

**ANL252**

**Python for Data Analytics**

# **End of Course Assessment**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Tea Ich Ngy** | **Y2210678** |
| Click to enter NAME. | Click to enter PI NO. |
| Click to enter NAME. | Click to enter PI NO. |
| Click to enter NAME. | Click to enter PI NO. |

# **Tutorial Group: ­­­­­­­­­­­­ T 03**

**Instructor’s Name: Mr. Munish Kumar**

**Submission Date: 03/11/2023**

**Question 1**

The first data pre-processing that was used is to find if there’s any missing values in the data set. For this, I have used the following code:

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

Results

PersonID 0

age 123

sex 0

bmi 0

children 0

smoker 0

region 0

charges 0

dtype: int64

Upon executing this code, I have found that out of all the categories, “age” has 123 missing data. To resolve this, I have decided to remove all data relating to missing “age”. The rationale behind this is that, “age” may be an important category for analysis and we want the data to be as true as possible, and replacing them with mean or median from the data set may give rise to inaccurate analysis later on. To remove the data, .dropna with how as “any” is used:

data\_df\_clean1 = data\_df.dropna(axis=0, how="any")

The second data pre-processing is to find for each category, if there’s any different values. After exploring the data, I have found that for “sex” category, there are different values when the following code:

data\_df\_clean1["sex"].str.get\_dummies(sep=', ').sum()

Results

F 4

M 2

female 598

male 613

dtype: int64

It was noted that there are 4 values that show “F” instead of “female” and 2 values that show “M” instead of “male. Although “F” and “M” carries the same meaning as “female” and “male, python will recognise this as 4 separate variables, instead of 2. If left untreated, this may affect any analytical models that would be carried out later. To rectify this, I have replace these values to its correct values using the following codes:

data\_df\_clean1["sex"] = data\_df\_clean1["sex"].replace({"F": "female"})

data\_df\_clean1["sex"] = data\_df\_clean1["sex"].replace({"M": "male"})

The third data pre-processing is to check for any outliers for numerical categories. Categories such as “bmi”, “charges” and “age” were checked, to ensure any outliers that do not usually belong in the data range will be remove. It is important to remove outliers in general because some analytical models like K-Means Clustering and Linear Regression models, which are sensitive to outliers, may produce inaccurate models. In our case, this method is use to filter out any potential figure that might have keyed in wrongly. To execute this, I used the following codes:

(Codes were repeating using different categories, “bmi”, “charges” and “age”)

q1 = data\_df\_clean1["bmi"].quantile(q=.25)

q3 = data\_df\_clean1["bmi"].quantile(q=.75)

iqr = q3-q1

data\_df\_clean1[~((data\_df\_clean1["bmi"]<q1-1.5\*iqr) | (data\_df\_clean1["bmi"]>q3+1.5\*iqr))]

data\_df\_clean1

The code uses interquartile range (IQR) to eliminate any lines of data that does not fit into data range. Upon executing the codes for all three numerical categories, I have noted that there are no outliers exist in the three categories as the number of rows were unchanged, telling us that no outliers data was found to be eliminated.

(385 words)

**Question 2**

Chart 1: Histogram

Codes:

# Importing pandas, matplotlib and seaborn library

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Using .read\_csv to read the dataframe

ECA\_df = pd.read\_csv("/Users/tea/Desktop/Python projects/School/ECA/cleaned\_ECA.csv")

# Histogram of BMI for male and female

sns.set\_style('darkgrid')

sns.histplot(data=ECA\_df, x="bmi", hue="sex", kde=True)

plt.xlabel("BMI")

plt.ylabel("Frequency")

plt.title("Histogram of BMI for Male and Female")

**A graph of a person and person

Description automatically generated**

The chart shows a histogram of the BMI for male (in orange) and female (in blue). From the histogram, we can see that firstly the BMI for male and female are normally distributed, as shown by the line graphs drawn on the diagram. Furthermore, we can see that in general, the frequency of females having BMI level from 30 to 35 seems to be higher than males. Conversely, the frequency of males having BMI level from 15 to 25 seems higher than the females. From this alone, we can see that males from this data set are generally healthier than the females based on their BMI level, which were at the ideal level of BMI from 18.5 to 24.9.

Chart 2: Scatter plot

Codes:

# Importing pandas, matplotlib and seaborn library

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Using .read\_csv to read the dataframe

ECA\_df = pd.read\_csv("/Users/tea/Desktop/Python projects/School/ECA/cleaned\_ECA.csv")

# Scatter plot of age and charges

sns.set\_style('darkgrid')

sns.scatterplot(data=ECA\_df, x="age", y="charges", hue="sex")

plt.xlabel("Age (years)")

plt.ylabel("Charges ($)")

plt.title("Scatterplot between Age and Charges for Male and Female")

A graph of age and charges

Description automatically generated with medium confidence

The chart shows a scatter plot between age and charges for males (in orange) and females (in blue). From the scatter plot, we can see that there is a simple positive linear relations between age and charges, meaning that the higher the age, the higher the charges will be for the medical costs. This is generally true because as people get older, there are more health complications to be treated, which can cost more.

However, interestingly, the scatter plot also seemed to show 3 distinct range of charges. This can indicate to us that these 3 range of charges may be the different medical insurance policies. For example, the lowest range, which also have the highest occurrence, maybe the basic insurance policy where most people will get for their health insurance. Whereas the higher 2 tiers, could be higher and more premium insurance policies that cover more health complications, which in turn costs more.

Chart 3: Bar chart

Codes:

# Importing pandas, matplotlib and seaborn library

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Using .read\_csv to read the dataframe

ECA\_df = pd.read\_csv("/Users/tea/Desktop/Python projects/School/ECA/cleaned\_ECA.csv")

sns.set\_style('darkgrid')

sns.barplot(ECA\_df, x="region", y="charges", hue="sex")

plt.xlabel("Region")

plt.ylabel("Charges ($)")

plt.title("Barchart of charges by region for Male and Female")

A graph of a bar chart

Description automatically generated

The chart shows a simple bar chart between the charges and the different regions for males (in orange) and females (in blue). At a glance, we can see that the southeast region generally incurred higher charges than the other 3 regions. This can indicate to us that the southeast region may have more premium insurance policy holders than the rest, which makes it the richest region. This supports the scatter plot in chart 2, showing 3 distinct charges, which relates to the different insurance polies covered

In addition to that, lowest charges for males is from the southwest region and lowest charges for females is from the northwest region. This can indicate to us that males and females of these regions may not have as much medical complications to be covered by insurance than the rest of the regions. This can be useful to medical researchers who may be looking into improving healthcare, as they can start with the specific demographic of that region.

(Total: 437 words)

**Question 3**

To plot a decision tree, the following libraries and sub-libraries were imported:

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import tree

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

from sklearn.tree import DecisionTreeClassifier

from sklearn.tree import plot\_tree

from sklearn.preprocessing import LabelEncoder

from sklearn.feature\_extraction import DictVectorizer

from sklearn.metrics import classification\_report

Then we load the data into data frame using .read\_csv:

# Loading data into dataframe using .read\_csv

ECA\_df = pd.read\_csv("/Users/tea/Desktop/Python projects/School/ECA/cleaned\_ECA.csv")

As the data frame have categorical values and decision tree can only work with numerical values, we have to encode those categorical values first. The categorical categories are “sex”, “smoker”, and “region”.

# Encode categorical values

ECA\_df['sex'] = LabelEncoder().fit\_transform(ECA\_df['sex'])

ECA\_df['smoker'] = LabelEncoder().fit\_transform(ECA\_df['smoker'])

ECA\_df['region'] = LabelEncoder().fit\_transform(ECA\_df['region'])

The next step is to define the dependent and independent variables. I have assigned X as independent variables and Y as dependent variable “smoker”:

# Define the independent variables

X = ECA\_df[['age','bmi','charges']]

# Define the dependent variable

Y = ECA\_df['smoker']

It is notable to mention that while exploring the modelling of the decision tree, I have found that using only “age”, “bmi” and “charges” will result in the least number of branches nodes possible (10 branches). I have also found that without independent variables “bmi” and “charges”, the model will result in a large tree that is not useful as a predictive model.

The next step is to split the data into training and testing sets. The training data sets is what we will use to train and model the decision tree, whereas the testing set will be use to evaluate the performance of the decision tree model:

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=5)

print(X\_train.shape)

print(X\_test.shape)

print(Y\_train.shape)

print(Y\_test.shape)

Results:

(973, 3)

(244, 3)

(973,)

(244,)

The next step is to fit the classifier into the training data set. I achieve this by doing the following:

# Fitting the classifier into the training data

dtc = tree.DecisionTreeClassifier()

dtc.fit(X\_train, Y\_train)

The following step is optional step which will not affect the modelling process of the decision tree, which is to show the classification report:

# Printing classification report

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

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

Results:

precision recall f1-score support

0 0.97 0.99 0.98 192

1 0.96 0.88 0.92 52

accuracy 0.97 244

macro avg 0.96 0.94 0.95 244

weighted avg 0.97 0.97 0.97 244

The final step is to plot the decision tree visually. I achieve this by using the following codes:

plt.figure(figsize=(12, 12))

tree.plot\_tree(dtc)

plt.show()

(Total: 247 words)

**Question 4:**

Decision Tree:

A diagram of a network

Description automatically generated

From the decision tree, there are a total of 10 branches nodes leading to the end of the tree. The root note started off with “charges” of <= to 14,525.11, which further branch out to other “charges” nodes. I have also noted that X[1] which is “bmi” and X[2] which is “charges” appears more frequently as the earlier leaf nodes than X[0] which is “age”. This tells us that “bmi” and “charges” are important to predict if someone is a smoker or not.

(83 words)

**Question 5:**

While decision trees are traditionally used for making predictions, I do think it can also be used effectively for exploratory data analysis (EDA). Taken from IBM, exploratory data analysis is generally “used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.” (What Is Exploratory Data Analysis? | IBM, n.d.).

It is true that the primary function of decision trees is to be used as a predictive model. However, the modelling process of decision tree, notably using Python, does indeed help with EDA. For example, using the decision tree modelled in question 3 and 4, I have learned that there are some independent variables that are more effective to be used in making the decision tree than others. If I am to explore a huge data set with many categories, modelling decision tree can help me to narrow into what specific independent variables that will play a huge part in determining the dependent variable that I have set for. In this way, modelling decision tree had helped me to discover patterns quicker.

(213 words)

**References:**

*Wu, K. Y., & Zhu, S. (2023). ANL252 Python for data analytics. Singapore University of Social Sciences.*

seaborn.barplot — seaborn 0.13.0 documentation. (n.d.). <https://seaborn.pydata.org/generated/seaborn.barplot.html>

seaborn.histplot — seaborn 0.13.0 documentation. (n.d.). <https://seaborn.pydata.org/generated/seaborn.histplot.html>

seaborn.scatterplot — seaborn 0.13.0 documentation. (n.d.). <https://seaborn.pydata.org/generated/seaborn.scatterplot.html>

What is Exploratory Data Analysis? | IBM. (n.d.). <https://www.ibm.com/topics/exploratory-data-analysis>

1.10. Decision Trees. (n.d.). Scikit-learn. <https://scikit-learn.org/stable/modules/tree.html>