Logo

Description automatically generated

**ANL 252**

**Python for Data Analytics**

**End of Course Assignment**

**(ECA01)**

Ong Sheng Da  
Z2281575

Prepared for: Professor Kumar Munish

Singapore University of Social Sciences

03 November 2023

Question 1

Before we begin to clean the ECA dataset I will have to determine the issues with the data. The original dataset has 1340 rows and 8 columns. After further investigation, I noticed that there are serval duplicate values in “PersonID”. There were also inconsistent values in “gender” and finally there was missing values in “age”.

The first issue in the dataset was the duplicate entries in “PersonID”. Each value in “PersonID” should be unique, however I found three similar values in “PersonID”. To rectify this issue, I removed the whole entries for all duplicates in the “PersonID” column. Have duplicate entries in the dataset can result in our analysis to be inaccurate as the values will be overstated.

Secondly, there are serval different values in “gender” mainly “f”, “m”, “female” and “male”. This inconsistent method of answering what gender you are makes it difficult to properly analyse the data later. Therefore, I decided to use substitution method to replace “f” and “m” with “female” and “male” respectively. By doing so we are able to have a consistent answer for the gender of the entries.

Finally, there are 123 missing values in the “age” variable. I decided to use the elimination method to remove any entries with missing values. It would be difficult to find a reliable substitute or use regression to get a suitable value. Additionally, since the number of missing values only constitute 9.17% of the total dataset, we can simply eliminate all of them.

After the rectification of the ECA dataset, we have 1,125 records in our dataset.

The code I used to rectify the issues mentioned above is as follows:

import pandas as pd

# Read file

medical\_cost = pd.read\_csv('ECA.csv')

# Remove duplicates

medical\_cost.drop\_duplicates(subset="PersonID", keep='first', inplace=True)

# Replace invalid values

medical\_cost['sex'] = medical\_cost['sex'].replace({'F': 'female', 'M': 'male'})

# Remove all missing values

medical\_cost\_clean = medical\_cost.dropna()

medical\_cost\_clean

Question 2

The first graph I decided to use is a heatmap that depicts the average age, BMI and gender of the data. I decided to group the BMI into 4 different categories namely underweight (< 18.5), normal (18.5 – 22.9), overweight (23.0 – 29.9) and obese (>= 30.0). With reference to the heatmap below, we can determine that as the person’s age increases their BMI is also likely to increase. With people being underweight around the ages of 34 and 33 for females and males respectively and obese people to be around the ages of 41 and 40 for females and males respectively.

A chart with different colors of squares

Description automatically generated with medium confidence

bmi\_category underweight normal overweight obese

sex

female 34.181818 34.964286 39.137168 40.550162

male 33.000000 36.530612 37.592593 40.374269

The code for the heat map is as follows:

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

# Create a new column 'bmi\_category'

bin\_ranges = [0, 18.5, 22.9, 29.9, float('inf')]

bin\_labels = ['underweight', 'normal', 'overweight', 'obese']

medical\_cost['bmi\_category'] = pd.cut(medical\_cost['bmi'], bins=bin\_ranges, labels=bin\_labels)

pivot\_table = medical\_cost\_clean.pivot\_table(values='age', index='sex', columns='bmi\_category', aggfunc='mean')

# Set the labels for x and y axis ticks

plt.xticks(np.arange(len(pivot\_table.columns)), pivot\_table.columns)

plt.yticks(np.arange(len(pivot\_table.index)), pivot\_table.index)

# Create the heatmap using imshow

plt.imshow(pivot\_table, cmap='coolwarm', vmin=pivot\_table.values.min(), vmax=pivot\_table.values.max())

# Display values inside the cells

for i in range(len(pivot\_table.index)):

for j in range(len(pivot\_table.columns)):

text = plt.text(j, i, f'{pivot\_table.values[i, j]:.2f}', ha="center", va="center", color="black")

# Set heatmap title and labels

plt.title('Average Age by Sex and BMI Category')

plt.xlabel('BMI Category')

plt.ylabel('Sex')

# Show the heatmap

plt.colorbar()

plt.show()

print(pivot\_table)

The next graph is a bar chart that depicts the relationship between the person’s average age and the number of children they have. With reference to the bar chart below, the average age range is relatively small the oldest being roughly 42 and youngest being roughly 37. However, what is interesting is that the people with 5 children are roughly 37 years old the youngest of the group. I would have expected people to have more children as the age, but this does not seem to be the case. People with 3 children have the highest average age of roughly 42 years old.

A graph of blue bars

Description automatically generated

children age

0 0 38.385496

1 1 39.560811

2 2 39.490826

3 3 41.500000

4 4 38.045455

5 5 36.529412

The code for the bar graph is as follows:

# Create table for average age and children

avg\_age\_by\_children = medical\_cost\_clean.groupby('children')['age'].mean().reset\_index()

# Create a bar chart

plt.figure(figsize=(8, 6))

plt.bar(avg\_age\_by\_children['children'], avg\_age\_by\_children['age'])

plt.xlabel('Number of Children')

plt.ylabel('Average Age')

plt.title('Average Age by Number of Children')

plt.xticks(avg\_age\_by\_children['children'])

plt.ylim(32, 42)

plt.show()

print(avg\_age\_by\_children)

The final graph is a stacked bar chart which shows the number of smokers and non-smokers in the different region. The number of non-smokers in the different regions are relatively small ranging between 238 and 246 people. As for the number of smokers in the Northeast, Northwest and Southwest have similar numbers, 58,53 and 54 respectively. However. Southeast has the highest number of smokers, 81. This could be due to a variety of reasons such as quality and price of cigarettes in the region. This also allows for more targeted anti-smoking campaigns in the Southeast region to reduce smoking.

A graph of smokers and non-smokers

Description automatically generated

smoker no yes

region

northeast 238 58

northwest 244 53

southeast 246 81

southwest 241 54

The code is as follows:

region\_smoker = medical\_cost\_clean.groupby(['region', 'smoker']).size().unstack(fill\_value=0)

# Create a stacked bar chart

region\_smoker.plot(kind='bar', stacked=True, figsize=(10, 6))

plt.xlabel('Region')

plt.ylabel('Number of Individuals')

plt.title('Number of Smokers and Non-Smokers in Each Region')

plt.legend(title='Smoker', loc='upper right', labels=['Non-Smoker', 'Smoker'])

plt.xticks(rotation=0)

plt.show()

print(region\_smoker)

Question 3

A decision tree is a machine learning algorithm that split a set of observations to reach certain decision points based on some criteria eventually. In this case we will be using a decision tree where the dependent variable or target is smoker and the independent variable are the others (age, sex, bmi\_category, children, region, charges).

Continuing from the previous codes, we will first have to import all the necessary libraries we will be using.

Next, I encoded all the categorical data (sex, smoker, region, bmi\_category) so that I will only have numeric information in my data set. Even though a decision tree algorithm can process categorial data, I wanted the information to be more accurate by changing the values into quantitative data.

Afterward I need to separate the “smoker” variable from the others by creating two lists with only the independent variables and a “smoker”. This way the machine learning algorithm can identify the dependent and independent variables.

Random state controls the randomness of the estimator. I set the random state to 0, because I wanted consistency in my various outputs. However, since I decided to remove the estimator from the code because I wanted more control over the output.

Before I decided on the max depth of the decision tree, attempted to obtain the best depth to get the most accurate output, however the when the decisions tree’s depth becomes to large it because hard to read. Therefore, I set the max depth to be “4”. I also set my criterion to "friedman\_mse" as I was creating a regression tree.

Question 4

The training set has a Mean Square Error (MSE) of 0.02. It can be considered so good that we might need to go back and check for overfitting. The testing set has a MSE of 0.04. Both training and testing sets have similar MSE, this indicates that the decision tree is working well, the slight increase in testing set is normal.

We can tell that the “charges” have the greatest influence on whether the target is a smoker, followed by “children”, “bmi category” and “age”. “sex” and “region” does not appear to have a big impact on whether the person is a smoker.

Looking at the leaves of the decision tree, the one with 85 samples and a value of 1.00 stands out. Tracking back, I can tell that they are 58 years old or older; have 3 or fewer children; and have medical charges more than $33,473.

This shows that a major of smokers will tend to have higher medical charges. Furthermore, those with more children are less likely to smoke. People who are younger tend to also not smoke. It could be due to younger generation perspective of smoking and their awareness of the health implications.

Decision Tree Results on Training and Testing Set

==============================================

MSE for Test set: 0.02

A diagram of a structure

Description automatically generatedMSE for Train set: 0.04

The code is as follows:

import pandas as pd

import numpy as np

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

from sklearn.tree import DecisionTreeRegressor

from sklearn import metrics

import matplotlib.pyplot as plt

from sklearn.tree import plot\_tree

from sklearn.preprocessing import LabelEncoder

medical\_cost\_train\_test\_set = medical\_cost\_clean.copy()

# Encoding data

label\_encoder = LabelEncoder()

medical\_cost\_train\_test\_set['sex'] = label\_encoder.fit\_transform(medical\_cost\_train\_test\_set['sex'])

medical\_cost\_train\_test\_set['smoker'] = label\_encoder.fit\_transform(medical\_cost\_train\_test\_set['smoker'])

medical\_cost\_train\_test\_set['region'] = label\_encoder.fit\_transform(medical\_cost\_train\_test\_set['region'])

medical\_cost\_train\_test\_set['bmi\_category'] = label\_encoder.fit\_transform(medical\_cost\_train\_test\_set['bmi\_category'])

Features = medical\_cost\_train\_test\_set[['age', 'sex', 'children', 'region', 'charges', 'bmi\_category']]

x = Features

y = medical\_cost\_train\_test\_set['smoker']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(

x, y, test\_size=test\_size, random\_state=random\_state

)

# Set the maximum depth to 4

max\_depth = 4

tree = DecisionTreeRegressor(max\_depth=max\_depth, criterion="friedman\_mse")

win\_tree = tree.fit(x\_train, y\_train)

yhat\_train = win\_tree.predict(x\_train)

yhat\_test = win\_tree.predict(x\_test)

print('Decision Tree Results on Training and Testing Set')

print('==============================================\n')

print("MSE for Test set: ", np.round(metrics.mean\_squared\_error(y\_train, yhat\_train), 2))

print("MSE for Train set: ", np.round(metrics.mean\_squared\_error(y\_test, yhat\_test), 2))

X\_label = Features.columns.tolist()

y\_label = ['Non-Smoker', 'Smoker']

plt.figure(figsize=(20, 12))

plot\_tree(win\_tree, feature\_names=X\_label, class\_names=y\_label, filled=True, rounded=True)

plt.show()

Question 5

Decision tree is a machine learning models that is traditionally employed in predictive modelling, especially in classification and regression tasks. It separates data into distinct segments based on a set of feature attributes. Each branch of the tree represents a decision point, and each leaf node corresponds to a prediction (Wu, 2023).

Decision trees are simple to understand, interpret and visualise; little effort is required for data preparation; it can handle both linear and categorical data; and nonlinear parameters do not affect its performance (Lee, 2022)

However, decision trees have its limitations which include overfitting occurs when the algorithm captures noise in the data; the model can become unstable due to small variation in data; a highly complicated decision tree tends to have a low bias which makes it difficult for the model to work with new data (Taylor, 2023).

Exploratory Data Analysis (EDA) is a critical phase in the data analysis process that focuses on understanding and summarizing the key characteristics, patterns, and insights within a dataset (Takyar, 2023). When the dataset is simple and a decision tree will be able to effectively identify features that influence the target giving the user key inside to the data.

However, when the data set is complex the decision trees may not capture the relationships, necessitating the use of ensemble methods. In EDA, overfit decision trees can lead to a misleading interpretation of data patterns. Finally, decision trees may struggle to uncover complex multivariate interactions.

Decision trees could be an effective and simple tool for EDA when the data set is simple. However, when the data set becomes more complexed it is best to utilise other methods. Decision trees may not be the most suitable method in handling complex relationships and overfitting, must be considered. While decision trees offer valuable insights, they are most powerful when integrated into a broader toolkit of EDA techniques, complementing their strengths and mitigating their weaknesses.

Reference

Lee, K. H. C. (2022). *ANL303 Fundamentals of data mining*. Singapore: Singapore

University of Social Sciences.

Takyar, A. (2023, October 19). *Exploratory Data Analysis - A Comprehensive Guide to EDA*. LeewayHertz - AI Development Company. https://www.leewayhertz.com/what-is-exploratory-data-analysis/

Taylor, S. (2023, October 6). *Decision tree*. Corporate Finance Institute. https://corporatefinanceinstitute.com/resources/data-science/decision-tree/

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

Social Sciences.