A red and blue text on a black background

Description automatically generated

**ANL252**

**Python For Data Analytics**

**T03**

**End Course Assignment**

**Submitted on 2 November 2023 by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Chua Yu Zhe** | J2311231 |

Question 1:

1. Handling missing or invalid data

Firstly, we will check for any missing or invalid data in the dataset. We can then drop or impute missing values based on the context. We can drop rows with missing values or impute them using the mean or median for numerical columns and the mode for categorical columns.

Thus, ensuring that all data is in the correct format (e.g., numerical, categorical) and that there are no outliers.

import pandas as pd

# Loading the dataset

df = pd.read\_csv('ECA.csv')

# Checking for missing values

missing\_values = df.isnull().sum()

print("Missing Values:")

print(missing\_values)

# Dropping the rows with missing values (if necessary)

df.dropna(inplace=True)

# Ensuring the data are in correct types

df['age'] = df['age'].astype(int)

df['bmi'] = df['bmi'].astype(float)

df['children'] = df['children'].astype(int)

1. Encoding categorical variables

Encoding categorical variables into a numerical format using techniques like one-hot encoding or label encoding. In this dataset, 'sex', 'smoker', and 'region' are categorical variables.

# Encoding categorical variables (one-hot encoding)

df = pd.get\_dummies(df, columns=['sex', 'smoker', 'region'])

1. Scaling Numerical Features

Standardize or normalize numerical features to ensure that they are on the same scale.

from sklearn.preprocessing import StandardScaler

# Selecting the numerical columns to scale

numerical\_columns = ['age', 'bmi', 'children']

# Initializing the StandardScaler

scaler = StandardScaler()

# Fitting and transforming the selected numerical columns

df[numerical\_columns] = scaler.fit\_transform(df[numerical\_columns])

These 3 tasks will help ensure that the dataset is ready for further analysis and modeling. Handling missing data, encoding categorical variables, and scaling numerical features are essential steps to prepare the data for machine learning tasks and to improve the quality of analysis and predictions.

Question 2:

Figure 1: Age Distribution

import matplotlib.pyplot as plt

# Plot the age distribution

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

plt.hist(df['age'], bins=20, color='skyblue')

plt.title('Age Distribution')

plt.xlabel('Age')

plt.ylabel('Frequency')

plt.show()

A graph of age distribution

Description automatically generated

Insights (Figure 1):

The age distribution shows that the dataset contains individuals of various ages, with a relatively even spread across different age groups. This information is essential for understanding the demographics of the dataset and can be useful when assessing how age relates to medical costs.

Figure 2: Charges by Smoking Status

# Plot charges by smoking status

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

smoker\_charges = df.groupby('smoker\_yes')['charges'].mean()

smoker\_charges.plot(kind='bar', color=['lightcoral', 'lightgreen'])

plt.title('Average Charges by Smoking Status')

plt.xlabel('Smoker')

plt.ylabel('Average Charges')

plt.xticks([0, 1], ['Non-Smoker', 'Smoker'], rotation=0)

plt.show()

A graph showing the average charge by smoking status

Description automatically generatedInsights (Figure 2):

This figure illustrates the average medical charges for smokers and non-smokers. Smokers tend to have significantly higher medical charges compared to non-smokers. It highlights the financial impact of smoking on healthcare costs, which can be a valuable insight for policymaking or health interventions.

Figure 3: BMI vs. Charges

# Plot BMI vs. charges

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

plt.scatter(df['bmi'], df['charges'], c='blue', alpha=0.5)

plt.title('BMI vs. Charges')

plt.xlabel('BMI')

plt.ylabel('Charges')

plt.show()

A blue dots on a white background

Description automatically generated

Insights (Figure 3):

This scatter plot shows the relationship between Body Mass Index (BMI) and medical charges. There is no strong linear relationship between BMI and charges, however it does show some clusters of higher charges for individuals with higher BMIs. This information can be used in understanding the impact of BMI on healthcare costs.

In summary, the figures provide insights into age distribution, the impact of smoking on charges, and the relationship between BMI and medical costs in the ECA dataset. Which can be valuable for healthcare providers, policymakers, and researchers in making informed decisions and developing strategies to manage medical costs and promote healthier lifestyles.

Question 3:

To use a decision tree, it will be broken down into three steps.

Data Preparation:

1. Load the dataset: First, load the ECA dataset containing information on medical costs.
2. Data Encoding: Encode categorical variables like 'sex', 'region', and 'smoker' into numerical values using techniques like one-hot encoding. This is necessary as decision trees require numerical data.
3. Data Splitting: Split the dataset into training and testing sets. This ensures that the model's performance can be evaluated effectively.

Building the Decision Tree:

1. Defining Dependent and Independent Variables: Set 'smoker' as the dependent variable and choose appropriate independent variables (features) that might influence the 'smoker' status, such as 'age', 'bmi', 'charges'.
2. Decision Tree Training: Train the decision tree model using the training data. The model will learn to make predictions based on the relationships between the independent and dependent variables.

Evaluation and Interpretation:

1. Model Evaluation: Assess the model's performance on the testing data using metrics like accuracy, precision and recall. Which will provide insights into how well the decision tree can predict 'smoker' status.
2. Visualizing the Tree: Visualizing the decision tree structure to understand how it makes decisions. This will allow us to understand which features are most important in determining the 'smoker' status.

By exploring the dataset with a decision tree, we can gain insights into the factors that influence an individual's smoking status. For instance, we can understand which features are more predictive of being a smoker or a non-smoker. Which can be valuable for healthcare professionals and policymakers when developing strategies to target and reduce smoking prevalence.

Question 4:

# Import necessary libraries

from sklearn.tree import DecisionTreeClassifier

from sklearn.tree import plot\_tree

import matplotlib.pyplot as plt

# Preparing the data, splitting it into features (X) and target variable (Y)

features = ['age','bmi','children',]

d = {'yes': 1, 'no': 0}

df['smoker'] = df['smoker'].map(d)

print(df)

X= df[features]

Y= df[smoker]

# Creating a decision tree classifier

clf = DecisionTreeClassifier()

# Fitting the decision tree on your data

clf.fit(X, y)

# Plotting the decision tree

plt.figure(figsize=(12, 8))

plot\_tree(clf, filled=True, feature\_names=X.columns, class\_names=["Non-Smoker", "Smoker"])

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

plt.show()

Question 5:

Even though Decision trees are traditionally employed as predictive models, but they can also be valuable tools for exploratory data analysis (EDA) by offering insights into the relationships that are found within the dataset provided.

Decision trees can be used to capture complex interactions between multiple variables. By exploring the branching structure of the tree, EDA can highlight how different variables combine to affect the target variable. Hence this can be helpful for detecting interactions that might not be evident in simple bivariate analyses.

It can identify outliers as data points that don't fit the general pattern of splits. Identifying and understanding these outliers can be part of EDA, as they might represent data quality issues or unique cases of interest.

EDA often involves searching for patterns and trends in the data. Thus, Decision trees can help uncover non-linear patterns and relationships between variables, which may be overlooked when using linear or simpler models. Another way it can be helpful is that it can handle missing data and inform EDA by showing how the missing values are treated. Which can then be used to assess how the missing data impacts the analysis.

Lastly, Decision trees can be visually appealing and easy to interpret. Visual representations of the tree structure can be used to communicate findings during EDA to stakeholders who might not be familiar with advanced statistical methods.

In summary, while decision trees are primarily predictive models, they offer valuable avenues for exploratory data analysis. It helps uncover relationships, patterns, and insights in the data beyond making predictions. This is particularly beneficial when dealing with complex, high-dimensional datasets where understanding the underlying structure is crucial. By exploring the decision tree's structure, features, and splits, EDA can extract meaningful insights that inform subsequent analyses and decision-making processes.