

**ANL252**

**Python for Data Analytics**

**End-of-Course Assignment (ECA- T03)**

**July 2023 Presentation**

Student PI: Z2270006

Name: Ong Lee Zhu

Submission Date: 03 November 2023

Question 1

The main purpose for data pre-processing tasks is to ensure a clean and more useful dataset for the outcome. The first step is to import the libraries such as *import pandas as pd, import numpy as np, random* in Python and read the ECA medical costs dataset (in csv format) by using pd.read\_csv( ) as below.

import pandas as pd

import numpy as np

import random

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

df = pd.DataFrame(df, columns=df.columns)

print(df)

df .head()

df .dtypes

The next task is to identify the missing data value and also categorical data. We are able to find out the missing numerical data/value by using df.isna( ) .sum( ). The output below shows that there are 123 missing values (numerical) for age column. Therefore, we can fixed the missing value data using df.dropna( ) and .isna( ) .sum( ). After that the output of age will be cleared and resolved to zero as below.

df.isna() .sum()

Output:

age 123

datadrop = df .dropna()

datadrop.isna() .sum()

Output:

age 0

Besides that, we can also use df. age. mean( ) to replace the missing data/value by fill up average age (39 years old) as below.

age\_mean = df .age .mean()

print("Mean of age colum: ",age\_mean)

Output: Mean of age colum: 39.19556285949055

df.age.fillna(age\_mean,inplace=True)

df.isna().sum()

For identify the categorical data from dataset, we can use df.select\_dtypes(exclude=[np.number]) to find out which are the categorical data. From the output of categorical\_df.head( ), the column of sex, smoker and region are under categorical data. Therefore, we can search any missing/error date for these three variables by using categorical\_df.sex.value\_counts(), categorical\_df.smoker.value\_counts(), and categorical\_df.region.value\_counts(). The categorical\_df.sex.replace({"male":0, "female":1}).head( ) to use to resolve the error as below.

categorical\_df = df.select\_dtypes(exclude=[np.number])

categorical\_df.head()

categorical\_df.sex.value\_counts()

Output:

F 4

M 2

categorical\_df.smoker.value\_counts()

categorical\_df.region.value\_counts()

df['sex'].unique()

Output:

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

categorical\_df.sex.replace({"male":0,"female":1}).head()

The output above shows that there are four ‘F’ was filled in for female and two ‘M’ for male in dataset. For region and smoker, no error was found.

Slitting the dataset into training and testing dataset by using scikit-learn from sklearn.model\_selection import train\_test\_split to build up the train test split as below.

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2

Question 2

import pandas as pd

import matplotlib.pyplot as plt

import numpy as py

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

df.shape

df.head()

df.dropna(inplace=True)

summary\_df = df.groupby(["age"]) ["charges"] .mean() .reset\_index()

summary\_df = summary\_df.sort\_values(by="charges")

plt.figure(figsize=(13, 2))

plt.bar(summary\_df["age"], summary\_df["charges"], color="yellow")

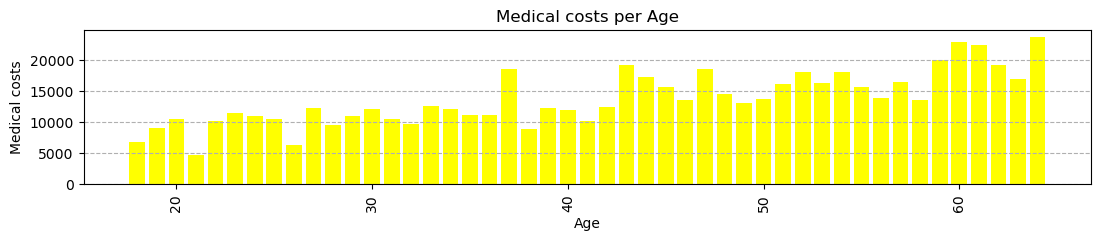
plt.title ("Medical costs per Age")

plt.xlabel ("Age")

plt.ylabel ("Medical costs")

plt.grid(axis="y", linestyle="--", alpha=1)

plt.show()



The chart above represented the relationship between medical costs against the age of patients. From our stereotype, the higher age will be incurred for higher medical costs. It is quite accurate if we refer to the chart for people age from 60 and above. The people age at 37 and 43 years old also incurred quite high medical cost. It may due to other reason like sex, high BMI or for smoker people. Therefore, age will be the most effective reason and factor to causes higher medical costs but other factors will need to take into the consideration.

summary\_df = df.groupby(["smoker"])["charges"] .mean() .reset\_index()

summary\_df = summary\_df.sort\_values(by="charges")

plt.figure(figsize=(13, 2))

plt.bar(summary\_df["smoker"], summary\_df["charges"], color="blue")

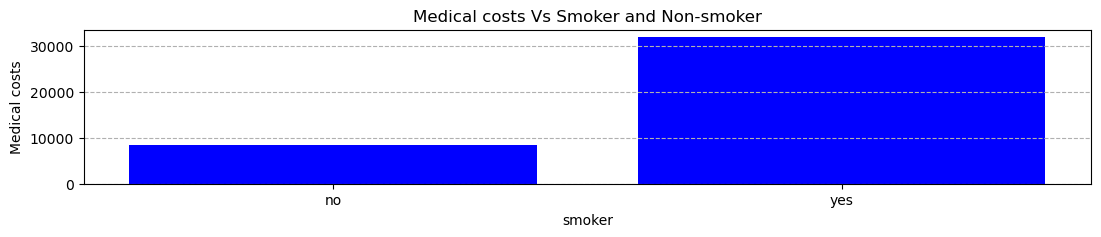
plt.title ("Medical costs Vs Smoker and Non-smoker")

plt.xlabel ("smoker")

plt.ylabel ("Medical costs")

plt.grid(axis="y", linestyle="--", alpha=1)

plt.show()



The chart above shows that the medical costs incurred for smoker patient is significate higher than non-smoker patient. Means that smoke will be one of the reasons effect the medical costs incurred.

summary\_df = df.groupby(["region"]) ["charges"] .mean() .reset\_index()

summary\_df = summary\_df.sort\_values(by="charges")

plt.figure(figsize=(13, 2))

plt.bar(summary\_df["region"], summary\_df["charges"], color="orange")

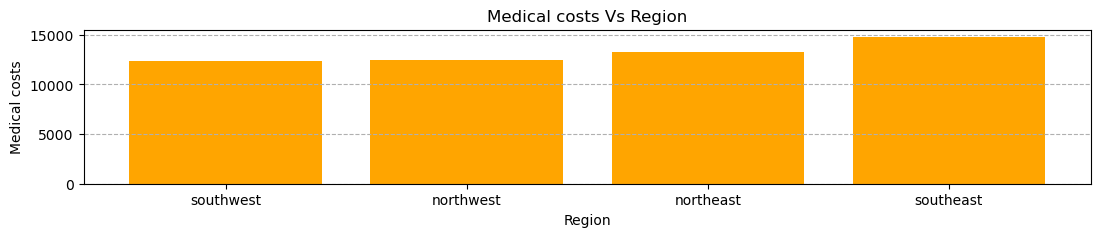
plt.title ("Medical costs Vs Region")

plt.xlabel ("Region")

plt.ylabel ("Medical costs")

plt.grid(axis="y", linestyle="--", alpha=1)

plt.show()



The chart above shows that the relationship between medical costs incurred against region of the patient and it seems insignificant.

Question 3

import numpy as np

import pandas as pd

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

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

from sklearn import tree

eca = pd.read\_csv("Desktop./ECA.csv")

eca.head()

x = eca.values[:, 1:5]

y = eca.values[:, 0]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=100)

clf\_entropy = DecisionTreeClassifier(criterion = "entropy", random\_state = 100, max\_depth=3, min\_samples\_leaf=5)

clf\_entropy.fit(X\_train, y\_train)

y\_pred\_en = clf\_entropy.predict(X\_test)

y-pred\_en

*Reference:*

Preprocessing data tasks:

<https://www.kdnuggets.com/2020/07/easy-guide-data-preprocessing-python.html>

<https://builtin.com/machine-learning/how-to-preprocess-data-python>

<https://realpython.com/python-data-cleaning-numpy-pandas/>

<https://ai-ml-analytics.com/data-preprocessing-data-cleaning-python/>

Plotting:

<https://www.w3schools.com/python/pandas/pandas_plotting.asp>

<https://pandas.pydata.org/docs/getting_started/intro_tutorials/04_plotting.html>

Decision tree:

<https://www.datacamp.com/tutorial/decision-tree-classification-python>

<https://www.simplilearn.com/tutorials/machine-learning-tutorial/decision-tree-in-python>