

**ANL252**

**Python for Data Analytics**

# **End-of-Course Assessment**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **BEVERLYN TAN JIAMIN** | **E2081428** |

**Tutorial Group: ­­­­­­­­­­­­ T03**

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

**Submission Date: 03/11/2023**

**Question 1**

The data pre-processing tasks revolve around the standardising data, imputing missing values and encoding of categorical values. For data standardisation; this does not refer to scaling or normalising numerical data, but ensuring duplicate values are removed and values which look different yet refer to the same outcome are homogenised. This is done for the “sex” and “PersonID” columns in the data-frame. When the unique values of the “sex” column are accessed using the code

print('Unique values of "sex" column:',', '.join(df['sex'].unique()))

4 unique values are produced:

Unique values of "sex" column: female, male, F, M

However, as the appendix notes, the sex column only has 2 distinct values. Thus, it can be inferred that ‘F’ and ‘M’ are acronyms for female and male respectively. To ensure that the data-frame only contains 2 unique values; the following code is implemented

dic={'F':'female','M':'male'}

df['sex'] = df['sex'].apply(lambda x: x if x not in dic else dic[x])

where if the recorded value in the dataframe is ‘F’ or ‘M’, it is changed to ‘female’ or ‘male’ respectively; while keeping instances of ‘female’ and ‘male’ unchanged. If not performed, Python will not treat instances of ‘F’ and ‘female’ to be equivalent despite referring to the same sex.

For the “PersonID” column, which should have unique IDs, the code below reveals that there is one ID that repeats itself in the dataframe

id\_counts=df['PersonID'].value\_counts()

print('\nPerson IDs repeated at least once (PersonID \ Count):',id\_counts[id\_counts>1],sep='\n')

With the ID being 100 as shown below

Person IDs repeated at least once (PersonID Count) after modification:

PersonID

100 3

For valuable information to be extracted from the dataframe; it is pertinent to ensure all the data comes from different people. As repeated IDs indicate that the data originates from the same source, repeats of such values are dropped; which is accomplished using the code below

df=df.iloc[df['PersonID'].drop\_duplicates().index]

With regards to missing data only present in the age column – as proven by the code

print('\nColumns with missing data:',', '.join(list(filter(lambda x:df[x].isna().sum()>0,df.columns))))

which yields the output

Columns with missing data: age

it is essential that, to make the most of the data provided, rows containing the missing data aren’t discarded but the data is instead imputed such that it can be used for analysis and prediction. This is accomplished using pandas’ built-in interpolate method,

df['age']=df['age'].interpolate().apply(lambda x: round(x,0))

which by default linearly interpolates missing values to be those between values in the rows before and after itself. This allows for the missing ages to be estimated; and the lambda function in the apply method rounds the value to one of an integer form, as expected for age inputs.

Finally, for the data to be fit into machine learning models such as Decision Tree models, categorical variables have to be encoded. Hence, One-Hot Encoding is performed on the columns that are determined to contain non-numeric values to convert it to a format which can be parsed my machine learning models. This is done in the code below

from sklearn.preprocessing import OrdinalEncoder

enc=OrdinalEncoder()

reference\_dic={}

non\_num=list(filter(lambda x:df[x].dtype=='O',df.columns))

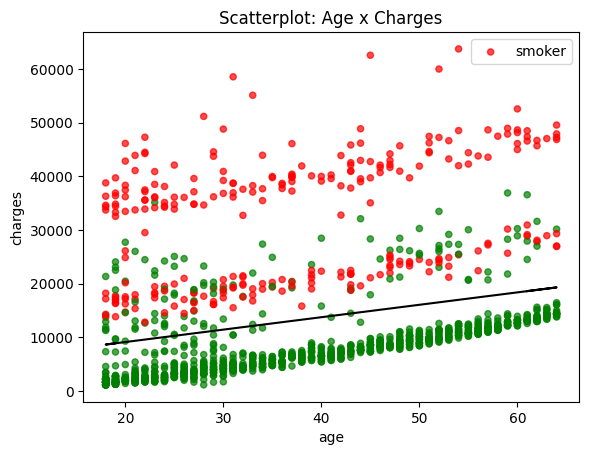
for e in non\_num:

df[e]=enc.fit\_transform(df[[e]])

reference\_dic[e]=list(enc.categories\_[0])

print(e,list(enc.categories\_[0]))

**Question 2**

****

***Figure 1***

ax = df.plot(kind='scatter',x='age',y='charges',

color=df['smoker'].map({1:'red',0:'green'}),alpha=0.7)

# observe comparison btw age and charges

# and also between smokers and non-smokers in the same demographic

plt.legend(['smoker'])

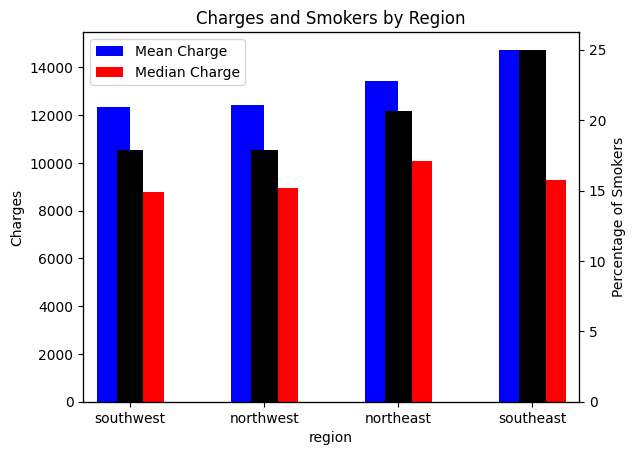
plt.title('Scatterplot: Age x Charges')

m, b = np.polyfit(df.age, df.charges, 1)

plt.plot(df.age, m\*df.age+b,color='black') # code for linear regression line

plt.show()

Figure 1 is a scatterplot depicting the change and variability in charges as age increases as well as by smoking status. It shows a weak linear relationship between age and charges. There is little change in variability of charges as age increases – indicating low heteroscedasticity. The solid line in black shows how as age increases – so too does medical charges; which can be attributed to increases problems in health and a necessity to have further medical check-ups as one becomes older. It is clear from the graph that in the same age demographic however, the charges for smokers (points in red) are generally significantly higher than those for non-smokers (points in green). This is likely due to the additional health problems smoking brings to these people.



***Figure 2***

fig, ax = plt.subplots()

grouped\_data=df.groupby(['region'])['charges'].aggregate(['mean','median']).sort\_values('mean') # get mean and median charges by region

ax2 = ax.twiny()# create a secondary axis for percentage of smokers

grouped\_data.plot(kind='bar',color=['blue','red'],width=.5,ax=ax)

ax.set\_ylabel('Charges')

(df.groupby(['region'])['smoker'].aggregate(['mean']).sort\_values('mean')\*100).plot(kind='bar', secondary\_y=True,

ax=ax,width=0.2,color='black').set\_ylabel('Percentage of Smokers')

ax2.axes.get\_xaxis().set\_visible(False) # remove 2nd x-axis

ax.legend(["Mean Charge", "Median Charge"]) # set legend for mean and median charges

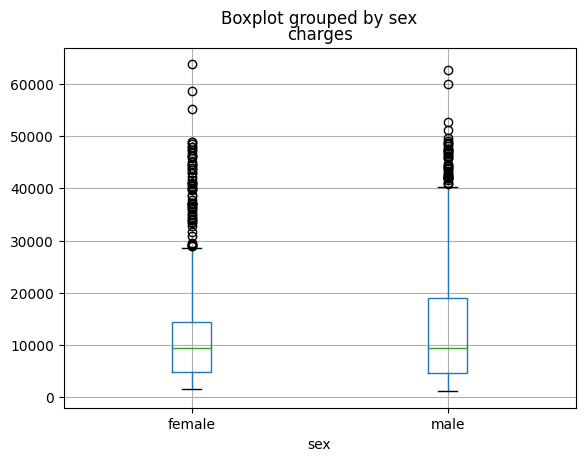
ax.set\_xticks(plt.xticks()[0],labels=list(map(lambda x:reference\_dic['region'][int(x)],indexes)),rotation=0, ha='center')

# code to swap numerically encoded values of region to human-readable data

plt.title('Charges and Smokers by Region')

plt.show()

Figure 2 shows how medical charges and percentage of smokers vary by different regions. From the data, it appears that the eastern regions (North-East and South-East) have a higher average (mean) medical charge per person of approximately $13500 and $14600 per person per region respectively – while in the North-West and South-West regions the mean medical charge per person for both places is about $12000. This seems to correlate with the percentage of smokers (black bar) in each of the regions, with the South-East having the greatest percentage of smokers at 25%, followed by the North-East’s ~21%, and the North-West and South-West both at ~18%. This enforces the observation of smokers having higher costs; as regions with greater percentages of smokers have greater mean medical charges. It also shows a distinction between the number of smokers in the East and West regions – owing to factors yet undetermined. What is notable is how the median charges across all regions are significantly lower than the mean charges; indicating a right-skewed distribution for charges.



***Figure 3***

df.boxplot('charges',by='sex') # boxplot of charges by sex

plt.xticks(plt.xticks()[0],reference\_dic['sex'])

#replace numeric x ticks values (0,1) with original categorical values #(female, male)

plt.show()

Figure 3 shows the distribution of charges by sex in a boxplot. It shows how both males and females have a similar median charge close to $10000, but the interquartile range of males (from $5000 to $20000) is significantly larger than that for females ($5000 to $15000). This may be due males being more likely to smoke and therefore resulting in higher medical costs for them.

**Question 3**

To explore the dataset whether the dependent variable is ‘smoker’ – the ‘PersonID’ column is dropped as it is unique to each specific data point but does not give any valuable information with respect to if someone is a smoker or not. This is accomplished in the try except block below

try:

df=df.drop("PersonID",axis=1)

except:

print("Axis already dropped")

As all categorical columns have already been One-Hot Encoded in Question 1, there is no need for re-encoding of variables. A standard DecisionTreeClassifier is used as the model; imported from sklearn as shown in the code below

from sklearn.tree import DecisionTreeClassifier

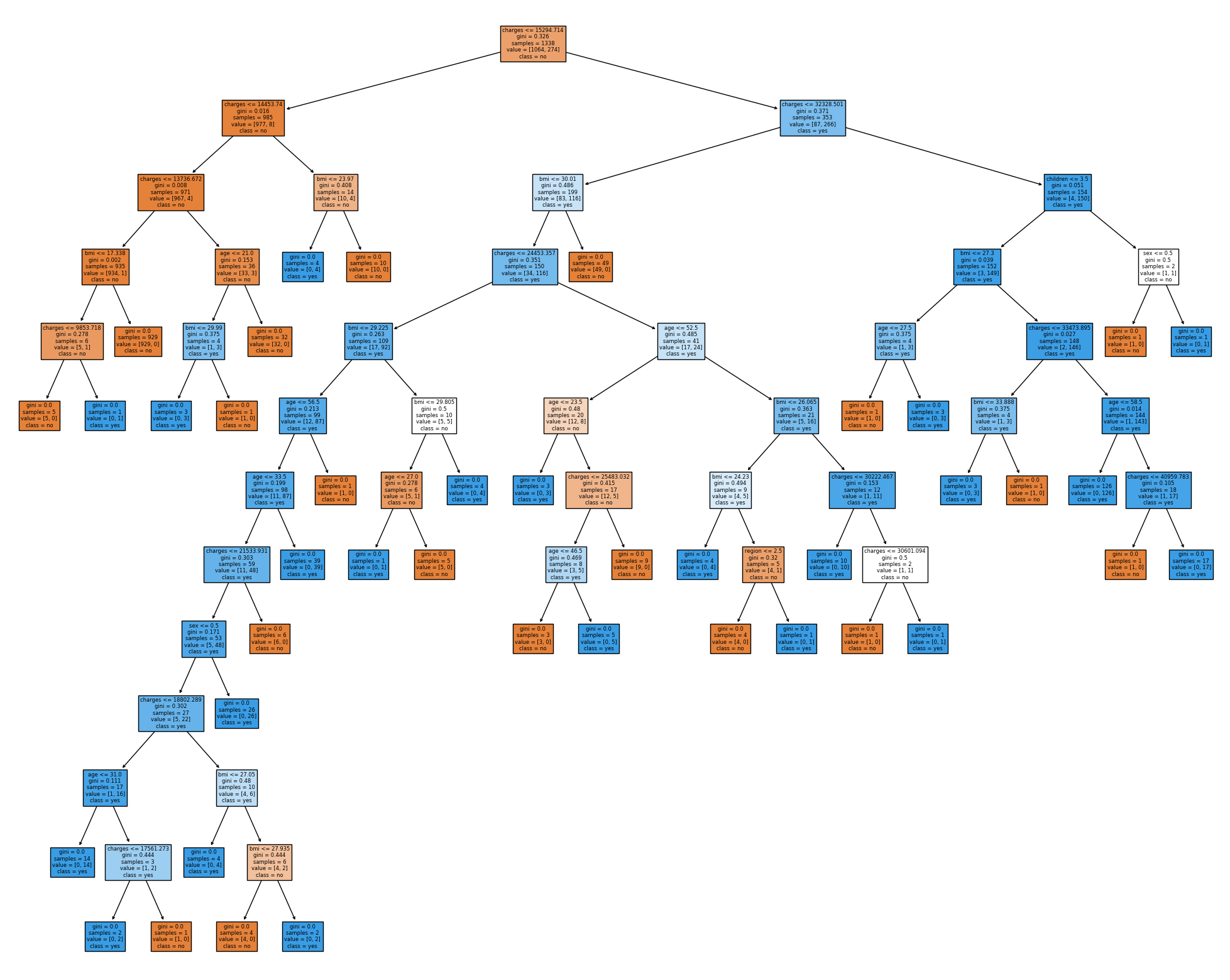
This is because smoker, while containing numerical values post One-Hot Encoding, is made up of categorical data. All columns that remain in the dataframe apart from smoker - ['age', 'sex', 'bmi', 'children', 'region', 'charges'] – are used for exploration as they are relevant to model evaluation; and insights as to whether or not a person is a smoker can be determined from disparities between the above features for smokers and non-smokers. Conclusions can also be drawn from how significant a feature is in classifying smokers/non-smokers based on how ‘high up’ the feature is on the Decision tree – acting as a feature-ranking method. Since data exploration is to be conducted and not model training; the data is not split into testing/training sets nor in cross validation employed – but rather the whole data set is used. Creating the model is thus accomplished in the code below

model=DecisionTreeClassifier()

features=[i for i in df.columns if i!=dep\_var]

model.fit(df[features],df[dep\_var])

**Question 4**



***Figure 4***

The figure above shows the Decision Tree created. Blue boxes are those classes with a majority for smokers while orange boxes are classes with a majority for non-smokers; a deeper colour intensity indicating a higher purity of smokers/non-smokers in the class respectively. From the Decision Tree, it can be determined that medical charges are a useful indicator for smoking status; with charges greater than a threshold of 15294.714 making it especially likely for a person to be a smoker and costs equal to or lower making it more likely for a person to be a non-smoker. BMI, the number of children a person has and one’s sex are also useful indicators. Region does not appear to play a significant role in determining a smoke’s status compared to the other features, age, BMI and charges are used more frequently as thresholds. BMI and charges are used especially often when a person has charges less than 15294.714 to weed out smokers, with people with lower BMIs and low medical charges being more likely to be smokers, while for people with charges greater than 15294.714, women with more than 3 children are not likely to be smokers, as are young people with low BMI.

**Question 5**

Decision trees can be used for exploratory data analysis (EDA), but not solely on their own. The advantage of incorporating Decision Trees in EDA is that, based on the recursive functions and thresholds that a Decision Tree uses, it implicitly ranks features (Karax et al.,2019) based on their significance in relation to a target variable; visualising a form of feature selection. This in turn highlights to the programmer the features that are especially significant to determine the value of a specific target variable (Brownlee, 2020), enabling them to focus on those features. However, using only decision trees for EDA is limiting in that it is unable to provide for other insights such as the range of numeric variables; distributions of features as well as the correlation between features that can be effectively rendered via boxplots, histograms and correlation matrices. Furthermore, there are other methods that exist in programming languages to determine the importance of a feature such as the variability of a feature (“Feature Selection”, n.d.) – thus the benefit that Decision Trees brings for EDA is by no means one that is unique to Decision Trees solely. Nonetheless, they are able to grant useful insights not only in terms of significant features, but also in recognising edge cases where even when a result may be expected to fall in a particular class, it belongs to another owing to a change in a particular feature. This is observed in the previous question for people with more than 3 children and charges greater than $32300 (3.s.f) – where if their sex is Male they are bound to be smokers yet if they are Female they are bound not to be smokers. Hence; decision trees can be used to obtain further insights during EDA, yet are not the gold standard of methodologies.

**References**

Brownlee, J. (2020, August 20). *How to Choose a Feature Selection Method For Machine Learning*. Machine Learning Mastery. <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/#:~:text=Feature%20selection%20methods%20are%20intended,redundant%20predictors%20from%20the%20model>

Feature Selection (n.d.). https://scikit-learn.org/stable/modules/feature\_selection.html

Karax, J., Malucelli A and Barddal J. (2019, April). Decision tree-based feature ranking in concept drifting data streams. *SAC '19: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (p. 590 – 592). <https://dl.acm.org/doi/10.1145/3297280.3297551>

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