

**ANL252 (Online)**

**PYTHON FOR DATA ANALYTICS**

**End of Course Assessment**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **DARRION GOH ING HEAN** | **E2310655** |

**Tutorial Group: ­­­­­­­­­­­­ T 03**

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

**Submission Date: 31/10/2023**

Q1)

Firstly, the most fundamental step is we would need to import libraries and load the dataset into the python codes at the beginning. Libraries like numpy and pandas provide a huge range of functions and data structures that make it easier to work and manipulate data. It also helps to format our code effectively and efficiently. (Hjelle, n.d.) Without importing these libraries, we would not have access to functions like ‘read.csv’ for data loading, data cleaning and data transformations. We are also able to reuse these for data analysis and we do not need to reinvent the wheel every time we work with the data.

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

from scipy import stats df\_medical = pd.read\_csv(r"C:\SUSS Work\ANL252\ECA\ECA.csv")

df\_medical

Next, we need to drop missing or null data and columns that are not useful. Having missing or null data is a very common thing that happen in a dataset. By removing those data, we can have a more accurate and unbiased results in our analysis. This is to allow us to have better consistency with our data as missing values could result in disrupting the uniformity of our data, making it harder for us to draw conclusions. By dropping not useful columns it makes it easier for us to focus on the remaining columns without any confusion.

df\_medical.dropna(inplace=True)

cols = ['region']

df\_medical = df\_medical.drop(cols, axis=1)

df\_medical

Lastly, there is a row, ‘sex’, which is supposed to be either male or female. However, there are different values compared to the rest. There are some data in that column which has ‘M’ and ‘F’, which is different than the values of ‘male’ and ‘female’. Hence, by replacing those with ‘male’ and ‘female’ values with ‘M’ and ‘F’, it clears up the confusion and the data will be more consistent. This will prevent the data to have a different groups of data that will fall under differently for ‘M’ and ‘male’ and the same for ‘F’ and ‘female’.

df\_medical['sex']=df\_medical['sex'].replace(['male', 'female'], ['M','F'])

df\_medical

(How to Replace Values in Pandas DataFrame, 2022)

(302 words)

Q2)

A red and blue dots

Description automatically generated

The chart above shows the medical charges by each person’s Body Mass Index (BMI) value and whether he or she is a smoker. We want to find out if the BMI value and smoker status affects the medical charges for each person. From the scatter plot, the red dots indicate that they are smokers and blue dots indicate that they are non-smokers. We can see that there are some outliers at the bottom right and top right of the graph. This scatter plot shows that even with high BMI values, it does not mean that the medical charges will be high as well, as we can see the charge for BMI above 50 to be one of the lowest. However, we can see that if the he or she is a smoker, the medical charges will be significantly higher than those that are non-smokers. Hence, we can conclude that the BMI values does not affect the medical charges but the smoking status of the person will affect the medical charges.

(170 words)

smoker\_yes = df\_medical[df\_medical['smoker'] == 'yes']

smoker\_no = df\_medical[df\_medical['smoker'] == 'no']

plt.figure(figsize=(10, 6))

plt.scatter(smoker\_yes['bmi'], smoker\_yes['charges'], c='red', label='Smoker: Yes', alpha=0.5)

plt.scatter(smoker\_no['bmi'], smoker\_no['charges'], c='blue', label='Smoker: No', alpha=0.5)

plt.title("Scatter Plot of Medical Costs by BMI")

plt.xlabel("BMI")

plt.ylabel("Medical Costs ($)")

plt.legend()

plt.show()

A graph with red and blue rectangles

Description automatically generated

The chart above shows the average medical charges of each gender, male and female, by their smoking status. The reason for doing this graph is because we want to see if males or females’ medical charges will be higher and if the smoking status of the person will result in higher medical charges. From the graph, the red bar will be smokers and the blue bar will be non-smokers. We can observe that the average medical charges for male and currently smoking is higher ($33091.66) than that of the average charges for females and smoking ($30598.03). However, we can see that for male and non-smoker, the average charges is $8194.33, which is lower than the females and non-smoker at $8720.38. Hence, we can conclude that if it is male and smoker, the medical charges will be higher than females and smoker, but the charges will be lower if it is male and non-smoker, compared to female and non-smoker.

(158 words)

male\_smoker\_yes = df\_medical[(df\_medical['sex'] == 'M') & (df\_medical['smoker'] == 'yes')]

male\_smoker\_no = df\_medical[(df\_medical['sex'] == 'M') & (df\_medical['smoker'] == 'no')]

female\_smoker\_yes = df\_medical[(df\_medical['sex'] == 'F') & (df\_medical['smoker'] == 'yes')]

female\_smoker\_no = df\_medical[(df\_medical['sex'] == 'F') & (df\_medical['smoker'] == 'no')]

plt.figure(figsize=(10, 6))

bar\_width = 0.2

bar1 = plt.bar([1, 2], [male\_smoker\_yes['charges'].mean(), female\_smoker\_yes['charges'].mean()], bar\_width, label='Smoker: Yes', color='r')

bar2 = plt.bar([1 + bar\_width, 2 + bar\_width], [male\_smoker\_no['charges'].mean(), female\_smoker\_no['charges'].mean()], bar\_width, label='Smoker: No', color='b')

plt.xlabel('Gender')

plt.ylabel('Average Medical Costs ($)')

plt.title('Average Medical Costs by Gender and Smoking Status')

plt.xticks([1.1, 2.1], ['Male', 'Female'])

plt.legend()

# Add values above the bars

for bar in [bar1, bar2]:

for rect in bar:

height = rect.get\_height()

plt.text(

rect.get\_x() + rect.get\_width() / 2, # x-coordinate of the value

height + 5, # y-coordinate of the value (adjusted for visibility)

f'{height:.2f}', # value to display (formatted to two decimal places)

ha='center', # horizontal alignment

va='bottom' # vertical alignment

)

plt.show()

A graph showing the average medical costs

Description automatically generated

The chart above shows the trend for the average medical charges by age and smoking status. We would want to see whether if the age increases, will the medical charges increase as well. From the chart, we can observe that there is a rising trend in medical charges when the age increases as well. Even though there were are few dips in medical charges, ultimately, the charges increases by age. The dip in medical charges could be due to other factors like health conditions. We could also observe that the average medical charges for non-smokers are significantly lesser compared to smokers. Hence, we can conclude that the higher the age, the higher the medical charges.

(115 words)

grouped = df\_medical.groupby(['age', 'smoker'])['charges'].mean().unstack().reset\_index()

plt.figure(figsize=(10, 6))

# Plot the line for 'smoker' == 'yes'

plt.plot(grouped['age'], grouped['yes'], marker='o', label='Smoker: Yes', color='b')

# Plot the line for 'smoker' == 'no'

plt.plot(grouped['age'], grouped['no'], marker='o', label='Smoker: No', color='r')

plt.title("Average Medical Costs by Age with Smoking Status")

plt.xlabel("Age")

plt.ylabel("Average Medical Costs")

plt.legend()

plt.grid(True)

plt.show()

Q3)

Accuracy: 0.9631147540983607

precision recall f1-score support

0 0.92 0.90 0.91 52

1 0.97 0.98 0.98 192

accuracy 0.96 244

macro avg 0.95 0.94 0.94 244

weighted avg 0.96 0.96 0.96 244

From the information above, we can observe that the accuracy for the prediction from the decision tree is approximately 0.96. The f1-score for both 0 and 1, which is ‘yes’ and ‘no’ under ‘smoker’ column respectively, is 0.91 and 0.98. This shows that the accuracy results for both is considered very high and does produce errors regularly. Next, we can observe that the precision of both values is 0.92 and 0.97 and thus we can say that the data can be useful. The recall rate for smoking status predicted and results rate is the same are high with 0.90 and 0.98. Therefore, we can say that the decision tree prediction results is accurate.

(113 words)

import pandas as pd

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

from sklearn.tree import DecisionTreeClassifier

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import accuracy\_score, classification\_report

X = df\_medical[[

'bmi',

'age',

'sex',

'charges',

]]

y = df\_medical['smoker']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create a decision tree classifier

clf = DecisionTreeClassifier()

# Fit the model on the training data

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

# Predict "smoker" on the test data

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

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

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

Q4)

A diagram of a diagram

Description automatically generated

From the decision tree diagram above, we can observe that it is split into two different child nodes. The parent node is first distinguished by the medical charges. It splits if the charges is lower than $15716.446. It will then split to the Body Mass Index (BMI) value than afterwards, it will be the age. We can see that most people are charges with below $15716.446 that are currently not smoking. However, on the right hand side of the decision tree, we can observe that when the charge is higher, the chances of the person being a smoker is also higher. Hence, we can make a conclusion that if the person is a smoker, the medical charges would be higher as well.

(122 words)

# Plot the decision tree

class\_names = ['Yes', 'No']

plt.figure(figsize=(15, 10)) # Adjust the figure size as needed

plot\_tree(clf, filled=True, feature\_names=X.columns.tolist(), class\_names=class\_names, fontsize = 10, rounded = False, impurity = False, max\_depth = 5)

plt.title("Decision Tree for 'Smoker' Prediction")

plt.show()

Q5)

Decision tree can be effective for the use for exploratory data analysis.

Firstly, the decision tree can be used to predict results and outcomes even in the presence of missing data. It is possible to predict the outcome with Classification & Regression Tree (CART) even when there are missing values as it relies on the most significant number of instances. (Duggal, 2023) In addition, we can be randomly assigned a node in the decision tree. In this way, it will continue to configure beyond the missing values and the algorithm won't halt.

However, there will be some ineffectiveness for exploratory data analysis.

Overfitting is a huge issue in the modern days. Decision tree is one of the most difficult problems to overcome overfitting. When overfitting occurs, there is a high chance that there is a high variance in output which can result in many errors in the final estimation, and it will show inaccuracy in the results. However, it can be overcome by setting restraints on the parameters. (Decision Tree Advantages and Disadvantages, 2023)

Nonetheless, I would say that decision tree will be effective for the use of exploratory data analysis even though there are some small issues. Decision tree can help to solve missing values and input prediction data while we do not need to do much data cleaning. It would move beyond the traditional roles in making predictions.

(230 words)

References:

Geir Arne Hjelle. (n.d.). *Python import: Advanced Techniques and Tips.* <https://realpython.com/python-import/#:~:text=In%20Python%2C%20you%20use%20the,while%20keeping%20your%20projects%20maintainable.>

*How to Replace Values in Pandas DataFrame.* (October 22, 2022). <https://datatofish.com/replace-values-pandas-dataframe/>

Nikita Duggal. (February 20, 2023). *Advantages of Decision Trees.* <https://www.simplilearn.com/advantages-of-decision-tree-article>

*Decision Tree Advantages and Disadvantages.* (March 10, 2023). <https://www.educba.com/decision-tree-advantages-and-disadvantages/>