# 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("2011-Q1-cabi-trip-history-data\_2011-Q1-cabi-trip-history-data.csv")  
df.head()

Duration Start date End date \  
0 0h 1min. 50sec. 3/31/2011 23:58 4/1/2011 0:00   
1 0h 16min. 21sec. 3/31/2011 23:52 4/1/2011 0:08   
2 0h 3min. 19sec. 3/31/2011 23:47 3/31/2011 23:50   
3 0h 5min. 44sec. 3/31/2011 23:45 3/31/2011 23:50   
4 0h 8min. 42sec. 3/31/2011 23:34 3/31/2011 23:43   
  
 Start station \  
0 14th & Harvard St NW (31105)   
1 19th & L St NW (31224)   
2 Lincoln Park / 13th & East Capitol St NE (31619)   
3 14th & R St NW (31202)   
4 Park Rd & Holmead Pl NW (31602)   
  
 End station Bike# Member Type   
0 16th & Harvard St NW (31103) W00749 Registered   
1 7th & Water St SW / SW Waterfront (31609) W01048 Casual   
2 13th & H St NE (31611) W00340 Registered   
3 Massachusetts Ave & Dupont Circle NW (31200) W00981 Registered   
4 15th & P St NW (31201) W00449 Registered

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()

['16th & Harvard St NW (31103)'  
 '7th & Water St SW / SW Waterfront (31609)' '13th & H St NE (31611)'  
 'Massachusetts Ave & Dupont Circle NW (31200)' '15th & P St NW (31201)'  
 'Adams Mill & Columbia Rd NW (31104)' '21st & M St NW (31212)'  
 '14th & V St NW (31101)' '7th & T St NW (31109)'  
 'Georgia Ave and Fairmont St NW (31207)' '20th & E St NW (31204)'  
 '14th St & Spring Rd NW (31401)' '14th & Rhode Island Ave NW (31203)'  
 '17th & Corcoran St NW (31214)' '25th St & Pennsylvania Ave NW (31237)'  
 '10th & Monroe St NE (31504)' 'C & O Canal & Wisconsin Ave NW (31225)'  
 '18th & M St NW (31221)' '11th & Kenyon St NW (31102)'  
 '3rd & H St NE (31616)' 'Park Rd & Holmead Pl NW (31602)'  
 'Calvert St & Woodley Pl NW (31106)'  
 'Georgia & New Hampshire Ave NW (31400)' '10th & U St NW (31111)'  
 'Potomac & Pennsylvania Ave SE (31606)'  
 'Columbus Circle / Union Station (31623)' 'Florida Ave & R St NW (31503)'  
 '14th & D St SE (31607)' 'McPherson Square - 14th & H St NW (31216)'  
 'Lincoln Park / 13th & East Capitol St NE (31619)'  
 '13th & D St NE (31622)' '21st & I St NW (31205)'  
 '16th & U St NW (31229)' '12th & Army Navy Dr (31008)'  
 '14th & R St NW (31202)' '5th St & K St NW (31600)'  
 '19th St & Pennsylvania Ave NW (31100)'  
 'Wisconsin Ave & Macomb St NW (31302)' '4th & East Capitol St NE (31618)'  
 '3rd & D St SE (31605)' 'Van Ness Metro / UDC (31300)'  
 'Tenleytown / Wisconsin Ave & Albemarle St NW (31303)'  
 '1st & M St NE (31603)' '36th & Calvert St NW / Glover Park (31304)'  
 'S Glebe & Potomac Ave (31010)' '8th & H St NW (31228)'  
 'New York Ave & 15th St NW (31222)' '13th St & New York Ave NW (31227)'  
 'Lamont & Mt Pleasant NW (31107)' 'Eckington Pl & Q St NE (31505)'  
 '4th St & Massachusetts Ave NW (31604)'  
 'Eastern Market / 7th & North Carolina Ave SE (31610)'  
 'Convention Center / 7th & M St NW (31223)'  
 'M St & New Jersey Ave SE (31208)' 'Kennedy Center (31211)'  
 '14th & Harvard St NW (31105)'  
 'Connecticut Ave & Newark St NW / Cleveland Park (31305)'  
 '17th & K St NW [formerly 17th & L St NW] (31213)'  
 '14th & G St NW (31238)' '4th & M St SW (31108)'  
 'Eastern Market Metro / Pennsylvania Ave & 7th St SE (31613)'  
 "L'Enfant Plaza / 7th & C St SW (31218)" '19th & L St NW (31224)'  
 '20th St & Florida Ave NW (31110)'  
 'Harvard St & Adams Mill Rd NW (31112)'  
 'Ward Circle / American University (31301)'  
 '19th St & Constitution Ave NW (31235)'  
 'Bladensburg Rd & Benning Rd NE (31617)' '5th & F St NW (31620)'  
 '27th & Crystal Dr (31009)' '18th & Hayes St (31004)'  
 'North Capitol St & F St NW (31624)' 'Good Hope & Naylor Rd SE (31700)'  
 '15th & Crystal Dr (31003)' '34th St & Wisconsin Ave NW (31226)'  
 '37th & O St NW / Georgetown University (31236)'  
 '20th & Crystal Dr (31002)' '18th & Bell St (31007)'  
 '23rd & Crystal Dr (31011)' 'USDA / 12th & Independence Ave SW (31217)'  
 'US Dept of State / Virginia Ave & 21st St NW (31220)'  
 '12th & Newton St NE (31501)' 'S Joyce & Army Navy Dr (31006)'  
 '12th & Hayes St (31001)' 'John McCormack Dr & Michigan Ave NE (31502)'  
 '10th St & Constitution Ave NW (31219)'  
 'Georgetown Harbor / 30th St NW (31215)'  
 '19th & East Capitol St SE (31601)' '1st & N ST SE (31209)'  
 '19th & E Street NW (31206)' 'Pennsylvania & Minnesota Ave SE (31805)'  
 'Anacostia Library (31804)' '8th & Eye St SE / Barracks Row (31608)'  
 '23rd & Eads (31013)' '26th & Crystal Dr (31012)'  
 'Anacostia Metro (31801)' '15th & Hayes St (31005)'  
 '4th & Adams St NE (31500)' '20th & Bell St (31000)'  
 '14th St Heights / 14th & Crittenden St NW (31402)' nan  
 'Minnesota Ave Metro/DOES (31703)'  
 'Randle Circle & Minnesota Ave NE (31702)'  
 'Good Hope Rd & MLK Ave SE (31802)' 'Benning Branch Library (31705)'  
 'Nannie Helen Burroughs & Minnesota Ave NE (31704)'  
 'Alta Bicycle Share Demonstration Station (31999)']

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

{0: '10th & Monroe St NE (31504)',  
 1: '10th & U St NW (31111)',  
 2: '10th St & Constitution Ave NW (31219)',  
 3: '11th & Kenyon St NW (31102)',  
 4: '12th & Army Navy Dr (31008)',  
 5: '12th & Hayes St (31001)',  
 6: '12th & Newton St NE (31501)',  
 7: '13th & D St NE (31622)',  
 8: '13th & H St NE (31611)',  
 9: '13th St & New York Ave NW (31227)',  
 10: '14th & D St SE (31607)',  
 11: '14th & G St NW (31238)',  
 12: '14th & Harvard St NW (31105)',  
 13: '14th & R St NW (31202)',  
 14: '14th & Rhode Island Ave NW (31203)',  
 15: '14th & V St NW (31101)',  
 16: '14th St & Spring Rd NW (31401)',  
 17: '14th St Heights / 14th & Crittenden St NW (31402)',  
 18: '15th & Crystal Dr (31003)',  
 19: '15th & Hayes St (31005)',  
 20: '15th & P St NW (31201)',  
 21: '16th & Harvard St NW (31103)',  
 22: '16th & U St NW (31229)',  
 23: '17th & Corcoran St NW (31214)',  
 24: '17th & K St NW [formerly 17th & L St NW] (31213)',  
 25: '18th & Bell St (31007)',  
 26: '18th & Hayes St (31004)',  
 27: '18th & M St NW (31221)',  
 28: '19th & E Street NW (31206)',  
 29: '19th & East Capitol St SE (31601)',  
 30: '19th & L St NW (31224)',  
 31: '19th St & Constitution Ave NW (31235)',  
 32: '19th St & Pennsylvania Ave NW (31100)',  
 33: '1st & M St NE (31603)',  
 34: '1st & N ST SE (31209)',  
 35: '20th & Bell St (31000)',  
 36: '20th & Crystal Dr (31002)',  
 37: '20th & E St NW (31204)',  
 38: '20th St & Florida Ave NW (31110)',  
 39: '21st & I St NW (31205)',  
 40: '21st & M St NW (31212)',  
 41: '23rd & Crystal Dr (31011)',  
 42: '23rd & Eads (31013)',  
 43: '25th St & Pennsylvania Ave NW (31237)',  
 44: '26th & Crystal Dr (31012)',  
 45: '27th & Crystal Dr (31009)',  
 46: '34th St & Wisconsin Ave NW (31226)',  
 47: '36th & Calvert St NW / Glover Park (31304)',  
 48: '37th & O St NW / Georgetown University (31236)',  
 49: '3rd & D St SE (31605)',  
 50: '3rd & H St NE (31616)',  
 51: '4th & Adams St NE (31500)',  
 52: '4th & East Capitol St NE (31618)',  
 53: '4th & M St SW (31108)',  
 54: '4th St & Massachusetts Ave NW (31604)',  
 55: '5th & F St NW (31620)',  
 56: '5th St & K St NW (31600)',  
 57: '7th & T St NW (31109)',  
 58: '7th & Water St SW / SW Waterfront (31609)',  
 59: '8th & Eye St SE / Barracks Row (31608)',  
 60: '8th & H St NW (31228)',  
 61: 'Adams Mill & Columbia Rd NW (31104)',  
 62: 'Alta Bicycle Share Demonstration Station (31999)',  
 63: 'Anacostia Library (31804)',  
 64: 'Anacostia Metro (31801)',  
 65: 'Benning Branch Library (31705)',  
 66: 'Bladensburg Rd & Benning Rd NE (31617)',  
 67: 'C & O Canal & Wisconsin Ave NW (31225)',  
 68: 'Calvert St & Woodley Pl NW (31106)',  
 69: 'Columbus Circle / Union Station (31623)',  
 70: 'Connecticut Ave & Newark St NW / Cleveland Park (31305)',  
 71: 'Convention Center / 7th & M St NW (31223)',  
 72: 'Eastern Market / 7th & North Carolina Ave SE (31610)',  
 73: 'Eastern Market Metro / Pennsylvania Ave & 7th St SE (31613)',  
 74: 'Eckington Pl & Q St NE (31505)',  
 75: 'Florida Ave & R St NW (31503)',  
 76: 'Georgetown Harbor / 30th St NW (31215)',  
 77: 'Georgia & New Hampshire Ave NW (31400)',  
 78: 'Georgia Ave and Fairmont St NW (31207)',  
 79: 'Good Hope & Naylor Rd SE (31700)',  
 80: 'Good Hope Rd & MLK Ave SE (31802)',  
 81: 'Harvard St & Adams Mill Rd NW (31112)',  
 82: 'John McCormack Dr & Michigan Ave NE (31502)',  
 83: 'Kennedy Center (31211)',  
 84: "L'Enfant Plaza / 7th & C St SW (31218)",  
 85: 'Lamont & Mt Pleasant NW (31107)',  
 86: 'Lincoln Park / 13th & East Capitol St NE (31619)',  
 87: 'M St & New Jersey Ave SE (31208)',  
 88: 'Massachusetts Ave & Dupont Circle NW (31200)',  
 89: 'McPherson Square - 14th & H St NW (31216)',  
 90: 'Minnesota Ave Metro/DOES (31703)',  
 91: 'Nannie Helen Burroughs & Minnesota Ave NE (31704)',  
 92: 'New York Ave & 15th St NW (31222)',  
 93: 'North Capitol St & F St NW (31624)',  
 94: 'Park Rd & Holmead Pl NW (31602)',  
 95: 'Pennsylvania & Minnesota Ave SE (31805)',  
 96: 'Potomac & Pennsylvania Ave SE (31606)',  
 97: 'Randle Circle & Minnesota Ave NE (31702)',  
 98: 'S Glebe & Potomac Ave (31010)',  
 99: 'S Joyce & Army Navy Dr (31006)',  
 100: 'Tenleytown / Wisconsin Ave & Albemarle St NW (31303)',  
 101: 'US Dept of State / Virginia Ave & 21st St NW (31220)',  
 102: 'USDA / 12th & Independence Ave SW (31217)',  
 103: 'Van Ness Metro / UDC (31300)',  
 104: 'Ward Circle / American University (31301)',  
 105: 'Wisconsin Ave & Macomb St NW (31302)'}

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

{0: '10th & Monroe St NE (31504)',  
 1: '10th & U St NW (31111)',  
 2: '10th St & Constitution Ave NW (31219)',  
 3: '11th & Kenyon St NW (31102)',  
 4: '12th & Army Navy Dr (31008)',  
 5: '12th & Hayes St (31001)',  
 6: '12th & Newton St NE (31501)',  
 7: '13th & D St NE (31622)',  
 8: '13th & H St NE (31611)',  
 9: '13th St & New York Ave NW (31227)',  
 10: '14th & D St SE (31607)',  
 11: '14th & G St NW (31238)',  
 12: '14th & Harvard St NW (31105)',  
 13: '14th & R St NW (31202)',  
 14: '14th & Rhode Island Ave NW (31203)',  
 15: '14th & V St NW (31101)',  
 16: '14th St & Spring Rd NW (31401)',  
 17: '14th St Heights / 14th & Crittenden St NW (31402)',  
 18: '15th & Crystal Dr (31003)',  
 19: '15th & Hayes St (31005)',  
 20: '15th & P St NW (31201)',  
 21: '16th & Harvard St NW (31103)',  
 22: '16th & U St NW (31229)',  
 23: '17th & Corcoran St NW (31214)',  
 24: '17th & K St NW [formerly 17th & L St NW] (31213)',  
 25: '18th & Bell St (31007)',  
 26: '18th & Hayes St (31004)',  
 27: '18th & M St NW (31221)',  
 28: '19th & E Street NW (31206)',  
 29: '19th & East Capitol St SE (31601)',  
 30: '19th & L St NW (31224)',  
 31: '19th St & Constitution Ave NW (31235)',  
 32: '19th St & Pennsylvania Ave NW (31100)',  
 33: '1st & M St NE (31603)',  
 34: '1st & N ST SE (31209)',  
 35: '20th & Bell St (31000)',  
 36: '20th & Crystal Dr (31002)',  
 37: '20th & E St NW (31204)',  
 38: '20th St & Florida Ave NW (31110)',  
 39: '21st & I St NW (31205)',  
 40: '21st & M St NW (31212)',  
 41: '23rd & Crystal Dr (31011)',  
 42: '23rd & Eads (31013)',  
 43: '25th St & Pennsylvania Ave NW (31237)',  
 44: '26th & Crystal Dr (31012)',  
 45: '27th & Crystal Dr (31009)',  
 46: '34th St & Wisconsin Ave NW (31226)',  
 47: '36th & Calvert St NW / Glover Park (31304)',  
 48: '37th & O St NW / Georgetown University (31236)',  
 49: '3rd & D St SE (31605)',  
 50: '3rd & H St NE (31616)',  
 51: '4th & Adams St NE (31500)',  
 52: '4th & East Capitol St NE (31618)',  
 53: '4th & M St SW (31108)',  
 54: '4th St & Massachusetts Ave NW (31604)',  
 55: '5th & F St NW (31620)',  
 56: '5th St & K St NW (31600)',  
 57: '7th & T St NW (31109)',  
 58: '7th & Water St SW / SW Waterfront (31609)',  
 59: '8th & Eye St SE / Barracks Row (31608)',  
 60: '8th & H St NW (31228)',  
 61: 'Adams Mill & Columbia Rd NW (31104)',  
 62: 'Alta Bicycle Share Demonstration Station (31999)',  
 63: 'Anacostia Library (31804)',  
 64: 'Anacostia Metro (31801)',  
 65: 'Benning Branch Library (31705)',  
 66: 'Bladensburg Rd & Benning Rd NE (31617)',  
 67: 'C & O Canal & Wisconsin Ave NW (31225)',  
 68: 'Calvert St & Woodley Pl NW (31106)',  
 69: 'Columbus Circle / Union Station (31623)',  
 70: 'Connecticut Ave & Newark St NW / Cleveland Park (31305)',  
 71: 'Convention Center / 7th & M St NW (31223)',  
 72: 'Eastern Market / 7th & North Carolina Ave SE (31610)',  
 73: 'Eastern Market Metro / Pennsylvania Ave & 7th St SE (31613)',  
 74: 'Eckington Pl & Q St NE (31505)',  
 75: 'Florida Ave & R St NW (31503)',  
 76: 'Georgetown Harbor / 30th St NW (31215)',  
 77: 'Georgia & New Hampshire Ave NW (31400)',  
 78: 'Georgia Ave and Fairmont St NW (31207)',  
 79: 'Good Hope & Naylor Rd SE (31700)',  
 80: 'Good Hope Rd & MLK Ave SE (31802)',  
 81: 'Harvard St & Adams Mill Rd NW (31112)',  
 82: 'John McCormack Dr & Michigan Ave NE (31502)',  
 83: 'Kennedy Center (31211)',  
 84: "L'Enfant Plaza / 7th & C St SW (31218)",  
 85: 'Lamont & Mt Pleasant NW (31107)',  
 86: 'Lincoln Park / 13th & East Capitol St NE (31619)',  
 87: 'M St & New Jersey Ave SE (31208)',  
 88: 'Massachusetts Ave & Dupont Circle NW (31200)',  
 89: 'McPherson Square - 14th & H St NW (31216)',  
 90: 'Minnesota Ave Metro/DOES (31703)',  
 91: 'Nannie Helen Burroughs & Minnesota Ave NE (31704)',  
 92: 'New York Ave & 15th St NW (31222)',  
 93: 'North Capitol St & F St NW (31624)',  
 94: 'Park Rd & Holmead Pl NW (31602)',  
 95: 'Pennsylvania & Minnesota Ave SE (31805)',  
 96: 'Potomac & Pennsylvania Ave SE (31606)',  
 97: 'Randle Circle & Minnesota Ave NE (31702)',  
 98: 'S Glebe & Potomac Ave (31010)',  
 99: 'S Joyce & Army Navy Dr (31006)',  
 100: 'Tenleytown / Wisconsin Ave & Albemarle St NW (31303)',  
 101: 'US Dept of State / Virginia Ave & 21st St NW (31220)',  
 102: 'USDA / 12th & Independence Ave SW (31217)',  
 103: 'Van Ness Metro / UDC (31300)',  
 104: 'Ward Circle / American University (31301)',  
 105: 'Wisconsin Ave & Macomb St NW (31302)'}

df['Start station'] = address\_labels

df.head()

Duration Start date End date Start station End station \  
0 150.0 3/31/2011 23:58 4/1/2011 0:00 12 12   
1 1621.0 3/31/2011 23:52 4/1/2011 0:08 30 30   
2 319.0 3/31/2011 23:47 3/31/2011 23:50 86 86   
3 544.0 3/31/2011 23:45 3/31/2011 23:50 13 13   
4 842.0 3/31/2011 23:34 3/31/2011 23:43 94 94   
  
 Bike# Member Type   
0 W00749 Registered   
1 W01048 Casual   
2 W00340 Registered   
3 W00981 Registered   
4 W00449 Registered

gle = LabelEncoder()  
bike\_labels = gle.fit\_transform(df['Bike#'])  
bike\_mappings = {index: label for index, label in  
 enumerate(gle.classes\_)}  
bike\_mappings

{0: '? (0x26B3BBA8)',  
 1: '? (0x3EB026B9)',  
 2: '? (0x4752DD3A)',  
 3: '? (0x7C120F6A)',  
 4: '? (0x9A5FEA16)',  
 5: '? (0xAAC5A4C0)',  
 6: '? (0xEBA95C18)',  
 7: '? (0xEE8DADF1)',  
 8: '? (0xFAF0B948)',  
 9: 'W00005',  
 10: 'W00006',  
 11: 'W00007',  
 12: 'W00008',  
 13: 'W00009',  
 14: 'W00010',  
 15: 'W00011',  
 16: 'W00012',  
 17: 'W00013',  
 18: 'W00014',  
 19: 'W00015',  
 20: 'W00017',  
 21: 'W00018',  
 22: 'W00019',  
 23: 'W00021',  
 24: 'W00024',  
 25: 'W00025',  
 26: 'W00026',  
 27: 'W00027',  
 28: 'W00028',  
 29: 'W00030',  
 30: 'W00031',  
 31: 'W00032',  
 32: 'W00033',  
 33: 'W00035',  
 34: 'W00036',  
 35: 'W00037',  
 36: 'W00038',  
 37: 'W00039',  
 38: 'W00040',  
 39: 'W00041',  
 40: 'W00042',  
 41: 'W00043',  
 42: 'W00044',  
 43: 'W00045',  
 44: 'W00046',  
 45: 'W00048',  
 46: 'W00049',  
 47: 'W00050',  
 48: 'W00051',  
 49: 'W00055',  
 50: 'W00057',  
 51: 'W00058',  
 52: 'W00060',  
 53: 'W00061',  
 54: 'W00062',  
 55: 'W00063',  
 56: 'W00064',  
 57: 'W00065',  
 58: 'W00066',  
 59: 'W00067',  
 60: 'W00068',  
 61: 'W00069',  
 62: 'W00070',  
 63: 'W00071',  
 64: 'W00072',  
 65: 'W00073',  
 66: 'W00074',  
 67: 'W00075',  
 68: 'W00080',  
 69: 'W00081',  
 70: 'W00082',  
 71: 'W00083',  
 72: 'W00084',  
 73: 'W00085',  
 74: 'W00086',  
 75: 'W00087',  
 76: 'W00088',  
 77: 'W00089',  
 78: 'W00091',  
 79: 'W00092',  
 80: 'W00094',  
 81: 'W00095',  
 82: 'W00096',  
 83: 'W00097',  
 84: 'W00098',  
 85: 'W00099',  
 86: 'W00100',  
 87: 'W00101',  
 88: 'W00102',  
 89: 'W00103',  
 90: 'W00105',  
 91: 'W00106',  
 92: 'W00107',  
 93: 'W00109',  
 94: 'W00110',  
 95: 'W00111',  
 96: 'W00112',  
 97: 'W00113',  
 98: 'W00114',  
 99: 'W00115',  
 100: 'W00117',  
 101: 'W00118',  
 102: 'W00120',  
 103: 'W00121',  
 104: 'W00122',  
 105: 'W00123',  
 106: 'W00124',  
 107: 'W00127',  
 108: 'W00128',  
 109: 'W00129',  
 110: 'W00130',  
 111: 'W00132',  
 112: 'W00133',  
 113: 'W00134',  
 114: 'W00136',  
 115: 'W00137',  
 116: 'W00138',  
 117: 'W00140',  
 118: 'W00142',  
 119: 'W00143',  
 120: 'W00145',  
 121: 'W00146',  
 122: 'W00147',  
 123: 'W00148',  
 124: 'W00149',  
 125: 'W00150',  
 126: 'W00153',  
 127: 'W00154',  
 128: 'W00155',  
 129: 'W00156',  
 130: 'W00157',  
 131: 'W00158',  
 132: 'W00159',  
 133: 'W00163',  
 134: 'W00164',  
 135: 'W00165',  
 136: 'W00166',  
 137: 'W00167',  
 138: 'W00168',  
 139: 'W00169',  
 140: 'W00170',  
 141: 'W00171',  
 142: 'W00172',  
 143: 'W00173',  
 144: 'W00174',  
 145: 'W00175',  
 146: 'W00176',  
 147: 'W00177',  
 148: 'W00179',  
 149: 'W00180',  
 150: 'W00181',  
 151: 'W00182',  
 152: 'W00183',  
 153: 'W00184',  
 154: 'W00185',  
 155: 'W00186',  
 156: 'W00188',  
 157: 'W00189',  
 158: 'W00190',  
 159: 'W00191',  
 160: 'W00192',  
 161: 'W00193',  
 162: 'W00194',  
 163: 'W00195',  
 164: 'W00196',  
 165: 'W00197',  
 166: 'W00198',  
 167: 'W00199',  
 168: 'W00200',  
 169: 'W00201',  
 170: 'W00202',  
 171: 'W00203',  
 172: 'W00204',  
 173: 'W00205',  
 174: 'W00206',  
 175: 'W00207',  
 176: 'W00208',  
 177: 'W00209',  
 178: 'W00210',  
 179: 'W00211',  
 180: 'W00212',  
 181: 'W00213',  
 182: 'W00214',  
 183: 'W00215',  
 184: 'W00216',  
 185: 'W00217',  
 186: 'W00218',  
 187: 'W00219',  
 188: 'W00220',  
 189: 'W00221',  
 190: 'W00222',  
 191: 'W00224',  
 192: 'W00225',  
 193: 'W00226',  
 194: 'W00228',  
 195: 'W00229',  
 196: 'W00230',  
 197: 'W00232',  
 198: 'W00233',  
 199: 'W00234',  
 200: 'W00235',  
 201: 'W00236',  
 202: 'W00238',  
 203: 'W00239',  
 204: 'W00240',  
 205: 'W00242',  
 206: 'W00244',  
 207: 'W00245',  
 208: 'W00246',  
 209: 'W00247',  
 210: 'W00249',  
 211: 'W00251',  
 212: 'W00252',  
 213: 'W00253',  
 214: 'W00254',  
 215: 'W00255',  
 216: 'W00256',  
 217: 'W00257',  
 218: 'W00259',  
 219: 'W00260',  
 220: 'W00263',  
 221: 'W00264',  
 222: 'W00266',  
 223: 'W00267',  
 224: 'W00268',  
 225: 'W00270',  
 226: 'W00272',  
 227: 'W00273',  
 228: 'W00274',  
 229: 'W00275',  
 230: 'W00276',  
 231: 'W00277',  
 232: 'W00278',  
 233: 'W00279',  
 234: 'W00280',  
 235: 'W00281',  
 236: 'W00282',  
 237: 'W00284',  
 238: 'W00285',  
 239: 'W00286',  
 240: 'W00287',  
 241: 'W00289',  
 242: 'W00290',  
 243: 'W00291',  
 244: 'W00293',  
 245: 'W00294',  
 246: 'W00295',  
 247: 'W00301',  
 248: 'W00302',  
 249: 'W00303',  
 250: 'W00304',  
 251: 'W00306',  
 252: 'W00307',  
 253: 'W00308',  
 254: 'W00309',  
 255: 'W00310',  
 256: 'W00312',  
 257: 'W00314',  
 258: 'W00316',  
 259: 'W00317',  
 260: 'W00318',  
 261: 'W00319',  
 262: 'W00320',  
 263: 'W00322',  
 264: 'W00324',  
 265: 'W00325',  
 266: 'W00326',  
 267: 'W00328',  
 268: 'W00331',  
 269: 'W00332',  
 270: 'W00333',  
 271: 'W00334',  
 272: 'W00335',  
 273: 'W00336',  
 274: 'W00339',  
 275: 'W00340',  
 276: 'W00341',  
 277: 'W00342',  
 278: 'W00343',  
 279: 'W00344',  
 280: 'W00345',  
 281: 'W00346',  
 282: 'W00347',  
 283: 'W00348',  
 284: 'W00349',  
 285: 'W00350',  
 286: 'W00351',  
 287: 'W00352',  
 288: 'W00353',  
 289: 'W00356',  
 290: 'W00357',  
 291: 'W00358',  
 292: 'W00359',  
 293: 'W00361',  
 294: 'W00362',  
 295: 'W00363',  
 296: 'W00365',  
 297: 'W00366',  
 298: 'W00367',  
 299: 'W00368',  
 300: 'W00369',  
 301: 'W00370',  
 302: 'W00371',  
 303: 'W00372',  
 304: 'W00373',  
 305: 'W00374',  
 306: 'W00375',  
 307: 'W00376',  
 308: 'W00377',  
 309: 'W00378',  
 310: 'W00380',  
 311: 'W00381',  
 312: 'W00382',  
 313: 'W00383',  
 314: 'W00384',  
 315: 'W00385',  
 316: 'W00386',  
 317: 'W00387',  
 318: 'W00388',  
 319: 'W00389',  
 320: 'W00390',  
 321: 'W00392',  
 322: 'W00393',  
 323: 'W00394',  
 324: 'W00395',  
 325: 'W00397',  
 326: 'W00398',  
 327: 'W00399',  
 328: 'W00400',  
 329: 'W00401',  
 330: 'W00402',  
 331: 'W00406',  
 332: 'W00407',  
 333: 'W00408',  
 334: 'W00409',  
 335: 'W00410',  
 336: 'W00411',  
 337: 'W00412',  
 338: 'W00414',  
 339: 'W00415',  
 340: 'W00416',  
 341: 'W00417',  
 342: 'W00418',  
 343: 'W00420',  
 344: 'W00421',  
 345: 'W00422',  
 346: 'W00425',  
 347: 'W00426',  
 348: 'W00427',  
 349: 'W00428',  
 350: 'W00429',  
 351: 'W00430',  
 352: 'W00431',  
 353: 'W00433',  
 354: 'W00435',  
 355: 'W00436',  
 356: 'W00437',  
 357: 'W00438',  
 358: 'W00439',  
 359: 'W00440',  
 360: 'W00442',  
 361: 'W00443',  
 362: 'W00444',  
 363: 'W00445',  
 364: 'W00446',  
 365: 'W00447',  
 366: 'W00448',  
 367: 'W00449',  
 368: 'W00450',  
 369: 'W00451',  
 370: 'W00453',  
 371: 'W00455',  
 372: 'W00456',  
 373: 'W00457',  
 374: 'W00460',  
 375: 'W00463',  
 376: 'W00464',  
 377: 'W00465',  
 378: 'W00466',  
 379: 'W00467',  
 380: 'W00468',  
 381: 'W00470',  
 382: 'W00471',  
 383: 'W00472',  
 384: 'W00473',  
 385: 'W00474',  
 386: 'W00475',  
 387: 'W00476',  
 388: 'W00477',  
 389: 'W00478',  
 390: 'W00479',  
 391: 'W00480',  
 392: 'W00481',  
 393: 'W00482',  
 394: 'W00483',  
 395: 'W00484',  
 396: 'W00485',  
 397: 'W00487',  
 398: 'W00488',  
 399: 'W00489',  
 400: 'W00490',  
 401: 'W00491',  
 402: 'W00492',  
 403: 'W00493',  
 404: 'W00494',  
 405: 'W00495',  
 406: 'W00498',  
 407: 'W00500',  
 408: 'W00501',  
 409: 'W00502',  
 410: 'W00503',  
 411: 'W00504',  
 412: 'W00506',  
 413: 'W00507',  
 414: 'W00508',  
 415: 'W00509',  
 416: 'W00511',  
 417: 'W00512',  
 418: 'W00514',  
 419: 'W00516',  
 420: 'W00517',  
 421: 'W00518',  
 422: 'W00519',  
 423: 'W00520',  
 424: 'W00522',  
 425: 'W00523',  
 426: 'W00524',  
 427: 'W00525',  
 428: 'W00526',  
 429: 'W00528',  
 430: 'W00529',  
 431: 'W00531',  
 432: 'W00532',  
 433: 'W00533',  
 434: 'W00534',  
 435: 'W00535',  
 436: 'W00537',  
 437: 'W00538',  
 438: 'W00540',  
 439: 'W00541',  
 440: 'W00542',  
 441: 'W00543',  
 442: 'W00544',  
 443: 'W00545',  
 444: 'W00546',  
 445: 'W00547',  
 446: 'W00548',  
 447: 'W00549',  
 448: 'W00550',  
 449: 'W00551',  
 450: 'W00552',  
 451: 'W00553',  
 452: 'W00554',  
 453: 'W00555',  
 454: 'W00556',  
 455: 'W00557',  
 456: 'W00558',  
 457: 'W00559',  
 458: 'W00560',  
 459: 'W00562',  
 460: 'W00563',  
 461: 'W00564',  
 462: 'W00565',  
 463: 'W00566',  
 464: 'W00567',  
 465: 'W00568',  
 466: 'W00569',  
 467: 'W00570',  
 468: 'W00571',  
 469: 'W00572',  
 470: 'W00573',  
 471: 'W00574',  
 472: 'W00575',  
 473: 'W00577',  
 474: 'W00579',  
 475: 'W00580',  
 476: 'W00581',  
 477: 'W00584',  
 478: 'W00585',  
 479: 'W00586',  
 480: 'W00587',  
 481: 'W00588',  
 482: 'W00589',  
 483: 'W00590',  
 484: 'W00592',  
 485: 'W00593',  
 486: 'W00594',  
 487: 'W00595',  
 488: 'W00596',  
 489: 'W00597',  
 490: 'W00598',  
 491: 'W00599',  
 492: 'W00600',  
 493: 'W00601',  
 494: 'W00604',  
 495: 'W00605',  
 496: 'W00606',  
 497: 'W00607',  
 498: 'W00608',  
 499: 'W00609',  
 500: 'W00610',  
 501: 'W00611',  
 502: 'W00612',  
 503: 'W00613',  
 504: 'W00614',  
 505: 'W00615',  
 506: 'W00616',  
 507: 'W00617',  
 508: 'W00618',  
 509: 'W00619',  
 510: 'W00620',  
 511: 'W00621',  
 512: 'W00622',  
 513: 'W00623',  
 514: 'W00624',  
 515: 'W00625',  
 516: 'W00627',  
 517: 'W00628',  
 518: 'W00629',  
 519: 'W00630',  
 520: 'W00631',  
 521: 'W00632',  
 522: 'W00633',  
 523: 'W00634',  
 524: 'W00635',  
 525: 'W00636',  
 526: 'W00637',  
 527: 'W00638',  
 528: 'W00640',  
 529: 'W00641',  
 530: 'W00642',  
 531: 'W00643',  
 532: 'W00645',  
 533: 'W00647',  
 534: 'W00648',  
 535: 'W00649',  
 536: 'W00650',  
 537: 'W00652',  
 538: 'W00653',  
 539: 'W00654',  
 540: 'W00655',  
 541: 'W00656',  
 542: 'W00657',  
 543: 'W00658',  
 544: 'W00659',  
 545: 'W00660',  
 546: 'W00661',  
 547: 'W00662',  
 548: 'W00663',  
 549: 'W00664',  
 550: 'W00665',  
 551: 'W00666',  
 552: 'W00667',  
 553: 'W00669',  
 554: 'W00670',  
 555: 'W00671',  
 556: 'W00672',  
 557: 'W00673',  
 558: 'W00674',  
 559: 'W00675',  
 560: 'W00676',  
 561: 'W00677',  
 562: 'W00678',  
 563: 'W00679',  
 564: 'W00680',  
 565: 'W00681',  
 566: 'W00682',  
 567: 'W00683',  
 568: 'W00684',  
 569: 'W00685',  
 570: 'W00686',  
 571: 'W00687',  
 572: 'W00688',  
 573: 'W00689',  
 574: 'W00691',  
 575: 'W00692',  
 576: 'W00693',  
 577: 'W00694',  
 578: 'W00695',  
 579: 'W00696',  
 580: 'W00697',  
 581: 'W00698',  
 582: 'W00699',  
 583: 'W00700',  
 584: 'W00701',  
 585: 'W00703',  
 586: 'W00704',  
 587: 'W00705',  
 588: 'W00706',  
 589: 'W00707',  
 590: 'W00708',  
 591: 'W00710',  
 592: 'W00713',  
 593: 'W00714',  
 594: 'W00715',  
 595: 'W00717',  
 596: 'W00718',  
 597: 'W00719',  
 598: 'W00720',  
 599: 'W00721',  
 600: 'W00722',  
 601: 'W00723',  
 602: 'W00724',  
 603: 'W00725',  
 604: 'W00726',  
 605: 'W00727',  
 606: 'W00728',  
 607: 'W00729',  
 608: 'W00730',  
 609: 'W00731',  
 610: 'W00733',  
 611: 'W00734',  
 612: 'W00735',  
 613: 'W00736',  
 614: 'W00739',  
 615: 'W00740',  
 616: 'W00741',  
 617: 'W00742',  
 618: 'W00743',  
 619: 'W00744',  
 620: 'W00745',  
 621: 'W00746',  
 622: 'W00747',  
 623: 'W00748',  
 624: 'W00749',  
 625: 'W00750',  
 626: 'W00751',  
 627: 'W00752',  
 628: 'W00754',  
 629: 'W00755',  
 630: 'W00756',  
 631: 'W00757',  
 632: 'W00758',  
 633: 'W00759',  
 634: 'W00760',  
 635: 'W00761',  
 636: 'W00762',  
 637: 'W00764',  
 638: 'W00766',  
 639: 'W00768',  
 640: 'W00769',  
 641: 'W00770',  
 642: 'W00771',  
 643: 'W00772',  
 644: 'W00773',  
 645: 'W00775',  
 646: 'W00776',  
 647: 'W00777',  
 648: 'W00778',  
 649: 'W00779',  
 650: 'W00780',  
 651: 'W00781',  
 652: 'W00782',  
 653: 'W00783',  
 654: 'W00784',  
 655: 'W00785',  
 656: 'W00786',  
 657: 'W00787',  
 658: 'W00789',  
 659: 'W00790',  
 660: 'W00792',  
 661: 'W00794',  
 662: 'W00795',  
 663: 'W00797',  
 664: 'W00798',  
 665: 'W00799',  
 666: 'W00800',  
 667: 'W00801',  
 668: 'W00802',  
 669: 'W00803',  
 670: 'W00804',  
 671: 'W00805',  
 672: 'W00806',  
 673: 'W00807',  
 674: 'W00808',  
 675: 'W00809',  
 676: 'W00810',  
 677: 'W00811',  
 678: 'W00812',  
 679: 'W00813',  
 680: 'W00814',  
 681: 'W00815',  
 682: 'W00816',  
 683: 'W00817',  
 684: 'W00818',  
 685: 'W00819',  
 686: 'W00820',  
 687: 'W00821',  
 688: 'W00822',  
 689: 'W00823',  
 690: 'W00825',  
 691: 'W00826',  
 692: 'W00827',  
 693: 'W00829',  
 694: 'W00831',  
 695: 'W00832',  
 696: 'W00833',  
 697: 'W00836',  
 698: 'W00837',  
 699: 'W00838',  
 700: 'W00839',  
 701: 'W00840',  
 702: 'W00841',  
 703: 'W00842',  
 704: 'W00843',  
 705: 'W00844',  
 706: 'W00845',  
 707: 'W00846',  
 708: 'W00847',  
 709: 'W00848',  
 710: 'W00849',  
 711: 'W00850',  
 712: 'W00851',  
 713: 'W00852',  
 714: 'W00853',  
 715: 'W00855',  
 716: 'W00856',  
 717: 'W00857',  
 718: 'W00858',  
 719: 'W00859',  
 720: 'W00860',  
 721: 'W00861',  
 722: 'W00862',  
 723: 'W00863',  
 724: 'W00864',  
 725: 'W00865',  
 726: 'W00867',  
 727: 'W00868',  
 728: 'W00869',  
 729: 'W00870',  
 730: 'W00872',  
 731: 'W00873',  
 732: 'W00874',  
 733: 'W00876',  
 734: 'W00877',  
 735: 'W00878',  
 736: 'W00879',  
 737: 'W00881',  
 738: 'W00882',  
 739: 'W00883',  
 740: 'W00884',  
 741: 'W00885',  
 742: 'W00886',  
 743: 'W00887',  
 744: 'W00888',  
 745: 'W00889',  
 746: 'W00890',  
 747: 'W00892',  
 748: 'W00893',  
 749: 'W00894',  
 750: 'W00895',  
 751: 'W00896',  
 752: 'W00897',  
 753: 'W00898',  
 754: 'W00899',  
 755: 'W00900',  
 756: 'W00901',  
 757: 'W00902',  
 758: 'W00903',  
 759: 'W00904',  
 760: 'W00905',  
 761: 'W00906',  
 762: 'W00907',  
 763: 'W00908',  
 764: 'W00909',  
 765: 'W00910',  
 766: 'W00911',  
 767: 'W00912',  
 768: 'W00913',  
 769: 'W00914',  
 770: 'W00915',  
 771: 'W00916',  
 772: 'W00917',  
 773: 'W00918',  
 774: 'W00920',  
 775: 'W00921',  
 776: 'W00922',  
 777: 'W00923',  
 778: 'W00924',  
 779: 'W00925',  
 780: 'W00926',  
 781: 'W00927',  
 782: 'W00928',  
 783: 'W00929',  
 784: 'W00930',  
 785: 'W00931',  
 786: 'W00932',  
 787: 'W00933',  
 788: 'W00935',  
 789: 'W00936',  
 790: 'W00937',  
 791: 'W00938',  
 792: 'W00939',  
 793: 'W00940',  
 794: 'W00941',  
 795: 'W00942',  
 796: 'W00943',  
 797: 'W00944',  
 798: 'W00945',  
 799: 'W00946',  
 800: 'W00948',  
 801: 'W00949',  
 802: 'W00950',  
 803: 'W00951',  
 804: 'W00952',  
 805: 'W00953',  
 806: 'W00954',  
 807: 'W00956',  
 808: 'W00957',  
 809: 'W00958',  
 810: 'W00959',  
 811: 'W00960',  
 812: 'W00961',  
 813: 'W00962',  
 814: 'W00963',  
 815: 'W00964',  
 816: 'W00965',  
 817: 'W00966',  
 818: 'W00968',  
 819: 'W00969',  
 820: 'W00970',  
 821: 'W00971',  
 822: 'W00972',  
 823: 'W00973',  
 824: 'W00974',  
 825: 'W00975',  
 826: 'W00976',  
 827: 'W00977',  
 828: 'W00978',  
 829: 'W00979',  
 830: 'W00980',  
 831: 'W00981',  
 832: 'W00982',  
 833: 'W00984',  
 834: 'W00985',  
 835: 'W00986',  
 836: 'W00987',  
 837: 'W00988',  
 838: 'W00989',  
 839: 'W00990',  
 840: 'W00992',  
 841: 'W00993',  
 842: 'W00994',  
 843: 'W00995',  
 844: 'W00996',  
 845: 'W00997',  
 846: 'W00998',  
 847: 'W00999',  
 848: 'W01000',  
 849: 'W01001',  
 850: 'W01002',  
 851: 'W01003',  
 852: 'W01004',  
 853: 'W01006',  
 854: 'W01007',  
 855: 'W01008',  
 856: 'W01010',  
 857: 'W01011',  
 858: 'W01012',  
 859: 'W01013',  
 860: 'W01014',  
 861: 'W01015',  
 862: 'W01016',  
 863: 'W01018',  
 864: 'W01019',  
 865: 'W01020',  
 866: 'W01021',  
 867: 'W01022',  
 868: 'W01023',  
 869: 'W01024',  
 870: 'W01025',  
 871: 'W01026',  
 872: 'W01027',  
 873: 'W01028',  
 874: 'W01030',  
 875: 'W01031',  
 876: 'W01032',  
 877: 'W01033',  
 878: 'W01034',  
 879: 'W01035',  
 880: 'W01036',  
 881: 'W01037',  
 882: 'W01038',  
 883: 'W01039',  
 884: 'W01041',  
 885: 'W01043',  
 886: 'W01044',  
 887: 'W01045',  
 888: 'W01046',  
 889: 'W01047',  
 890: 'W01048',  
 891: 'W01049',  
 892: 'W01050',  
 893: 'W01051',  
 894: 'W01053',  
 895: 'W01054',  
 896: 'W01055',  
 897: 'W01056',  
 898: 'W01058',  
 899: 'W01059',  
 900: 'W01062',  
 901: 'W01063',  
 902: 'W01065',  
 903: 'W01066',  
 904: 'W01067',  
 905: 'W01068',  
 906: 'W01069',  
 907: 'W01070',  
 908: 'W01071',  
 909: 'W01072',  
 910: 'W01073',  
 911: 'W01074',  
 912: 'W01075',  
 913: 'W01076',  
 914: 'W01079',  
 915: 'W01080',  
 916: 'W01081',  
 917: 'W01082',  
 918: 'W01083',  
 919: 'W01084',  
 920: 'W01085',  
 921: 'W01087',  
 922: 'W01088',  
 923: 'W01089',  
 924: 'W01090',  
 925: 'W01091',  
 926: 'W01093',  
 927: 'W01094',  
 928: 'W01095',  
 929: 'W01096',  
 930: 'W01097',  
 931: 'W01098',  
 932: 'W01099',  
 933: 'W01101',  
 934: 'W01102',  
 935: 'W01104',  
 936: 'W01105',  
 937: 'W01107',  
 938: 'W01108',  
 939: 'W01109',  
 940: 'W01110',  
 941: 'W01112',  
 942: 'W01113',  
 943: 'W01115',  
 944: 'W01116',  
 945: 'W01119',  
 946: 'W01120',  
 947: 'W01121',  
 948: 'W01122',  
 949: 'W01123',  
 950: 'W01124',  
 951: 'W01125',  
 952: 'W01129',  
 953: 'w00059',  
 954: 'w00223',  
 955: 'w00527',  
 956: 'w00583',  
 957: 'w00765',  
 958: 'w01100',  
 959: 'w01117'}

df['Bike#'] = bike\_labels

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

print(y)

['Registered' 'Casual' 'Registered' ... 'Registered' 'Casual' 'Registered']

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))

Accuracy of our model is: 88.4%  
[[ 1264 3100]  
 [ 379 25269]]  
 precision recall f1-score support  
  
 Casual 0.77 0.29 0.42 4364  
 Registered 0.89 0.99 0.94 25648  
  
 accuracy 0.88 30012  
 macro avg 0.83 0.64 0.68 30012  
weighted avg 0.87 0.88 0.86 30012