|  |  |
| --- | --- |
|  | Logo  Description automatically generated |

**ANL252**

**Python for Data Analytics**

**End-Course Assignment**

**July 2023 Presentation**

Submitted By:

|  |  |  |
| --- | --- | --- |
| S.N. | Name | PI. No |
| 1. | Ng Ling Ying | E2181019 |

Tutorial Group: TG03

Submission Date: 3 November 2023

Table of Contents

[Question 1 3](#_Toc149075908)

[Treating Missing Values 5](#_Toc149075909)

[Treating Outliers 6](#_Toc149075910)

[Treat Numerical Values 8](#_Toc149075911)

[Question 2 9](#_Toc149075912)

[Charges Distribution by Smoker Status 9](#_Toc149075913)

[Scatter Plot of Age vs Charges 10](#_Toc149075914)

[Box Plot Grouped by Region 11](#_Toc149075915)

[Question 3 12](#_Toc149075916)

[Question 4 13](#_Toc149075917)

[Question 5 14](#_Toc149075918)

[References 15](#_Toc149075919)

# Question 1

Firstly, I imported the data set and pandas into Python.

import pandas as pd

eca\_data = pd.read\_csv("ECA.csv")

display (eca\_data)

A table with numbers and text

Description automatically generated

I have decided to make all of the columns with decimal points reflect in 2d.p. This ensures consistency in the dataset and eliminates variation in the number of decimal places, making it easier to work with and present the data uniformly.

eca\_data['bmi'] = eca\_data['bmi'].round(2)

eca\_data['charges'] = eca\_data['charges'].round(2)

display(eca\_data)

Output:

A table with numbers and text

Description automatically generated

I have also checked that column that does not have numerical value to ensure that the data is consistent. I have found out that the ‘sex’ column contains different values when they are supposed to be female and male only. After, I have decided to replace 'M' with 'male' and 'F' with 'female' in the 'sex' column and double-check whether it has been replaced.

unique\_sex\_values = eca\_data['sex'].unique()

print(unique\_sex\_values)

Output: ['female' 'male' 'F' 'M']

unique\_region\_values = eca\_data['region'].unique()

print(unique\_region\_values)

Output: ['southwest' 'southeast' 'northwest' 'northeast']

unique\_smoker\_values = eca\_data['smoker'].unique()

print(unique\_smoker\_values)

Output: ['yes' 'no']

#Replacing

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

Double-checked:

unique\_sex\_values = eca\_data['sex'].unique()

print(unique\_sex\_values)

Output: ['female' 'male']

## Treating Missing Values

After this, I have decided to identify variables with missing values using the code below.

eca\_data.isnull().sum(axis = 0/1)

Output:

PersonID 0

age 123

sex 0

bmi 0

children 0

smoker 0

region 0

charges 0

dtype: int64

The output indicates that there are 123 ‘null’ values in ‘age’. Hence, I have decided to treat the missing data by using the mean age based on gender to replace the missing data. After adding the code, I double-checked using eca\_data.isnull().sum(axis = 0/1)to ensure that the missing data is replaced.

#Calculate mean age based on gender

gender\_mean\_age = eca\_data.groupby('sex')['age'].mean()

#Replace the missing data with the calculated mean

for index, row in eca\_data.iterrows():

if pd.isnull(row['age']):

eca\_data.at[index,'age'] = gender\_mean\_age[row['sex']]

This is a practical way to handle missing data, as it ensures that the imputed values will be relevant to the data set. By using the ‘gender’ means, it ensures that it preserves the gender-related patterns in the data. Double-checking the data will help to confirm that no missing values are left out.

## Treating Outliers

Secondly, I went to check if there were any outliers in the data.

from scipy import stats

z\_scores\_bmi = stats.zscore(eca\_data['bmi'])

z\_scores\_charges = stats.zscore(eca\_data['charges'])

outliers\_bmi = eca\_data[abs(z\_scores\_bmi) > 3]

outliers\_charges = eca\_data[abs(z\_scores\_charges) > 3]

display("Outliers in 'bmi' column:")

display(outliers\_bmi)

display("Outliers in 'charges' column:")

display(outliers\_charges)

Output:

A screenshot of a table

Description automatically generated

The Python code will remove a number of outliers. This code will remove rows from the DataFrame where the values in the 'charges' column are considered outliers using the IQR method. Rows with 'charges' values less than or greater than the lower\_bound will be filtered out. After that, I used the above code to double-check the data.

This will help with dealing with outliers, which can skew statistical analysis and modelling. It is a critical data preprocessing step to ensure that the data accurately represents the underlying patterns and relationships, especially when dealing with numerical variables like 'charges'.

# Calculate Interquartile range for 'charges'

q1\_charges = eca\_data['charges'].quantile(0.25)

q3\_charges = eca\_data['charges'].quantile(0.75)

iqr\_charges = q3\_charges - q1\_charges

# Calculate Interquartile range for 'bmi'

q1\_bmi = eca\_data['bmi'].quantile(0.25)

q3\_bmi = eca\_data['bmi'].quantile(0.75)

iqr\_bmi = q3\_bmi - q1\_bmi

# Define bounds for 'charges'

lower\_bound\_charges = q1\_charges - 1.5 \* iqr\_charges

upper\_bound\_charges = q3\_charges + 1.5 \* iqr\_charges

# Define bounds for 'bmi'

lower\_bound\_bmi = q1\_bmi - 1.5 \* iqr\_bmi

upper\_bound\_bmi = q3\_bmi + 1.5 \* iqr\_bmi

# Filter data to exclude outliers for 'charges' and 'bmi' separately

eca\_data = eca\_data[~((eca\_data['charges'] < lower\_bound\_charges) | (eca\_data['charges'] > upper\_bound\_charges))]

eca\_data = eca\_data[~((eca\_data['bmi'] < lower\_bound\_bmi) | (eca\_data['bmi'] > upper\_bound\_bmi))]

Double-checked data:

A close-up of a white background

Description automatically generated

## Treat Numerical Values

For treating numerical data, I have decided to split them into different categories based on their age. I decided to find out what is the maximum age and minimum age to determine the range I could create for the respective category. After finding out the range, I have decided to make the range for young adults (18 to 25), adults (26 to 50) and the elderly (51 to 64) This will help to simplify data analysis and allow for easier comparisons between different age groups

max\_age = eca\_data['age'].max()

min\_age = eca\_data['age'].min()

print("Maximum age:", max\_age)

print("Minimum age:", min\_age)

Output:

Maximum age: 64.0

Minimum age: 18.0

age\_bins = [17, 26, 36, 65]

age\_labels = ['young adult', 'adult', 'elderly']

eca\_data['age\_category'] = pd.cut(eca\_data['age'], bins=age\_bins, labels=age\_labels)

display (eca\_data)

Output:

A screenshot of a data table

Description automatically generated

379 words

# Question 2

## Charges Distribution by Smoker Status

The first graph is a histogram that depicts the charge distribution by smoker status. Nonsmokers make up the majority of the sample. This indicates that the vast majority of people in the dataset do not smoke, which is an important demographic insight.

A significant difference in healthcare costs between smokers and nonsmokers is also shown by the histogram. Nonsmokers' bars are significantly more common at the lower end of the charge scale, implying that nonsmokers have lower healthcare costs. The bars for smokers, on the other hand, are higher, indicating that smokers generally incur higher medical costs.

plt.hist(eca\_data[eca\_data['smoker'] == 'yes']['charges'], bins=30, alpha=0.5, label='Smoker', color='red')

plt.hist(eca\_data[eca\_data['smoker'] == 'no']['charges'], bins=30, alpha=0.5, label='Non-Smoker', color='green')

plt.xlabel('Charges')

plt.ylabel('Frequency')

plt.title('Charges Distribution by Smoker Status')

plt.legend()

plt.show()

Output:

A graph of a number of charge levels

Description automatically generated with medium confidence

## Scatter Plot of Age vs Charges

The scatter plot depicts the relationship between a person's age and charges. There is an observable trend of charges increasing as a person's age increases. In general, people may require more extensive healthcare services as they age due to the natural ageing process, which frequently results in a higher prevalence of chronic conditions and a greater need for medical care.

The scatter plot uses colour to distinguish between smokers and nonsmokers. It demonstrates that smokers face higher charges than nonsmokers at various ages. This finding emphasises smoking's negative impact on healthcare costs. Smokers are more likely to develop health issues such as respiratory diseases, heart disease, and certain types of cancer, which can lead to higher medical costs. This disparity is effectively depicted by the scatter plot, which shows that smoking is associated with higher healthcare costs.

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

sns.scatterplot(x='age', y='charges', data=eca\_data, hue='smoker')

plt.title('Scatter Plot of Age vs. Charges')

plt.xlabel('Age')

plt.ylabel('Charges')

plt.show()

Output:

A graph of a scatter plot of age versus charges

Description automatically generated

## Box Plot Grouped by Region

The final graph will be a boxplot that depicts the relationship between charges and region. The distribution is skewed to the left in all four graphs. This implies that the majority of the data points are clustered at the lower end of the charge scale, with a long tail extending to higher charges. Left-skewed in this context means that the tail of the distribution extends towards the lower charges.

It also states that the 'northeast' region has the highest maximum charges as well as the highest median value. This indicates that people in the northeast have higher medical expenses than people in other regions. The 'southeast' region, on the other hand, has the lowest median charges, implying that people in this region typically have lower healthcare costs. Furthermore, the 'southwest' region has the lowest maximum charges, implying that the most extreme high-cost cases are less common in this region.

eca\_data.boxplot(column='charges', by='region', showfliers=False)

plt.title('Charges by Region')

plt.xlabel('Region')

plt.ylabel('Charges')

plt.suptitle("")

plt.show()

Output:

**A graph with lines and numbers

Description automatically generated with medium confidence**

399 words

# Question 3

To construct the decision tree, the 'children' feature is chosen as the primary independent variable. Using a single feature like 'children' keeps the model simple and easy to visualize, making it an accessible starting point for understanding. The decision tree model is built using a DecisionTreeClassifier from the Scikit-Learn library, I will encode the 'smoker' variable, replacing 'non-smoker' with 0 and 'smoker' with 1, as it allows for the classification of individuals into non-smokers and smokers.

The feature matrix, denoted by the letter X, contains only one feature, 'children.' The number of children a person has is chosen as the predictor variable because it is hypothesised to influence one's smoking status. The target variable, y, represents whether a person is a smoker or not.

A decision tree classifier is built and trained on the dataset with the primary goal of revealing relationships between the number of children and smoking habits. The decision tree is built by selecting the feature that best separates the data based on a criterion such as Gini impurity or entropy, which measures the impurity or randomness of the data.

The resulting decision tree depicts how the presence or absence of children affects an individual's likelihood of smoking. It divides the data into subsets and assigns a class label to each leaf node, with the two classes being 'non-smoker' and 'smoker'.

In conclusion, the decision tree approach in this code is useful in investigating the impact of having children on one's 'smoker' status. It provides a clear and interpretable model for determining the complex relationships between variables, which can be useful for making informed decisions about targeted public health interventions and tailored awareness campaigns for specific demographic groups.

281 words

# Question 4

# Question 5

Decision trees can be used effectively for exploratory data analysis, expanding their role beyond prediction. It provides a visual and intuitive representation of decision-making, making it easy for humans to interpret and understand.

By forming splits and branches, decision trees can determine the importance of various features (Janbandhu, 2023). This not only provides a clear understanding of how decisions are made but also reveals which variables have the most influence on the outcome. Decision trees assist organisations and data scientists in making more informed decisions (Janbandhu, 2023).

The ability of decision trees to interpret and visualise nonlinear data patterns is their most significant advantage (Kapil, 2023). They perform well, particularly for exploratory data analysis, delivering high accuracy on small datasets and assisting in the handling of messy, non-normalized data by excluding non-essential features (Berezovsky, 2023). Decision trees allow you to improve accuracy by defining the logic for branch splits.

Additionally, by utilising decision tree algorithms, organisations can gain valuable insights into their data, assisting in risk assessment, understanding the impact of choices, and optimising operations (Emeritus, 2023).

Based on the relationships in the dataset, decision trees can determine which attributes of the data are most informative in predicting missing values (Tierney et al., 2015). Decision trees facilitate the process of dealing with missing data by identifying the most relevant predictors for imputation, ensuring that imputed values are as close to the actual values as possible.

Incorporating decision trees into exploratory data analysis improves the process significantly, especially for complex datasets. However, it is critical to recognise their limitations, such as overfitting and the possibility of missing all data patterns. To gain a comprehensive understanding of the data, it is recommended to supplement decision tree analysis with other exploratory data analysis techniques (Janbandhu, 2023).

293 words

# References

Berezovsky, O. (2023, October 4). *Supervised machine learning - binary logistic regression overview*. Logic20/20. https://logic2020.com/insight/decision-tree-classifier-overview

Emeritus. (2023, May 22). *Decision tree: A powerful tool for analytical decision making*. https://emeritus.org/in/learn/data-science-decision-tree/

Janbandhu, M. (2023, June 11). *Why decision tree algorithm ?*. LinkedIn. https://www.linkedin.com/pulse/why-decision-tree-algorithm-mohit-janbandhu/

Kapil, A. R. (2023, October 9). *Advantages and disadvantages of Decision Tree in machine learning*. Blogs & Updates on Data Science, Business Analytics, AI Machine Learning. https://analytixlabs.co.in/blog/decision-tree-algorithm/

Tierney, N. J., Harden, F. A., Harden, M. J., & Mengersen, K. L. (2015, June 29). *Using decision trees to understand structure in missing data*. BMJ open. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4486966/