A red and blue text on a black background

Description automatically generated

## **Assignment Title Page**

|  |  |
| --- | --- |
| **STUDENT NAME:** | Goh Qing Feng |
| **P.I. NO.:** | Q2181339 |
| **TITLE:** | ECA |
| **COURSE CODE:** | ANL252 |
| **TG:** | T03 |
| **SUBMISSION DATE:** | 31/10/2023 |

**Plagiarism and Collusion**

**Plagiarism** is the act of using or passing as one’s own, the ideas or writings of another without acknowledging or crediting the source from which the ideas are taken from.

**Collusion** is the act of submitting any academic work (including assignment, project or report) that was completed by another person and pass these work off as one’s own.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Declaration** | | | | |
|  | | | | |
| I declare that this assignment is my own work, unless otherwise acknowledge or credited by appropriate referencing. I have read and abide by the SUSS Honour Code and I am aware of the penalties associated with plagiarism and collusion listed in the Student Handbook. | | | | |
|  | | | | |
| Initial: | G.Q.F |  | Date: | 31/10/2023 |
|  | | | | |

Question 1)

After importing pandas and reading the dataset, the first pre-processing task would be to check for duplicate data and remove them. Duplicate data will affect the integrity of a dataset and may lead to inaccurate data analysis. Hence, removing duplicates is a crucial data pre-processing task.

import pandas as pd

ECA = pd.read\_csv("ECA.csv")

ECA = pd.DataFrame(ECA)

ECA.index[ECA.duplicated(keep=False)]

Output:

Int64Index([99, 100, 101], dtype='int64')

Using the duplicated() function, we are able to identify any duplicated data in the dataset. By setting ‘keep=False’, the first occurrence and its duplicates would return ‘True’. Additionally, the ‘.index’ attribute allows us to identify the row labels of the duplicated data. In this case, the output indicates that row indices 99, 100, and 101 contain identical data. Thus, row indices 100 and 101, which are duplicates of row index 99, shall be removed using the drop() function.

ECA = ECA.drop(axis = 0, index = [100,101])

Setting ‘axis = 0’ and ‘index = [100,101]’ allows us to specify that row indices 100 and 101 will be removed from the dataset.

ECA.iloc[99:102]

Output:

A screenshot of a graph

Description automatically generated

The output generated from the .iloc() function allows us to check if there are still duplicates from indices 99 to 101.

The second pre-processing task would be to identify missing values using the isnull() function and replace or delete them. Missing values can lead to data inconsistencies which create difficulties in data analysis. It is also important to assess whether it is appropriate to replace or delete missing values based on the nature of the data and the significance of the missing values.

ECA.isnull().any()

Output:

PersonID False

age True

sex False

bmi False

children False

smoker False

region False

charges False

dtype: bool

The output indicates missing values in the column ‘age’. The following code would specify how many missing values there are.

ECA['age'].isnull().sum()

Output:

123

Furthermore, using the ‘.shape’ attribute, we are able to assess how significant these missing values are to the dataset.

ECA.shape

Output:

(1338, 8)

This shows that out of 1338 rows of data, 123 rows contain missing values. The number of missing values is about 9.2% of the dataset, suggesting that it is relatively significant. Hence, removing the data containing missing values may be inappropriate. Furthermore, the variables in the ‘age’ column are quantitative variables that allow us to replace the missing values with common measures of central tendency, such as mean, median, and mode. In this case, I will replace the missing values with the median using the fillna() function as age is typically not normally distributed. The median is also less sensitive to outliers as compared to the mean.

import numpy as np

age\_median = ECA['age'].median()

ECA['age'] = ECA['age'].fillna(age\_median)

After running the above code, all missing values are replaced and running the same isnull() function will display ‘False’ for all variables.

The last pre-processing task would be to standardize the values in the ‘sex’ column. Using the unique() function, it can be seen that the unique variables are ‘female’, ‘male’, ‘F’, and ‘M’.

ECA['sex'].unique()

Output:

array(['female', 'male', 'F', 'M'], dtype=object)

The variables ‘F’ and ‘M’ will be replaced with ‘female’ and ‘male’ respectively to ensure uniformity in the dataset. If left alone, this inconsistency may lead to errors and misinterpretation of the dataset. Standardising the variables also ensures an accurate visual representation as the variable ‘sex’ will only display two unique variables (female and male) as intended.

ECA['sex'] = ECA['sex'].replace('F','female')

ECA['sex'] = ECA['sex'].replace('M','male')

ECA['sex'].unique()

Output:

array(['female', 'male'], dtype=object)

(Word count: 495)

Question 2)

After importing matplotlib.pyplot, three charts are created from the dataset that was cleaned after running the codes stated in Question 1. The first chart would display the histogram of medical charges.

import matplotlib.pyplot as plt

plt.hist(ECA['charges'],bins=10)

plt.ylabel('Frequency')

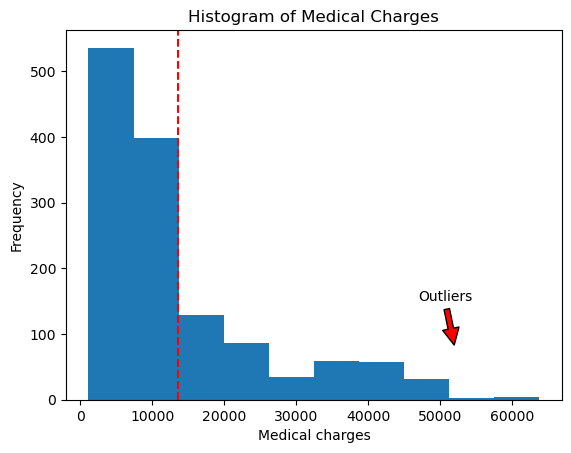
plt.xlabel('Medical charges')

plt.title('Histogram of Medical Charges')

plt.axvline(x = 13600,color = 'red', linestyle = '--')

plt.annotate('Outliers', xy = (52000,80), xytext = (47000,150), arrowprops = dict(facecolor = 'red', shrink = 0.05))

plt.show()



*Figure 1. Histogram of Medical Charges*

The histogram of medical charges displays a distribution that is skewed to the right. This indicates that the mean value is higher than the median, which in turn is higher than the mode. Such a distribution suggests that there are some data points with unusually large medical charges, which are considered outliers. These outliers are shown by the red arrow on the right side of the chart. The red dotted line represents the majority of the data points, which have medical charges of up to $14,000. Due to the presence of outliers, the mean may not be the best measure of central tendency since it may be distorted. Median, which is less affected by outliers than the mean, is considered a more robust measure of central tendency and thus, would be the preferred measure of central tendency for medical charges.

The second chart would display a line chart comparing medical charges and age. A regression line is also generated using the Scikit-learn package to display the overall trend (Biswal, 2023).

from sklearn.linear\_model import LinearRegression

#To find median charge for each age

median\_charges\_by\_age = ECA.groupby('age')['charges'].median()

plt.plot(median\_charges\_by\_age.index, median\_charges\_by\_age.values)

x = median\_charges\_by\_age.index.values.reshape(-1, 1)

y = median\_charges\_by\_age.values

regressor = LinearRegression()

regressor.fit(x, y)

trend\_line = regressor.predict(x)

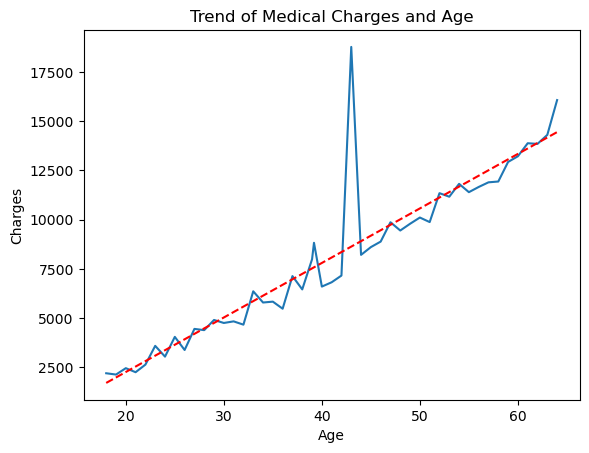
plt.plot(median\_charges\_by\_age.index, trend\_line, linestyle='--', color='red')

plt.title('Trend of Medical Charges and Age')

plt.xlabel('Age')

plt.ylabel('Charges')

plt.show()



*Figure 2. Trend of Medical Charges and Age*

The chart depicts a consistent upward trend, indicating that medical charges tend to increase with age. At the age of 20, the charges amounted to approximately $2,500, while at the age of 60, they reached around $12,500, signifying a rise of almost 400%. This underlines the fact that older people are more vulnerable to health problems and, consequently, incur more medical fees. However, there is a significant spike in charges at the age of 43, where the amount rose to about $17,500. Since most data points closely follow the regression trendline, we can assume that this data point is an outlier.

The last chart is a clustered column chart that displays the relationship between medical charges, gender, and smoking status.

grouped\_data = ECA.groupby(['sex', 'smoker'])['charges'].median().unstack()

grouped\_data.plot(kind='bar', color = ['lightblue','gray'])

plt.title('Medical Charges by Gender and Smoking Status')

plt.xlabel('Gender and Smoking Status')

(male,female) = ECA['sex'].value\_counts()

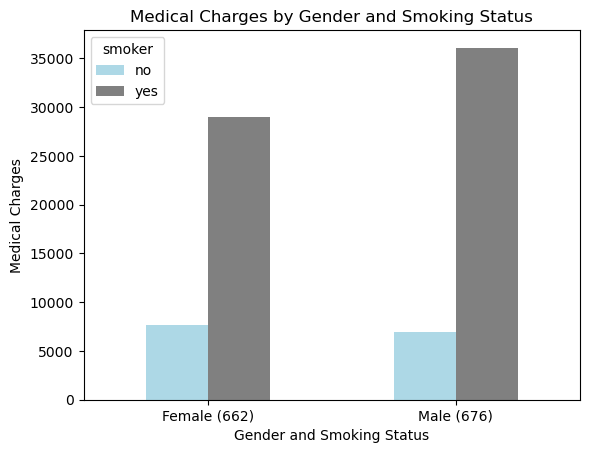
Label\_f = f'Female ({female})'

Label\_m = f'Male ({male})'

plt.xticks([0,1],labels = [Label\_f,Label\_m], rotation=0)

plt.ylabel('Medical Charges')

plt.show()



*Figure 3. Medical Charges by Gender and Smoking Status*

In the dataset, there are 662 females and 676 males. Males have a higher (median) medical charge overall, and smokers incur higher medical charges than non-smokers. This indicates that smoking harms the immune system, leading to more health problems and higher medical fees. If we compare the light blue columns, we can see that the medical charges for female non-smokers are almost the same as those of male non-smokers. Therefore, the difference in the overall medical charges between genders is mostly due to male smokers having a higher medical charge than female smokers. This may suggest gender differences in smoking habits, with males smoking more than females, being more susceptible to health issues, and incurring more medical fees.

(Word count: 439)

Question 3)

from sklearn import tree

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler, Normalizer

As a good practice, I would first create a copy of the cleaned data set. This is to ensure that any other data transformation needed to generate a decision tree would not modify the original data.

train\_test\_set = ECA.copy()

Since the decision tree would be used to predict if a person is a smoker or non-smoker, it is a classification problem that would involve the use of the estimator DecisionTreeClassifier. This estimator would expect numerical input data and is unable to handle categorical variables. Hence, I would need to convert the categorical variables ‘sex’ and ‘region’ into numerical form through one-hot encoding by using the get\_dummies() function.

train\_test\_set = pd.get\_dummies(train\_test\_set, columns=['sex','region'])

Then, I would categorise the variables into independent variable ‘x’ and dependent variable ‘y’.

Feature = train\_test\_set[['age',

'bmi',

'children',

'charges',

'sex\_female',

'sex\_male',

'region\_northeast',

'region\_northwest',

'region\_southeast',

'region\_southwest']]

x = Feature

y = train\_test\_set['smoker'].values

In this case, the variable ‘PersonID’ is an index variable only used to identify the rows in the dataset and does not contain meaningful information for the decision tree modelling. Hence, the variable is excluded from the model to avoid irrelevant information. The remaining variables will be included as only the more important variables will be ultimately selected and used in the decision tree.

Next, the train\_test\_split function is used to split the dataset into a training set and a testing set. The test\_size will be set to 0.3 to reserve 30% of the data set for testing. The random\_state will be allocated the seed of 0.

x\_train,X\_test,y\_train,Y\_test = train\_test\_split(x,y,test\_size = 0.3, random\_state = 0)

dtc = tree.DecisionTreeClassifier(max\_leaf\_nodes = 6, random\_state = 0)

dtc.fit(x\_train, y\_train)

y\_predict = dtc.predict(X\_test)

plt.figure(figsize=(7,8))

tree.plot\_tree(dtc, filled = True, feature\_names = x.columns.to\_list(), class\_names = ['non-smoker','smoker'])

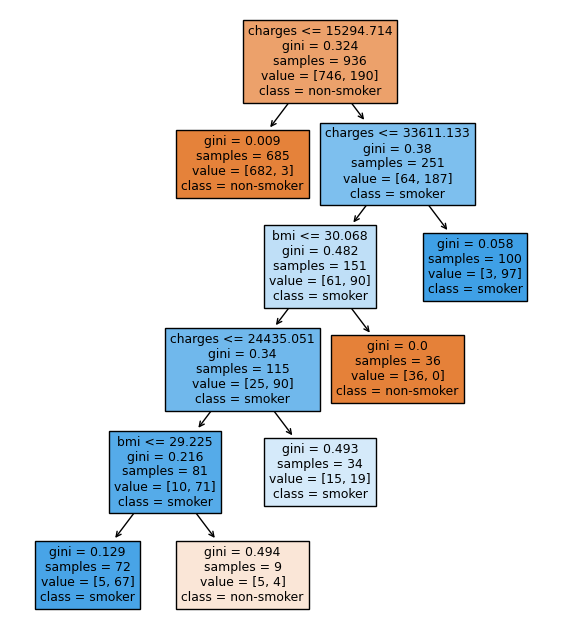
plt.show()

Data transformation steps such as normalisation or standardisation are unnecessary and not implemented as the dataset mainly consists of binary variables with a few continuous variables and decision trees are not sensitive to feature scaling. Furthermore, having the decision point as “charges <= 0.1294” would worsen the model interpretability instead of improving it.

The ‘max\_leaf\_nodes’ parameter is set to 6 to prevent the decision tree from overfitting and to control the complexity of the model.

(Word count: 300)

Question 4)



*Figure 4. Decision Tree Output*

First, a person with medical charges not exceeding $15,294.714 is predicted as a non-smoker (with a confidence of 99.56%).

Second, a person with medical charges exceeding $33,611.133 is predicted to be a smoker (with a confidence of 97.00%).

Third, a person with medical charges above $15,294.714 but not exceeding $33,611.133 and has a bmi above 30.068 is predicted to be a non-smoker (with a confidence of 100.00%).

Fourth, a person with medical charges above $24,435.051 but not exceeding $33,611.133 and has a bmi not exceeding 30.068 is predicted to be a smoker (with a confidence of 55.88%).

Fifth, a person with medical charges above $15,294.714 but not exceeding $24,435.051 and has a bmi not exceeding 29.225 is predicted as a smoker (with a confidence of 93.06%).

Last, a person with medical charges above $15,294.714 but not exceeding $24,435.051 and has a bmi exceeding 29.225 is predicted as a non-smoker (with a confidence of 55.56%).

(Word count: 154)

Question 5)

Decision trees can be effectively used for exploratory data analysis (EDA), but it has limitations and drawbacks that need to be considered.

EDA involves analysing and summarizing the main characteristics of the data for further analysis (Nabriya, 2023). Decision trees are useful tools as they can break down complex data into a visual representation that is easy to interpret and analyse. With decision trees, we can identify influential variables that have a significant impact on the dependent variable, and we can focus on these variables to gain insights into their relationship with the dependent variable. Moreover, analysing the splits and branches in the decision tree can help identify non-linear and complex relationships between various independent and dependent variables. Consequently, we can also identify the lack of significant relationships between variables as irrelevant independent variables may be filtered out of the decision tree, thus simplifying the data, and streamlining the analysis process.

EDA also involves data preparation steps, such as handling outliers (GeeksForGeeks, 2023). Decision trees are suitable for EDA as they’re robust to outliers – splits are based on sample proportions.

However, decision trees may not be appropriate for EDA as they are prone to overfitting, which occurs when the model captures the noise of the training data instead of meaningful patterns (Nadar, 2023). An overfitted decision tree introduces complications, making it difficult to interpret and analyse the data. Additionally, decision trees are highly sensitive to small changes in the data, which can result in drastically different models (Thevapalan & Le, 2023). This instability can lead to inconsistent findings, thus making it challenging to establish a stable understanding of the data’s underlying patterns and relationships. Decision trees also do not provide insights into descriptive statistics, which could be essential for understanding the central tendency and distribution of data.

(Word count: 297)

**List of references**

Biswal, Avijeet. “Sklearn Linear Regression.” *Simplilearn.Com*, 3 Apr. 2023, <https://www.simplilearn.com/tutorials/scikit-learn-tutorial/sklearn-linear-regression-with-examples>.

Nabriya, Pratik. “Exploratory Data Analysis and Visualization Techniques in Data Science.” *Analytics Vidhya*, 16 May 2023, <https://www.analyticsvidhya.com/blog/2021/08/exploratory-data-analysis-and-visualization-techniques-in-data-science/>.

Nadar, Joel Bhaskar. “Overfitting in Decision Tree Models: Understanding and Overcoming the Pitfalls.” *Medium*, 22 June 2023, <https://joelnadarai.medium.com/overfitting-in-decision-tree-models-understanding-and-overcoming-the-pitfalls-880cf7af7d8b>.

Thevapalan, Arunn, and James Le. “Decision Trees in Machine Learning Using R.” *Datacamp*, June 2023, <https://www.datacamp.com/tutorial/decision-trees-R>.

“What Is Exploratory Data Analysis ?” *GeeksforGeeks*, 1 Sept. 2023, <https://www.geeksforgeeks.org/what-is-exploratory-data-analysis/>.