****

**ANL252**

**Python for Data Analytics**

**End-of-Course Assessment- July Semester 2023**

**July 2023 Presentation**

|  |  |
| --- | --- |
| **Student Name** | **PI Number** |
| Lee Wei Xuan | E2310018 |

**Submission Date: 31/10/2023**

1. The three preprocessing tasks to clean a data set are to handle missing data, data encoding (numerical to categorical) and to normalize the data. Missing data treatment is important because it could lead to inaccurate insights and reduce the accuracy of machine learning. Data encoding helps improve the efficiency of machine learning by separating small intervals with a single representative value for that interval. Sometimes binning improves accuracy in predictive models (Duca, 2021). Data normalization helps machine learning to process the data values faster by ranging them to smaller value ranges and making them easier to be compared.

Missing value

#import pandas into python

In: import pandas as pd

#uploading the csv file into python

In: data = pd.read\_csv("ECA.csv")

#Using .shape to identify the original matrix of the data set

In: data.shape

Out: (1340, 8)

#Using isnull() to find out which variable has missing value inside

In: data.isnull().any(axis=0)

missing = data.isnull().any(axis=0)

missing[missing==True].index

Out: Index(['age'], dtype='object')

#Using .drop() to remove rows with missing values instead of removing the entire column causing huge loss of data

In: data.isnull().any(axis=1)

missing = data.isnull().any(axis=1)

missing[missing==True].index

Data.cleaned = data.drop(axis=0, index = missing[missing==True].index)

Out: 1217 rows × 8 columns

Data Encoding (Binning ages into variable range) (Binning a Column With Python Pandas | Saturn Cloud Blog, 2023)

#Creating bins for numerical values, 0-20( Young ), 20-40( Middle-aged), 40-60 ( Old ), 60-80 ( Very-Old )

In: bins = [0, 20, 40, 60, 80]

labels = ['young', 'middle-aged', 'old', 'very-old']

df['age\_bin'] = pd.cut(df['age'], bins, labels=labels)

print(df)

Out: age\_bin

0 young

1 young

2 middle-aged

3 middle-aged

4 middle-aged

... ...

1335 old

1336 young

1337 young

1338 middle-aged

1339 very-old

Data Normalization

In: from sklearn.preprocessing import StandardScaler

#Identifying the variable that requires normalization

In: num\_variables = ['age', 'bmi', 'charges']

X = data[num\_variables]

scaler = StandardScaler()

X\_normalized = scaler.fit\_transform(X)

print(X\_normalized)

Out: [[-1.43896294 -0.44970968 0.29848232]

[-1.51021438 0.51146156 -0.95468473]

[-0.79769998 0.38537947 -0.72950957]

...

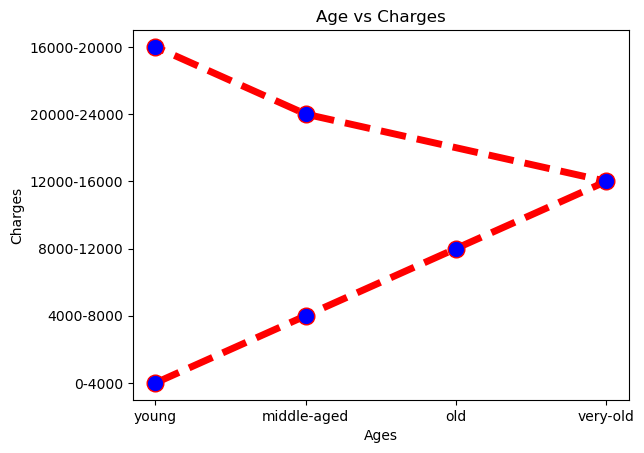
[-1.51021438 1.01578991 -0.96259744]

[-1.29646006 -0.79356992 -0.93134041]

[ 1.55359755 -0.25813041 1.31167485]]

1. Figure 1 is a line chart done to compare age with charges. This is to prove that the older you get, the higher the expenses. The line chart shows an upwards chart as the ages of patients get older. However, there is also another upwards back towards the younger patient which might be due to the outliers.

Figure 1



import matplotlib.pyplot as plt

age= df['age\_bin']

charges= df['charges\_bin']

plt.plot(age,charges, color='red', linewidth=5 ,marker='o',markerfacecolor='blue',markersize=12, linestyle='dashed')

plt.xlabel('Ages')

plt.ylabel('Charges')

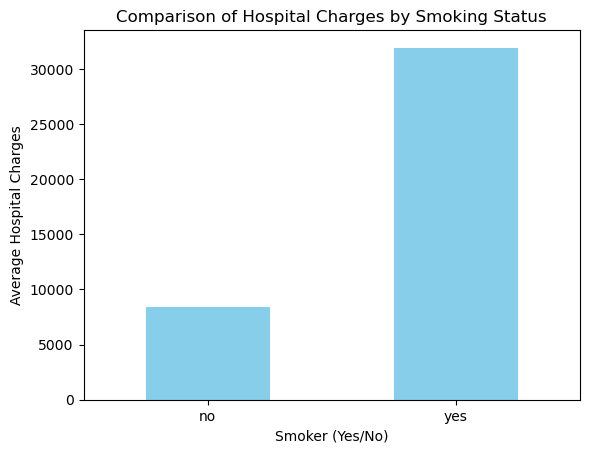
plt.title('Age vs Charges')

plt.show()

Figure 2 shows a bar chart that compares smoker status and the average hospital

charges. As shown from Figure 2, for non-smokers, the average hospital bill is at $8,434 across all ages. On the other hand, for smokers, the average hospital bill is at $31,932 across all ages. This proves that the smoker will incur a higher hospital bill over a lifetime.

Figure 2



import pandas as pd

import matplotlib.pyplot as plt

# Reading data from a CSV file

file\_path = 'ECA.csv' # Replace with your CSV file path

df = pd.read\_csv(file\_path)

# Extracting two variables from the the data

smoker = df['smoker']

charges = df['charges']

# Calculate the mean of the "charges" variable

mean\_charges = charges.mean()

print(mean\_charges)

# Group the data by "smoker" and calculate the mean or sum of charges for each group

grouped = df.groupby('smoker')['charges'].mean()

# Create a bar chart

grouped.plot(kind='bar', color='skyblue')

plt.xlabel('Smoker (Yes/No)')

plt.ylabel('Average Hospital Charges')

plt.title('Comparison of Hospital Charges by Smoking Status')

plt.xticks(rotation=0)

# Show the chart

plt.show()

print(grouped)

smoker

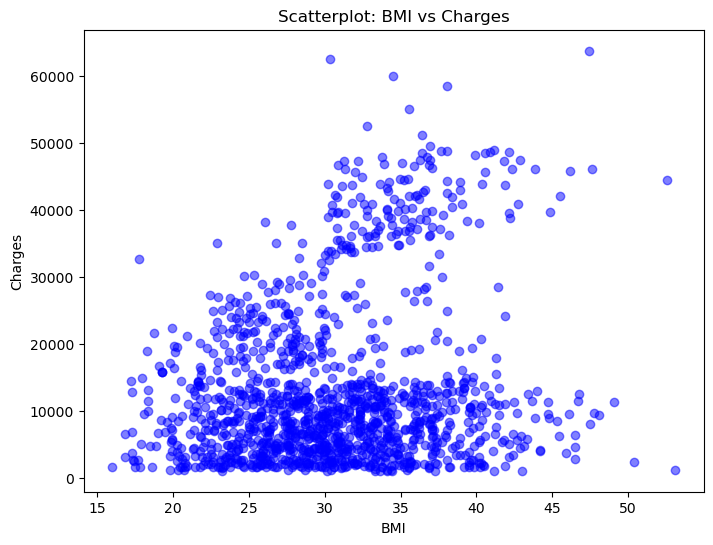
no 8434.268298

yes 31932.626521

Name: charges, dtype: float64

Figure 3 shows a scatterplot that compares BMI and charges to see if there is any correlation between higher bmi and higher hospital bill. As shown, we can see that there is a vast population of between the bmi of 25 to 35 which is in the overweight and obese range. This does not prove that higher bmi will equate to a higher hospital charge. However, this shows that most of the patients who have visited the hospital are mostly in the overweight and obese range.

Figure 3



import pandas as pd

import matplotlib.pyplot as plt

# Reading data from a CSV file

file\_path = 'ECA.csv' # Replace with your CSV file path

df = pd.read\_csv(file\_path)

# Extracting two variables from the the data

bmi = df['bmi']

charges = df['charges']

# Creating a scatterplot

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

plt.scatter(bmi, charges, alpha=0.5, color='b', marker='o')

plt.title('Scatterplot: BMI vs Charges')

plt.xlabel('BMI')

plt.ylabel('Charges')

# Plotting the graph

plt.show()

1. To predict if a person is a smoker by making the ‘smoker’ as a dependent variable. We would first have to load in the data with the different variables. Afterwards, we would have to treat the missing values in the data set as it might affect the result of the decision tree by filtering out the rows with missing data. Next, we must select the independent features that we want to use for this decision tree analysis. Data splitting would be the next step, the dataset will be split into training and testing sets. We will then start to build the decision tree model and visualize the tree structure to understand if the model has learnt to predict. Lastly, we will analyze the tree model and gain insights into the factors that influence whether a person is a smoker or not. (Python Machine Learning Decision Tree, n.d.)

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier, plot\_tree

from sklearn.metrics import accuracy\_score, classification\_report

from sklearn.preprocessing import LabelEncoder

import matplotlib.pyplot as plt

# Load your dataset (replace 'your\_data.csv' with the actual file path)

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

# Handle missing values

missing = df.isnull().any(axis=1) # Check for missing values in rows

df\_cleaned = df[~missing] # Filter out rows with missing values

# Encode categorical variables into numerical format

le = LabelEncoder()

df\_cleaned['smoker'] = le.fit\_transform(df\_cleaned['smoker']) # Encode 'smoker' as 0 or 1

variables = ['age', 'bmi', 'charges']

# Define independent variables (features)

X = df\_cleaned[variables] # Use df\_cleaned instead of df

# Define the dependent variable (target)

y = df\_cleaned['smoker'] # Use df\_cleaned instead of df

dtree = DecisionTreeClassifier()

dtree = dtree.fit(X, y)

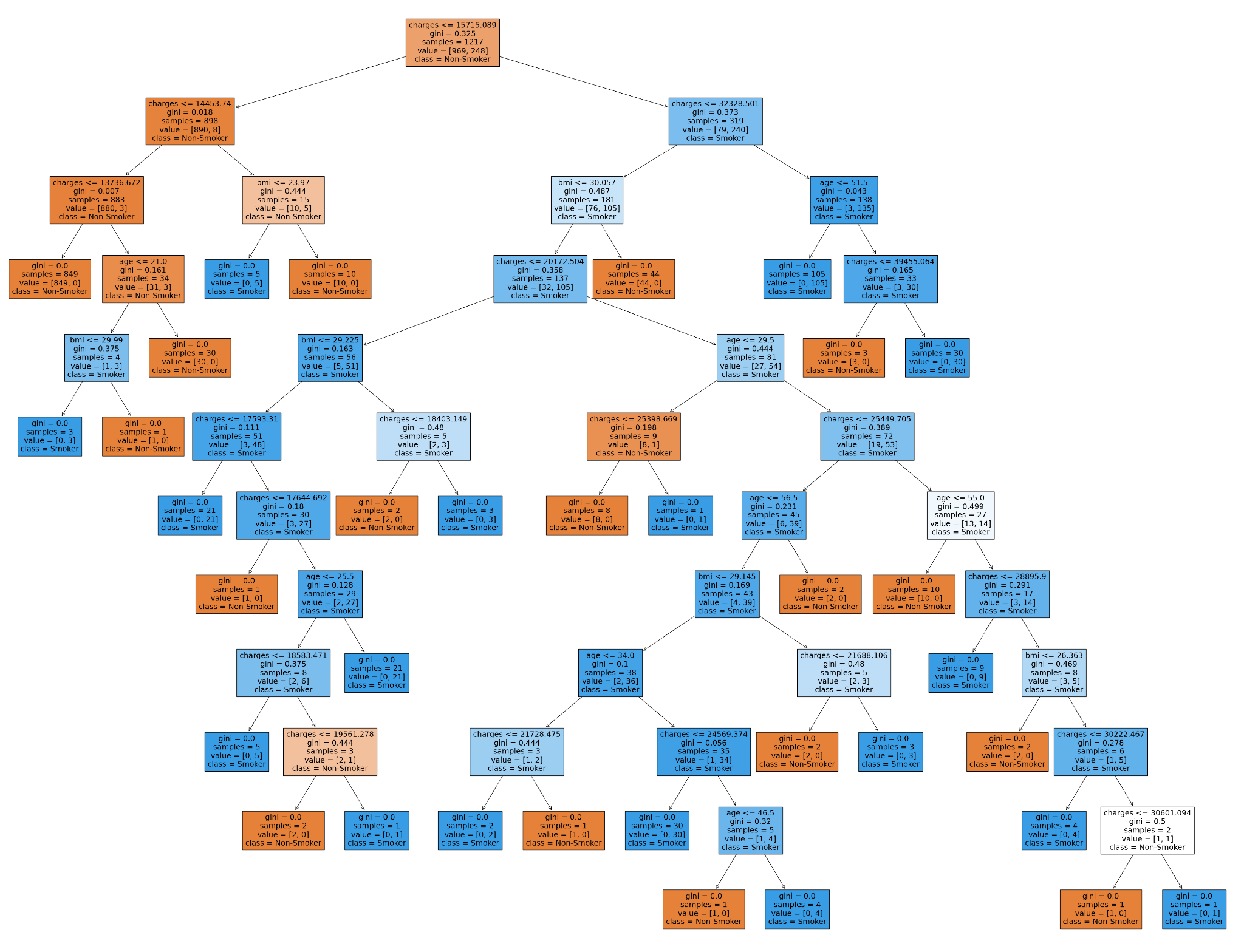
# Plot the decision tree

plt.figure(figsize=(52, 40)) # Set the figure size (optional)

plot\_tree(dtree, feature\_names=variables, class\_names=['Non-Smoker', 'Smoker'], filled=True)

plt.show()

4.



This is a decision tree diagram plotted from question 3. We can see that for ‘smoker’ being the dependent variable. The decision tree diagram begins at the root node where ‘smoker’ is used as the initial attribute that we are trying to predict. It is then split into 2 ranges of hospital charges, charges below $14,453 on the left and the rest on the right.  
  
The complexity of the decision tree starts after the first parent node where BMI was taken into account of. This decision tree provides valuable insights on the probability of a client being a smoker. In conclusion, this decision tree effectively separates patients into ‘smoker’ and ‘non-smoker’ based on age, bmi and charges. This offers hospitals to apply this to their healthcare system and for insurance companies to do risk assessments.

5. Decision trees can be effective in exploratory data analysis. This is because decision trees can highlight the importance of different variables and identify the key variable driving trends in the dataset. It can be easily visualized and understandable too because it is displayed in a flowchart manner (Kapil, 2022). Decision trees also can help create decision boundaries which means that it is able to solve nonlinear problems (Kapil, 2022). However, there might be some downsides of using a decision tree, despite the decision tree being efficient in predicting, the decision tree might suffer dramatical change if the data has some changes to it (Inside Learning Machines, 2023).

In conclusion, though there are some flaws when it comes to using a decision tree to predict the outcome and the importance of a certain variable, it is still a powerful tool that can easily be self explanatory for data analysis.

References

1. Binning a Column with Python Pandas | Saturn Cloud Blog. (2023, June 19). Binning a Column With Python Pandas | Saturn Cloud Blog. Retrieved from <https://saturncloud.io/blog/binning-a-column-with-python-pandas/>
2. Duca, A. L. (2021, October 24). Data Preprocessing with Python Pandas — Part 5 Binning. Medium. Retrieved from <https://towardsdatascience.com/data-preprocessing-with-python-pandas-part-5-binning-c5bd5fd1b950>
3. Python Machine Learning Decision Tree. (n.d.). Python Machine Learning Decision Tree. <https://www.w3schools.com/python/python_ml_decision_tree.asp>
4. Kapil, A. R. (2022, October 1). Advantages and disadvantages of decision tree in machine learning. Blogs & Updates on Data Science, Business Analytics, AI Machine Learning. <https://www.analytixlabs.co.in/blog/decision-tree-algorithm/>
5. 8 Key Advantages and Disadvantages of Decision Trees - Inside Learning Machines. (2023, April 7). Inside Learning Machines. <https://insidelearningmachines.com/advantages_and_disadvantages_of_decision_trees/>