Course Code: ANL252

Assignment Title: ECA

T-Group: T03

SUSS PI No.: Z1910329

Name: Muhammad Fithri Bin Fadiliah

Submission Date: 2nd November 2023

**Question 1.**

**[Coding]**

# Import pandas data analysis library, with alias “pd”.

import pandas as pd

# Under variable “medcost\_df”, run ‘pd.read\_csv’ to load and read ECA.csv in Python.

medcost\_df = pd.read\_csv(“ECA.csv”)

**[First pre-processing task – identifying and locating missing/incompatible values.]**

# Execute .isnull() function to check cells for null/missing values; True = 1, False = 0.

# Execute .sum(axis = 0) to check sum of all cells; “axis = 0” parameter checking values in columns.

# In output, "age" variable has 123 null values.

medcost\_df.isnull().sum(axis = 0)

# To check incompatible values in categorical data (e.g., "M" or "F" in Gender instead of "Male" or "Female")

gender = medcost\_df["sex"].unique()

print(gender)

region = medcost\_df["region"].unique()

print(region)

children = medcost\_df["children"].unique()

print(children)

smoker = medcost\_df["smoker"].unique()

print(smoker)

**[Second pre-processing task – replacing missing/incompatible values]**

# .fillna() function to replace missing values; parameter "value" to change missing values to age mean.

mean\_age = medcost\_df['age'].mean()

medcost\_df.fillna(value = mean\_age, inplace = True)

medcost\_df.isnull().sum(axis = 0)

# Execute .replace() to replace incompatible values in Gender; "M" or "F" to "male" or "female"

# Parameters "to\_replace" for current value, "value" for intended value, and "inplace" to...

# ...switch out values without a copy of pre-replace value.

medcost\_df.replace(to\_replace = "M",

value = "male",

inplace = True)

medcost\_df.replace(to\_replace = "F",

value = "female",

inplace = True)

**[Third pre-processing task – detecting and removing outliers]**

# For “charges” variable, values below first quartile “q1” and third quartile “q3” are ignored for processing.

q1 = medcost\_df["charges"].quantile(q=.25)

q3 = medcost\_df["charges"].quantile(q=.75)

iqr = q3-q1

# Under new variable “treated\_medcost\_df”, finding and removing outliers for “charges” variable in medcost\_df.

treated\_medcost\_df = medcost\_df[~((medcost\_df["charges"]<q1-1.5\*iqr) | (medcost\_df["charges"]>q3+1.5\*iqr))]

treated\_medcost\_df

# Export cleaned dataframe into .csv format for processing.

treated\_medcost\_df.to\_csv('ECA\_Treated.csv',

index = False)

**Explanations:**

1. **Identifying and locating missing/incompatible values**
   1. As there are highly numerous rows indicating observations, it would be time-consuming and difficult to go through the whole file and discover missing values. For this purpose, we use the .isnull() function to uncover if there are any missing values under any of the column variables (e.g., 123 null values in the “age” variable). Missing values may cause a bias in statistical analysis, especially in large numbers (e.g., 123 null values in “age” is sizable; close to 10% of all observations).
   2. In addition, as data may not be clean or homogenous, certain categorical values may be considered by python as a different category altogether (e.g., “M” instead of “male” or “F” instead of “female”) and this can cause some inaccuracies in analysis. As such, categorical values that are incompatible should also be identified and located.
   3. By identifying and locating what variable types have missing values, we can also decide how to deal with different variable types (e.g., numerical value vs categorical value) as some variables may not be required for analysis tasks.
2. **Replacing missing/incompatible values**
   1. While deleting whole observations with missing values via the .drop() function can be efficient, it may cause loss of data that are important in other analysis tasks (e.g., BMI-Charges or Smoker-Charges analysis). Unfortunately, missing values would still exist and replacing such values (e.g., with 0) will only act as proxies while the true values remain obscure.
   2. To preserve values in other variables, the .fillna() function allows us to keep other values beyond the “age” variable, still allowing us to perform analysis tasks of other variables. Missing values in the “age” variable is then ignored during the analysis process through exclusion of outliers.
   3. As for categorical values like in the “gender” variable, values that are considered to be out of the norm like “M” or “F” are subsumed into “male” or “female” categories to standardise values for a more accurate analysis.
3. **Detecting and removing outlier values**
   1. As outliers within a dataset can cause statistical estimation inaccuracies, negatively affecting goodness of fit models, outliers also need to be detected and ignored for analysis tasks.
   2. In deciding which variable to base this outlier exclusion decision, the “charges” variable was chosen as all other variables were found to exist within satisfactory ranges (e.g., ages between 18 to 64), and also because it is the dependant variable we are looking to examine.
   3. In executing the outlier exclusion pre-analysis process, the .quantile() function allows us to determine what the first and third quartiles of the “charges” variable are – thus allowing the exclusion of those below and above the first and third quartiles respectively.

**Question 2.**

**Line plot:**

**[Coding]**

treated\_medcost\_df = pd.read\_csv("ECA\_Treated.csv")

import matplotlib.pyplot as plt

import seaborn as sns

# Graph style, addition of gridlines

sns.set(style = "whitegrid")

# Creating line plot for age vs charges

sns.lineplot(data = treated\_medcost\_df, x = "age", y = "charges")

# Detailing lineplot x and y axes, and title

plt.xlabel("Age of Clients")

plt.ylabel("Medical Costs")

plt.title("Medical Costs based on Age")

plt.show

**[Diagram]**



**Medical Costs vs Age**

Line plot suggests a positive correlation between age and charges, where the older the patient is, the more their medical cost would be. This is an expected relation as extant literature suggests the older population show more susceptibility to medical issues – showing how age can be a prime factor in rising medical costs. As such, clients can expect to have larger expenditure towards healthcare as they grow older.

**Scatter plot:**

**[Coding]**

# Creating scatter plot for BMI vs charges

sns.scatterplot(data = treated\_medcost\_df, x = "bmi", y = "charges")

# Detailing scatter plot x and y axes, and title

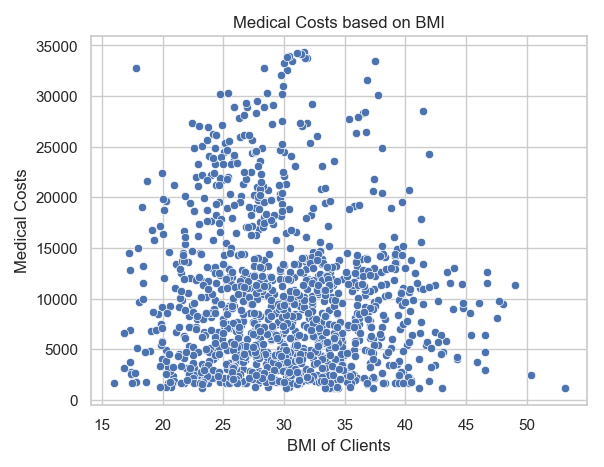
plt.xlabel("BMI of Clients")

plt.ylabel("Medical Costs")

plt.title("Medical Costs based on BMI")

plt.show

**[Diagram]**



**Medical Costs vs BMI**

Despite assertions in extant literature that a higher body mass index (BMI) would show higher susceptibility to medical issues (and therefore higher medical costs), the scatterplot shows no discernible patterns of this relationship. This considers that the healthy BMI range is between 18.5 and 22.9. While a small number of outliers exist on both the medical cost and BMI axes, the large majority of the distribution show that there is little correlation between medical costs and the BMI of the clients. As such, while BMI can be a predictor of higher medical costs, it is difficult to assert that it is a significant one with this sample.

**Strip plot:**

**[Coding]**

# Creating scatterplot for smoker vs charges

sns.stripplot(data = treated\_medcost\_df, x = "smoker", y = "charges", jitter = True, hue = "smoker")

# Detailing lineplot x and y axes, and title

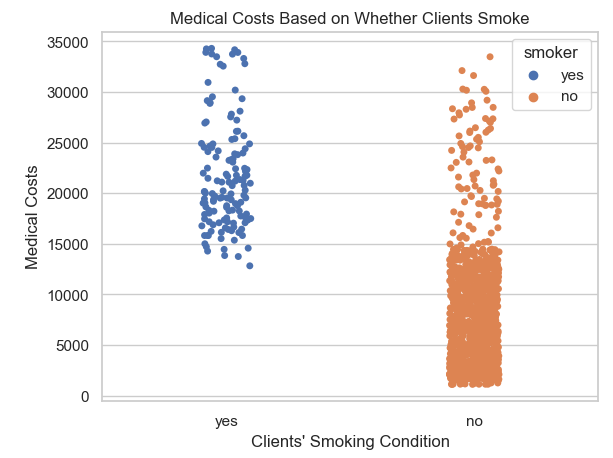
plt.xlabel("Clients' Smoking Condition")

plt.ylabel("Medical Costs")

plt.title("Medical Costs Based on Whether Clients Smoke")

plt.show

**[Diagram]**



**Medical Costs vs Smoker**

In agreement with extant literature, the above strip plot seems to suggest that most who do not smoke show lower medical cost with majority of those paying below $15000 being non-smokers, and almost all smokers paying beyond $15000. This suggests that being a smoker may come with higher medical costs due to more medical complications related to smoking, while being a non-smoker is related to lesser medical costs. Despite that, there is a skewed proportion between smokers and non-smokers in this sample (~ 300 smokers vs ~1000 non-smokers). That being said, the proportion of smokers to non-smokers in this sample may be representative of the population if the sampling method is randomised.

**Question 3.**

**[Coding]**

# Load libraries for decision tree - https://www.datacamp.com/tutorial/decision-tree-classification-python

import numpy as np

from sklearn import tree

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

from sklearn.metrics import classification\_report

smoker\_model = tree.DecisionTreeClassifier(max\_depth = 5, max\_features = 3)

# DecisionTreeClassifier cannot work with categorical variables, therefore dummies from pd.get\_dummies needed to...

# ...replace categorical variables with numericals of either 1 or 0.

treated\_medcost\_df = pd.get\_dummies(treated\_medcost\_df, drop\_first = True)

treated\_medcost\_df

# Separating independent and dependent variables, dropping irrelevant variables

# E.g., dependent variable = smoker, independent variables = age, bmi, children, charges, sex

X = treated\_medcost\_df.drop(['PersonID','smoker\_yes','region\_northwest','region\_southeast','region\_southwest'], axis = 1)

Y = treated\_medcost\_df['smoker\_yes']

# Proportioning data for training to learn patterns, and test data to model predictions (e.g., 70% to training...

# ...30% to testing)

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.3, random\_state=5)

# Fitting training data onto decision tree through .fit() function

smoker\_model.fit(X\_train,Y\_train)

# Thereafter using trained data to inform predictions via .predict() function

Y\_predict = smoker\_model.predict(X\_test)

print(classification\_report(Y\_predict,Y\_test))

**Approach explanation:**

* As the dependent variable is whether the client is a smoker or a non-smoker, a categorical decision tree was employed (decisiontreeclassifier) to allow leaf nodes to arrive at either decisions.
  + The smoker variable is assigned to Y as the dependent variable, while the rest of the variables are independent variables assigned to X.
* In preparing the dataset for the decision tree process, the categorical variables are converted into numerical variables (e.g., male indicated by 1 in male\_sex variable, while female indicated by 0) through the onehotencoder method to indicate their lack of ordinal value.
  + To facilitate the separation between variables, the smoker variable is also converted into numerical values (smokers indicated by 1 in the smoker\_yes variable, while non-smokers are indicated by 0).
* In avoiding overfitting and examining too many variables at once, certain variables are dropped out of the decision tree process (e.g., PersonID, Region) so that other more proximal variables can be explored first.
  + Furthermore, the number of features and maximum depth was also balanced so that the decision tree would not be visually complex and allowing better generalisation of the test data.
* To facilitate a more accurate machine learning for prediction, the dataset is then proportioned towards training and testing – where majority (70%) of the dataset facilitates the training processes to inform the remaining 30% for testing.

**Question 4.**

**[Coding]**

# Prepping visualisation of decision tree, installing and importing graphviz

%conda install graphviz

import graphviz

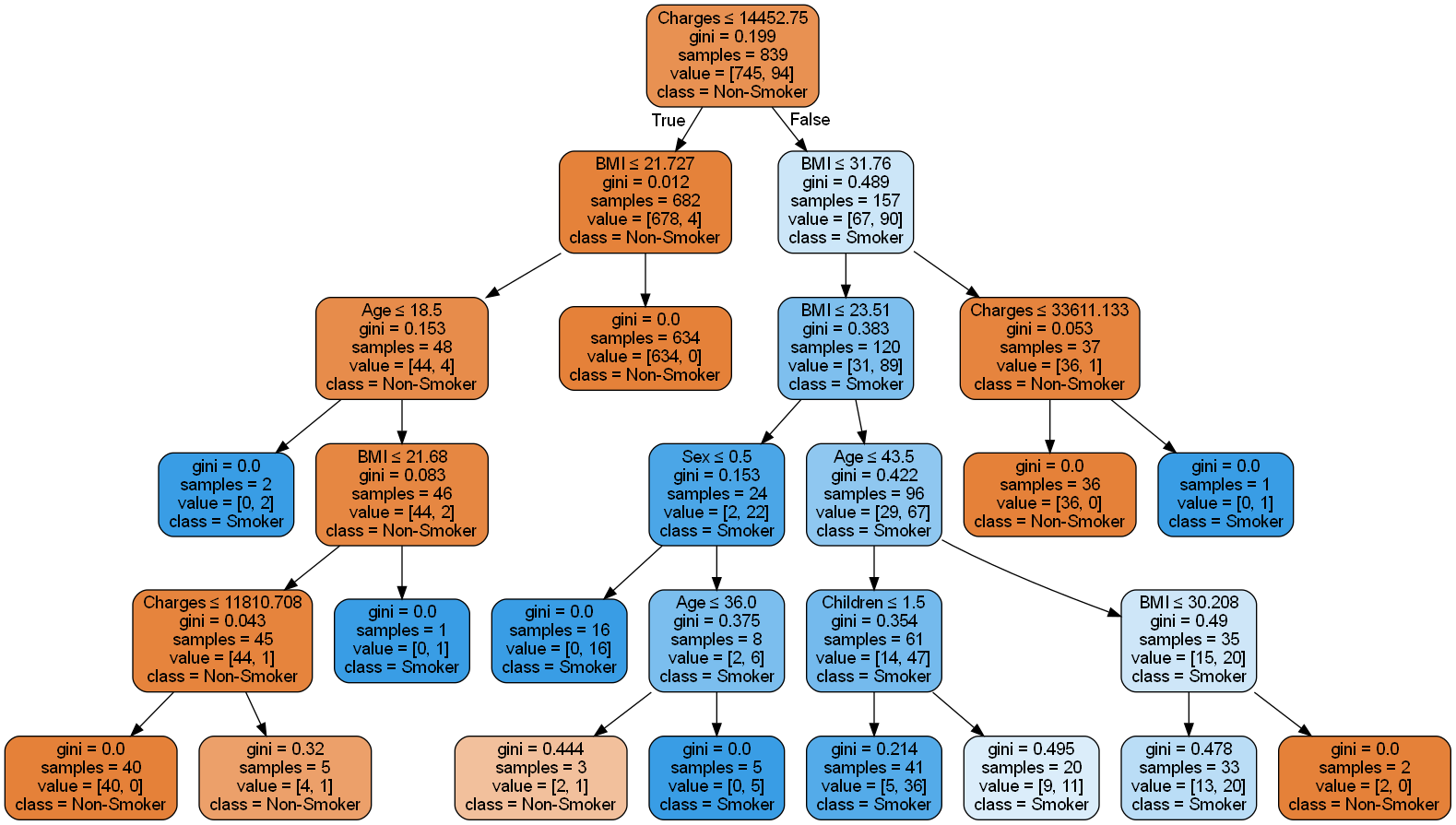
# Visualisation of decision tree, with filled (color), variable names (independent variable),...

# and class names (dependent variable) on nodes.

dot\_data = tree.export\_graphviz(smoker\_model, out\_file = None, feature\_names = x\_variables, class\_names = ['Non-Smoker','Smoker'], filled = True, rounded = True,special\_characters = True)

graph = graphviz.Source(dot\_data)

graph



**Insight:**

According to the decision tree above, those with much lesser medical costs (e.g., charges ≤ 14452.75) are most likely to be non-smokers. This is especially so if their BMI is within the healthy range (e.g., BMI ≤ 21.727). With lower significance, for those whose medical costs are much higher (e.g., charges ≥ 14452.75), the decision tree suggests that clients are more likely to be smokers if they have below 32 BMI, are female, aged below 43.5 and have lesser than 2 children. This suggests that while the chances of being a smoker can have numerous variables, the decision tree asserts how those with higher charges are more likely those who are classified as smokers.

**Question 5.**

Via decision trees, exploratory data analysis can be effectively employed by allowing us a better understanding of underlying relationships and patterns in the data.

According to Breiman (2017), one manner how trees aid exploratory data analysis is the discovery and visualisation of relationships and patterns between large sets of variables by describing how dependent variables interact with numerous different variables. Decision trees help discern complex interactions and non-linear correlations through recursive partitioning processes in classification and regression tree algorithm, of which cannot be easily detected in traditional statistical methods. And this is exceptionally helpful when handling datasets with high numbers of variables.

Secondly, Song and Lu (2015) suggested that exploratory analysis can be effectively done through decision trees by understanding how significant variables are. In examining the splits in the trees and the variables therein, we can understand which variables are the most significant predictors of the dependent variable, thus helping us identify key factors of the patterns in the data. Thusly, decision trees also help us visually see likely costs, results, and implications of complex decisions (Asana, 2023).

However, decision trees can complicate exploratory analysis when it becomes too overly complex due to overfitting – most likely when there are too many leaf nodes and when datasets contain noise (Song & Lu, 2015). To counter this, regularisation techniques like pruning or employing ensemble techniques (e.g., random forest) that help calibrate the decision tree model to minimise errors can help mitigate overfitting (Slatery, 2020).

Summarily, decision trees can be effectively used for exploratory purposes beyond just traditional purposes of prediction.

**References**

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: Applications for classification and prediction. *Shanghai Archives of Psychiatry*, *27*(2), 130–135. <https://doi.org/10.11919/j.issn.1002-0829.215044>

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Slatery, L. (2020, March 8) *Decision trees: Understanding the basis of ensemble methods*. Medium. <https://towardsdatascience.com/decision-trees-understanding-the-basis-of-ensemble-methods-e075d5bfa704>

Asana. (2023, February 27). *What is decision tree analysis? 5 steps to make better decisions*. <https://asana.com/resources/decision-tree-analysis>