A picture containing logo

Description automatically generated

## ANL252 ECA

|  |  |
| --- | --- |
| **STUDENT NAME:** | Angela Liu Qiyu |
| **P.I. NO.:** | J2181353 |
| **TITLE:** | ECA |
| **COURSE CODE:** | ANL252 |
| **TG:** | T03 |
| **SUBMISSION DATE:** | 3rd November 2023 |

**Plagiarism and Collusion**

**Plagiarism** is the act of using or passing as one’s own, the ideas or writings of another without acknowledging or crediting the source from which the ideas are taken from.

**Collusion** is the act of submitting any academic work (including assignment, project or report) that was completed by another person and pass these work off as one’s own.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Declaration** | | | | |
|  | | | | |
| I declare that this assignment is my own work, unless otherwise acknowledge or credited by appropriate referencing. I have read and abide by the SUSS Honour Code and I am aware of the penalties associated with plagiarism and collusion listed in the SUSS Student Handbook. | | | | |
|  | | | | |
| Initial: | Angela Liu |  | Date: | 1 Nov 2023 |

**Question 1**

Before we conduct data pre-processing tasks, we need to import the excel file.

import pandas as pd

#Load Medical cost dataset

data = pd.read\_csv('Medical\_cost.csv')

The first way to use Python to conduct a data pre-processing task to clean and prepare the dataset is to deal with the missing data by checking for them and replacing them with a specific mean value calculated from the rest of the data in the dataset.

#Check for missing values

missing\_values = data.isnull().sum()

#Filling the missing values in ‘age’ column with a specific mean value

data['age'].fillna(data['age'].mean(), inplace=True)

The second way to use Python to conduct a data pre-processing task is to use scaling. Numeric features with different scales can impact the performance of Python. Scaling ensures that all features contribute equally. In this case, Scikit-Learn and its MinMaxScaler will be imported to scale. Afterwards, we will define scaler. MinMaxScaler will then transform the values under the ‘age’ and ‘charges’ column to a range of [0, 1].

from sklearn.preprocessing import MinMaxScaler

# Using Min-Max scaling

scaler = MinMaxScaler()

data[['age','charges']] = scaler.fit\_transform(data[['age','charges']])

The third way to use Python to conduct a data pre-processing task is to use lambda to replace the values for standardisation. We can utilize the apply() function to employ the lambda function on both columns of the dataset. Since the column indicated was 'sex', the function gets applied to each row. Under the ‘sex’ column, there are 6 values that are recorded differently from the rest, which are recorded as ‘F’ and ‘M’ instead of ‘female’ and ‘male’. Standardisation of how the sex of the patient is recorded is required. Lambda with apply will go through every row in the ‘sex’ column. If the value is ‘F’, it will be replaced with ‘female’. If the value is not ‘F’, Lambda will check if the value is ‘M’. If the value is ‘M’, it will be replaced with ‘male’. Otherwise, Lambda will return the original value which was defined as x.

#Replace F with female and M with male

data['sex'] = data['sex'].apply(lambda x: 'female' if x =='F' else ( 'male' if x=='M' else x))

(349 words)

**Question 2**

Bar chart: relationship between bmi and charges.

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

#Load dataframe

data2 = pd.read\_csv('Medical\_cost.csv')

data2['sex'] = data2['sex'].apply(lambda x: 'female' if x=='F' else ('male' if x =='M' else x))

#Bin for bmi and labels for each category

data2['bmi\_bin'] = pd.cut(data2['bmi'], [10, 20, 30, 40, 50, 60], labels = ['10-20', '20-30', '30-40', '40-50', '50-60'])

#Pivot dataframe so there is column for each gender, each row represents a bmi\_bin, and the cells have mean charges for bmi\_bin and gender

data2\_pivot = pd.pivot\_table(data2, values = 'charges', index = 'bmi\_bin', columns = 'sex', aggfunc = np.mean)

#Plot bar chart

ax = data2\_pivot.plot(kind = 'bar')

#Get Matplotlib figure from the axis object for formatting

fig = ax.get\_figure()

#Change plot dimensions

fig.set\_size\_inches(7,6)

#Change axis labels

ax.set\_xlabel('BMI')

ax.set\_ylabel('Average Medical Cost')

A graph of different colored bars

Description automatically generated with medium confidence

From here, it can be seen that below 30, the average medical charges for male and female are similar, however, as the age increases from 30 to 50, there is a significant increase in the charges for males, compared to that of the females. For ages 50 to 60, only males went to the hospital. This might indicate that for younger people, the average medical charges generally stay the same for both female and male as the health problems in both genders tends to be similar, however, as the age increases, males have a higher charge than females as older males’ health problems might vary from older females.

Boxplot: relationship between number of children and charges.

import pandas as pd

#Load dataframe

data2 = pd.read\_csv('Medical\_cost.csv')

#Import libraries

import matplotlib.pyplot as plt

import numpy as np

#Defining individual arrays with different number of children

nda\_c0 = data2.loc[data2['children'] == 0, 'charges'].values

nda\_c1 = data2.loc[data2['children'] == 1, 'charges'].values

nda\_c2 = data2.loc[data2['children'] == 2, 'charges'].values

nda\_c3 = data2.loc[data2['children'] == 3, 'charges'].values

nda\_c4 = data2.loc[data2['children'] == 4, 'charges'].values

nda\_c5 = data2.loc[data2['children'] == 5, 'charges'].values

#Plot boxplot with labels.

plt.boxplot([nda\_c0, nda\_c1, nda\_c2, nda\_c3, nda\_c4, nda\_c5], labels = ['c\_0', 'c\_1', 'c\_2', 'c\_3', 'c\_4', 'c\_5'])

A graph of a number of boxes

Description automatically generated with medium confidence

From this, median values of medical charges across people from 0 to 5 children stays around the same. However, as the number of children increase, minimum medical charges increase, as usually people with more children are of a higher age, therefore, medical costs are higher as medical costs associated with aging are higher. However, people with 2 and 3 children have the highest medical cost and people with no children have the most outliers mostly due to the large base.

Scatterplot: relationship between age and charges.

import pandas as pd

#Load dataframe

data2 = pd.read\_csv('Medical\_cost.csv')

#Import libraries

import matplotlib.pyplot as plt

import numpy as np

plt.scatter(data2.age,data2.charges)

plt.title('Medical charges and Age')

plt.xlabel('Age')

plt.ylabel('Medical charges')

A graph of blue dots

Description automatically generated

From this, generally, as age increases, medical costs increase. The different sections in the scatterplot might indicate different health problems. The bottom group may include mostly outpatients, whereas the middle group may include inpatients with short hospital stays, without surgery. The upper group might be inpatients with fatal injuries with more complicated surgeries for example for heart failures.

(500 words)

**Question 3**

In [1]: import pandas as pd

import numpy as np

In [2]: #Importing dataframe.

df = pd.read\_csv('Medical\_cost.csv')

df.head()

Out[2]: A screenshot of a graph

Description automatically generated

In [3]: # Splitting the data set into data and predicted.

y = df['smoker'].astype('category')

X = df.drop('smoker',axis = 1)

X = X.drop('PersonID',axis = 1)

X['age'].fillna(X['age'].mean(), inplace=True)

X.head()

Out[3]: A screenshot of a table

Description automatically generated

In [4]: #Converting the object datatype to category

X['sex'] = X['sex'].astype('category')

X['region'] = X['region'].astype('category')

#Standardise the representation of male and females.

X['sex'].replace('F', 'female', inplace=True)

X['sex'].replace('M', 'male', inplace=True)

X['sex'].unique()

Out[4]: 

In [5]: #Encoding Categorical data

X = pd.get\_dummies(X, columns = ['sex','region'])

In [6]: X.head()

Out[6]: A table with numbers and letters

Description automatically generated

In [7]: #Splitting data

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=1, test\_size=.3)

In [8]: from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()

In [9]: #Fitting model

model.fit(X\_train,y\_train)

In [10]: #Testing accuracy, precision, and recall

from sklearn.metrics import accuracy\_score, precision\_recall\_fscore\_support

pred = model.predict(X\_test)

print(accuracy\_score(pred,y\_test))

print(precision\_recall\_fscore\_support(pred,y\_test))



In [11]: import matplotlib.pyplot as plt

fig = plt.figure(figsize = (35,25))

from sklearn.tree import plot\_tree

a = plot\_tree(model, fontsize = 13)

Since dependent variable is ‘smoker’, I remove ‘smoker’ to separate the data used to predict ‘smoker’ and the predicted ‘smoker’. Since ‘PersonID’ does not affect the results, ‘PersonID’ is dropped. Next, I converted the object datatype of ‘sex’ and ‘region’ into category as they are non-numeric, followed by standardisation of the representation by replacing ‘F’ and ‘M’ with ‘female’ and ‘male’ respectively. Afterwards, I encode the categorical data, so the machine is able to understand and extract values followed by splitting the data into train and tests sets where their variables will be used as input. Then, I import the DecisionTreeClassifier to use all the values calculated for prediction. Next, I import precision\_recall\_fscore\_support to return the individual values for the array. Lastly, I resize the Decision Tree by using Matplotlib and import plot\_tree to show the Decision Tree and the predictions of ‘smoker’.

(300 words)

**Question 4**

A diagram of a network

Description automatically generated

402 samples are used to train the machine and 938 samples are used to test. For medical charges less than or equals to $14,525.11, if the patient’s medical charge is lesser than $14,525.11, move left, else, move right. Gini shows that there is 32.4% that the patient would have a medical charge of more than $14,525.11, hence moving right. Focusing on the left side where the patient has a medical charge of less than $14,525.11, there is 1.2% in 686 samples that he has a bmi smaller than 21.727. If he has a bmi larger than 21.727 in 633 samples, it would be concluded that he is not a smoker as gini is 0.0. If he has a bmi smaller than 21.727, move to the left side. In 53 samples, there is 14% that he has a medical charge more than $12,652.40. If he has a medical charge of more than $12,625.40, move left and conclude that he is not a smoker. Otherwise, move to the right where there is a 50% chance in 8 people that he is a smoker. If he is younger than 50.098 years old, he is a smoker, else he is not a smoker.

(199 words)

**Question 5**

Yes, decision trees can be effectively used for exploratory data analysis beyond their traditional role in making predictions. Decision trees offer an efficient approach for decision-making by presenting a structured depiction of the problem and its various potential outcomes, which enables developers to consider the possible consequences of a decision, and as the algorithm becomes more familiar with extra data, it gains the ability to predict results for future data points. (Coursera, 2023).

Decision trees are usually known for classification and regression, they offer several advantages such as identifying the important features in a dataframe, simple visualisation of the distribution of data and show how different variables impact the dependent variables.

Firstly, identifying the important features in a dataframe. Decision trees can show us which parts of the data are the most important by how the tree divides into different sections. This helps you see which factors have the most impact on what we are trying to predict, and it reveals possible connections and patterns in the data.

Next, decision trees help to provide a simple visualisation of how the data is being distributed. The divisions, paths and nodes in the tree shows how data is separated based on different characteristics, showing us how data is spread and how it changes. It also shows us how different variables affects the dependent variable. Complex tree branches and deep divisions hint at strong impacts from certain factor combinations, which requires us to discover the hidden data relationships and patterns.

In conclusion, decision trees extend beyond predictive modelling and serve as versatile instruments for exploratory data analysis (EDA). They provide valuable insights into the significance of features and clear visualization of pattern. Leveraging decision trees for EDA empowers analysts to attain a profound comprehension of their data, rendering it a valuable approach in data exploration.

(300 words)

**Bibliography:**

Coursera. Oct 30, 2023. Decision Trees in Machine Learning: Two Types (+ Examples). Retrieved from Coursera: <https://www.coursera.org/articles/decision-tree-machine-learning>