

**ANL252**

**Python for Data Analytics**

# **End-of-Course Assessment**

**July 2023 Report**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Nur Hamizah Binte Ridwan** | M2311185 |

**Tutorial Group: ­­­­­­­­­­­­ T03**

**Instructor’s Name: Dr. Munish Kumar**

**Submission Date: 02/11/2023**

**Question 1**

The three pre-processing tasks are as follows: exploratory data analysis, data cleaning and data preparation. It will be elaborated below.

1. **Exploratory data analysis**

**#**Importing of libraries needed for this report  
import pandas as pd  
import numpy as np  
from sklearn.tree import DecisionTreeClassifier  
from sklearn.model\_selection import train\_test\_split  
from sklearn.tree import plot\_tree  
from sklearn.model\_selection import GridSearchCV  
from sklearn import tree  
import plotly.express as px

#Inserting dataset  
df = pd.read\_csv(‘ECA.csv’)  
df = df.head()

#Finding out data types, null values, shape, content of columns (categorical columns)

df.info()

df.shape

df[‘region’].unique()

df[‘smoker’].unique()

df[‘sex’].unique() #naming conventions need to be standardized in step 2.

1. **Data cleaning**

#Handling missing data  
df.isnull().sum()

There are 123 missing null values that were found in ‘age’ column, as it’s less than 10% of the total dataset of 1300+ rows. I have decided to drop the column with df\_new = df.dropna() to create a new Dataframe and not distort the original dataframe.

#Handling data errors  
In the exploratory data analysis, for the ‘sex’ column there was ‘F’, ‘M’, ‘female’ and ‘male’. I’ve decided to shorten it to ‘F’ and ‘M’ instead.

df\_new['sex\_short'] = df\_new["sex"].replace({"male": "M", "female": "F"})

df\_new['sex\_short'].unique() *#check if the column has been updated.*

#Handling outliers (CF Blog, 2023)

df\_new.describe()

After running the above code, the mean, and the max for ‘charges’ was noticeably rather large. A boxplot was created to visualize suspected outliers with a boxplot by plotly library according to region.

fig = px.box(df\_new, x='region', y="charges", color='region')

fig.show()

A graph with different colored squares

Description automatically generated  
Figure 1. Finding outliers in ‘charges’ column according to region

As suspected, there are too many outliers in said dataset, to remove it will render it not usable for analysis, therefore I decided to leave it and continue with the modelling.

1. **Preparing the data by transforming it to make it usable for data modeling.**

#Encoding categorical data

# Change sex to int

df\_new['sex\_short'] = df\_new['sex\_short'].map({'M':0,

'F':1})

# Change region to int

df\_new['region'] = df\_new['region'].map({'southwest':0,

'southeast':1,

'northwest':2,

'northeast':3})

# Change smoker to int

df\_new['smoker'] = df\_new['smoker'].map({'no':0,

'yes':1})

There are a few categorical columns in this data set and when the dataset is used for modelling it needs to be in a numerical form by encoding it, there are a few methods to do so, but this method is called ‘mapping’, this will allow the data to be fed into the model to give the predictions.

(389 words)

**Question 2**

The figures below will reflect what the charges are based on and how it impacts the individuals from their region, gender, BMI and if they are a smoker and how many dependents they have.

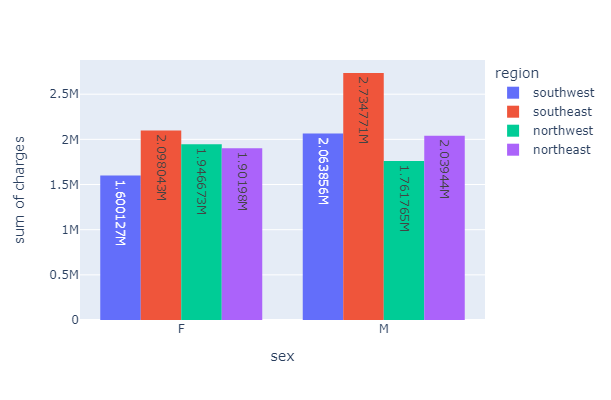


Figure 2. Distribution of demographic based on the charges according to region in the dataset.

fig = px.histogram(df\_new, x="sex\_short", y="charges",

color='region', barmode='group', text\_auto=True,

height=400, labels={

"sex\_short": "sex"})

fig.show()

For figure 2 (above), the different bars represent the different regions while the x axis is the sex of the insurance contractor while the y axis is the sum of charges. As we can deduce from the following, in most regions the males are charged more by the insurance companies except for the northwest region. However while checking the sample size for both with df\_new['sex\_short'].value\_counts(), the sample size for male is more by 13, this could affect the sum of charges. To rectify this we can use a statistical method or reduce the male sample size to make it the same. However which rows to remove is uncertain, as the outliers are too many.

A graph showing a line and a line

Description automatically generated with medium confidence

Figure 3. Trend of being charged more if individual has a high BMI and who smokes.

fig = px.scatter(df\_new, x="bmi", y="charges", color="smoker", trendline="ols")

fig.show()

For Figure 3 (above), insurance companies would relatively charge smokers more along with a high BMI. This individuals have the risks of developing cancer and chronic diseases in the future requiring the use of their insurance. Therefore, the trend makes sense despite some outliers on the chart. Non-smokers with high BMI generally pay similar amounts to non smokers with low BMI individuals, this can be seen from the horizontal line in red. The overlapping of smoker and non-smokers are very few which means all the regions practice this way of evaluation for charging their insurance buyers.

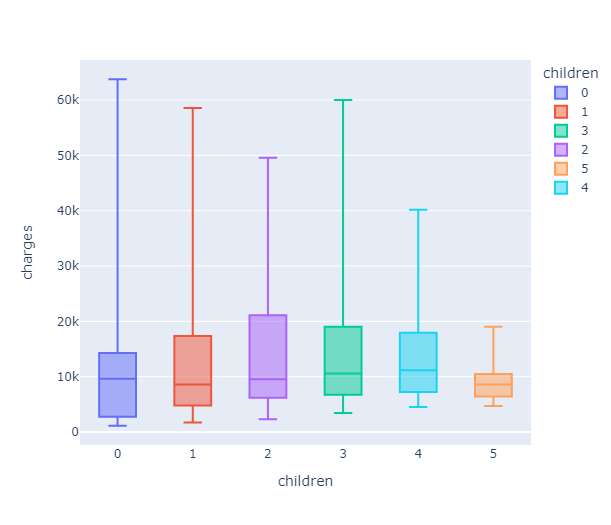


Figure 4. Variability of charges based on the number of children the insurance contractor has.

fig = px.box(df\_new, x='children', y="charges", color='children', points = False)

fig.show()

For Figure 4 (above), insurance companies would usually charge more for individuals who have more dependents, in this case it is their children. Individuals with 2 children see varied number of charges due to the size of the box in the boxplot. However individuals with no children are charged lower than individuals with children but their charges can very varied as well due to the outliers in this case its the upper whisker. (Jim Frost, n.d.) (398 words)

**Question 3**

With any type of modelling, we need to select the features that we are considering for the dependent, y which is ‘smoker.

features = ['age', 'sex\_short', 'bmi', 'children', 'region', 'charges']

X = df\_new[features]

y = df\_new['smoker']

print(X)

print(y)

#Error message when running model

df\_new = df\_new[np.isfinite(df\_new).all(1)]

This line input is to counter the error message after running the model where there’s a ValueError: ValueError: Input contains NaN, infinity or a value too large for dtype('float32’)

#Split for train and test set

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

model = DecisionTreeClassifier(random\_state = 42)

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

#Evaluation of model

print(f'Score on training set: {model.score(X\_train, y\_train)}')

print(f'Score on testing set: {model.score(X\_test, y\_test)}')

#Using GridSearch to find the best estimator

GS = GridSearchCV(estimator = DecisionTreeClassifier(),

param\_grid = {'max\_depth': [2, 3, 5],

'min\_samples\_split': [5, 10, 15],

'min\_samples\_leaf': [2, 3, 4, 5],

'ccp\_alpha': [0, 0.001, 0.01, 0.1, 1, 10]}, verbose = 1)

# import time library  
# Start our timer.

import time

tt = time.time()

# Gridsearch the conditions above for training

grid\_fit = GS.fit(X\_train, y\_train)

# Stop the clock to show outcome.  
print(time.time() - tt)

# Determining the best estimator for the tree  
GS.best\_estimator

Out[]: DecisionTreeClassifier(ccp\_alpha=0, max\_depth=3, min\_samples\_leaf=2,min\_samples\_split=5)

**(**199 words)

**Question 4**

A diagram of a mathematical system

Description automatically generated with medium confidence

Figure 5. Plot of Decision Tree Classifier model. The tree best estimator for maximum depth is 3.

# Use the plot\_tree library from sklearn.tree

#Plot decision tree

plt.figure(figsize = (40, 20))

plot\_tree(GS.best\_estimator,feature\_names = X\_train.columns, filled = True, class\_names =['Non Smoker', ‘Smoker’]);  
  
Based on the above plot, the colours in the tree represents that the darker it is the lower the gini impurity means better split (KDnuggets, 2022).

All the charges <= 14453.74 goes to the left while the charges > 14453.74 goes to the right.

Gini is the score for the node as mentioned, the lower the Gini, gives better results.

The samples are number of samples used in the node.

The values represent number of samples based non smoker and smoker, non smoker comes first as according to mapping. Therefore, at the root, there are 729 non smokers while there are 183 smokers.

The class shows which it represents more off, at the root, the majority is non smoker.

Finally, there are the leaves and nodes, non smoker is in orange while smoker is in blue.

As we can see the node and leaf are more biased towards non smoker as the Gini impurity is lower.

**Question 5**

Yes. Decision trees are not just useful for predicting models; they can also be used in exploratory data analysis (EDA).

Decision trees depict how data can be divided into distinct groups or outcomes based on the variables provided. This structure helps in understanding the underlying patterns and relationships in the data, making it easier to find errors on anomalies that could be missed in the initial pre-processing during EDA (exploratory.io, 2022).

Furthermore, decision trees can be interpreted. They make it easier to communicate your findings to non-technical audiences (Simplilearn, 2023), by providing a clearer and simple way to describe the elements and variables that have the most impactful on the data.

By identifying fewer common branches or leaves in the tree, decision trees can also help detect outliers and abnormalities in data. This can be critical for EDA quality control and data cleaning.

It also aids in feature selection which is helpful to rid of redundant data. Examining the depth of different variables in the tree might help you focus on the most important features of your data during analysis.

To summarize, decision trees can be included as one of the EDA tools on top of the other conventional ways that allows you to analyze and interpret the data more effectively.

(211 words)

References:

Jim, Frost. (n.d.) Box Plot Explained with Examples | Statistics By Jim. Retrieved from <https://statisticsbyjim.com/graphs/box-plot/>

Nagesh Singh, C. (2022, February 9). Decision Tree Algorithm, Explained | KDnuggets. Retrieved from <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html>

Kei, Saito. (2020, February 22). Building Decision Tree and Visualize in Exploratory | Exploratory.io. Retrieved from <https://exploratory.io/note/kei/Visualizing-a-decision-tree-using-R-packages-7185899605672132>

Learning, M. (2022, June 13). Advantages of Decision Trees: Everything You Need to Know | Simplilearn. Retrieved from <https://www.simplilearn.com/advantages-of-decision-tree-article>