 are correctly classified	No instances of label 0 misclassified as label 1	No instances of label 0 misclassified as label 2	No instances of label 0 misclassified as label 3	No instances of label 0 misclassified as label 4
Actual Label 1 (AB)	No instances of label 1 misclassified	15835 instances of label 1	3849 instances of label 1 misclassified	323 instances of label 1 misclassified	892 instances of label 1 misclassified

	as label 0	correctly classified	as label 2	as label 3	as label 4
Actual Label 2 (C1)	No instances of label 2 misclassified as label 0	3226 instances of label 2 misclassified as label 1	33198 instances of label 2 correctly classified	904 instances of label 2 misclassified as label 3	2551 instances of label 2 misclassified as label 4
Actual Label 3 (C2)	No instances of label 3 misclassified as label 0	364 instances of label 3 misclassified as label 1	1002 instances of label 3 misclassified as label 2	13896 instances of label 3 correctly classified	4704 instances of label 3 misclassified as label 4
Actual Label 4 (DE)	No instances of label 4 misclassified as label 0	369 instances of label 4 misclassified as label 1	1429 instances of label 4 misclassified as label 2	2969 instances of label 4 misclassified as label 3	26192 instances of label 4 correctly classified

Error Metrics	
Mean Absolute Error	Predictions made by our model vary by around 0.209 units on average from the actual values.
Mean Squared Error	The metric suggests that on average the squared difference between the actual value and predicted value predicted by our model is 0.327. This measure has more impact on the larger errors made
Root Mean Squared Error	This value suggests that the square root of the mean square error is 0.572. This value gives more weight to larger errors making it more sensitive to large prediction mistakes.

Analysis of the Classification Report:

Target Labels (Approximated Social Grade)	Analysis
0 (No code)	The model perfectly classifies label 0 with 100% precision, recall, and F1 Score. There are 30732 actual instances of label 0 in total.
1 (AB)	A precision of 0.80 suggests that when the model classifies label 1 (AB), it is correct 80% of the time. A recall of 0.76 suggests that the model is able to capture 76% of actual label 1 (AB) instances. An f-1 score of 0.78 suggests the overall balance between precision and recall. There are 20899 actual instances of label 1 in total.
2 (C1)	A precision of 0.84 suggests that when the model classifies label 2 (C1), it is correct 84% of the time. A recall of 0.83 suggests that the model is able to capture 83% of actual label 2(C1) instances. An f-1 score of 0.84 suggests the overall efficiency in classifying label 2 (C1) and a tradeoff between precision and recall. There are a total of 39879 instances of label 2 (C1)
3 (C2)	A precision of 0.77 suggests that when the model classifies label 3 (C2), it correctly classifies 77% of the time. A recall of 0.70 suggests that the model captures 70% of total actual cases of label 3(C2). The F-1 score of 0.73 suggests a lower ability of the model to classify label 3 (C2) as compared to other labels. There are 19966 actual overall cases of label 3(C2)
4 (DE)	A precision of 0.76 suggests that when the trained model classifies label 4 (DE), it correctly classifies 76% of the time. A recall of 0.85 suggests that the model captures 85% of the total actual instances of label 4 (DE). There are a total 30959 actual instances of label 4(DE)

Overall Model Performance:

Accuracy	The accuracy before performing cross-validation was **84.14%**, and after performing cross-validation with 6 folds, the accuracy dropped to **70%** (average accuracy of 6 folds). This figure gives us more reliable information on the overall percentage of correct predictions.
Macro Average	This is the average of precision, recall, and F1-score across all classes, it is calculated without considering the support of each class. The macro average of **83 %**, for precision, recall and f1 score suggests that on average, each class performs at a similar level.

Weighted Average	This is the average of precision, recall, and F1-score and it takes into account the support of each class. The weighted average in this case is **84 %** for precision, recall and f1 score, reflecting the overall performance weighted by class size.
AUC (One-vs-Rest)	A score of **0.9746** indicates the ability of our model to distinguish each class from the other classes, suggesting a very high predictive accuracy across the different classes

Comparative Analysis of all the models used above for classification:

Model	**Accuracy (6 fold cross validation)**	**Analysis**
Random Forest	72 %	It has the best accuracy along with MLP Classifier. Performs slightly better than Bagging Classifier because it introduces randomness while selecting features as a result of which it reduces overfitting of the model.
Bagging Classifier	71 %	This model performs slightly less than Random Forest because it doesn't select random features for each tree, making it less capable of capturing complex patterns.
MLP Classifier	72 %	It's performance is similar to Random Forest. It is effectively captures complex patterns in the data.
CatBoost Classifier	70 %	This model has the least accuracy compared to the other models.

Task 3: Regression on "No of hours"

1. Preprocessing for Regression

All the major steps of preprocessing are already performed in Task 2 (Preprocessing for Classification) on the 'census_num' dataset. The only major step that needs to be performed now is to segregate the features and the target column again to perform regression.

SL No.	Preprcosessing
1.1 Segregating Features and Target Variable	The features (represented by census_X) include all the columns except 'No of hours' which is our target variable (represented by y) to perform regression.
1.2 Feature Scaling	census_X which contains all the features are normalised using MinMaxScaler and stored as variable X. The target variable (y) is not scaled.
1.3 Partitioning into Train and Test data	The scaled features(X) and target variable (y) are split into 75% for training and 25% for testing where **X_train** contains **75%** of the features for training, **X_test** contains **25%** of the features for testing, **y_train** contains **75%** of the target data for training, and **y_test** contains **25%** of the target data for testing.

2. Regression Models

2.1 Random Forest Regressor

Model Training and Evaluation:

- The RandomForestRegressor is first imported from the ensemble module from the sklearn library.
- The RandomForestRegressor is then initialised as rf_regressor and set the random state to 0 for reproducibility.
- The intialised model is then trained on X_train (which contains 75% features for training) and y_train (which contains 75% of the target data training).
- After successful training, the result is predicted using X_test (which contains 25% of the features for testing).
- The overall performance of the model is analysed using y_test (actual values) and y_pred (predicted values)
- 6 fold cross-validation is done for better generalisation of the model

Result (Prediction on X_train (75% features) and y_train (25% target data)

```
Mean Absolute Error: 2.038094775355554
Mean Squared Error: 12.514534051472438
Root Mean Squared Error: 3.537588734077555
R-squared Score: 0.9683329268199997
Adjusted R2 score: 0.9683291467920251
```

Result (Prediction on X (features) and y (target variable) using 6-Fold cross-validation)

```
Mean Absolute Error: 2.9307304994831562
Mean Squared Error: 29.602948081168215
Root Mean Squared Error: 5.440859130796185
R-squared Score: 0.9249800430500469
Adjusted R2 score: 0.9249778045209605
```

Analysis of the Result (Before and After Cross-Validation)

Mean Absolute Error	The predictions made by our model vary by around **2.038** units (when trained on X_train and y_train) on average from the actual values. This value increased to **2.930** when performed 6 fold-cross validation on X (features) and y (target variable)
Mean Squared Error	This metric suggests that on average the squared difference between the actual values and the predicted values is **12.514** (when trained on X_train and y_train). This measure has more impact on the large errors made by our model. This value increased to **29.60** when performed 6 fold-cross validation on X (features) and y (target variable)
Root Mean Square Error	This metric suggests that the square root of the mean squared error is **3.537** (when trained on X_train and y_train). This measure gives more weight to the larger errors made by our model making it more sensitive to large prediction mistakes. This value increased to **5.44** when performed 6 fold-cross validation on X (features) and y (target variable)
R-squared Score	This measure suggests that **96.83%** (when trained on X_train and y_train) of the fluctuations

	or variance in the target variable can be explained by our trained model. This high score also suggests that our model has been trained on the data quite well. However, this percentage decreased to **92.49%** when performed 6 fold-cross validation on X (features) and y (target variable)
Adjusted R-squared Score	This measure adjusts the R-squared score based on the number of features in the model and it penalises the score for unwanted features which makes this measure more reliable. An adjusted R2 score of **96.83 %** (when trained on X_train and y_train) suggests that our model is able to explain the fluctuations or variance taking into consideration the features used to train the model. This high score indicates a very good training. However, this percentage decreased to **92.49%** when performed 6 fold-cross validation on X (features) and y (target variable)

2.2 Linear Regression

Model Training and Evaluation:

- The LinearRegression model is first imported from the linear_model module from the sklearn library.
- The LinearRegression model is then intialised as linear_regressor
- The intialised model is then trained on X_train (which contains 75 % of the features for training) and y_train (which contains 75 % of the target data for training).
- After successfully training the model, the result is predicted using X_test (which contains 25 % of the features for testing).
- The overall performance of the model is analysed using y_test (actual values) and y_pred (predicted values)
- 6 fold cross-validation is done for better generalisation of the model

Result (Prediction on X_train (75% features) and y_train (25% target data)

```
Mean Absolute Error: 2.361059320885778
Mean Squared Error: 13.20286206611145
Root Mean Squared Error: 3.633574282454048
R-squared Score: 0.9665911653191908
Adjusted R2 score: 0.9665871773810263
```

Result (Prediction on X (features) and y (target variable) using 6-Fold cross-validation)

```
Mean Absolute Error: 2.524592931806926
Mean Squared Error: 14.79949257141106
Root Mean Squared Error: 3.847010861878487
R-squared Score: 0.9624950429753077
Adjusted R2 score: 0.9624939238606002
```

Analysis of the Result (Before and After Cross-Validation)

Mean Absolute Error	The predictions made by our model vary by around **2.361** units (when trained on X_train and y_train) on average from the actual values. This value increased to **2.524** when performed 6 fold-cross validation on X (features) and y (target variable)
Mean Squared Error	This metric suggests that on average the squared difference between the actual values and the predicted values is **13.202** (when trained on X_train and y_train). This measure has more impact on the large errors made by our model. This value increased to **14.799** when performed 6 fold-cross validation on X (features) and y (target variable)
Root Mean Square Error	This metric suggests that the square root of the mean squared error is **3.633** (when trained on X_train and y_train). This measure gives more weight to the larger errors made by our model making it more sensitive to large prediction mistakes. This value increased to **3.847** when performed 6 fold-cross validation on X (features) and y (target variable)
R-squared Score	This measure suggests that **96.65%** (when trained on X_train and y_train) of the fluctuations or variance in the target variable can be explained by our trained model. This high score also suggests that our model has been trained on the data quite well. However, this percentage decreased to **96.24%** when performed 6 fold-cross validation on X (features) and y (target variable)
Adjusted R-squared Score	This measure adjusts the R-squared score based on the number of features in the model and it

	penalises the score for unwanted features which makes this measure more reliable. An adjusted R2 score of **96.65 %** (when trained on X_train and y_train) suggests that our model is able to explain the fluctuations or variance taking into consideration the features used to train the model. This high score indicates a very good training. However, this percentage decreased to **96.24%** when performed 6 fold-cross validation on X (features) and y (target variable)

2.3 Cat Boost Regressor

Model Training and Evaluation:

- The CatBoostRegressor is first imported from the catboost module.
- The CatBoostRegressor is then initialised as catboost_reg with the following parameters: iterations=500, learning_rate=0.1, depth=6, eval_metric= 'RMSE', random_seed=0 for reproducibility and verbose=100.
- The initialsed model is then trained on X_train (which contains 75% of the features for training) and y_train (which contains 75% of the target data for training)
- X_test (which contains 25 % of the features for testing) and y_test (which contains 25% of the target data for testing) are used as an evaluation set. By setting the use_best_model=True, the model automatically selects the optimal number of iterations.
- After successful training, the result is predicted using X_test (which contains 25% of the features for testing).
- The overall performance of the model is analysed using y_test (actual values) and y_pred (predicted values)
- 6 fold cross-validation is done for better generalisation of the model

Result (Prediction on X_train (75% features) and y_train (25% target data)

```
0:      learn: 17.9341150       test: 17.9537826        best: 17.9537826 (0)    total: 44.7ms   remaining: 22.3s
100:    learn: 3.2480407        test: 3.2717674 best: 3.2717422 (98)    total: 3.86s    remaining: 15.3s
200:    learn: 3.2370625        test: 3.2689121 best: 3.2687578 (192)   total: 7.55s    remaining: 11.2s
300:    learn: 3.2312088        test: 3.2693812 best: 3.2687578 (192)   total: 11.2s    remaining: 7.43s
400:    learn: 3.2258228        test: 3.2698443 best: 3.2687578 (192)   total: 15s      remaining: 3.71s
499:    learn: 3.2213746        test: 3.2705145 best: 3.2687578 (192)   total: 18.8s    remaining: 0us

bestTest = 3.268757814
bestIteration = 192

Shrink model to first 193 iterations.
Mean Absolute Error: 1.9424086059868026
Mean Squared Error: 10.68477764388738
Root Mean Squared Error: 3.268757813587201
R-squared Score: 0.972962985743668
Adjusted R2 score: 0.9729597583955119
```

Result (Prediction on X (features) and y (target variable) using 6-Fold cross-validation)

```
Mean Absolute Error: 2.898545981891888
Mean Squared Error: 27.719668413967895
Root Mean Squared Error: 5.264947142561632
R-squared Score: 0.9297526609383296
Adjusted R2 score: 0.929750564819935
```

Analysis of the Result (Before and After Cross-Validation)

bestTest	This value **(3.268)** indicates the best value of root mean squared error achieved during the training process.
bestIteration	This value **(192)** refers to the iteration number in which the best root mean squared error **(3.268)** is obtained.
Shrink model to first 193 iterations	The model is iterated over the first **193** iterations to retain the best root mean squared error at iteration number **192.**
Mean Absolute Error	The predictions made by our model vary by around **1.942** units (when trained on X_train and y_train) on average from the actual values. This value increased to **2.898** when performed 6 fold-cross validation on X (features) and y (target variable)
Mean Squared Error	This metric suggests that on average the squared difference between the actual values and the predicted values is **10.684** (when trained on X_train and y_train). This measure has more impact on the large errors made by our model. This value increased to **27.719** when performed 6 fold-cross validation on X (features) and y (target variable)
Root Mean Square Error	This metric suggests that the square root of the mean squared error is **3.268** (when trained on X_train and y_train). This measure gives more weight to the larger errors made by our model making it more sensitive to large prediction mistakes. This value increased to **5.264** when performed 6 fold-cross validation on X (features) and y (target variable)
R-squared Score	This measure suggests that **97.29 %** (when trained on X_train and y_train) of the fluctuations

	or variance in the target variable can be explained by our trained model. This high score also suggests that our model has been trained on the data quite well. However, this percentage decreased to **92.97 %** when performed 6 fold-cross validation on X (features) and y (target variable)
Adjusted R-squared Score	This measure adjusts the R-squared score based on the number of features in the model and it penalises the score for unwanted features which makes this measure more reliable. An adjusted R2 score of **97.29 %** (when trained on X_train and y_train) suggests that our model is able to explain the fluctuations or variance taking into consideration the features used to train the model. This high score indicates a very good training. However, this percentage decreased to **92.97%** when performed 6 fold-cross validation on X (features) and y (target variable)

Comparative Analysis of all the algorithms used above for Regression:

Model	**R-squared score(from 6 fold cross-validation)**	**Analysis**
Random Forest Regressor	0.924	This model underperforms and the possible reason might be the lack of complex non-linear pattern in the data.
Linear Regression	0.962	This model performs the best which suggests that the features and the target variable are more linearly related and the linear regression model is able to fit the data quite well
CatBoost Regressor	0.929	This model performs moderately well and the score indicates that the model is able to capture the non-linear patterns in the data but the model doesn't perform better than the simple linear regression model

Task 4: Association Rule Mining

1. Preprocessing for Association Rule Mining

SL No.	Preprcosessing
1.1	The 'census_num' dataset (preprocessed dataset used in classification and regression) is used and created a copy of it which is stored as 'census_cat'.
1.2	Mapped the actual names of the values in the 'census_num' dataset for easy interpretation and understanding while retrieving rules after performing Association Rule Mining.
1.3	'census_cat_sample' contains 1 % of the entire dataset and is used to perform Association Rule Mining.

2. Association Rule Mining Models

2.1 Using 'Apriori' algorithm to perform Association Rule Mining

- The apriori function is first imported from the apyori module
- The data is converted into a list of transactions which contains values from all the 18 columns of the 'census_cat_sample' dataset.
- Apriori algorithm is applied to generate rules from the created list. The minimum support and minimum confidence chosen is 25 % and the minimum number of items to include in an itemset is chosen as 2.
- The results after performing association rule mining are stored in a dataframe for easy view of the rules generated. A filter is applied to show only the rules where the lift is greater than 1.
- A particular index is chosen to review a particular rule.

Rule 1 (Association Rule at index 4129)

```
print(filter_rules.loc[[4129]])
```

```
                                                                                Items  \
4129  {UK, White, No, Married/same-sex civil partnership couple family, Usual resident}

                                                                            Antecedent  \
4129  {White, No, Married/same-sex civil partnership couple family, Usual resident}

     Consequent   Support  Confidence      Lift
4129       {UK}  0.341583    0.934229  1.094449
```

Rule 2 (Association Rule at index 109)

```
print(filter_rules.loc[[109]])
```

```
Items  \
109  {Married or in a registered same-sex civil partnership, UK, No, White, Not resident in a communal establishment, Married/s
ame-sex civil partnership couple family, Usual resident}

                                                                                Antecedent
\
109  {Married or in a registered same-sex civil partnership, Not resident in a communal establishment, White, Usual resident}

                                          Consequent   Support  \
109  {UK, No, Married/same-sex civil partnership couple family}  0.292084

     Confidence      Lift
109    0.896069  2.495066
```

Rule 3 (Association Rule at index 4123)

```
print(filter_rules.loc[[4123]])
```

```
                                                                    Items  \
4123  {White, Not resident in a communal establishment, UK, Good health}

                                                                Antecedent  \
4123  {White, Not resident in a communal establishment, Good health}

     Consequent   Support  Confidence      Lift
4123       {UK}  0.266807    0.934235  1.094455
```

Rule 4 (Association Rule at index 100)

```
print(filter_rules.loc[[100]])
```

```
                                                                                Items
\
100  {Married or in a registered same-sex civil partnership, UK, White, No, Married/same-sex civil partnership couple family}

                                            Antecedent  \
100  {White, No, Married/same-sex civil partnership couple family}

                                          Consequent   Support  \
100  {Married or in a registered same-sex civil partnership, UK}  0.292084

     Confidence      Lift
100    0.798848  2.571207
```

Rule 5 (Association Rule at index 1478)

```
print(filter_rules.loc[[1478]])
```

```
Items  \
1478  {No code required (People aged under 16, people not working, and students or schoolchildren living away during term-tim
e), UK, (-0.001, 24.0], Single (never married or never registered a same-sex civil partnership)}


Antecedent  \
1478  {No code required (People aged under 16, people not working, and students or schoolchildren living away during term-tim
e), UK, Single (never married or never registered a same-sex civil partnership)}

           Consequent  Support  Confidence      Lift
1478  {(-0.001, 24.0]}  0.26435    0.998674  1.571236
```

Interpretation of the above Rules	
Rule No.	**Interpretation**
Rule 1 (Association Rule at index 4129)	If a person's ethnic group is white, not a student, belongs to a married/ same-sex civil partnership couple family, and is a usual resident then there is a 93.4229 % chance that the country of birth of the person is the UK.
Rule 2 (Association Rule at index 109)	If a person is married or in a registered same-sex civil partnership, doesn't reside in a communal establishment, has white as the ethnic group, and is a usual resident then there is 89.60 % chance that the person's country of birth is the UK, is not a student and belongs to a married/ same-sex civil partnership couple family.
Rule 3 (Association Rule at index 4123)	If a person's ethnic group is white, doesn't reside in a communal establishment and has a good health then there is 93.4235 % chance that the country of birth of the person is the UK.
Rule 4 (Association Rule at index 100)	If a person's ethnic group is white, is not a student, belongs to married/ same-sex civil partnership couple family then there is 79.88 % chance that the person's country of birth is the UK and the marital status is married or in a registered same-sex civil partnership.
Rule 5 (Association Rule at index 1478)	If a person belongs to the no code required category (People aged under 16, people not working, and students or schoolchildren living away during term-time), country of birth is the UK and marital status is Single (never married or never registered a same-sex civil partnership) then there is 99.86 % chance that the person falls in the age group of 0 to 24 years.

2.2 Using "Frequent-Pattern Growth (FP-Growth)" algorithm to perform Association Rule Mining

- The fpgrowth and association_rules functions are first imported from the mlxtend.frequent_patterns module.
- The census_cat_sample dataset is converted into a one-hot encoded data frame suitable for performing FP-Growth and stored as data_encoded.
- The FP-Growth algorithm is applied to retrieve frequent itemsets from the 'data_encoded' data frame with a minimum support of 25% to filter the itemsets.
- A minimum confidence of 80 % is applied to generate rules from the frequent itemsets.
- The results after performing association rule mining are stored in a dataframe for easy view of the rules generated. A filter is applied to show only the rules where the lift is greater than 1.
- A particular index is chosen to review a particular rule.

Rule 1 (Association Rule at index 3101)

```
specific_rule = result_df.loc[[3101]]
print(specific_rule)
```

```
      Items                                                 Antecedent  \
3101   True  (Sex_Male, Country of Birth_UK, Population Base_Usual resident)

                                                 Consequent  Support  \
3101  (Residence Type_Not resident in a communal establishment)  0.42654

      Confidence      Lift
3101    0.981818  1.000075
```

Rule 2 (Association Rule at index 3097)

```
specific_rule = result_df.loc[[3097]]
print(specific_rule)
```

```
      Items            Antecedent  \
3097   True  (Ethnic Group_White)

                                                 Consequent   Support  \
3097  (Residence Type_Not resident in a communal establishment)  0.831841

      Confidence      Lift
3097    0.982176  1.000439
```

Rule 3 (Association Rule at index 214)

```
specific_rule = result_df.loc[[214]]
print(specific_rule)
```

```
     Items  \
214   True

                                                                                Antecedent  \
214  (Country of Birth_UK, Student_No, Marital Status_Married or in a registered same-sex civil partnership)


Consequent  \
214  (Ethnic Group_White, Family Composition_Married/same-sex civil partnership couple family, Population Base_Usual resident,
Residence Type_Not resident in a communal establishment)

      Support  Confidence      Lift
214  0.292084    0.944381  2.088564
```

Rule 4 (Association Rule at index 829)

```
specific_rule = result_df.loc[[829]]
print(specific_rule)
```

```
     Items  \
829   True


Antecedent  \
829  (Country of Birth_UK, Economic Activity_Economically active: Employee, Population Base_Usual resident, Residence Type_Not
resident in a communal establishment)

                        Consequent   Support  Confidence      Lift
829  (Ethnic Group_White, Student_No)  0.304722    0.946565  1.388766
```

Rule 5 (Association Rule at index 1479)

```
specific_rule = result_df.loc[[1479]]
print(specific_rule)
```

```
      Items  \
1479   True

                                                                    Antecedent  \
1479  (Residence Type_Not resident in a communal establishment, Religion_Christian)

                                              Consequent   Support  \
1479  (Ethnic Group_White, Population Base_Usual resident)  0.519747

      Confidence     Lift
1479    0.923869  1.09242
```

Rule 6 (Association Rule at index 2158)

```
specific_rule = result_df.loc[[2158]]
print(specific_rule)
```

```
      Items                                                  Antecedent  \
2158   True  (Student_No, Economic Activity_Economically active: Employee)

                                                                Consequent  \
2158  (Ethnic Group_White, Country of Birth_UK, Population Base_Usual resident)

       Support  Confidence      Lift
2158  0.305424    0.811946  1.031361
```

Interpretation of the above Rules	
Rule No.	**Interpretation**
Rule 1 (Association Rule at index 3101)	If a person's country of birth is the UK, the gender is male and a usual resident then there is 98.18% chance that the person doesn't reside in a communal establishment
Rule 2 (Association Rule at index 3097)	If a person's ethnic group is white then there is 98.21% chance that the person doesn't reside in a communal establishment.
Rule 3 (Association Rule at index 214)	If a person's country of birth is the UK, is not a student and the marital status is married or in a registered same-sex civil partnership then there is 94.43% chance that the ethnic group of the person is white, belongs to married/ same-sex civil partnership couple family, a usual resident and the person doesn't reside in a communal establishment.
Rule 4 (Association Rule at index 829)	If a person's country of birth is the UK, doesn't reside in a communal establishment and an economically active employee then there is 94.65% chance that the person's ethnic group is white and not a student
Rule 5 (Association Rule at index 1479)	If a person's religion is christianity and doesn't reside in a communal establishment then there is 92.38% chance that the ethnic group of the person is white and is a usual resident.
Rule 6 (Association Rule at index 2158)	If a person is not a student and an economically active employee then there is 81.19% chance that the country of birth of the person is the UK , the person's ethnic group is white and the person is a usual resident.

Task 5: Clustering

1. Preprocessing for Clustering

SL No.	Preprcosessing
1.1	The 'census_num' dataset is used (preprocessed dataset used in classification and regression).
1.2	The data within 'census_num' is standardised using StandardScaler.
1.3	1 % of the standardised data is taken as a sample and stored as 'sampled_data' which will be used for clustering.

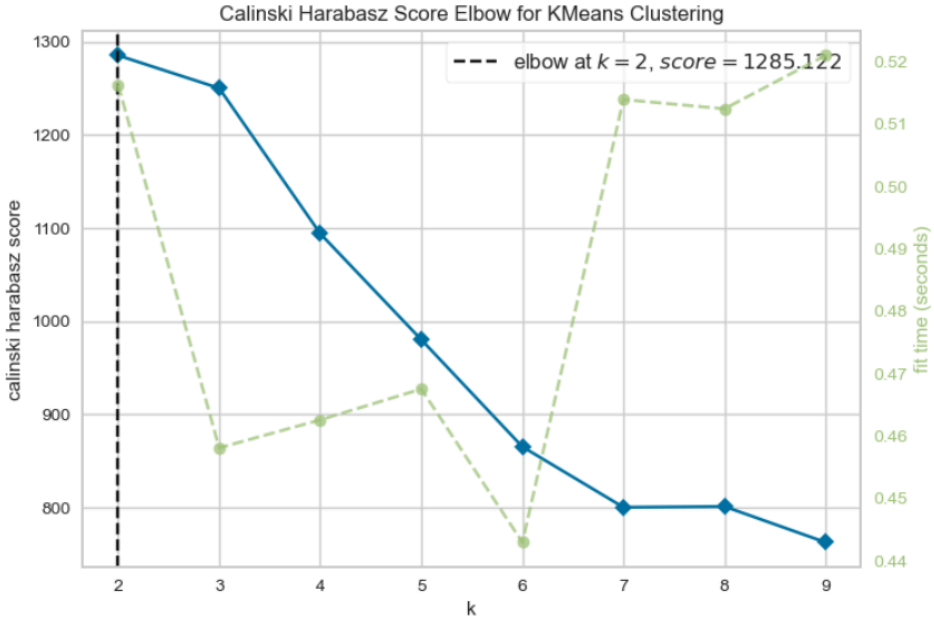
2. Clustering Models

2.1 Using 'k-means' algorithm to perform clustering

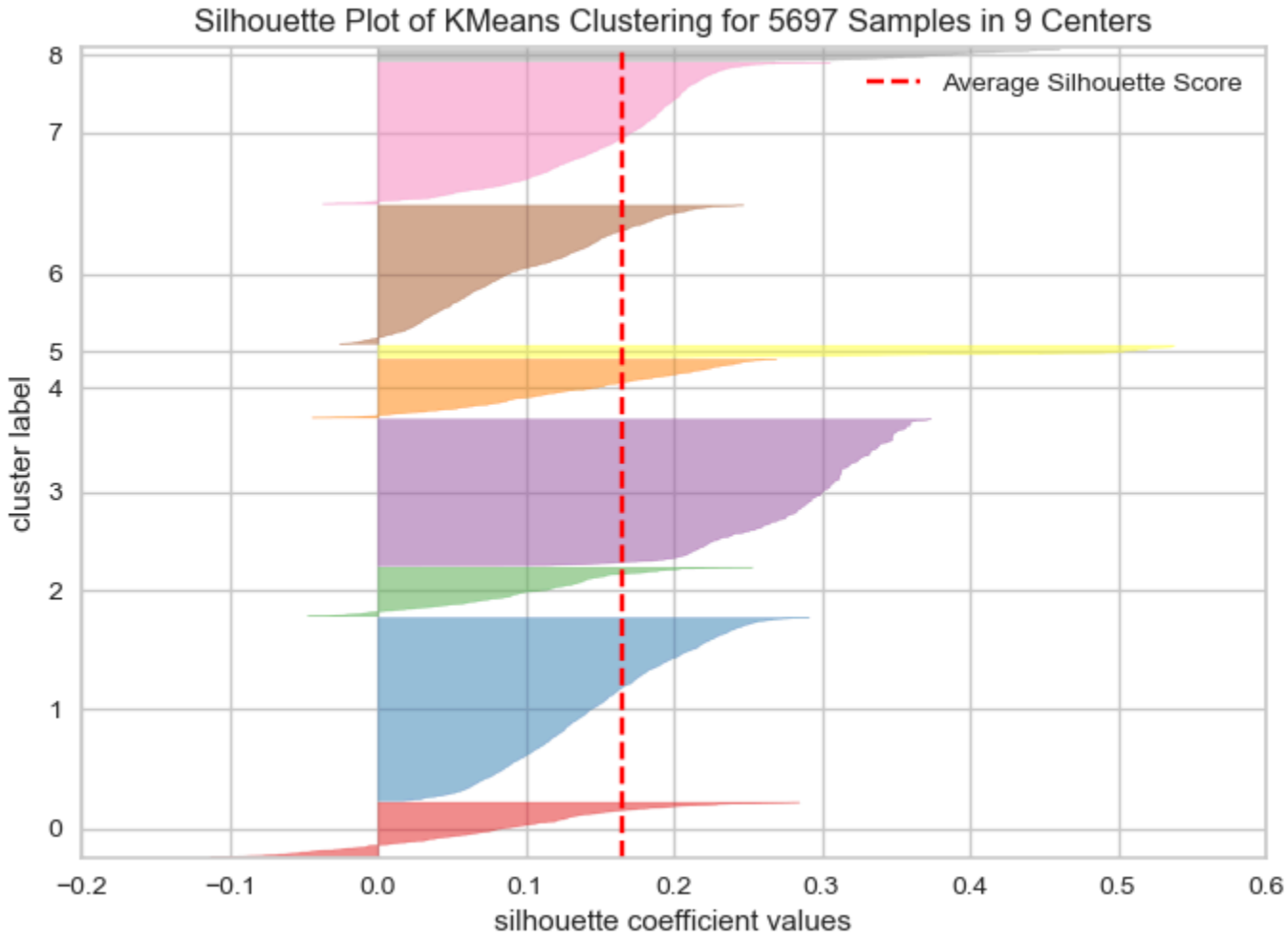
- The KMeans clustering algorithm is first initialised with two clusters.
- The model is trained on the 'sampled_data'
- After modelling, the cluster centers are viewed.
- The silhouette score is computed for the clustering.
- The elbow curve is plotted to determine the optimum number of clusters for better clustering.
- The silhouette visualiser is plotted to graphically analyse the performance of clustering.
- A new column named 'cluster' is appended to the 'sampled_data' to store the cluster labels assigned to each data point by the KMeans algorithm.
- Properties of both the clusters are viewed.
- For both the cluster, mean values of all attributes are computed from the cluster centers
- A bar chart is plotted to analyse the mean values for both the clusters, first for the first 10 columns and then for the last 8 columns.

Results:

Elbow Curve

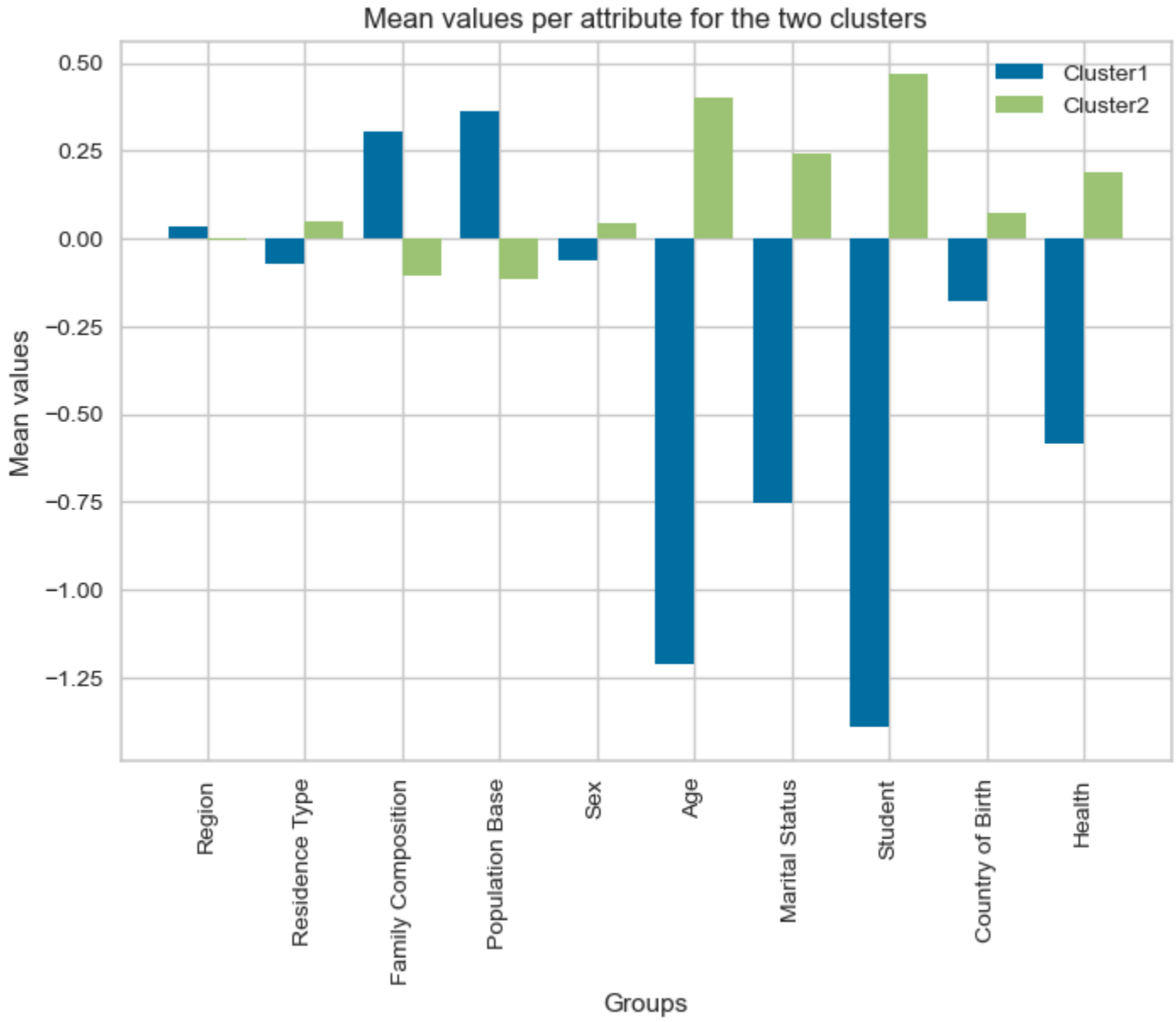


Silhouette Plot (for 9 clusters)

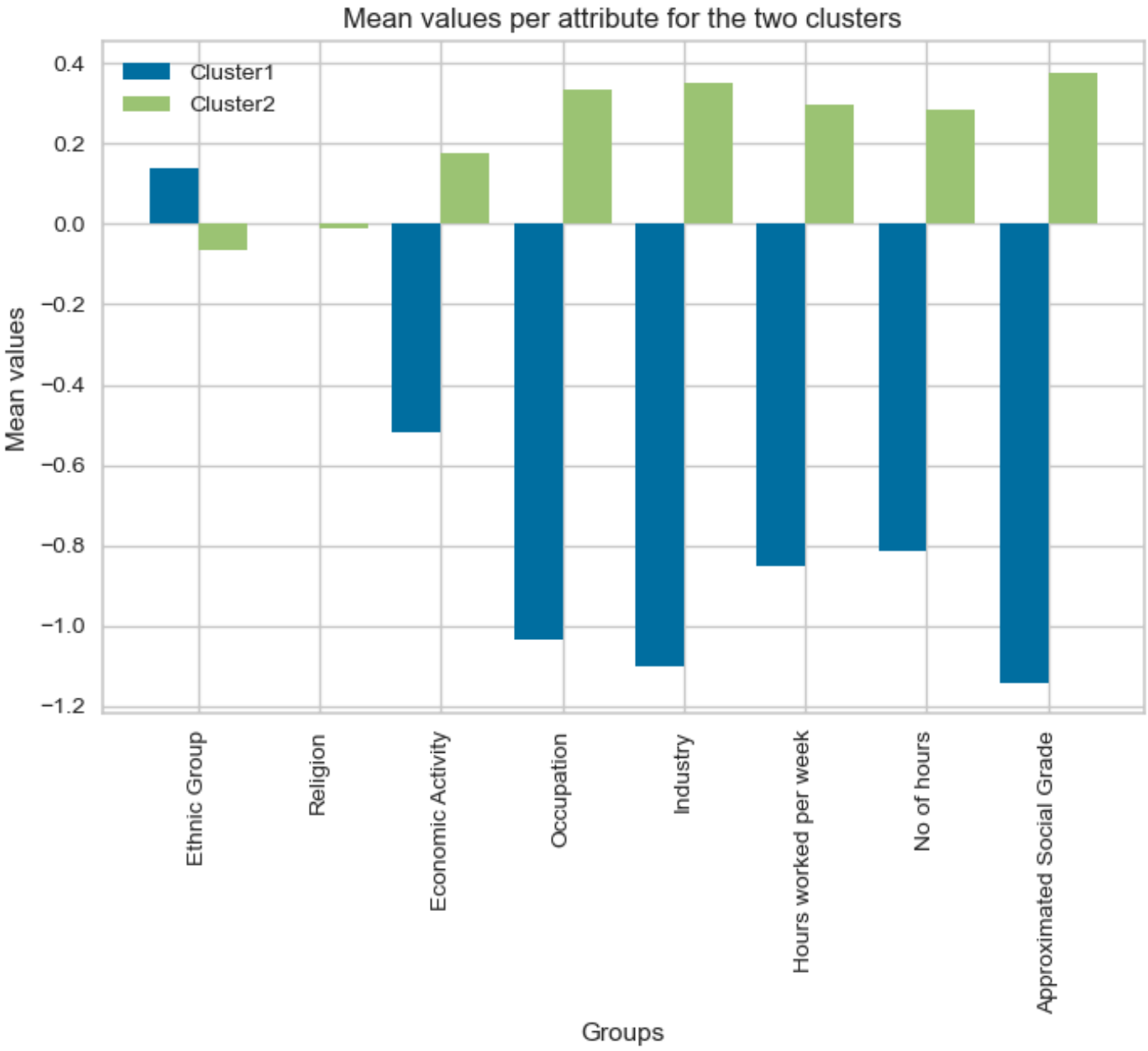


Silhouette Plot (for 2 clusters)

Visualisation of mean values per attribute of the two clusters for the first 10 columns:



Visualisation of mean values per attribute of the two clusters for the last 8 columns



Interpretation and Analysis of the above plots:

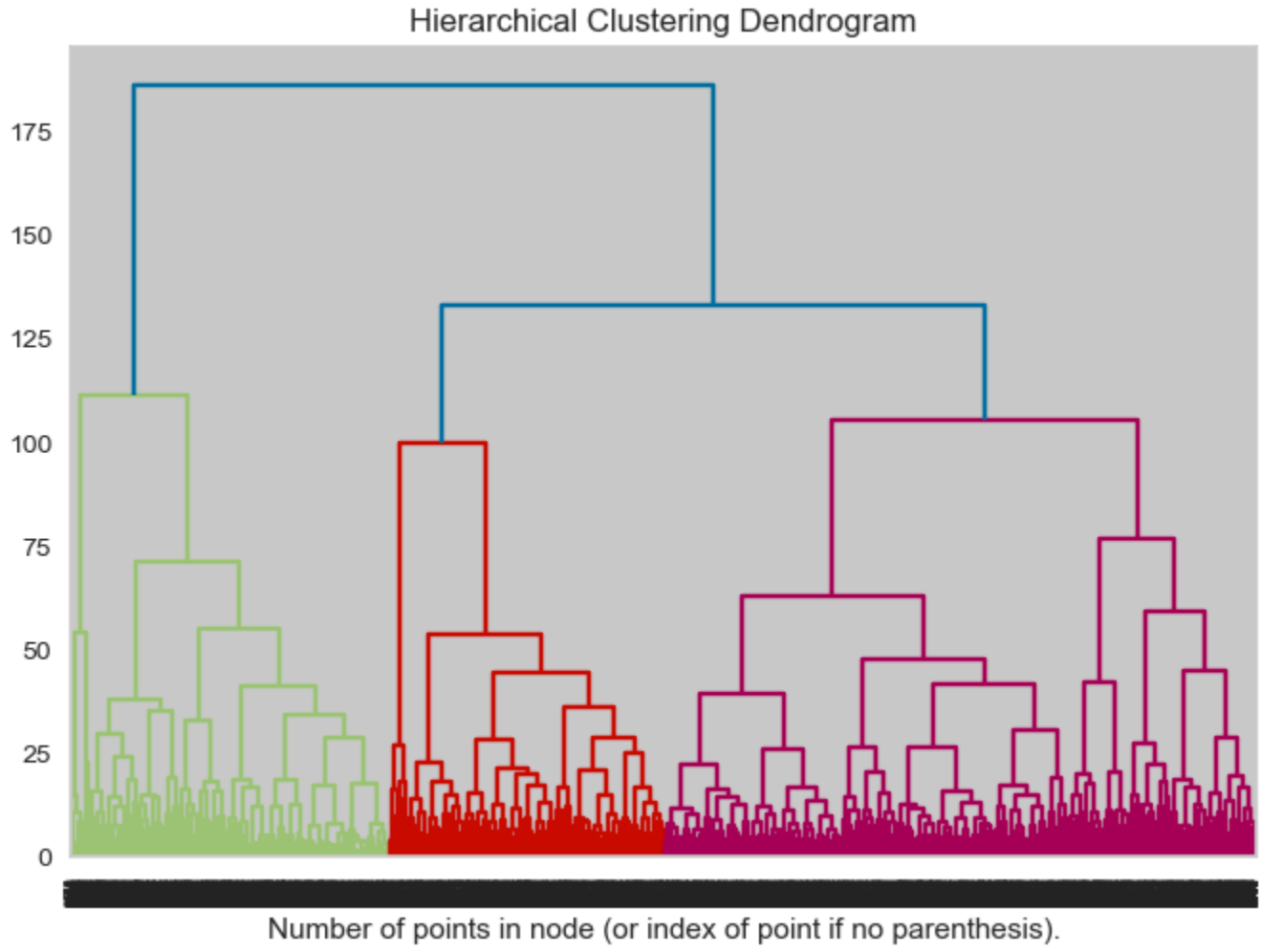
Plot	Interpretation
Elbow Curve	The elbow curve suggest that the optimal number of clusters for better clustering performance is 2 with the highest calinski harabasz score; this point is the elbow point.
Silhouette Plot (for 9 clusters)	This plot visualises the silhouette score for 9 different clusters and gives the average silhouette score.
Silhouette Plot (for 2 clusters)	This plot helps us visualise the silhouette scores for the two clusters; indicating an average silhouette score of 0.24 which shows the moderate clustering quality with a significant room for improvement. A score nearer to 1 is more desirable.
Visualisation of mean values per attribute for the two clusters for the first 10 columns:	We can see that some attributes show a significant difference in mean values between the two clusters which indicates better clustering and more distinct separation between the groups. These include- Family Composition, Population Base, Age, Marital Status, Student and Health. While, features like Country of Birth,Sex, Region and Residence Type show moderate to less difference in mean values between the two clusters which indicates less influence on the clustering process and contribute less to the separation of the two clusters.
Visualisation of mean values per attribute for the two clusters for the last 8 columns:	Features like Economic Activity, Occupation, Industry, Hours worked per week, No of hours and Approximated Social Grade show significant difference in the mean values between the two clusters which indicates better clustering and more distinct separation between the two groups. On the other hand, features like Ethnic Group and Region show moderate to less difference between the mean values which indicates less influence on the clustering process and contribute less to the separation of the two clusters.

2.2 Using 'AgglomerativeClustering' algorithm to perform clustering

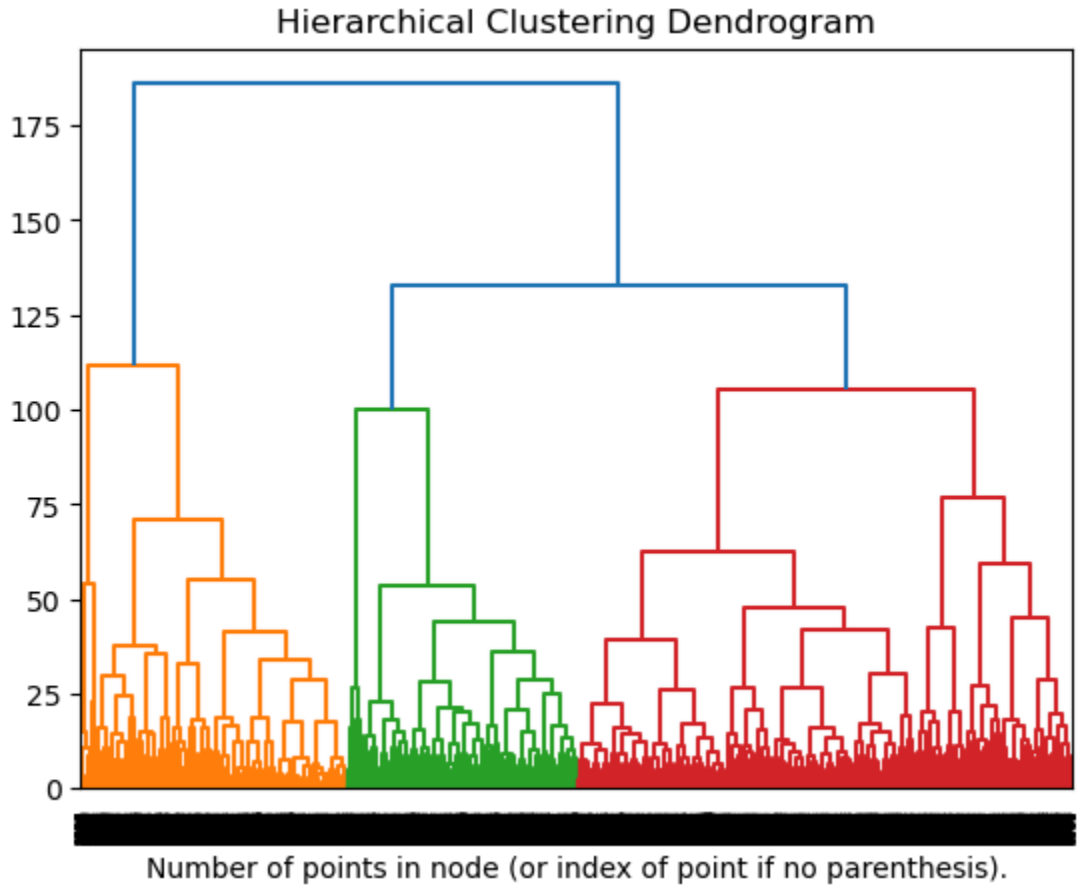
- 1 % of the standardised data is taken as a sample and stored as 'X' which will be used for clustering.
- The AgglomerativeClustering is first initialised with 2 clusters. The Silhouette score for single, average, complete and ward linkage are computed.
- The AgglomerativeClustering is then initialised with 3 clusters. The Silhouette score is again computed for single, average, complete and ward linkage.
- The top 4 levels of the dendogram is visualised for 2 clusters (ward linkage) and 3 clusters (ward linkage)
- A new column 'cluster_no' is appended to the 'X' dataframe to store the cluster labels assigned to each data points by the AgglomerativeClustering algorithm.
- Properties of the 3 clusters are viewed (since 3 clusters are chosen)
- For the three clusters, mean values of all attributes are computed.
- A bar chart is plotted to analyse the mean values for the three clusters, first for the first 10 columns and then for the last 8 columns.

Results:

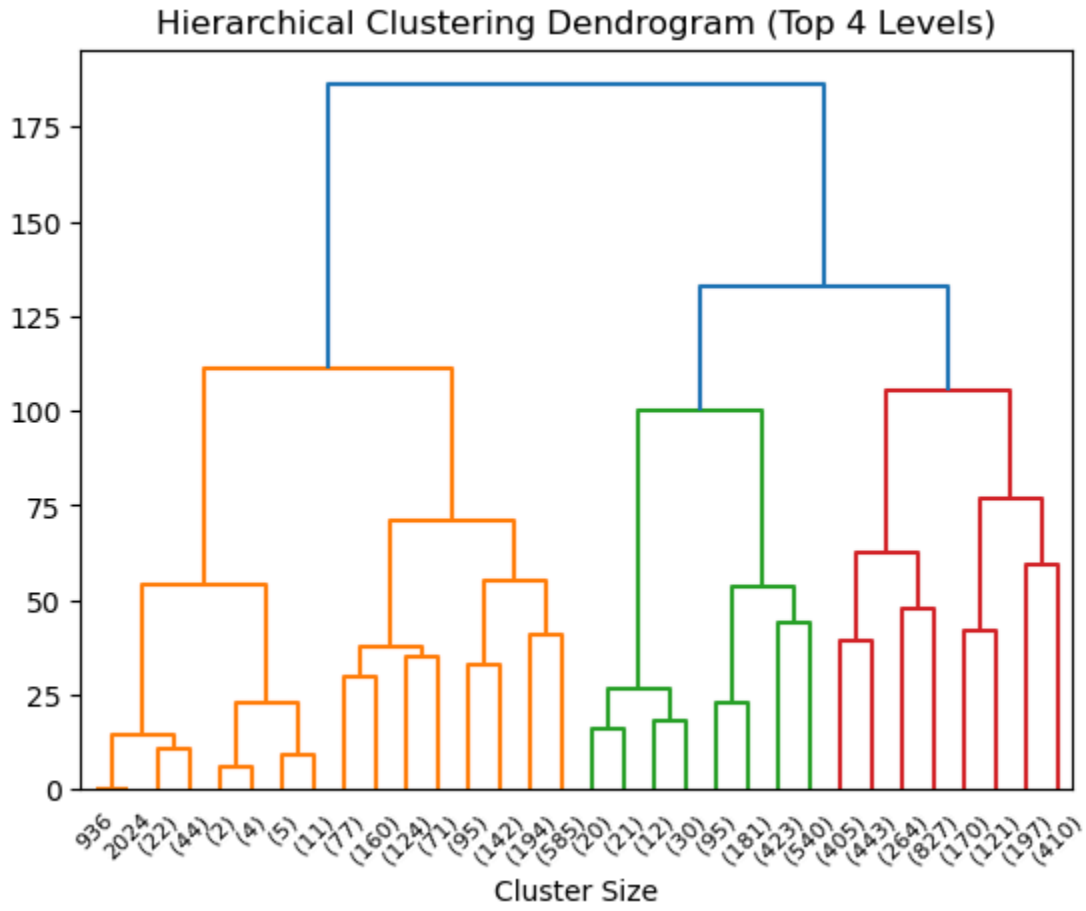
Dendogram to visualise the hierarchical clustering of all the levels for 2 clusters



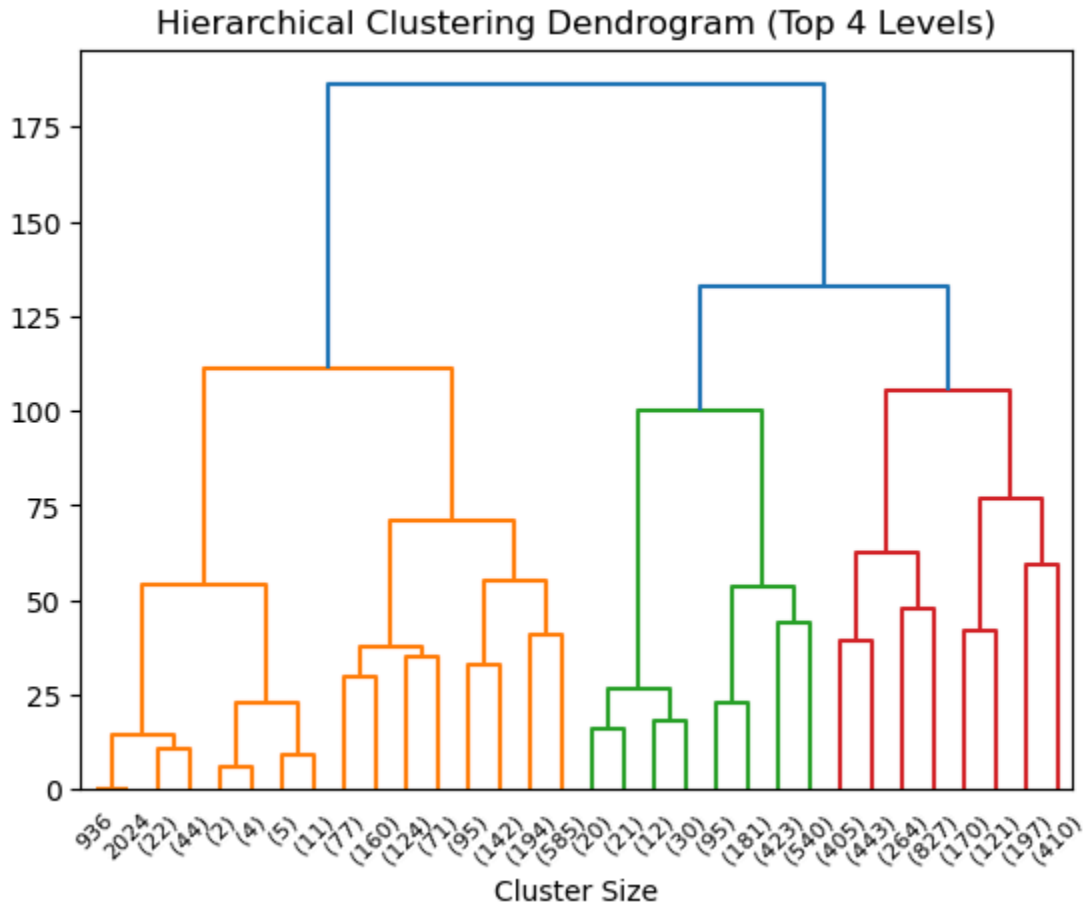
Dendogram to visualise the hierarchical clustering of all the levels for 3 clusters



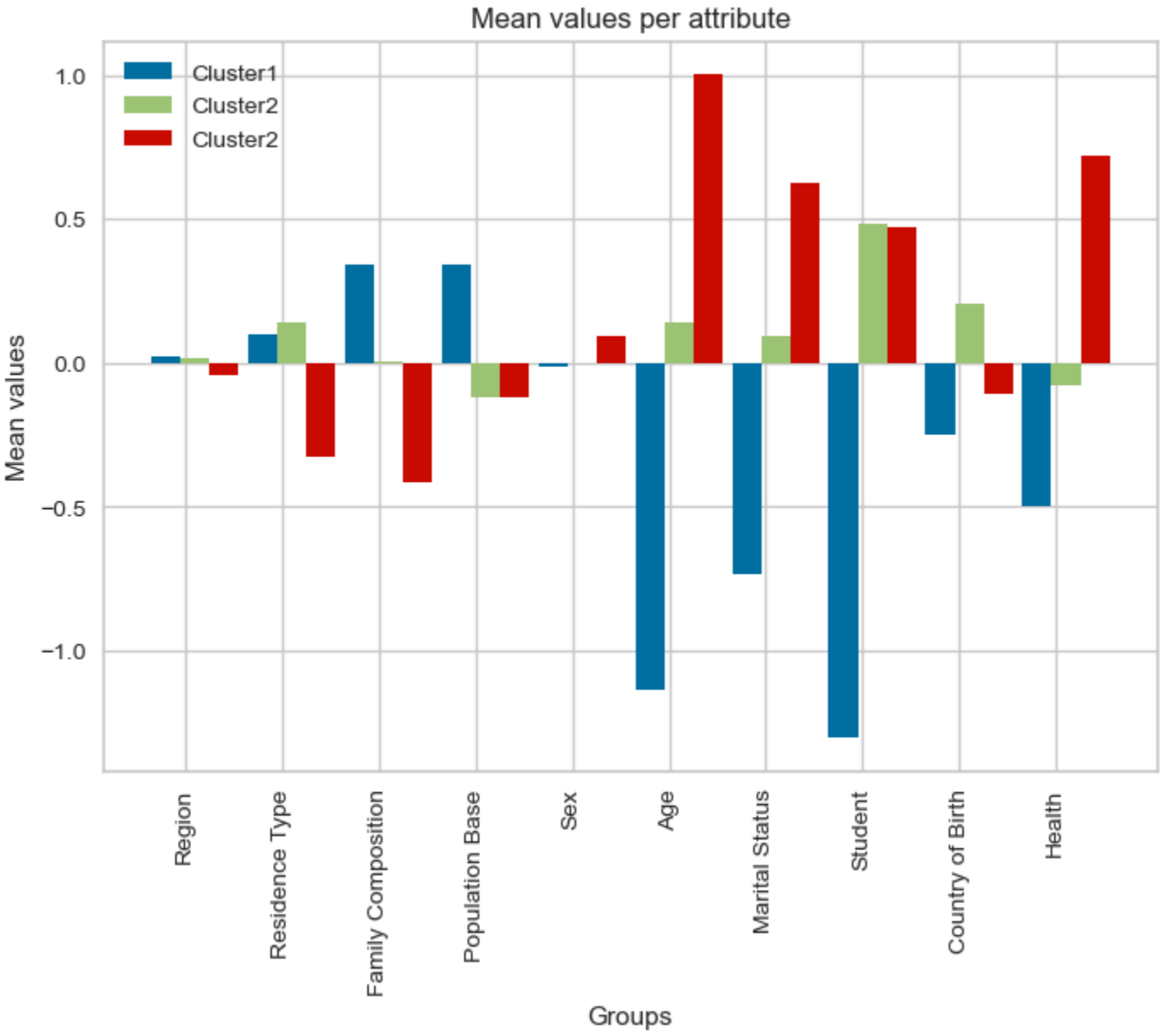
Dendogram to visualise the hierarchical clustering of the top 4 levels for 2 clusters (ward linkage)



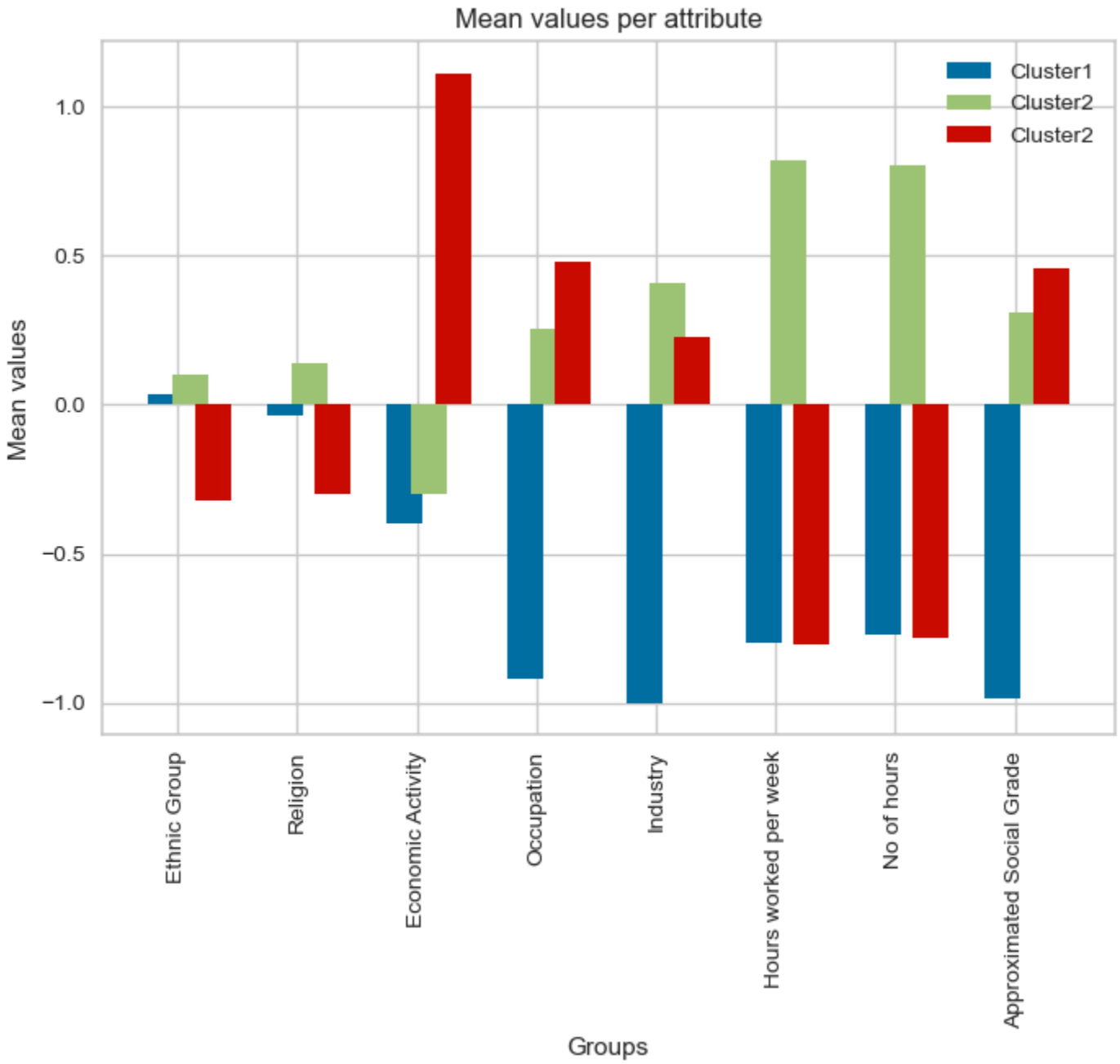
Dendogram to visualise the hierarchical clustering of the top 4 levels for 3 clusters (ward linkage)



Visualisation of mean values per attribute of the three clusters for the first 10 columns:



Visualisation of mean values per attribute of the three clusters for the last 8 columns:



Interpretation and Analysis of the above plots:

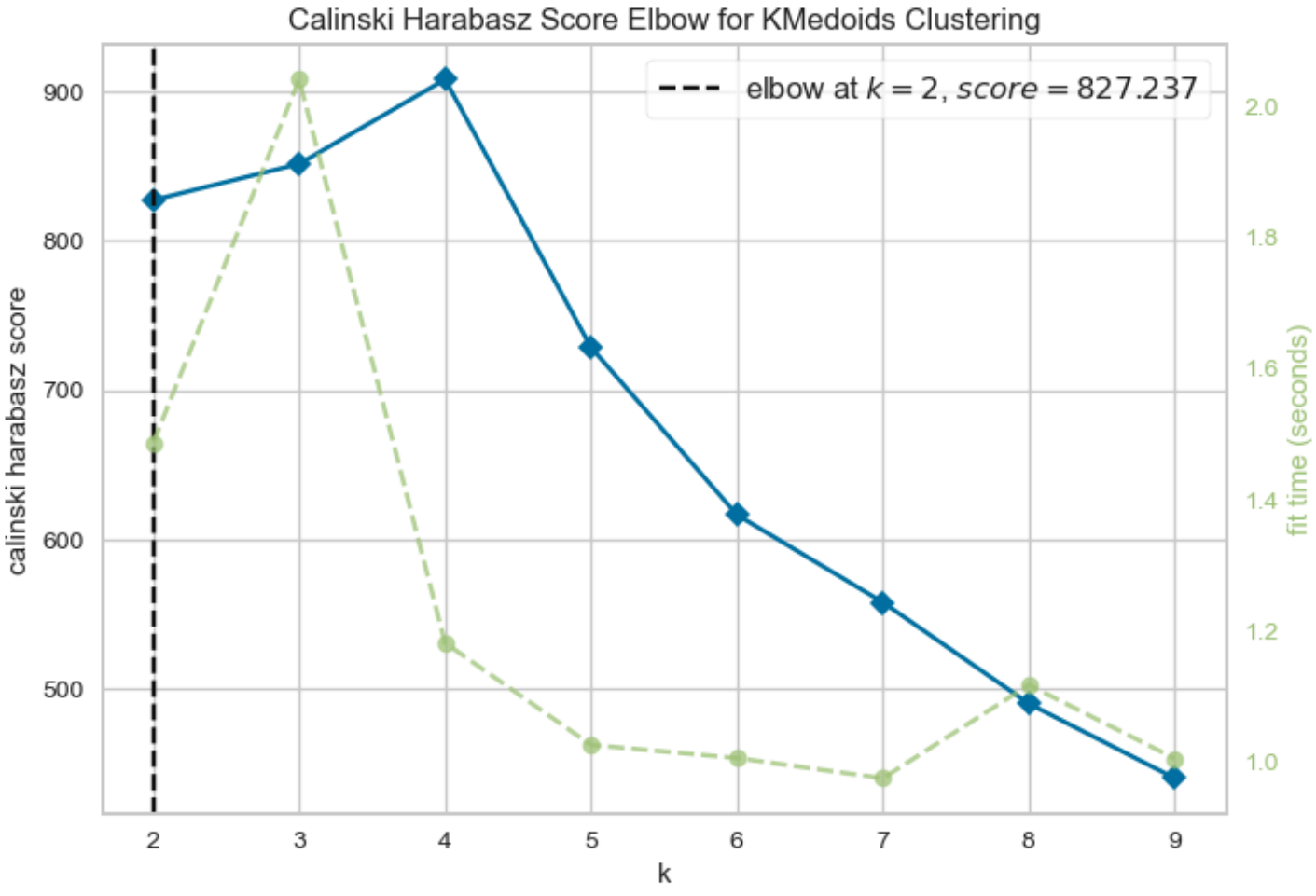
Plot	Interpretation
Dendogram to visualise the hierarchical clustering of all the levels for 2 clusters	This visualisation shows how the data merges hierarchically for the two clusters with ward as the linkage for all the levels
Dendogram to visualise the hierarchical clustering of all the levels for 2 clusters	This visualisation shows how the data merges hierarchically for the three clusters with ward as the linkage for all the levels
Dendogram to visualise the top 4 levels for 2 clusters (ward linkage)	This visualisation shows how the data merges hierarchically for the two clusters with ward as the linkage for the top 4 levels
Dendogram to visualise the top 4 levels for 3 clusters (ward linkage)	This shows how the data is merged hierarchically for the three clusters with ward as the linkage for the top 4 levels
Visualisation of mean values per attribute for the three clusters for the first 10 columns	We can see that some attributes show a significant difference in mean values between the three clusters which indicates better clustering and more distinct separation between the groups. These include- Age, Marital Status, Health and Student. While, other features show moderate to less difference in mean values between the three clusters which indicates less influence on the clustering process and contribute less to the separation of the three clusters.
Visualisation of mean values per attribute for the three clusters for the last 8 columns:	Features like Economic Activity, Occupation, Industry and Approximated Social Grade show significant difference in the mean values between the three clusters which indicates better clustering and more distinct separation between the groups. On the other hand, other features show moderate to less difference between the mean values which indicates less influence on the clustering process and contribute less to the separation of the two clusters.

2.3 Using ‘KMedoids’ algorithm to perform clustering

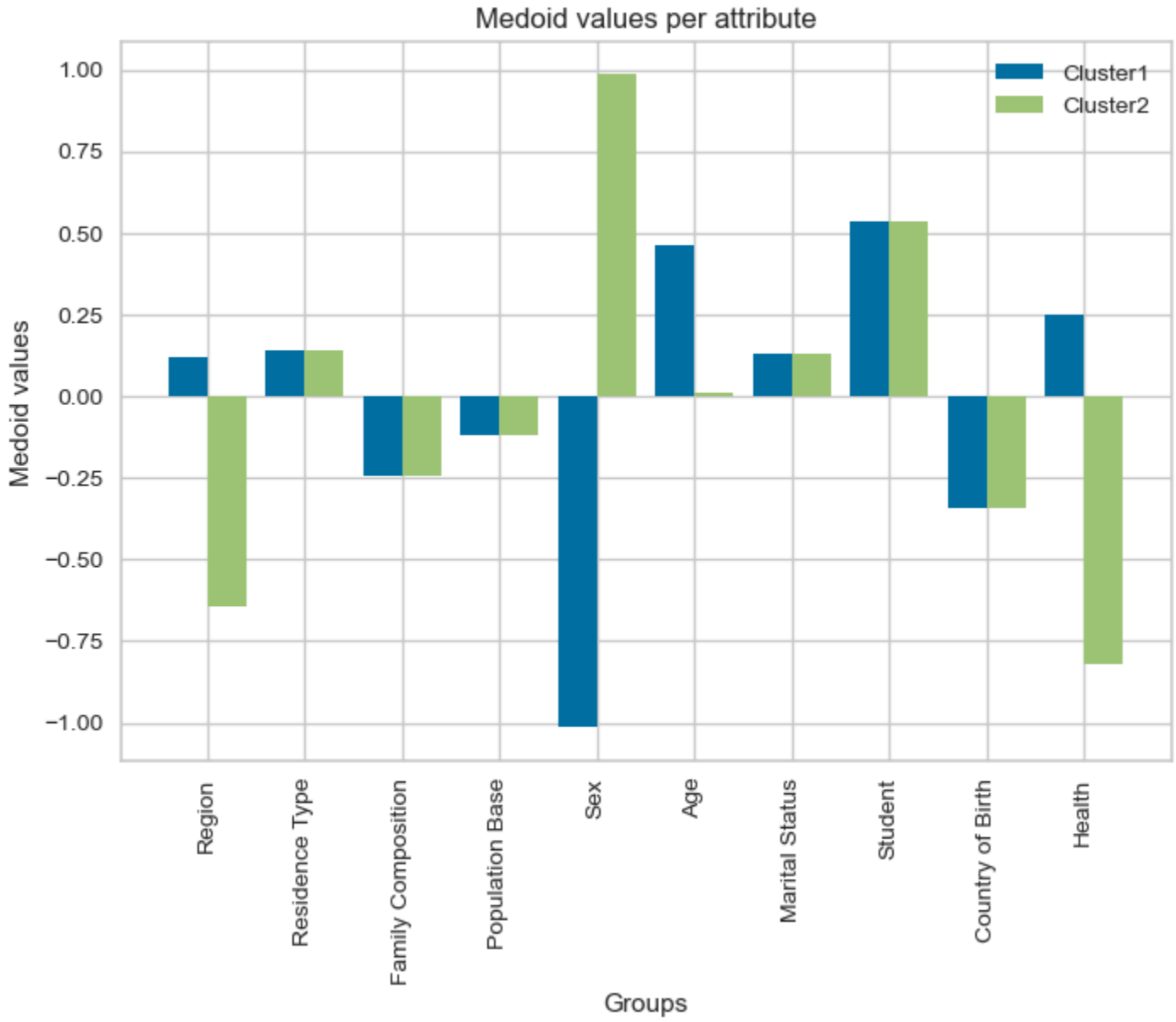
- The KMedoids clustering algorithm is first initialised with two clusters.
- The model is trained on the ‘sampled_data’
- The silhouette score is computed for the clustering (with 2 clusters) and comes out to be 0.14
- The elbow curve is plotted to determine the optimum number of clusters for better clustering.
- The elbow point appeared at k=2. So, the optimum number of clusters chosen is 2.
- A new column named ‘cluster’ is appended to the ‘sampled_data’ to store the cluster labels assigned to each data point by the KMeans algorithm.
- Properties of both the clusters are viewed.
- For both the cluster, medoid values of all attributes are computed from the cluster centers
- A bar chart is plotted to analyse the medoid values for both the clusters, first for the first 10 columns and then for the last 8 columns.

Results:

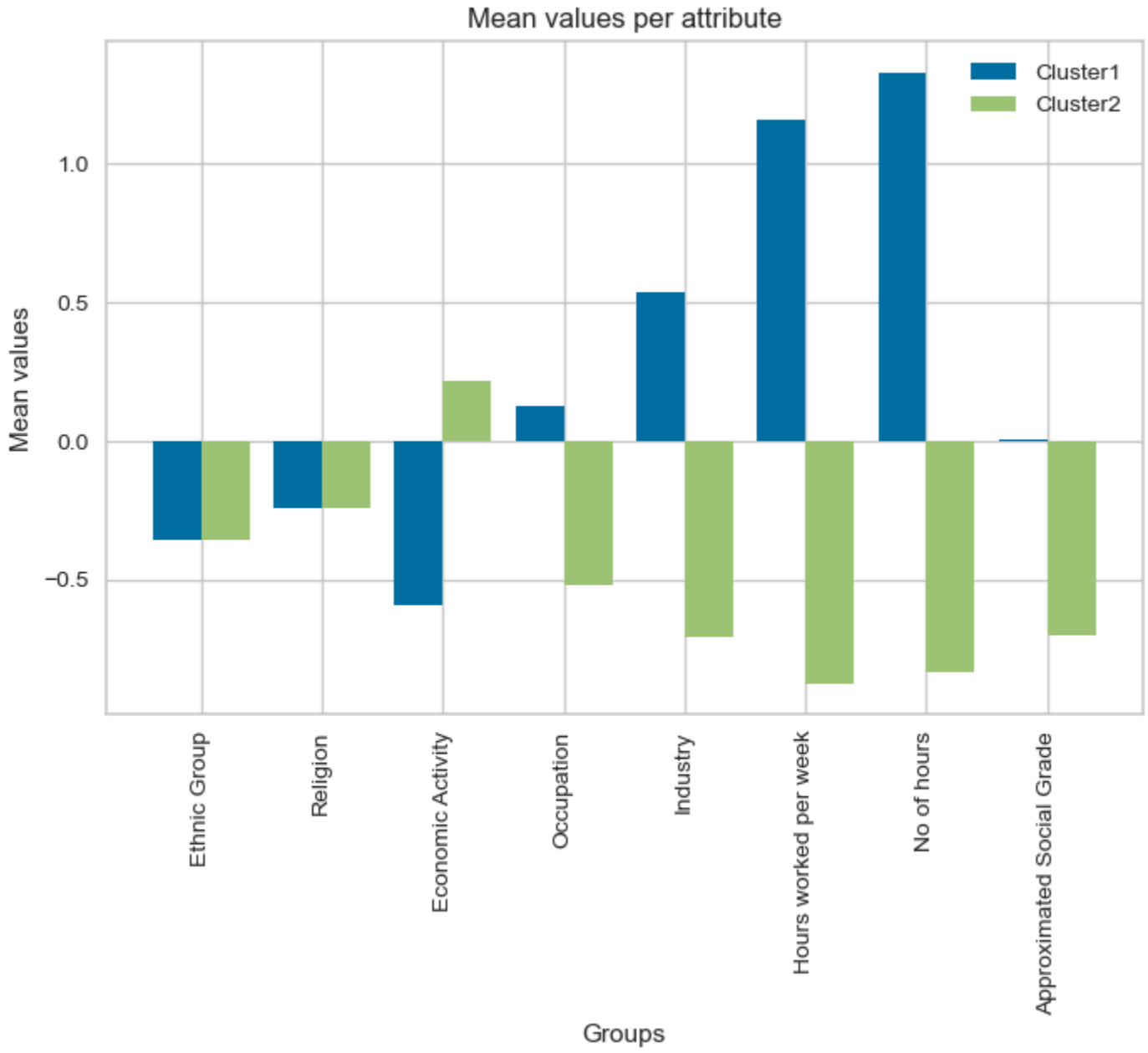
Elbow Curve:



Visualisation of medoid values per attribute of the two clusters for the first 10 columns:



Visualisation of medoid values per attribute of the two clusters for the last 8 columns:



Interpretation and Analysis of the above plots:

Plot	Interpretation
Elbow curve	The elbow curve suggest that the optimal number of clusters for better clustering performance is 2
Visualisation of medoid values per attribute of the two clusters for the first 10 columns	We can see that some attributes show a significant difference in medoid values between the two clusters which indicates better clustering and more distinct separation between the groups. These include- Region, Sex, Age and Health. While, other features show very less difference in medoid values between the two clusters which indicates less influence on the clustering process and contribute less to the separation of the two clusters.
Visualisation of medoid values per attribute of the two clusters for the last 8 columns	Features like Economic Activity, Occupation, Industry, Hours worked per week, No of hours and Approximated Social Grade show significant difference in the medoid values between the two clusters which indicates better clustering and more distinct separation between the groups. On the other hand, other features show less difference between the medoid values which indicates less influence on the clustering process and contribute less to the separation of the two clusters.

Comparative Analysis of all the algorithms used above for clustering

Clustering Method	No. of clusters	Silhouette Score	Analysis
K-Means	2	0.24	It has moderate clustering quality and suggests some overlapping between the clusters
Agglomerative Clustering	2 (single linkage)	0.61	It has excellent clustering quality with clear and distinct separation between the clusters.
	2 (average linkage)	0.61	Similar to single linkage, the clusters are well defined and has distinct separation between the two clusters.
	2 (complete linkage)	0.48	It has good clustering quality but the clusters are less well defined and distinct as compared to single and average linkage.
	2 (ward linkage)	0.21	Considerably less clustering quality, with moderate to less separation between the clusters
	3 (single linkage)	0.59	It has high clustering quality, the clusters have clear separation between them
	3 (average linkage)	0.43	Moderate clustering quality, clusters start to overlap
	3 (complete linkage)	0.42	Similar to average linkage, clusters are less well defined and

			distinct
	3 (ward linkage)	0.18	It has the least clustering quality, with high overlaps between the three clusters with no clear separation.
K-Medoids	2	0.14	The clustering quality needs significant improvement, the clusters have minimal separation between them.

References

1. scikit-learn developers. (n.d.). *scikit-learn: Machine learning in Python*. Retrieved November 17, 2024, from https://scikit-learn.org/stable/
2. Seaborn developers. (n.d.). *seaborn.violinplot.* Retrieved November 17, 2024, from https://seaborn.pydata.org/generated/seaborn.violinplot.html
3. Coding Infinite. (n.d.). *Implement FP-growth algorithm in Python. Coding Infinite*. Retrieved November 17, 2024, from https://codinginfinite.com/implement-fp-growth-algorithm-in-python/
4. University of Portsmouth. (n.d.). *Intelligent Data and Text Analytics.* University of Portsmouth. Retrieved November 17, 2024, from https://moodle.port.ac.uk/course/view.php?id=25669
5. JavaTpoint. (n.d.). *K-medoids clustering*: Theoretical explanation. JavaTpoint. Retrieved November17, 2024, from https://www.javatpoint.com/k-medoids-clustering-theoretical-explanation