```python
# Importing necessary libraries
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report, confusion_matrix

df = pd.read_csv("/content/2011-Q1-cabi-trip-history-data.csv")
df.head()

df["Duration"] = df['Duration'].str.replace('min', '')
df["Duration"] = df['Duration'].str.replace('sec', '')
df["Duration"] = df['Duration'].str.replace('h', '')
df["Duration"] = df['Duration'].str.replace(' ', '')
df["Duration"] = df['Duration'].str.replace('.', '').astype(float)

address=df['End station'].unique()
print(address)
df = df.dropna()

from sklearn.preprocessing import LabelEncoder
gle = LabelEncoder()
address_labels = gle.fit_transform(df['End station'])
address_labels = gle.fit_transform(df['Start station'])
address_mappings = {index: label for index, label in
                  enumerate(gle.classes_)}
address_mappings

df['End station'] = address_labels

gle = LabelEncoder()
address_labels = gle.fit_transform(df['Start station'])
address_mappings = {index: label for index, label in
                  enumerate(gle.classes_)}
address_mappings

df['Start station'] = address_labels

df.head()

gle = LabelEncoder()
bike_labels = gle.fit_transform(df['Bike#'])
bike_mappings = {index: label for index, label in
                  enumerate(gle.classes_)}
bike_mappings

df['Bike#'] = bike_labels

# Assign X and y
X = df.iloc[:,[0,3,5]].values
y = df.iloc[:, -1].values
```

```
print(y)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=0)

tree = DecisionTreeClassifier(max_leaf_nodes=3, random_state=0)
tree.fit(X_train, y_train)
y_pred = tree.predict(X_test)

# Check the Accuracy
score = metrics.accuracy_score(y_test, y_pred)
print("Accuracy of our model is: {:.1f}%".format(score*100))
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```