Logo, company name

Description automatically generated

**ANL 252**

**Python for Data Analytics**

**End-Of-Course Assessment (ECA)**

**July 2023 Presentation**

**Tan Wee Huat (K2270304)**

**Question 1**

The first data pre-processing task is to address continuous variables “sex” where there are inconsistencies so it can be analysed. Variable “sex” consist entries of “M” or “F” which creates inconsistencies when analysing. First, convert the entries connotated by “M” or “F” and convert to “male” or “female”. Next, quantify this dataset by converting “male” and “female” into equivalent values of “1” and “0” respectively.

import pandas as pd

# Read ECA CSV file into a DataFrame

Dataframe\_ECA = pd.read\_csv("ECA.csv")

# Converted "sex" male and female into arbitrary equivalent values of 1 and 0

# Define a mapping dictionary for gender conversion

gender\_mapping = {"M": "male", "F": "female"}

sex\_mapping = {"male": 1, "female": 0}

# Use the 'replace' function to apply the gender mapping to the 'sex' column

Dataframe\_ECA['sex'] = Dataframe\_ECA['sex'].replace(gender\_mapping)

Dataframe\_ECA['sex'] = Dataframe\_ECA['sex'].replace(sex\_mapping)

# Save the DataFrame back to a CSV file

Dataframe\_ECA.to\_csv("Converted1.csv", index=False)

The second pre-data processing task is to treat numerical values of variable “BMI” and convert the arbitrary values it into the equivalent categorical value of {“Underweight”, “Normal Weight”, “Overweight”, “Obese”}.

import pandas as pd

# Read the CSV file into a DataFrame

Dataframe\_BMI = pd.read\_csv("Converted1.csv")

# Define BMI categories and corresponding bins

bmi\_bins = [0, 18.5, 24.9, 29.9, float('inf')]

# Define labels for the BMI categories

bmi\_labels = ['Underweight', 'Normal Weight', 'Overweight', 'Obese']

# Create a new column 'bmi\_category'

Dataframe\_BMI.loc[:, 'bmi\_category'] = pd.cut(df['bmi'], bins=bmi\_bins, labels=bmi\_labels, right=False)

# Save the updated DataFrame to updated CSV file

Dataframe\_BMI.to\_csv(Converted2.csv', index=False)

The third pre-data processing task is more sophisticated as there are missing entries under “age” variables that would affect the accuracy of the data analysis. Instead of deleting these data entries, linear regression can be performed to predict the missing values of “age” by using sklearn. The independent variables used to determine the dependent variable “age” are “charges” and “BMI” as the have the highest correlation among the other independent variables.

import pandas as pd

import numpy as np

from sklearn.linear\_model import LinearRegression

ok

# Load the dataset from Converted2 CSV file

Dataframe\_LR = pd.read\_csv('Converted2.csv')

# Separate the dataset into two parts: one with missing ages and one without

Dataframe\_missing = df[df['age'].isnull()]

Dataframe\_not\_missing = df[~df['age'].isnull()]

# Prepare the features (X) and target (y) for linear regression

X = Dataframe\_not\_missing[['charges', 'bmi']]

y = Dataframe\_not\_missing['age']

# Create a Linear Regression model and fit it to the data

model = LinearRegression()

model.fit(X, y)

# Predict missing ages based on 'charges' and 'bmi'

X\_missing = Dataframe\_missing[['charges', 'bmi']]

predicted\_ages = model.predict(X\_missing)

# Round the predicted ages to the nearest whole number

predicted\_ages = np.round(predicted\_ages).astype(int)

Dataframe\_LR.loc[df['age'].isnull(), 'age'] = predicted\_ages

# Save the updated DataFrame to the latest CSV file

Dataframe\_LR.to\_csv('Converted3.csv', index=False)

**(504 Words)**

**Question 2**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Load data from ECA CSV file into Dataframe\_ECA

Dataframe\_ECA = pd.read\_csv('Converted3.csv')

# Create Box Plot

plt.figure(figsize=(18, 9))

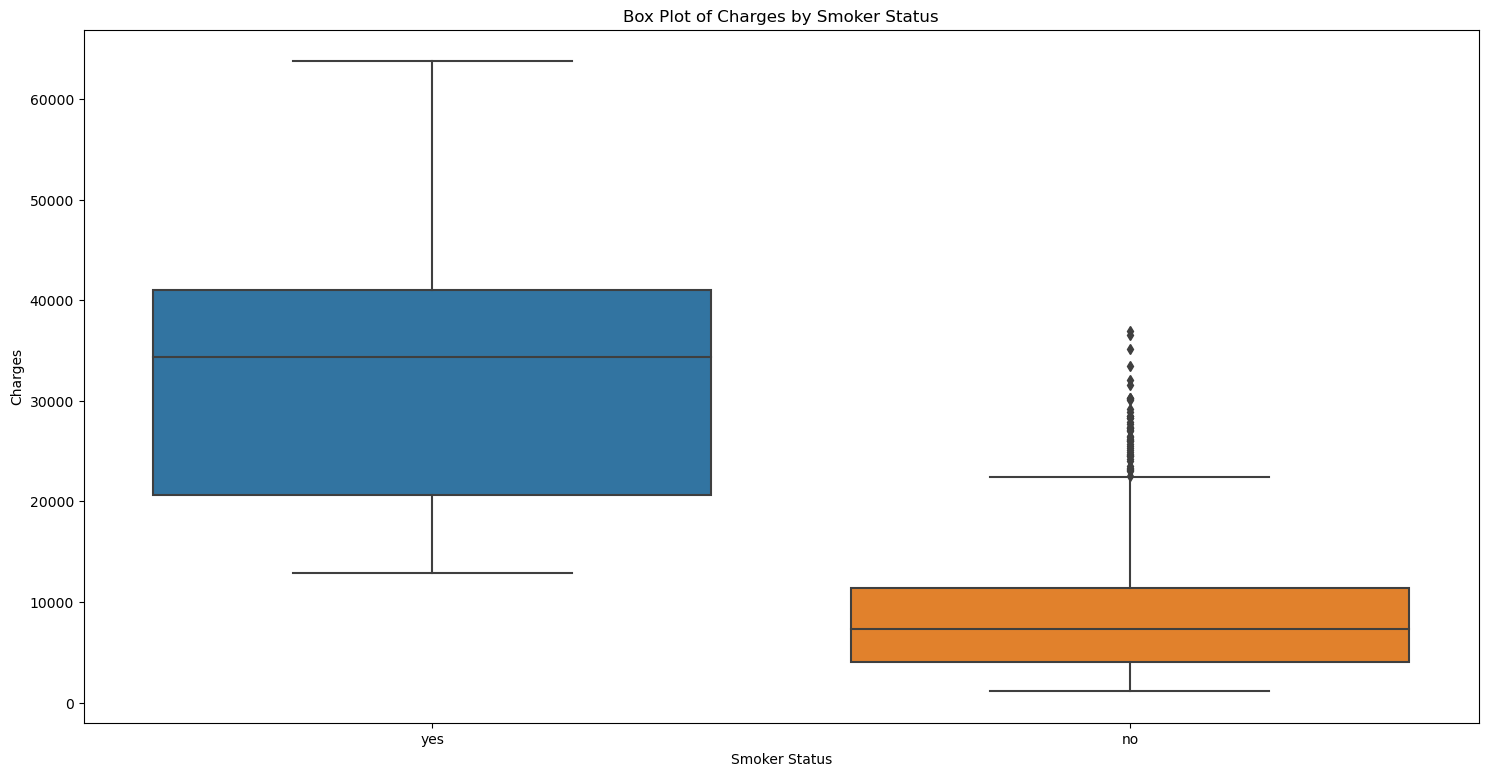
sns.boxplot(x='smoker', y='charges', data=Dataframe\_ECA)

plt.title('Box Plot of Charges by Smoker Status')

plt.xlabel('Smoker Status')

plt.ylabel('Charges')

plt.show()



*Figure 1. Box plot of charges by smoker status*

In Figure 1 charges for smokers is higher than non-smokers. Interquartile range for smokers is wider than non-smokers because the variability of the central portion of data is wider than non-smokers. The outliers on the higher end for non-smokers means there are extreme data points that is not considered. The median and quartiles for non-smoker are evenly distributed than smoker category.

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Load the dataset from the CSV file

Dataframe\_ECA = pd.read\_csv('Converted3.csv')

# Scatter plot between 'charges' and 'bmi'

plt.figure(figsize=(20, 10))

sns.scatterplot(x='charges', y='bmi', data=Dataframe\_ECA, alpha=0.5)

sns.regplot(x='charges', y='bmi', data=Dataframe\_ECA, scatter=False, ci=None)

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

plt.grid(True)

plt.show()

# Calculate and display the correlation coefficient

corr\_bmi = Dataframe\_ECA['charges'].corr(Dataframe\_ECA['bmi'])

print("Correlation Coefficient (Charges vs. BMI):", corr\_bmi)

# Scatter plot between 'charges' and 'age'

plt.figure(figsize=(20, 10))

sns.scatterplot(x='charges', y='age', data=Dataframe\_ECA, alpha=0.5)

sns.regplot(x='charges', y='age', data=Dataframe\_ECA, scatter=False, ci=None)

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

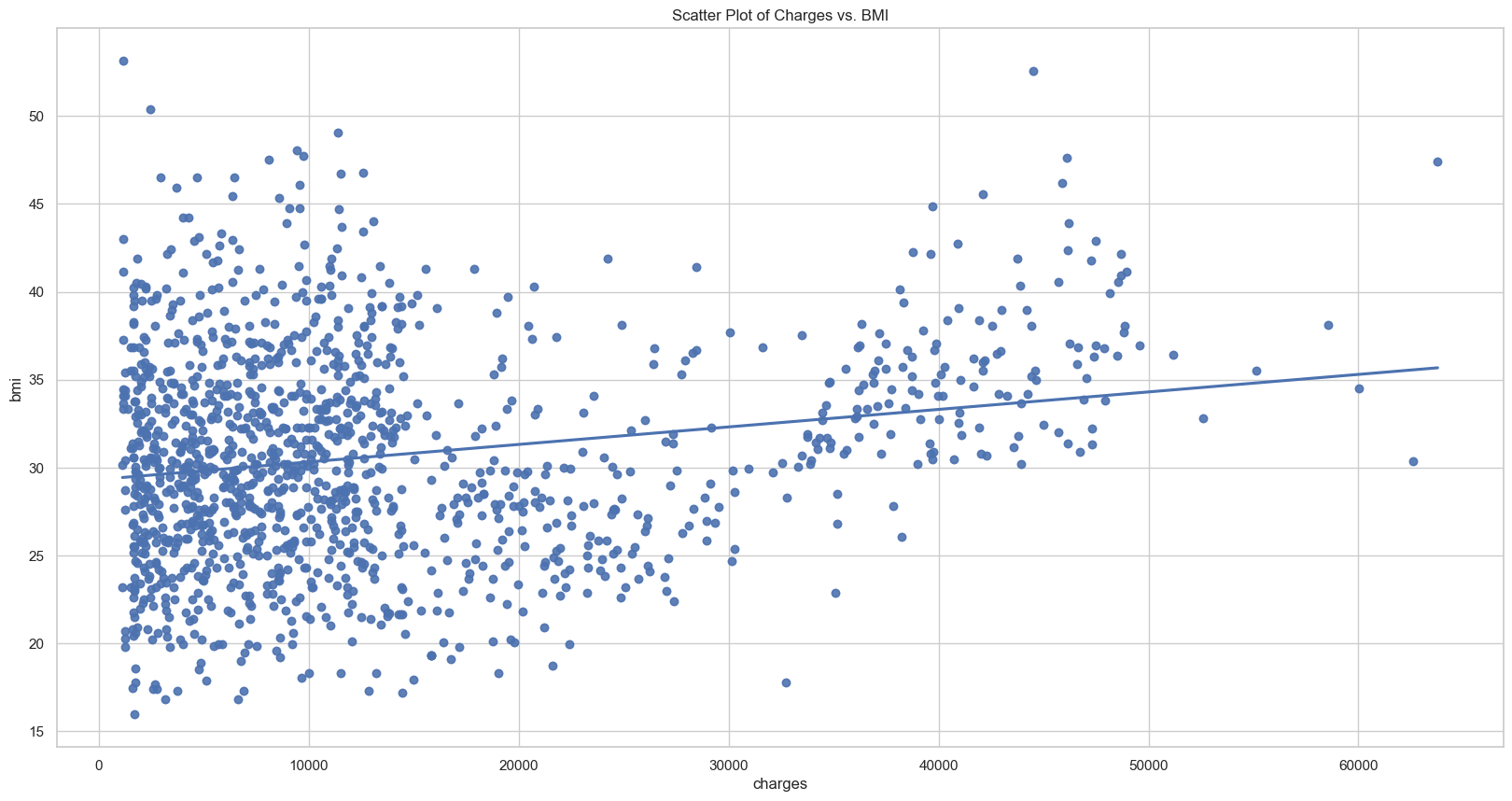
plt.grid(True)

plt.show()

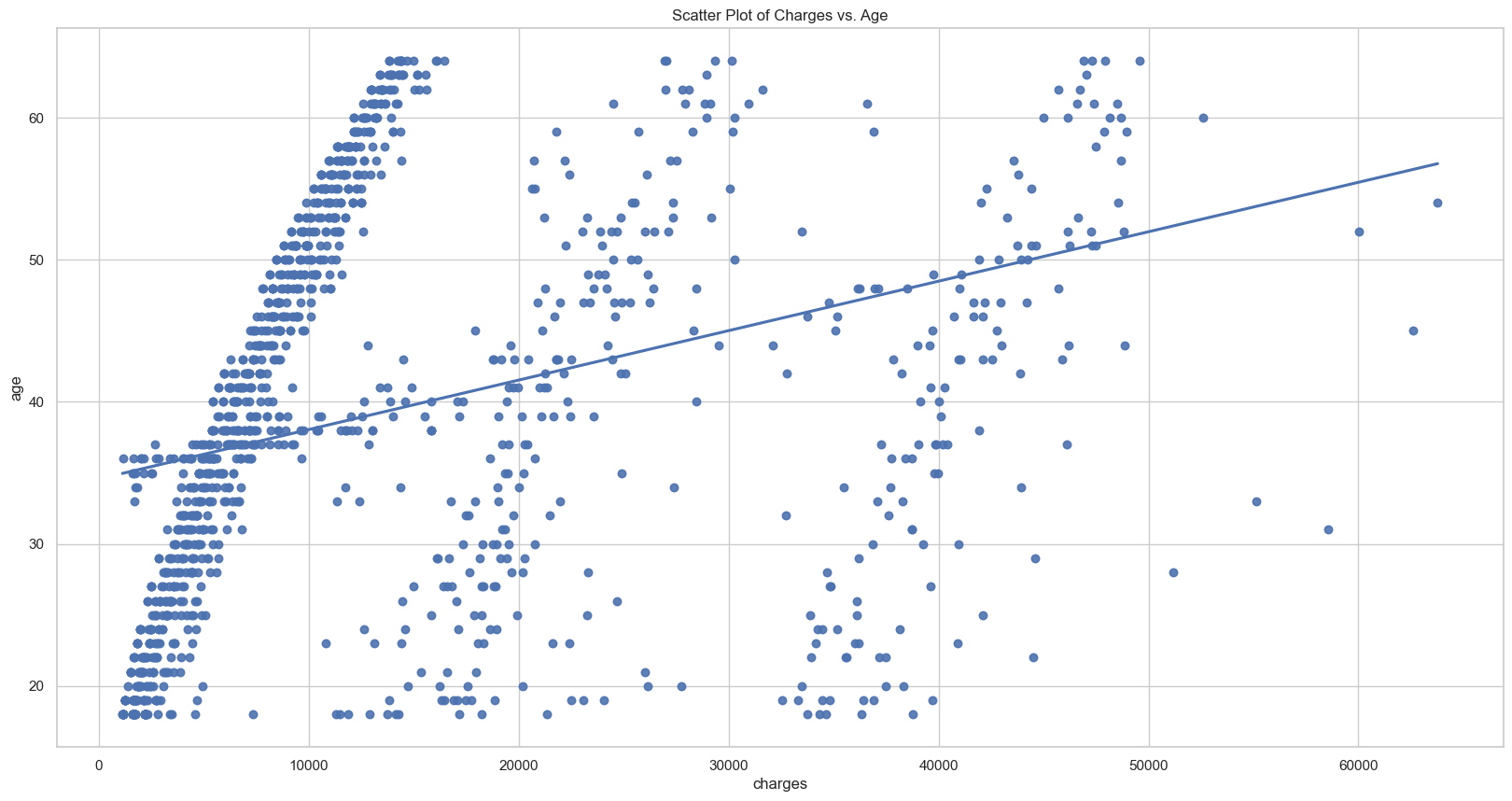
# Calculate and display the correlation coefficient

corr\_age = Dataframe\_ECA['charges'].corr(Dataframe\_ECA['age'])

print("Correlation Coefficient (Charges vs. Age):", corr\_age)



*Figure 2.1. Scatter plot between bmi and charges with coefficient line*

**

*Figure 2.2. Scatter plot between age and charges with coefficient line*

Figure 2.1 depicts the relationship between bmi and charges while Figure 2.2 depicts the relationship between age and charges. The correlation coefficient for both figures are 0.197 and 0.313 respectively. As both figures have a positive correlation within their own comparisons, Figure 2.2 has a stronger correlation based on the correlation coefficient and a steeper gradient. Figure 2.1 suggest that there is clustering of data points charges below 10000 and between bmi of 20 and 40 which reveals a population subset that are more concentrated. Figure 2.2 suggests that there is a heavy subset of clustering of data points across the entire age range and charges below 15000, subsequently in the 2 other subset of datapoints, there are more scattered as the charges increases. This could suggest that in the dataset between age and charges, there are heavier datapoints for the lower charges and decreases as the charges increases. Generally, the data points for Figure 2.1 are more concentrated and has lesser outliers.

import pandas as pd

# Load the dataset from the CSV file

Dataframe\_ECA = pd.read\_csv('Converted3.csv')

#Histogram between region ‘distribution’ and ‘count’

plt.figure(figsize=(18, 9))

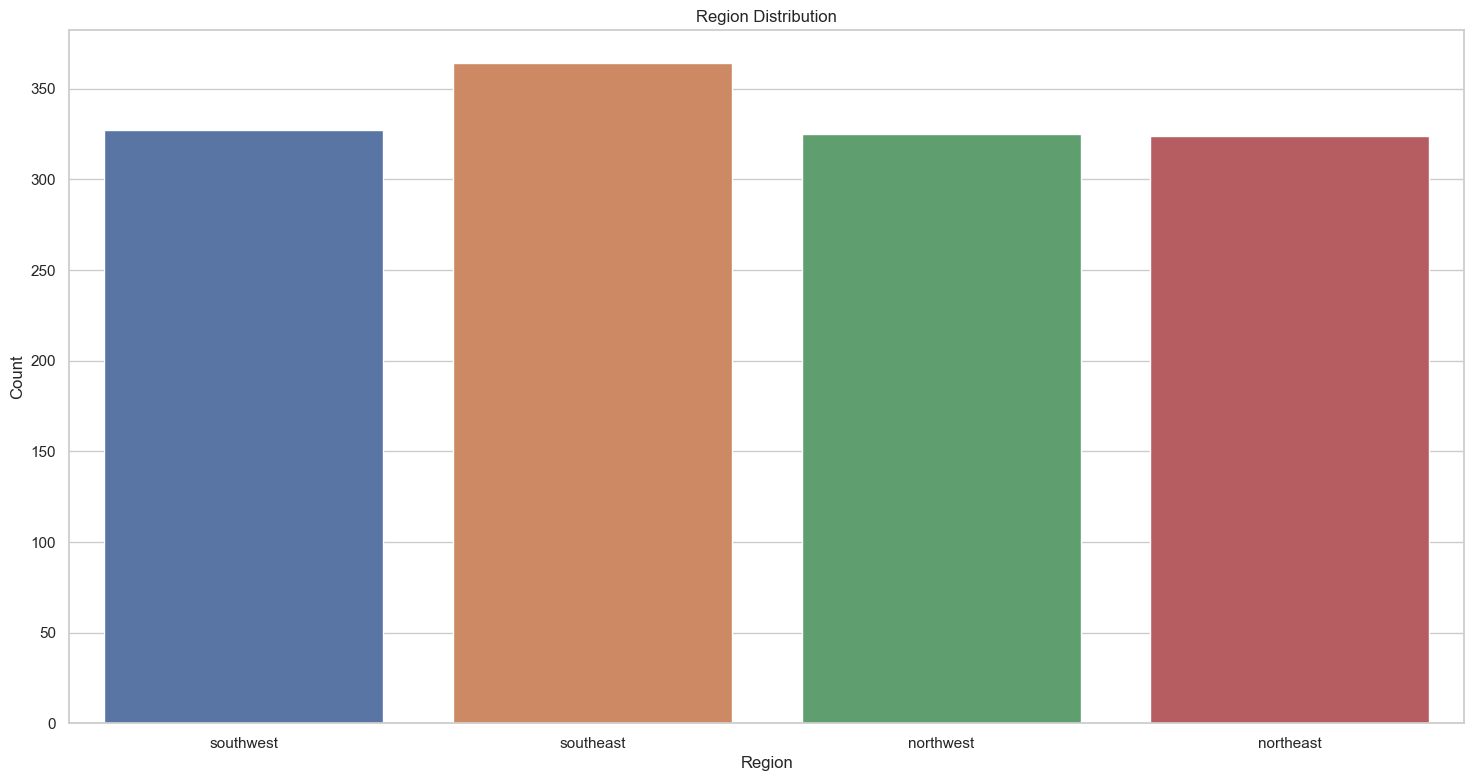
sns.countplot(x='region', data=Dataframe\_ECA)

plt.title('Region Distribution')

plt.xlabel('Region')

plt.ylabel('Count')

plt.show()



*Figure 3. Histogram between count and region distribution*

Figure 3 visually shows an even distribution of data entries across the 4 regions which suggests that the data sample is diversified. This ensures that the data analysis would be more justifiable and has a lower tendency for extreme outcomes. Essential to ensure that the data sampled covers enough aspects to achieve a greater objective outcome.

**(500 Words)**

**Question 3**

The dependent variable “smoker” is the target variable. A decision tree allows us to understand the relationship between the target variable and the various features. The first step is to import the pandas library to provide data structures and functions. Then import sklearn.tree module to enable implementation of decision trees and machine learning algorithms. The next step would be to prepare our dataset which is the CSV format file. However, we will be using the updated dataset where the data has been pre-processed. The next step is to pre-process our data by defining the dependent variable and features, where the features are the other variables represented in columns from the CSV file. We assign a variable name for the dependent variable and features for further purposes. The next step is essential, to split our dataset into train and test sets. The purpose of this is to evaluate how well the machine learning model performs. For test\_size, we decided to set the dataset allocated for testing to be 20% as the value usually varies between 20% and 30% depending on the size of dataset. For random\_state, we decided to set as 0 since we are unsure how it influences the decision tree but doing so maintains consistency and reproducibility. The next step is to create the decision tree, in this case is a classification decision tree since we are trying to predict which category is more closely related to “smoker”. The model evaluation make predictions on the testing data and compare predictions with actual values. It includes a confusion matrix and classification matrix which provides metrics such as accuracy, precision, recall, and F1-score for both classes of “smoker”. Lastly, plot\_tree is executed to define the parameters, plt.figure function to set the size of the decision tree, and plot.show() to display graph.

**(300 Words)**

**Question 4**

Confusion Matrix:

[[196 10]

[ 6 56]]

Classification Report:

precision recall f1-score support

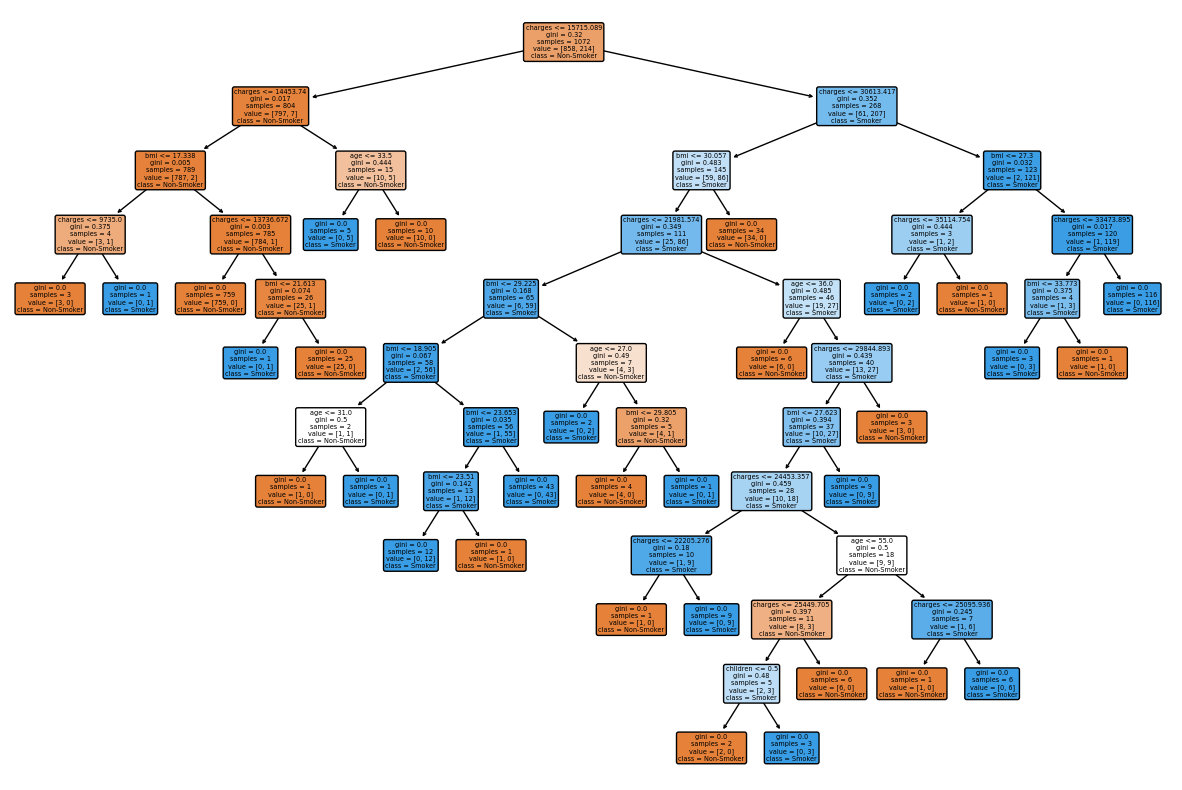
no 0.97 0.95 0.96 206

yes 0.85 0.90 0.88 62

accuracy 0.94 268

macro avg 0.91 0.93 0.92 268

weighted avg 0.94 0.94 0.94 268



From the above decision tree, we can see that the root node branches away for charges <= 14453.74 are non-smokers while charges <= 30613.417 are smokers. This model predicts that the feature “chargers” with its threshold value distinguishes between “smokers” and “non-smokers”.

The subsequent child node for “non-smokers” is split between bmi <= 17.338 and age <= 33.5. This suggests that further refinement is required and has determined other features to predict the outcomes.

We can observe there are 31 leaf nodes which is quite excessive given the scope of this model. Perhaps one of the reasons is due to lack of pruning which retains its predictive power while simplifying the model.

Based on the classification report, it shows that this decision tree model appears to perform effectively through high score in high precision, recall, and F1-scores for both “yes” and “no” categories. The high values in precision and recall suggests that the model is making accurate and consistent predictions. The overall accuracy score of 94% suggests that the model is very accurate in classifying the instances. The balance between precision and recall values suggest that this model is effective in making both accurate positive and negative predictions.

**(198 Words)**

**Question 5**

A decision tree visualizes how an underlying data predicts a selected target and highlights unique insights. Exploratory data analysis (EDA) refers to the methodology of analysing datasets to summarise their main characteristics through 2 methods (Restori). The first method is via graphical or non-graphical and the second method is either univariate or bivariate.

A main feature of the decision tree is the visualization of the nodes and how each node are formed determines which variable is the most impactful. This can be useful in EDA when deciding which variables should be prioritised and collected.

Another feature is the visualization of the entire data set into a diagram that prioritises from the most priority to the least priority with a clear structure. This allows for a more comprehensive analysis in which conjuncture does a key decision takes place.

Another feature is the data segregation since the decision tree predicts and categorises the information into subsets where it is more fitting based on the impact of the variables. It could potentially identify new subsets or trends that can lead to new discovery about the data set which could be useful for EDA.

A key aspect is how decision handles missing data or outliers. The decision tree is capable of treating missing data in an appropriate manner via the Classification and Regression Tree CART algorithm (Attard, 2023). They can guide the imputation process by determining at which data point is crucial for making certain decisions.

Another feature is the ability to visualise how the nodes interact with each other. Through observing how the nodes branch out based on the variables offer insights as to how it splits. It can be useful to evaluate which variables are more influential.

Decision trees are effective for EDA provided that it is utilised in an appropriate manner.

**(300 Words)**

# References

Attard, M. (2023, May 29). *Can decision trees handle missing values?*. Inside Learning Machines. <https://insidelearningmachines.com/decision_trees_handle_missing_values/>

Decision tree. visualization in an exploration. (n.d.). <https://www.ibm.com/docs/en/cognos-analytics/12.0.0?topic=type-decision-tree>

Pol, I. (2022, May 25). *How to split a dataset into train and test sets using Python*. GeeksforGeeks. <https://www.geeksforgeeks.org/how-to-split-a-dataset-into-train-and-test-sets-using-python/>

Restori, M. (n.d.). *What is exploratory data analysis*. Chartio. <https://chartio.com/learn/data-analytics/what-is-exploratory-data-analysis/>