

**ANL252 (Online)**

**Python for Data Analytics**

# **End-of-Course Assessment - July Semester 2023**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Don Lee Yang** | **W2270423** |

**Instructor’s Name: Mr. MUNISH KUMAR**

**Submission Date: 01/11/2022**

**Question 1**

#importing of libraries  
import pandas as pd   
import numpy as np

#importing of csv file to python  
eca\_csv = pd.read\_csv("ECA.csv")  
eca\_csv.shape

Output:(1340, 8)

#first pre-processing task, removing missing data from data set   
#Verify if there are missing data in the data set

missing\_data = eca\_csv.isnull().any()  
missing\_data

PersonID False

age True

sex False

bmi False

children False

smoker False

region False

charges False

dtype: bool

#Observe there are missing datas in "age"  
#Extract rows in age with empty data

missing\_data\_age = eca\_csv.loc[eca\_csv['age'].isnull()]  
missing\_data\_age.shape

Output:(123, 8)

#dropping all rows with empty data set   
eca\_csv.dropna(axis = 0 , how = 'any')

new\_eca\_csv\_dataset = eca\_csv.dropna(axis = 0 , how = 'any')  
new\_eca\_csv\_dataset.shape

Output: (1217, 8)

The first task is to remove any empty rows in the dataset. Removing empty rows in the dataset improves the quality of data. Empty data might lead to inaccurate analysis, affecting the whole analysis. Thus, we need to remove any missing data before conducting any analysis.

I first imported the “pandas” library to read the .csv file. To verify if there is any missing data in the dataset, I used the “isnull ()” function and noticed that there are missing data in the column “age”. After this, I used the “.shape” function to determine the missing data in the dataset.

I utilised the “dropna” function to remove rows containing empty data. This function will drop all rows that contain any empty data. After this, I used the “.shape” function to determine that all the empty rows were removed.

#Assuming that dataset has already been imported   
#Standardise the genders in the data set   
#Checking for the unqique values in "sex"

unique\_sex = (eca\_csv['sex'].unique())  
unique\_sex

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

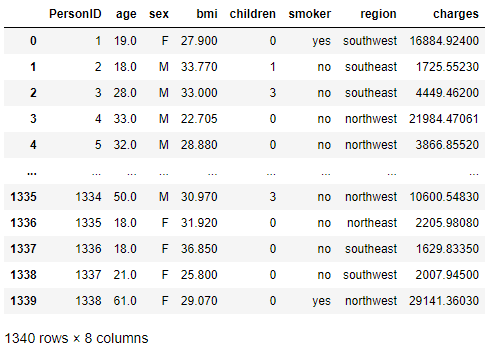
#Replacing the genders with just 2 unique values

eca\_csv\_updated\_female = eca\_csv.replace(to\_replace="female", value="F", inplace = False)

eca\_csv\_updated\_male = eca\_csv.replace(to\_replace="male", value="M", inplace = True)

eca\_csv

Output:



The second task is to handle the categorical data in the dataset. When looking at the dataset, I noticed that the “sex” column has four different unique values. However, those unique values are “F,” “M,” “female,” and “male.” Where “F” equates to “female” and “M” equates to male. To fix this, I decided to ensure that the category “sex” only has two unique values.

I used “F” and “M” as my two unique values. To replace all the “female” and “male” in the “sex” category, I used the “replace()” function as per my code above. The function replaced all the “male” with “M” and “female” with “F.” After which, I printed the data frame to ensure that all data in the “sex” column had been replaced accordingly. By standardising the dataset, I made it more consistent and easier to work with.

#Last task is normalising. I choose "charges" to normalise

#import standardscaler from library

from sklearn import preprocessing  
from sklearn.preprocessing import StandardScaler

#Convert chargers from string to float

eca\_csv['charges'] = eca\_csv['charges'].astype(float)

#create a duplucate of charges

eca\_csv['charges\_normalise'] = eca\_csv.loc[:, 'charges']

#normalise charges duplicate

norm\_charges = eca\_csv['charges\_normalise']

norm\_charges\_array = np.array(norm\_charges)

# Reshape the NumPy array to 2D

norm\_charges\_array = norm\_charges\_array.reshape(-1, 1)

#Normalising the data

scaler = preprocessing.StandardScaler()

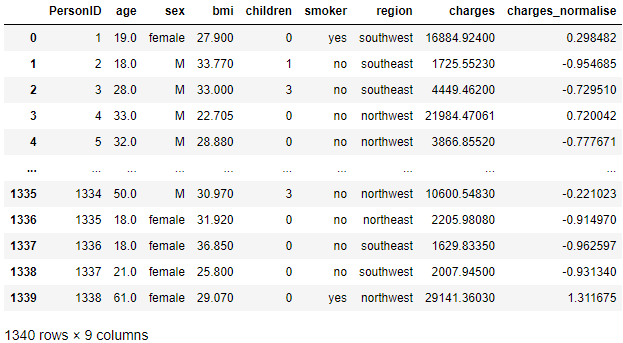
scaler.fit(norm\_charges\_array)

norm\_charges\_array = scaler.transform(norm\_charges\_array)

#inserting the normalised data back into the data frame

eca\_csv['charges\_normalise'] = norm\_charges\_array

eca\_csv



The last task for would be normalising the data. Normalising data is part of data pre-processing for better machine learning  (Jaitley, 2019). For this task, I have selected “charges” as the column to normalise.

I imported the "standardscaler" from the library to normalize the data. After that, I converted the "charges" column from a string to a float. I duplicated the "charges" column and named the column "charge-normalise." Through the "charges-normalised" column, I normalised the data through the "preprocessing.StandardScaler()" function to get the normalise data.

In conclusion, preparing our data beforehand is an essential step to attain our dataset's best and most accurate analysis. The three steps I presented, removing empty rows, standardising data, and normalising data, are just some ways we can prepare our data for analysis.

(407 Words)

**Question 2**

#importing of libraries

import pandas as pd   
import matplotlib.pyplot as plt

#importing of csv file to python

medical\_cost = pd.read\_csv("ECA.csv", usecols = ["charges"])

#Plotting of histogram chart   
plt.title ("Medical Cost Histogram")  
plt.hist (medical\_cost, bins = None , range = None , align = "mid", orientation = "vertical", rwidth = None , color = None)

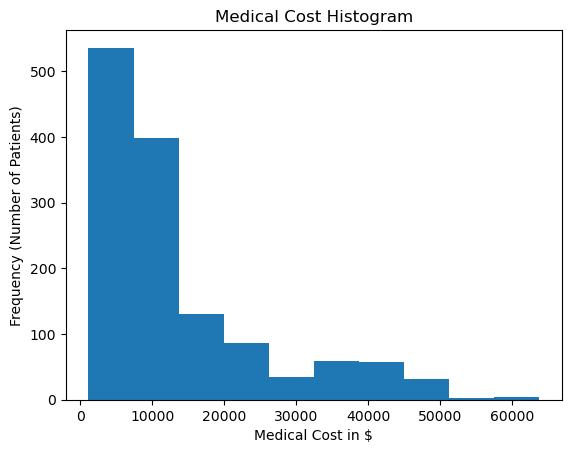
# Labelling the chart

plt.xlabel("Medical Cost in $")  
plt.ylabel("Frequency (Number of Patients)")

# Display the plot

plt.show()

Histogram



The first chart presented is a histogram. In the histogram above, the height of each bin, measured on the y-axis, indicates the number of patients, while the x-axis shows the medical cost for each patient.

The medical cost histogram chart above also has a right-skewed distribution, indicated by the highest peak on the left before tapering towards the right. I observe that most patients are being charged less than $20,000 for their medical costs. While there are very few data points between the $50,000 to $60,000 range, which may indicate that only a few patients pay more than $50,000 for their medical costs. A right-skewed distribution also means that the mean of the medical cost is more than the median, and the median is more than the mode of the medical cost.

However, as there are a few outliers above the $50,000 mark, it is worth checking the data set for any data entry errors before properly analysing the whole data set.

#Plotting scatter plot

plt.title ("BMI vs Medical Cost")

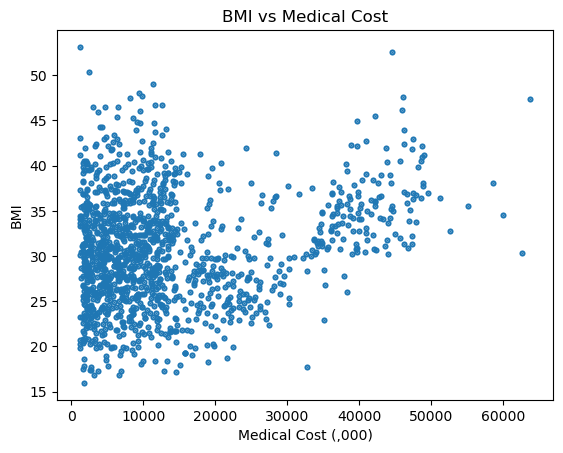
plt.xlabel("Medical Cost (,000)")

plt.ylabel("BMI")

plt.scatter (medical\_cost , bmi , color = None , marker = None , linewidths = 3 , edgecolors = None, s = 2)

plt.show()

Scatter Plot (BMI vs Medical Cost)



The second chart presented above is a scatter plot of the body mass index (BMI) of each patient versus their medical costs. The purpose of plotting a scatter plot is to observe any correlation between the two variables.

When observing the scatter plot, I noticed that most of the data points are scattered towards the left side of the chart, which could indicate that factors are keeping the medical cost low for these patients.

However, when looking toward the right side of the chart, I noticed that there is a slight positive correlation between BMI and medical cost as the data sets are trending upwards, which indicates a relationship between the patient's BMI and medical cost. There are also 5 data points above the $50,000 mark, which suggests that there may be other factors affecting the medical cost aside from the patient's BMI.

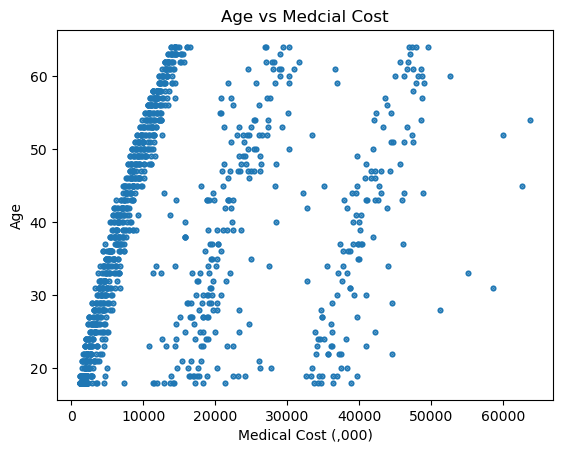
#Creating scatter plot Age vs Charges   
#Importing csv files  
medical\_cost = pd.read\_csv("ECA.csv", usecols = ["charges"])  
age = pd.read\_csv("ECA.csv", usecols = ["age"])

#Plotting scatter plot

plt.title ("Medical Cost vs Age")  
plt.xlabel("Medical Cost (,000)")  
plt.ylabel("Age")  
plt.scatter (medical\_cost , age , color = None , marker = None , linewidths = 3 , edgecolors = None, s = 2)

plt.show()

Scatter Plot (Age vs Medical Cost)



The third chart presented above is a scatter plot of the age of each patient versus their medical costs. The purpose of plotting a scatter plot is to observe any correlation between the two variables.

When observing the scatter plot, I noticed that there is no correlation between the both variables. This is evident as the data points are over the charts and there no positive nor negative slop in the scatter plot, which indicates that there might be no relation between the age of patient and medical costs.

(394 Words)

**Question 3**

#importing of the necessary libraries   
import pandas as pd  
from sklearn import tree  
from sklearn.tree import DecisionTreeClassifier   
from sklearn.model\_selection import train\_test\_split   
from sklearn import metrics  
from sklearn.metrics import classification\_report

#importing csv  
eca\_csv = pd.read\_csv("ECA.csv")

#preparation of dataset

#checking of any null value   
eca\_csv.isnull().sum()

Output:

PersonID 0

age 123

sex 0

bmi 0

children 0

smoker 0

region 0

charges 0

dtype: int64

#removing rows with null values

eca\_csv = eca\_csv.dropna (axis = 0 , how = 'any'

eca\_csv.isnull().sum()

Output:

PersonID 0

age 0

sex 0

bmi 0

children 0

smoker 0

region 0

charges 0

dtype: int64

#Replacing the genders with just 2 unique values

eca\_csv = eca\_csv.replace(to\_replace="female", value="F", inplace=False)

eca\_csv.replace(to\_replace="male", value="M", inplace = True)

#converting all str to numeric form to allow machine learning

eca\_csv['smoker'] = eca\_csv['smoker'].replace({'yes': 0, 'no': 1})

eca\_csv['sex'] = eca\_csv['sex'].replace({'M': 0, 'F': 1})

eca\_csv['region'] = eca\_csv['region'].replace({'northeast': 0, 'southeast': 1, 'southwest': 2, 'northwest': 3 })

#splitting our dataset between dependent and independent variables

x = eca\_csv.drop (['smoker'] , axis = 1) #independent

y = eca\_csv[['smoker']] #dependent

#splitting of data between training and testing with 20% testing

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state = 5)

dtc = tree.DecisionTreeClassifier(max\_depth = 6)

#fit decision tree on training data

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

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

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

Output:

precision recall f1-score support

0 0.92 0.85 0.88 52

1 0.96 0.98 0.97 192

accuracy 0.95 244

macro avg 0.94 0.91 0.92 244

weighted avg 0.95 0.95 0.95 244

I first imported the necessary libraries to enable a decision tree. After that, I prepare the dataset through the following methods. Removal of all null values through the use of the “.drop” function and converting all string values to integer values through the “.replace” function. This allows for better machine learning.

After this, I split the dataset between independent and dependent variables. In this case, we use “smokers” as the dependent variable. I labelled the independent variable as “x” and dependent variable as “y”. The independent variables used are age, sex, BMI, children and chargers. After splitting the two types of variables, I utilised the “train\_test\_split” function to evaluate the model performance, where 80% is for training while 20% is for testing for optimal model performance.

Lastly, I used the decision tree classifier to fit and train the model on the training data. I also limit the “max\_depth” to six for a less complex tree. The model is then used to make predictions and evaluate its performance against the testing data to assess its classification abilities.

(176 Words)

**Question 4**

import matplotlib.pyplot as plt  
from sklearn import plot\_tree

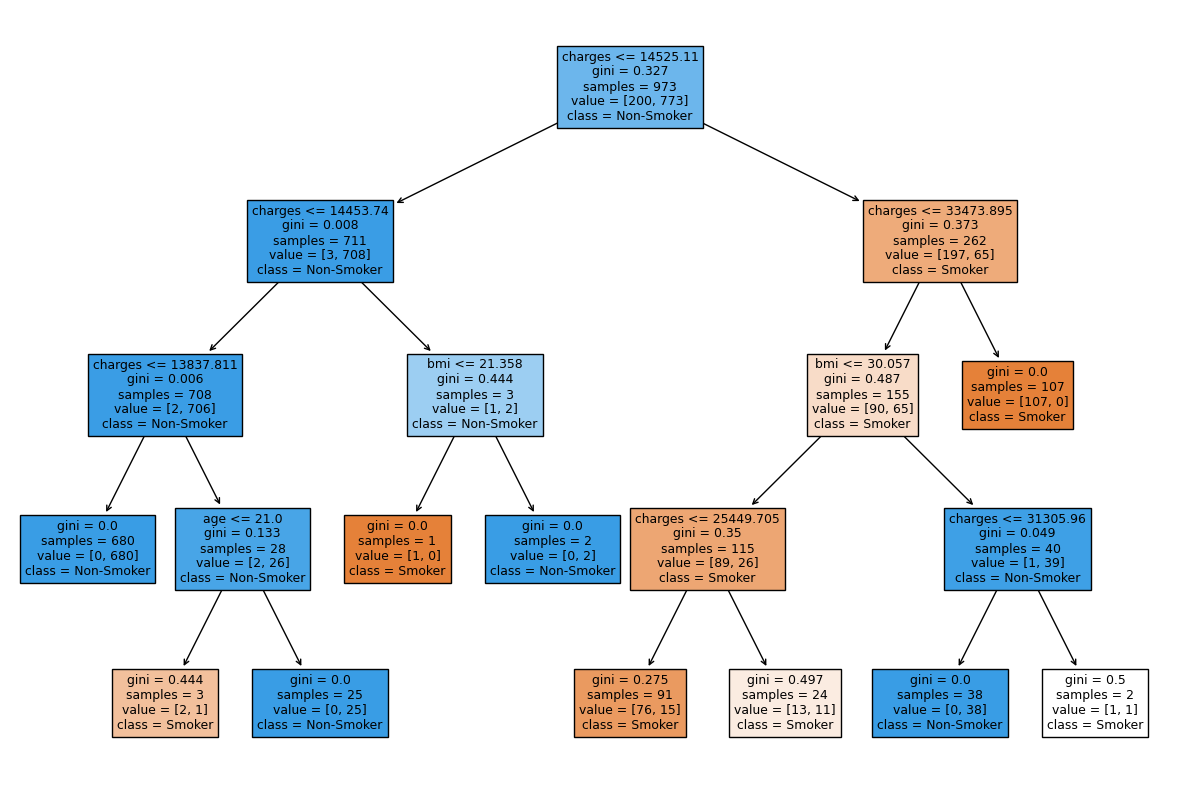
#Control of size

plt.figure(figsize=(15,10))

# plot the decision tree

tree.plot\_tree(dtc, feature\_names=x.columns.tolist(), class\_names=['Smoker', 'Non-Smoker'], filled=True)

plt.show()



The decision tree above is the result obtain from *Question 3*. When studying the decision tree above, I noticed that “charges” seem to be the most important independent variable in predicting if the patient is a smoker or not as it sits at the highest node of the decision tree.

When looking at the two top nodes in the decision tree, I noticed that the threshold for charges for non-smoker is at approximately $14,453 while smokers is at approximately $33,473. This could indicate that smokers usually have a higher medical cost as compared to non-smokers. Furthermore, 680 patients with charges less than or equal to $13,873, are non-smokers.

(108 Words)

**Question 5**

Decision tree in machine learning is mainly used for prediction. However, it can go beyond its predictive capabilities and be used for exploratory data analysis (EDA).

EDA refers to the initial investigation of a dataset. It is where data is pre-processed. Evaluating abnormalities in conducting hypotheses through summary statistics and graphical representations is part of EDA (Patil, 2022). All of these were conducted on the previous questions. Using a decision tree, we can eliminate some of the processes for EDA.

For example, decision tree models require less data cleaning. Particularly, the normalization of data before the machine learning phase as decision tree models can normalize both numerical and categorical variables (Castillo, 2023).

Another example of decision tree models requiring less data cleaning is that the models can handle missing data in the data set. Decision tree models automatically handle missing data during the tree-building process (Kurre, 2023). Thus, there is no need to remove missing data during dataset preparation.

Lastly, decision tree models are easy to interpret. We can quickly determine which variable has the highest impact by looking at the variables on the top nodes. Furthermore, as the nodes splits, we can see the relative importance of each variable.

In conclusion, a decision tree can go beyond its predictive capabilities and use for EDA as it helps eliminate many data pre-processing tasks as well as allows other people to easily interpret their results, thus making them a valuable tool not only for predictive modelling but also for exploratory data analysis.

(251 Words)

**References**

Jaitley, U. (2019, April 9). Why Data Normalization is necessary for Machine Learning models. *Medium*. <https://medium.com/@urvashilluniya/why-data-normalization-is-necessary-for-machine-learning-models-681b65a05029>

Patil, P. (2022, May 30). What is Exploratory Data Analysis? - Towards Data Science. [*Medium*. https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15](Medium.%20https:/towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15)

Castillo, D. (2023, April 7). *Decision trees in machine learning explained*. Seldon. <https://www.seldon.io/decision-trees-in-machine-learning>

Kurre, A. (2023, August 21). *How decision trees handle missing values? @Pickl.AI*. Pickl.AI. <https://www.pickl.ai/blog/how-decision-trees-handle-missing-values-a-comprehensive>