Report

Name. : Sanskriti Bahl Roll Number : 045050

Description of Data 2.1. Data Source, Size, Shape Data Source: The dataset used for analysis is sourced from Kaggle. Data Size: The dataset size is approximately 28.83 MB. Data Shape: The dataset comprises 11 variables with 97712 records. 2.2. Description of Variables Index Variable(s): None Variables or Features with Categories (CV): model transmission fuelType Manufacturer Variables or Features with Nominal Categories | Categorical Variables or Features - Nominal Type: model fuelType Manufacturer Variables or Features with Ordinal Categories | Categorical Variables or Features - Ordinal Type: transmission Non-Categorical Variables or Features: srno year price mileage tax mpg engineSize 2.3. Descriptive Statistics 2.3.1. Descriptive Statistics: Categorical Variables or Features Count | Frequency Statistics: Model: Fiesta: 6509 Golf: 4797 Focus: 4555 C Class: 3694 Corsa: 3285 ... Transmission: Manual: 55502 Semi-Auto: 22296 Automatic: 19905 Other: 9 FuelType: Petrol: 53982 Diesel: 40419 Hybrid: 3059 Other: 246 Electric: 6 Manufacturer: Ford: 17811 Volkswagen: 14893 Vauxhall: 13258 Merc: 12860 BMW: 10664 ... 2.3.2. Descriptive Statistics: Non-Categorical Variables or Features Measures of Central Tendency: Year: Mean - 2017.066502, Median - 2017.0 Price: Mean - 16773.487555, Median - 14470.0 Mileage: Mean - 23219.475499, Median - 17682.5 Tax: Mean - 120.142408, Median - 145.0 MPG: Mean - 55.205623, Median - 54.3 EngineSize: Mean - 1.664913, Median - 1.6 Measures of Dispersion: Year: Standard Deviation - 2.118661, Range - 4.488726 Price: Standard Deviation - 9868.552222, Range - 9.738832e+07 Mileage: Standard Deviation - 21060.882301, Range - 4.435608e+08 Tax: Standard Deviation - 63.357250, Range - 4.014141e+03 MPG: Standard Deviation - 16.181659, Range - 2.618461e+02 EngineSize: Standard Deviation - 0.558574, Range - 3.120049e-01 Correlation Statistics (with Test of Correlation): Year vs. Price: Correlation coefficient - 0.492491 Year vs. Mileage: Correlation coefficient - -0.739664 Year vs. Tax: Correlation coefficient - 0.192058 Year vs. MPG: Correlation coefficient - -0.130547 Year vs. EngineSize: Correlation coefficient - -0.035639 Price vs. Mileage: Correlation coefficient - -0.417996 Price vs. Tax: Correlation coefficient - 0.307003 Price vs. MPG: Correlation coefficient - -0.295507 Price vs. EngineSize: Correlation coefficient - 0.639356 Mileage vs. Tax: Correlation coefficient - -0.215558 Mileage vs. MPG: Correlation coefficient - 0.183070 Mileage vs. EngineSize: Correlation coefficient - 0.107299 Tax vs. MPG: Correlation coefficient - -0.451370 Tax vs. EngineSize: Correlation coefficient - 0.280770 MPG vs. EngineSize: Correlation coefficient - -0.249346

Observations on Data Analysis

Pre-processing:

No columns have NaN values. Outliers are observed in the 'price', 'mileage', 'tax', 'mpg', and 'engineSize' variables. Clustering:

K = 4 is determined as the appropriate number of clusters for the dataset.

Feature Importance :

1. Engine Size (0.503): Identified as the most important feature in predicting the target variable. Indicates that variations in engine size have the highest impact on the

classification outcome. Vehicles with different engine sizes may belong to distinct categories, influencing their classification.

2. Miles Per Gallon (MPG) (0.267): Ranked as the second most important feature in determining the target variable. Suggests that fuel efficiency, as represented by MPG, plays a significant role in the classification process. Vehicles with higher MPG may tend to belong to certain categories compared to those with lower MPG.

3. Model (0.066) and Manufacturer (0.053): While contributing less to the overall importance compared to engine size and MPG, model and manufacturer still hold notable significance. Indicates that specific models and manufacturers have distinct characteristics influencing their classification. Certain models or manufacturers may be more prevalent in certain categories.

4. Transmission (0.040): Ranked lower in importance compared to engine size, MPG, model, and manufacturer. Suggests that while transmission type influences classification, its impact is less pronounced compared to other features.

5. Price (0.022), Year (0.021), Tax (0.018), Mileage (0.009): Ranked lower in importance compared to engine size, MPG, model, manufacturer, and transmission. These features still contribute to the classification process but have relatively lesser impact compared to others. Price, year, tax, and mileage may play secondary roles in distinguishing between different categories.

Model Performance:

Decision Tree: Achieves perfect classification accuracy (100%) on both Gini and Entropy criteria. Support Vector Machine (SVM): Achieves high accuracy (98%) on both training and testing sets. Logistic Regression: Achieves perfect classification accuracy (100%) on both training and testing sets. K Nearest Neighbors (KNN): Achieves perfect classification accuracy (100%) on both training and testing sets.

Managerial Insights :

1. Decision Tree provides a reliable classification method with perfect accuracy, suitable for scenarios where interpretability is important.

2. Support Vector Machine (SVM) demonstrates strong performance on both training and testing sets, indicating its effectiveness in classifying data points into distinct categories.

3. Logistic Regression offers a simple yet powerful classification approach with perfect accuracy, making it suitable for scenarios where interpretability and performance are equally important.

4. K Nearest Neighbors (KNN) shows excellent performance with perfect accuracy, especially beneficial when dealing with a large dataset and no clear separation between classes.

```
import pandas as pd
import numpy as np
import time
from memory_profiler import memory_usage
from sklearn.metrics import pairwise_distances
from sklearn.preprocessing import LabelEncoder, OrdinalEncoder # For
Encoding Categorical Data [Nominal | Ordinal]
from sklearn.preprocessing import OneHotEncoder # For Creating Dummy
Variables of Categorical Data [Nominal]
from sklearn.impute import SimpleImputer, KNNImputer # For Imputation
of Missing Data
from sklearn.preprocessing import StandardScaler, MinMaxScaler,
RobustScaler # For Rescaling Data
from sklearn.model_selection import train_test_split # For Splitting
Data into Training & Testing Sets
import seaborn as sns
import matplotlib.pyplot as plt # For Data Visualization
import scipy.cluster.hierarchy as sch # For Hierarchical Clustering
from sklearn.cluster import AgglomerativeClustering as agclus, KMeans
as kmclus # For Agglomerative & K-Means Clustering
from sklearn.metrics import silhouette_score as sscore,
davies_bouldin_score as dbscore # For Clustering Model Evaluation
from sklearn.model_selection import train_test_split # For Splitting
Data into Training & Testing Sets
from sklearn.tree import DecisionTreeClassifier, export_text,
plot_tree  # For Decision Tree Model
from sklearn.metrics import confusion_matrix, classification_report #
For Decision Tree Model Evaluation
import plotly.express as px
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, OrdinalEncoder # For
Encoding Categorical Data [Nominal | Ordinal]
from sklearn.preprocessing import OneHotEncoder # For Creating Dummy
Variables of Categorical Data [Nominal]
from sklearn.impute import SimpleImputer, KNNImputer # For Imputation
of Missing Data
from sklearn.preprocessing import StandardScaler, MinMaxScaler,
RobustScaler # For Rescaling Data
from sklearn.model_selection import train_test_split # For Splitting
Data into Training & Testing Sets
```

```
import seaborn as sns
import matplotlib.pyplot as plt # For Data Visualization
import scipy.cluster.hierarchy as sch # For Hierarchical Clustering
from sklearn.cluster import AgglomerativeClustering as agclus, KMeans
as kmclus # For Agglomerative & K-Means Clustering
from sklearn.metrics import silhouette_score as sscore,
davies_bouldin_score as dbscore # For Clustering Model Evaluation
from sklearn.model_selection import train_test_split # For Splitting
Data into Training & Testing Sets
from sklearn.tree import DecisionTreeClassifier, export_text,
plot_tree  # For Decision Tree Model
from sklearn.metrics import confusion_matrix, classification_report #
For Decision Tree Model Evaluation
import plotly.express as px
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score
from sklearn.preprocessing import StandardScaler

import seaborn as sns
import numpy as np
from sklearn.preprocessing import LabelEncoder, OrdinalEncoder # For
Encoding Categorical Data [Nominal | Ordinal]
from sklearn.preprocessing import OneHotEncoder # For Creating Dummy
Variables of Categorical Data [Nominal]
from sklearn.impute import SimpleImputer, KNNImputer # For Imputation
of Missing Data
from sklearn.preprocessing import StandardScaler, MinMaxScaler,
RobustScaler # For Rescaling Data
from sklearn.model_selection import train_test_split # For Splitting
Data into Training & Testing Sets
import seaborn as sns
import scipy.cluster.hierarchy as sch # For Hierarchical Clustering
from sklearn.cluster import AgglomerativeClustering as agclus, KMeans
as kmclus # For Agglomerative & K-Means Clustering
from sklearn.metrics import silhouette_score as sscore,
davies_bouldin_score as dbscore # For Clustering Model Evaluation
from sklearn.model_selection import train_test_split # For Splitting
Data into Training & Testing Sets
from sklearn.tree import DecisionTreeClassifier, export_text,
plot_tree  # For Decision Tree Model
from sklearn.metrics import confusion_matrix, classification_report #
For Decision Tree Model Evaluation
import plotly.express as px
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score
```

```
from sklearn.preprocessing import StandardScaler

# Load CSV file
df = pd.read_csv('CarsData.csv')
df.head()
```

```
           model  year  price transmission  mileage fuelType  tax
mpg  \
0             I10  2017   7495       Manual    11630   Petrol  145
60.1
1            Polo  2017  10989       Manual     9200   Petrol  145
58.9
2       2 Series  2019  27990    Semi-Auto     1614   Diesel  145
49.6
3   Yeti Outdoor  2017  12495       Manual    30960   Diesel  150
62.8
4         Fiesta  2017   7999       Manual    19353   Petrol  125
54.3

   engineSize Manufacturer
0         1.0       hyundi
1         1.0   volkswagen
2         2.0          BMW
3         2.0        skoda
4         1.2         ford
```

```
total_memory = df.memory_usage(deep=True).sum()
print(f"Total memory usage of dataframe: {total_memory} bytes")
```

```
Total memory usage of dataframe: 29446708 bytes
```

```
df.insert(0, 'srno', range(0, len(df))) # Adding Index Variable
df_ah = df.copy()
df.head(10)
```

```
   srno          model  year  price transmission  mileage fuelType
tax   mpg  \
0     0            I10  2017   7495       Manual    11630   Petrol
145  60.1
1     1           Polo  2017  10989       Manual     9200   Petrol
145  58.9
2     2       2 Series  2019  27990    Semi-Auto     1614   Diesel
145  49.6
3     3   Yeti Outdoor  2017  12495       Manual    30960   Diesel
150  62.8
4     4         Fiesta  2017   7999       Manual    19353   Petrol
125  54.3
5     5           C-HR  2019  26791    Automatic     2373   Hybrid
135  74.3
6     6           Kuga  2019  17990       Manual     7038   Petrol
145  34.4
```

```
7     7          Tiguan  2019  27490    Semi-Auto     3000   Petrol
145  30.4
8     8          Fiesta  2018   9891       Manual    31639   Petrol
145  65.7
9     9         A Class  2017  17498       Manual     9663   Diesel
30  62.8

   engineSize Manufacturer
0         1.0       hyundi
1         1.0   volkswagen
2         2.0          BMW
3         2.0        skoda
4         1.2         ford
5         1.8       toyota
6         1.5         ford
7         2.0   volkswagen
8         1.0         ford
9         2.1         merc
```

```
# Data Source
data_source = "https://www.kaggle.com/datasets/meruvulikith/90000-cars-data-from-1970-to-2024/data"  # Fill in the actual data source if available

# Data Size
data_size = df.memory_usage(deep=True).sum() / (1024 ** 2)  # Convert to MB

# Data Shape
num_variables = df.shape[1]
num_records = df.shape[0]

# Print Description of Data
print("2. Description of Data")
print("\t2.1. Data Source, Size, Shape")
print(f"\t\t2.1.1. Data Source: {data_source}")
print(f"\t\t2.1.2. Data Size: {data_size:.2f} MB")
print(f"\t\t2.1.3. Data Shape: Dimension: {num_variables} variables | {num_records} records")
```

```
2. Description of Data
     2.1. Data Source, Size, Shape
          2.1.1. Data Source: https://www.kaggle.com/datasets/meruvulikith/90000-cars-data-from-1970-to-2024/data
          2.1.2. Data Size: 28.83 MB
          2.1.3. Data Shape: Dimension: 11 variables | 97712 records
```

```
# 2.2. Description of Variables
print("2.2. Description of Variables")
```

```
# 2.2.1. Index Variable(s)
index_variables = df.index.name
print(f"\t2.2.1. Index Variable(s): {index_variables}")

# 2.2.2. Variables or Features having Categories | Categorical
Variables or Features (CV)
categorical_variables =
df.select_dtypes(include=['object']).columns.tolist()
print("\t2.2.2. Variables or Features having Categories | Categorical
Variables or Features (CV):")
for var in categorical_variables:
    print(f"\t\t{var}")

# 2.2.2.1. Variables or Features having Nominal Categories |
Categorical Variables or Features - Nominal Type
nominal_variables = ['model', 'fuelType', 'Manufacturer']
print("\t2.2.2.1. Variables or Features having Nominal Categories |
Categorical Variables or Features - Nominal Type:")
for var in nominal_variables:
    print(f"\t\t{var}")

# 2.2.2.2. Variables or Features having Ordinal Categories |
Categorical Variables or Features - Ordinal Type
ordinal_variables = ['transmission']
print("\t2.2.2.2. Variables or Features having Ordinal Categories |
Categorical Variables or Features - Ordinal Type:")
for var in ordinal_variables:
    print(f"\t\t{var}")

# 2.2.3. Non-Categorical Variables or Features
non_categorical_variables =
df.select_dtypes(exclude=['object']).columns.tolist()
print("\t2.2.3. Non-Categorical Variables or Features:")
for var in non_categorical_variables:
    print(f"\t\t{var}")
```

```
2.2. Description of Variables
     2.2.1. Index Variable(s): None
     2.2.2. Variables or Features having Categories | Categorical
Variables or Features (CV):
          model
          transmission
          fuelType
          Manufacturer
     2.2.2.1. Variables or Features having Nominal Categories |
Categorical Variables or Features - Nominal Type:
          model
          fuelType
          Manufacturer
```

```
     2.2.2.2. Variables or Features having Ordinal Categories | Categorical Variables or Features - Ordinal Type:
          transmission
     2.2.3. Non-Categorical Variables or Features:
          srno
          year
          price
          mileage
          tax
          mpg
          engineSize
```

```
# 2.3. Descriptive Statistics

# 2.3.1. Descriptive Statistics: Categorical Variables or Features
print("2.3.1. Descriptive Statistics: Categorical Variables or Features")

# 2.3.1.1. Count | Frequency Statistics
print("\t2.3.1.1. Count | Frequency Statistics:")
categorical_variables = ['model', 'transmission', 'fuelType', 'Manufacturer']
for var in categorical_variables:
    print(f"\t\t{var}:")
    print(df[var].value_counts())

# 2.3.1.2. Proportion (Relative Frequency) Statistics
print("\t2.3.1.2. Proportion (Relative Frequency) Statistics:")
for var in categorical_variables:
    print(f"\t\t{var}:")
    print(df[var].value_counts(normalize=True))

# 2.3.2. Descriptive Statistics: Non-Categorical Variables or Features
print("2.3.2. Descriptive Statistics: Non-Categorical Variables or Features")

# 2.3.2.1. Measures of Central Tendency
print("\t2.3.2.1. Measures of Central Tendency:")
non_categorical_variables = ['year', 'price', 'mileage', 'tax', 'mpg', 'engineSize']
print(df[non_categorical_variables].mean())
print(df[non_categorical_variables].median())

# 2.3.2.2. Measures of Dispersion
print("\t2.3.2.2. Measures of Dispersion:")
print(df[non_categorical_variables].std())
print(df[non_categorical_variables].var())

# 2.3.2.3. Correlation Statistics (with Test of Correlation)
```

```
print("\t2.3.2.3. Correlation Statistics (with Test of Correlation):")
print(df[non_categorical_variables].corr())
```

```
2.3.1. Descriptive Statistics: Categorical Variables or Features
     2.3.1.1. Count | Frequency Statistics:
           model:
 Fiesta     6509
 Golf       4797
 Focus      4555
 C Class    3694
 Corsa      3285
            ...
E Class        1
200            1
 Ranger        1
180            1
220            1
Name: model, Length: 196, dtype: int64
           transmission:
Manual       55502
Semi-Auto    22296
Automatic    19905
Other            9
Name: transmission, dtype: int64
           fuelType:
Petrol      53982
Diesel      40419
Hybrid       3059
Other         246
Electric        6
Name: fuelType, dtype: int64
           Manufacturer:
ford          17811
volkswagen    14893
vauxhall      13258
merc          12860
BMW           10664
Audi          10565
toyota         6699
skoda          6188
hyundi         4774
Name: Manufacturer, dtype: int64
     2.3.1.2. Proportion (Relative Frequency) Statistics:
           model:
 Fiesta     0.066614
 Golf       0.049093
 Focus      0.046617
 C Class    0.037805
 Corsa      0.033619
              ...
```

```
E Class     0.000010
200         0.000010
 Ranger     0.000010
180         0.000010
220         0.000010
Name: model, Length: 196, dtype: float64
           transmission:
Manual       0.568016
Semi-Auto    0.228181
Automatic    0.203711
Other        0.000092
Name: transmission, dtype: float64
           fuelType:
Petrol      0.552460
Diesel      0.413654
Hybrid      0.031306
Other       0.002518
Electric    0.000061
Name: fuelType, dtype: float64
           Manufacturer:
ford          0.182281
volkswagen    0.152417
vauxhall      0.135684
merc          0.131611
BMW           0.109137
Audi          0.108124
toyota        0.068559
skoda         0.063329
hyundi        0.048858
Name: Manufacturer, dtype: float64
2.3.2. Descriptive Statistics: Non-Categorical Variables or Features
     2.3.2.1. Measures of Central Tendency:
year           2017.066502
price         16773.487555
mileage       23219.475499
tax             120.142408
mpg              55.205623
engineSize        1.664913
dtype: float64
year           2017.0
price         14470.0
mileage       17682.5
tax             145.0
mpg              54.3
engineSize        1.6
dtype: float64
     2.3.2.2. Measures of Dispersion:
year              2.118661
price          9868.552222
mileage       21060.882301
```

```
tax              63.357250
mpg              16.181659
engineSize        0.558574
dtype: float64
year          4.488726e+00
price         9.738832e+07
mileage       4.435608e+08
tax           4.014141e+03
mpg           2.618461e+02
engineSize    3.120049e-01
dtype: float64
     2.3.2.3. Correlation Statistics (with Test of Correlation):
                year     price   mileage       tax       mpg
engineSize
year        1.000000  0.492491 -0.739664  0.192058 -0.130547   -
0.035639
price       0.492491  1.000000 -0.417996  0.307003 -0.295507
0.639356
mileage    -0.739664 -0.417996  1.000000 -0.215558  0.183070
0.107299
tax         0.192058  0.307003 -0.215558  1.000000 -0.451370
0.280770
mpg        -0.130547 -0.295507  0.183070 -0.451370  1.000000   -
0.249346
engineSize -0.035639  0.639356  0.107299  0.280770 -0.249346
1.000000

#Analysis of Data

#Data Pre-Processing Data
#Missing data in each of the variables
df.info() # Dataframe Information (Provide Information on Missing
Data)
variable_missing_data = df.isna().sum(); variable_missing_data #
Variable-wise Missing Data Information

##None of the variables contain any missing data, hence there isn't
any need of missing treatmet.

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 97712 entries, 0 to 97711
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   srno          97712 non-null  int64
 1   model         97712 non-null  object
 2   year          97712 non-null  int64
 3   price         97712 non-null  int64
 4   transmission  97712 non-null  object
 5   mileage       97712 non-null  int64
```

```
 6   fuelType      97712 non-null  object
 7   tax           97712 non-null  int64
 8   mpg           97712 non-null  float64
 9   engineSize    97712 non-null  float64
 10  Manufacturer  97712 non-null  object
dtypes: float64(2), int64(5), object(4)
memory usage: 8.2+ MB
```

```
srno            0
model           0
year            0
price           0
transmission    0
mileage         0
fuelType        0
tax             0
mpg             0
engineSize      0
Manufacturer    0
dtype: int64
```

```
record_missing_data =
df.isna().sum(axis=1).sort_values(ascending=False).head();
record_missing_data # Record-wise Missing Data Information (Top 5)
```

```
0        0
65138    0
65147    0
65146    0
65145    0
dtype: int64
```

```
# Check for NaN values in the entire DataFrame
nan_values = df.isna().any()

# Print the columns with NaN values
print("Columns with NaN values:")
print(nan_values[nan_values].index)

##So it is evident from the result that all columns in the DataFrame
are free from missing values, and there are no NaN values present.
```

```
Columns with NaN values:
Index([], dtype='object')
```

```
df_cat = df[['model','transmission','fuelType','Manufacturer','srno']]
# Categorical Data [Nominal | Ordinal]
df_noncat = df[['year','price',
'mileage','tax','mpg','engineSize','srno']] # Non-Categorical Data
```

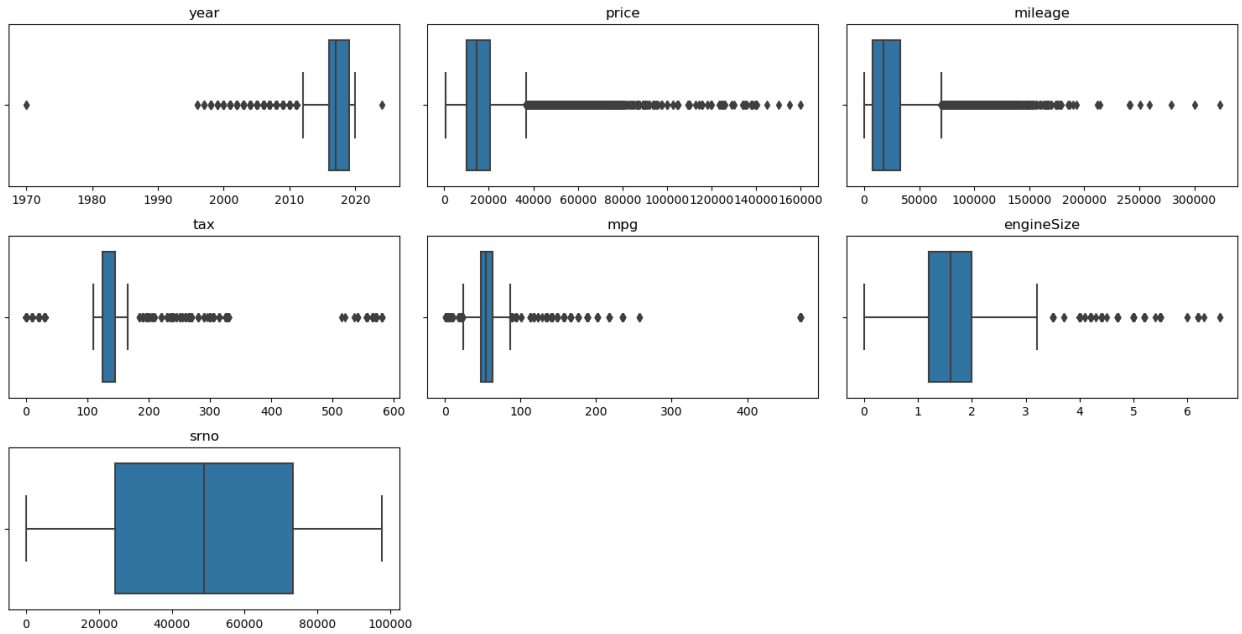
```python
from sklearn.preprocessing import LabelEncoder

# Define mapping dictionaries
transmission_mapping = {'Manual': 0, 'Semi-Auto': 1, 'Automatic': 2,
'Other': 3}
fuelType_mapping = {'Petrol': 0, 'Diesel': 1, 'Hybrid': 2, 'Electric':
3, 'Other': 4}
Manufacturer_mapping = {'ford': 0, 'volkswagen': 1, 'vauxhall': 2,
'merc': 3, 'BMW': 4, 'Audi': 5, 'toyota': 6, 'skoda': 7, 'hyundi': 8}

# Perform numerical encoding for 'transmission'
df_cat['transmission_enc'] =
df_cat['transmission'].map(transmission_mapping)

# Perform numerical encoding for 'fuelType'
df_cat['fuelType_enc'] = df_cat['fuelType'].map(fuelType_mapping)

# Perform numerical encoding for 'Manufacturer'
df_cat['Manufacturer_enc'] =
df_cat['Manufacturer'].map(Manufacturer_mapping)

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Encode 'model'
df_cat['model_enc'] = label_encoder.fit_transform(df_cat['model'])

# Display the updated DataFrame
print(df_cat)
```

```
               model transmission fuelType Manufacturer   srno  \
0                I10       Manual   Petrol       hyundi      0
1              Polo        Manual   Petrol   volkswagen      1
2          2 Series    Semi-Auto   Diesel          BMW      2
3       Yeti Outdoor       Manual   Diesel        skoda      3
4             Fiesta       Manual   Petrol         ford      4
...              ...          ...      ...          ...    ...
97707         Fiesta    Automatic   Petrol         ford  97707
97708       3 Series       Manual   Diesel          BMW  97708
97709         Fiesta       Manual   Petrol         ford  97709
97710          Astra    Automatic   Petrol     vauxhall  97710
97711    Grandland X       Manual   Diesel     vauxhall  97711

       transmission_enc  fuelType_enc  Manufacturer_enc  model_enc
0                     0             0                 8         81
1                     0             0                 1        115
2                     1             1                 4          1
3                     0             1                 7        184
4                     0             0                 0         60
...                 ...           ...               ...        ...
```

```
97707                  2             0                  0          60
97708                  0             1                  4           2
97709                  0             0                  0          60
97710                  2             0                  2          25
97711                  0             1                  2          79

[97712 rows x 9 columns]

/var/folders/j4/9pz_sc_x7298x60tw9kbd6jw0000gn/T/
ipykernel_89397/2033554636.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_cat['transmission_enc'] =
df_cat['transmission'].map(transmission_mapping)
/var/folders/j4/9pz_sc_x7298x60tw9kbd6jw0000gn/T/ipykernel_89397/20335
54636.py:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_cat['fuelType_enc'] = df_cat['fuelType'].map(fuelType_mapping)
/var/folders/j4/9pz_sc_x7298x60tw9kbd6jw0000gn/T/ipykernel_89397/20335
54636.py:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_cat['Manufacturer_enc'] =
df_cat['Manufacturer'].map(Manufacturer_mapping)
/var/folders/j4/9pz_sc_x7298x60tw9kbd6jw0000gn/T/ipykernel_89397/20335
54636.py:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_cat['model_enc'] = label_encoder.fit_transform(df_cat['model'])
```

```
# Create boxplots for each non-categorical variable
plt.figure(figsize=(15, 10))
for i, var in enumerate(df_noncat):
```

```
    plt.subplot(4, 3, i+1)
    sns.boxplot(x=df_noncat[var])
    plt.title(var)
    plt.xlabel('')
    plt.ylabel('')

plt.tight_layout()
plt.show()
```




```
#Outlier treatment

def outlier_statistics(df):
    # Define non-categorical variables
    non_cat_vars = ['price', 'mileage', 'tax', 'mpg', 'engineSize']

    # Compute the first and third quartiles
    Q1 = df[non_cat_vars].quantile(0.25)
    Q3 = df[non_cat_vars].quantile(0.75)

    # Compute the interquartile range
    IQR = Q3 - Q1

    # Compute lower and upper bounds for outliers
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Count outliers for each variable
    outliers_count = ((df[non_cat_vars] < lower_bound) |
(df[non_cat_vars] > upper_bound)).sum()
```

```
    return outliers_count

# Compute outlier statistics
outliers_stats = outlier_statistics(df_noncat)
print("Outlier Statistics for Non-Categorical Variables:")
print(outliers_stats)

##For the variable 'price', there are 3825 outliers.
##For the variable 'mileage', there are 3836 outliers.
##For the variable 'tax', there are 28594 outliers.
##For the variable 'mpg', there are 930 outliers.
##For the variable 'engineSize', there are 648 outliers.
```

```
Outlier Statistics for Non-Categorical Variables:
price          3825
mileage        3836
tax           28594
mpg             930
engineSize      648
dtype: int64
```

```
from sklearn.preprocessing import MinMaxScaler

# Define the non-categorical variables
non_cat_vars = ['price', 'mileage', 'tax', 'mpg', 'engineSize']

# Step 1: Outlier Statistics
outlier_stats = df_noncat[non_cat_vars].describe(percentiles=[0.25,
0.75]).T
outlier_stats['IQR'] = outlier_stats['75%'] - outlier_stats['25%']
outlier_stats['Lower Bound'] = outlier_stats['25%'] - 1.5 *
outlier_stats['IQR']
outlier_stats['Upper Bound'] = outlier_stats['75%'] + 1.5 *
outlier_stats['IQR']

# Step 2: Outlier Treatment (Winsorization)
for var in non_cat_vars:
    lower_bound = outlier_stats.loc[var, 'Lower Bound']
    upper_bound = outlier_stats.loc[var, 'Upper Bound']
    df_noncat[var] = df_noncat[var].clip(lower=lower_bound,
upper=upper_bound)

# Step 3: Min-Max Scaling
scaler = MinMaxScaler()
df_noncat_scaled = df_noncat.copy()
df_noncat_scaled[non_cat_vars] =
scaler.fit_transform(df_noncat_scaled[non_cat_vars])

# Display outlier statistics
```

```
print("Outlier Statistics:")
print(outlier_stats)

# Display the first few rows of the scaled DataFrame
print("\nScaled DataFrame:")
print(df_noncat_scaled.head())
```

```
Outlier Statistics:
              count          mean           std    min     25%
50%  \
price       97712.0  16773.487555   9868.552222  450.0  9999.0
14470.0
mileage     97712.0  23219.475499  21060.882301    1.0  7673.0
17682.5
tax         97712.0    120.142408     63.357250    0.0   125.0
145.0
mpg         97712.0     55.205623     16.181659    0.3    47.1
54.3
engineSize  97712.0      1.664913      0.558574    0.0     1.2
1.6

                 75%       max      IQR   Lower Bound  Upper Bound
price       20750.0  159999.0  10751.0 -6.127500e+03     36876.50
mileage     32500.0  323000.0  24827.0 -2.956750e+04     69740.50
tax           145.0     580.0     20.0  9.500000e+01       175.00
mpg            62.8     470.8     15.7  2.355000e+01        86.35
engineSize      2.0       6.6      0.8 -2.220446e-16         3.20

Scaled DataFrame:
   year     price   mileage     tax       mpg  engineSize  srno
0  2017  0.193403  0.166749  0.6250  0.582006      0.3125     0
1  2017  0.289322  0.131905  0.6250  0.562898      0.3125     1
2  2019  0.756043  0.023129  0.6250  0.414809      0.6250     2
3  2017  0.330666  0.443923  0.6875  0.625000      0.6250     3
4  2017  0.207239  0.277490  0.3750  0.489650      0.3750     4

/var/folders/j4/9pz_sc_x7298x60tw9kbd6jw0000gn/T/
ipykernel_89397/1838598279.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_noncat[var] = df_noncat[var].clip(lower=lower_bound,
upper=upper_bound)
```

```
# Pre-Processed Dataset
merged_df = pd.merge(df_cat, df_noncat_scaled, on='srno');
merged_df # Pre-Processed Dataset
```

```
               model transmission fuelType Manufacturer   srno  \
0                I10       Manual   Petrol       hyundi      0
1               Polo       Manual   Petrol   volkswagen      1
2           2 Series    Semi-Auto   Diesel          BMW      2
3       Yeti Outdoor       Manual   Diesel        skoda      3
4             Fiesta       Manual   Petrol         ford      4
...              ...          ...      ...          ...    ...
97707         Fiesta    Automatic   Petrol         ford  97707
97708       3 Series       Manual   Diesel          BMW  97708
97709         Fiesta       Manual   Petrol         ford  97709
97710          Astra    Automatic   Petrol     vauxhall  97710
97711    Grandland X       Manual   Diesel     vauxhall  97711

       transmission_enc  fuelType_enc  Manufacturer_enc  model_enc
year  \
0                     0             0                 8         81
2017
1                     0             0                 1        115
2017
2                     1             1                 4          1
2019
3                     0             1                 7        184
2017
4                     0             0                 0         60
2017
...                 ...           ...               ...        ...   .
..
97707                 2             0                 0         60
2017
97708                 0             1                 4          2
2014
97709                 0             0                 0         60
2017
97710                 2             0                 2         25
2017
97711                 0             1                 2         79
2019

          price   mileage     tax       mpg  engineSize
0      0.193403  0.166749  0.6250  0.582006     0.31250
1      0.289322  0.131905  0.6250  0.562898     0.31250
2      0.756043  0.023129  0.6250  0.414809     0.62500
3      0.330666  0.443923  0.6875  0.625000     0.62500
4      0.207239  0.277490  0.3750  0.489650     0.37500
...         ...       ...     ...       ...         ...
97707  0.274443  0.119531  0.6250  0.489650     0.31250
97708  0.399297  0.363797  0.0000  0.602707     0.62500
97709  0.233347  0.285477  0.3750  0.489650     0.37500
97710  0.281389  0.350834  0.3750  0.427548     0.43750
97711  0.421342  0.151779  0.6875  0.400478     0.46875
```

```
[97712 rows x 15 columns]
```

```
# Assuming 'df' is your DataFrame containing the dataset

# Selecting only the normalized and numeric encoded variables
numeric_encoded_df = merged_df[['model_enc','transmission_enc',
'fuelType_enc', 'Manufacturer_enc', 'year', 'price', 'mileage', 'tax',
'mpg', 'engineSize']]

# Displaying the new DataFrame
print(numeric_encoded_df)
```

```
       model_enc  transmission_enc  fuelType_enc  Manufacturer_enc
year  \
0             81                 0             0                 8
2017
1            115                 0             0                 1
2017
2              1                 1             1                 4
2019
3            184                 0             1                 7
2017
4             60                 0             0                 0
2017
...          ...               ...           ...               ...   .
..
97707         60                 2             0                 0
2017
97708          2                 0             1                 4
2014
97709         60                 0             0                 0
2017
97710         25                 2             0                 2
2017
97711         79                 0             1                 2
2019

          price   mileage     tax       mpg  engineSize
0      0.193403  0.166749  0.6250  0.582006     0.31250
1      0.289322  0.131905  0.6250  0.562898     0.31250
2      0.756043  0.023129  0.6250  0.414809     0.62500
3      0.330666  0.443923  0.6875  0.625000     0.62500
4      0.207239  0.277490  0.3750  0.489650     0.37500
...         ...       ...     ...       ...         ...
97707  0.274443  0.119531  0.6250  0.489650     0.31250
97708  0.399297  0.363797  0.0000  0.602707     0.62500
97709  0.233347  0.285477  0.3750  0.489650     0.37500
97710  0.281389  0.350834  0.3750  0.427548     0.43750
97711  0.421342  0.151779  0.6875  0.400478     0.46875
```

```
[97712 rows x 10 columns]
```

```
numeric_encoded_df1 = merged_df[['transmission_enc', 'fuelType_enc',
'Manufacturer_enc', 'year', 'price', 'mileage', 'tax', 'mpg',
'engineSize']]
```

```
from sklearn.cluster import KMeans
```

```
k = 4
```

```
kmeans = KMeans(n_clusters=k, random_state=42)
clusters = kmeans.fit_predict(numeric_encoded_df)
```

```
/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
cluster/_kmeans.py:870: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  warnings.warn(
```

```
numeric_encoded_df['Cluster'] = clusters
```

```
/var/folders/j4/9pz_sc_x7298x60tw9kbd6jw0000gn/T/
ipykernel_89397/3918485800.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  numeric_encoded_df['Cluster'] = clusters
```

```
# Import necessary libraries
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix
```

```
X = numeric_encoded_df.drop(columns=['Cluster'])  #
y = numeric_encoded_df['Cluster']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42, stratify=y)
```

```
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (78169, 10)
X_test shape: (19543, 10)
y_train shape: (78169,)
y_test shape: (19543,)
```

```python
# Fit the Decision Tree classifier with Gini criterion on the training data
dt_model_gini = dt_gini.fit(X_train, y_train)

# Predict on the testing data
y_pred_gini = dt_gini.predict(X_test)

# Calculate accuracy
accuracy_gini = accuracy_score(y_test, y_pred_gini)

# Display classification report
print("Classification Report for Decision Tree with Gini Criterion:")
print(classification_report(y_test, y_pred_gini))

# Initialize Decision Tree classifier with Entropy criterion
dt_entropy = DecisionTreeClassifier(criterion='entropy', random_state=42)

# Fit the Decision Tree classifier with Entropy criterion on the training data
dt_model_entropy = dt_entropy.fit(X_train, y_train)

# Predict on the testing data
y_pred_entropy = dt_entropy.predict(X_test)

# Calculate accuracy
accuracy_entropy = accuracy_score(y_test, y_pred_entropy)

# Display classification report
print("\nClassification Report for Decision Tree with Entropy Criterion:")
print(classification_report(y_test, y_pred_entropy))
```

```
Classification Report for Decision Tree with Gini Criterion:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4141
           1       1.00      1.00      1.00      5907
           2       1.00      1.00      1.00      3021
           3       1.00      1.00      1.00      6474

    accuracy                           1.00     19543
   macro avg       1.00      1.00      1.00     19543
weighted avg       1.00      1.00      1.00     19543


Classification Report for Decision Tree with Entropy Criterion:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4141
           1       1.00      1.00      1.00      5907
```

```
           2       1.00      1.00      1.00      3021
           3       1.00      1.00      1.00      6474

    accuracy                           1.00     19543
   macro avg       1.00      1.00      1.00     19543
weighted avg       1.00      1.00      1.00     19543
```

```
from sklearn.tree import export_text

# Generate and print decision tree rules for Gini criterion
tree_rules_gini = export_text(dt_model_gini,
feature_names=X.columns.tolist())
print("Decision Tree Rules for Gini Criterion:\n", tree_rules_gini)

# Generate and print decision tree rules for Entropy criterion
tree_rules_entropy = export_text(dt_model_entropy,
feature_names=X.columns.tolist())
print("\nDecision Tree Rules for Entropy Criterion:\n",
tree_rules_entropy)
```

```
Decision Tree Rules for Gini Criterion:
 |--- Manufacturer_enc <= 2.50
|   |--- year <= 2015.50
|   |   |--- Manufacturer_enc <= 0.50
|   |   |   |--- year <= 2014.50
|   |   |   |   |--- class: 0
|   |   |   |--- year >  2014.50
|   |   |   |   |--- fuelType_enc <= 1.50
|   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |--- mileage <= 0.89
|   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |--- mileage >  0.89
|   |   |   |   |   |   |   |--- mpg <= 0.51
|   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |--- mpg >  0.51
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- fuelType_enc >  1.50
|   |   |   |   |   |--- class: 0
|   |   |--- Manufacturer_enc >  0.50
|   |   |   |--- class: 0
|   |--- year >  2015.50
|   |   |--- transmission_enc <= 1.50
|   |   |   |--- year <= 2019.50
|   |   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |   |--- class: 3
|   |   |   |   |--- transmission_enc >  0.50
```

```
|   |   |   |   |   |--- Manufacturer_enc <= 1.50
|   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |--- Manufacturer_enc >  1.50
|   |   |   |   |   |   |--- price <= 0.48
|   |   |   |   |   |   |   |--- fuelType_enc <= 0.50
|   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |--- fuelType_enc >  0.50
|   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- price >  0.48
|   |   |   |   |   |   |   |--- engineSize <= 0.45
|   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |--- engineSize >  0.45
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- year >  2019.50
|   |   |   |   |   |--- Manufacturer_enc <= 1.50
|   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |--- Manufacturer_enc >  1.50
|   |   |   |   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |--- transmission_enc >  0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- transmission_enc >  1.50
|   |   |   |   |--- Manufacturer_enc <= 1.50
|   |   |   |   |   |--- class: 3
|   |   |   |   |--- Manufacturer_enc >  1.50
|   |   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |--- year >  2017.50
|   |   |   |   |   |   |--- price <= 0.23
|   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |--- price >  0.23
|   |   |   |   |   |   |   |--- engineSize <= 0.19
|   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |--- engineSize >  0.19
|   |   |   |   |   |   |   |   |--- class: 1
|--- Manufacturer_enc >  2.50
|   |--- Manufacturer_enc <= 5.50
|   |   |--- year <= 2015.50
|   |   |   |--- Manufacturer_enc <= 4.50
|   |   |   |   |--- class: 0
|   |   |   |--- Manufacturer_enc >  4.50
|   |   |   |   |--- year <= 2014.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- year >  2014.50
|   |   |   |   |   |--- class: 2
|   |   |--- year >  2015.50
```

```
|   |   |   |--- Manufacturer_enc <= 4.50
|   |   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |   |--- mpg <= 0.45
|   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.50
|   |   |   |   |   |   |   |   |--- price <= 0.45
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.22
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- mileage >  0.22
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.28
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.28
|   |   |   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |   |--- price >  0.45
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.43
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- mpg >  0.43
|   |   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |--- Manufacturer_enc >  3.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- mpg >  0.45
|   |   |   |   |   |   |   |--- mpg <= 0.49
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.50
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.25
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- mileage >  0.25
|   |   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  3.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- mpg >  0.49
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- year >  2016.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- transmission_enc >  0.50
|   |   |   |   |   |--- class: 1
|   |   |   |--- Manufacturer_enc >  4.50
|   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |--- price <= 0.89
|   |   |   |   |   |   |   |--- engineSize <= 0.86
|   |   |   |   |   |   |   |   |--- price <= 0.68
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- price >  0.68
|   |   |   |   |   |   |   |   |   |--- price <= 0.68
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- price >  0.68
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- engineSize >  0.86
```

```
|   |   |   |   |   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- year >  2016.50
|   |   |   |   |   |   |   |   |   |--- fuelType_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- fuelType_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.64
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- price >  0.64
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |--- price >  0.89
|   |   |   |   |   |   |   |   |--- mpg <= 0.21
|   |   |   |   |   |   |   |   |   |--- tax <= 0.84
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- tax >  0.84
|   |   |   |   |   |   |   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- year >  2016.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- mpg >  0.21
|   |   |   |   |   |   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- year >  2016.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- year >  2016.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- year >  2017.50
|   |   |   |   |   |   |--- class: 1
|   |   |--- Manufacturer_enc >  5.50
|   |   |   |--- year <= 2013.50
|   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |   |--- year <= 2011.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- year >  2011.50
|   |   |   |   |   |   |--- year <= 2012.50
|   |   |   |   |   |   |   |--- Manufacturer_enc <= 7.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Manufacturer_enc >  7.50
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |--- year >  2012.50
|   |   |   |   |   |   |   |--- class: 2
|   |   |   |--- year >  2013.50
|   |   |   |   |--- price <= 0.58
```

```
|   |   |   |   |--- fuelType_enc <= 1.50
|   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |   |   |   |   |--- mpg <= 0.12
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- mpg >  0.12
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- transmission_enc >  0.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |--- fuelType_enc >  1.50
|   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |   |   |   |--- class: 2
|   |   |   |--- price >  0.58
|   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |   |   |--- engineSize <= 0.69
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- engineSize >  0.69
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |--- fuelType_enc <= 0.50
|   |   |   |   |   |   |   |   |--- price <= 0.65
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- price >  0.65
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- fuelType_enc >  0.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |   |--- class: 2


Decision Tree Rules for Entropy Criterion:
 |--- Manufacturer_enc <= 2.50
```

```
|   |--- year <= 2015.50
|   |   |--- Manufacturer_enc <= 0.50
|   |   |   |--- year <= 2014.50
|   |   |   |   |--- class: 0
|   |   |   |--- year >  2014.50
|   |   |   |   |--- fuelType_enc <= 1.50
|   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |--- mileage <= 0.89
|   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |--- mileage >  0.89
|   |   |   |   |   |   |   |--- mpg <= 0.51
|   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |--- mpg >  0.51
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- fuelType_enc >  1.50
|   |   |   |   |   |--- class: 0
|   |   |--- Manufacturer_enc >  0.50
|   |   |   |--- class: 0
|   |--- year >  2015.50
|   |   |--- transmission_enc <= 1.50
|   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |--- class: 3
|   |   |   |--- transmission_enc >  0.50
|   |   |   |   |--- Manufacturer_enc <= 1.50
|   |   |   |   |   |--- class: 3
|   |   |   |   |--- Manufacturer_enc >  1.50
|   |   |   |   |   |--- price <= 0.44
|   |   |   |   |   |   |--- engineSize <= 0.45
|   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |--- engineSize >  0.45
|   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- price >  0.44
|   |   |   |   |   |   |--- mpg <= 0.24
|   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |--- mpg >  0.24
|   |   |   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |   |   |--- mpg <= 0.35
|   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |   |--- mpg >  0.35
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |--- transmission_enc >  1.50
|   |   |   |--- Manufacturer_enc <= 1.50
```

```
|   |   |   |   |--- class: 3
|   |   |   |--- Manufacturer_enc >  1.50
|   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |--- class: 3
|   |   |   |   |--- year >  2017.50
|   |   |   |   |   |--- engineSize <= 0.19
|   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |--- engineSize >  0.19
|   |   |   |   |   |   |--- price <= 0.23
|   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |--- price >  0.23
|   |   |   |   |   |   |   |--- class: 1
|--- Manufacturer_enc >  2.50
|   |--- Manufacturer_enc <= 4.50
|   |   |--- year <= 2015.50
|   |   |   |--- class: 0
|   |   |--- year >  2015.50
|   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |--- mpg <= 0.45
|   |   |   |   |   |   |--- Manufacturer_enc <= 3.50
|   |   |   |   |   |   |   |--- mpg <= 0.42
|   |   |   |   |   |   |   |   |--- price <= 0.44
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.28
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.38
|   |   |   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |   |   |   |--- price >  0.38
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- mileage >  0.28
|   |   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |   |--- price >  0.44
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- mpg >  0.42
|   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |--- Manufacturer_enc >  3.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- mpg >  0.45
|   |   |   |   |   |   |--- mpg <= 0.49
|   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.50
|   |   |   |   |   |   |   |   |--- price <= 0.39
|   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |   |--- price >  0.39
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- Manufacturer_enc >  3.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- mpg >  0.49
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- year >  2016.50
|   |   |   |   |   |--- class: 1
```

```
|   |   |   |--- transmission_enc >  0.50
|   |   |   |   |--- class: 1
|   |--- Manufacturer_enc >  4.50
|   |   |--- price <= 0.52
|   |   |   |--- year <= 2014.50
|   |   |   |   |--- Manufacturer_enc <= 5.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |   |--- year <= 2013.50
|   |   |   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |   |   |   |--- year <= 2012.50
|   |   |   |   |   |   |   |   |--- year <= 2011.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- year >  2011.50
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 7.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  7.50
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- year >  2012.50
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- year >  2013.50
|   |   |   |   |   |   |--- class: 2
|   |   |   |--- year >  2014.50
|   |   |   |   |--- Manufacturer_enc <= 5.50
|   |   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- year >  2016.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- year >  2017.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |--- fuelType_enc <= 1.50
|   |   |   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |   |   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.61
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- engineSize >  0.61
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- transmission_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- fuelType_enc >  1.50
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |--- price >  0.52
|   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |   |--- year <= 2014.50
|   |   |   |   |   |   |   |--- Manufacturer_enc <= 5.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |   |   |   |   |--- year <= 2013.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- year >  2013.50
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |--- year >  2014.50
|   |   |   |   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |--- year >  2016.50
|   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |--- engineSize <= 0.92
|   |   |   |   |   |   |   |   |   |--- tax <= 0.94
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- tax >  0.94
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.35
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.35
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |--- engineSize >  0.92
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.25
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.10
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.10
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- mpg >  0.25
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.64
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- price >  0.64
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |--- transmission_enc >  1.50
```

```
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 5.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |--- year >  2017.50
|   |   |   |   |   |--- Manufacturer_enc <= 5.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |   |   |   |--- fuelType_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- fuelType_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |--- fuelType_enc <= 0.50
|   |   |   |   |   |   |   |   |   |--- price <= 0.65
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- price >  0.65
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- fuelType_enc >  0.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |--- class: 2
```

```
# Subset numeric_encoded_df based on Inputs as {mpg, hp, cyl, vs} &
Output as {am}
numeric_encoded_df_inputs = numeric_encoded_df[['transmission_enc',
'model_enc', 'Manufacturer_enc', 'year', 'price', 'mileage', 'tax',
'mpg', 'engineSize']]; numeric_encoded_df
numeric_encoded_df_output = numeric_encoded_df[['fuelType_enc']];
numeric_encoded_df_output
```

```
       fuelType_enc
0                 0
1                 0
2                 1
3                 1
4                 0
...             ...
97707             0
97708             1
97709             0
97710             0
```

```
97711              1

[97712 rows x 1 columns]
```

```
numeric_encoded_df_input_names = numeric_encoded_df_inputs.columns;
numeric_encoded_df_input_names
numeric_encoded_df_output_labels =
numeric_encoded_df_output['fuelType_enc'].unique().astype(str);
numeric_encoded_df_output_labels
```

```
array(['0', '1', '2', '4', '3'], dtype='<U21')
```

```
# Split the mtcars Data Subset into Training & Testing Sets
from sklearn.model_selection import train_test_split

# Splitting the dataset with stratification
train_cars_inputs, test_cars_inputs, train_cars_output,
test_cars_output = train_test_split(
    numeric_encoded_df_inputs,
    numeric_encoded_df_output,
    test_size=0.2,  # 20% for testing
    stratify=numeric_encoded_df_output,  # Stratify by the output
variable
    random_state=5050
)

# Printing the shapes of the resulting sets
print("Training inputs shape:", train_mtcars_inputs.shape)
print("Testing inputs shape:", test_mtcars_inputs.shape)
print("Training outputs shape:", train_mtcars_output.shape)
print("Testing outputs shape:", test_mtcars_output.shape)
```

```
Training inputs shape: (78169, 9)
Testing inputs shape: (19543, 9)
Training outputs shape: (78169, 1)
Testing outputs shape: (19543, 1)
```

```
#Decision Tree : Model (Training Subset)
dtc = DecisionTreeClassifier(criterion='gini', random_state=1234) #
Other Criteria : Entropy,  Log Loss
dtc_model = dtc.fit(train_cars_inputs, train_cars_output); dtc_model

# Decision Tree : Model Rules
dtc_model_rules = export_text(dtc_model, feature_names =
list(numeric_encoded_df_input_names)); print(dtc_model_rules)

# Decision Tree : Feature Importance
dtc_imp_features = pd.DataFrame({'feature':
numeric_encoded_df_input_names, 'importance':
np.round(dtc_model.feature_importances_, 3)})
```

```
dtc_imp_features.sort_values('importance', ascending=False,
inplace=True); dtc_imp_features
```

```
|--- engineSize <= 0.45
|   |--- mpg <= 0.76
|   |   |--- engineSize <= 0.16
|   |   |   |--- price <= 0.34
|   |   |   |   |--- mileage <= 0.74
|   |   |   |   |   |--- mpg <= 0.70
|   |   |   |   |   |   |--- mpg <= 0.59
|   |   |   |   |   |   |   |--- tax <= 0.84
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- tax >  0.84
|   |   |   |   |   |   |   |   |--- mpg <= 0.36
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- mpg >  0.36
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- mpg >  0.59
|   |   |   |   |   |   |   |--- mileage <= 0.42
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 7.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  7.50
|   |   |   |   |   |   |   |   |   |--- model_enc <= 82.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  82.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- mileage >  0.42
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- mpg >  0.70
|   |   |   |   |   |   |--- mileage <= 0.29
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- mileage >  0.29
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- mileage >  0.74
|   |   |   |   |   |--- Manufacturer_enc <= 4.00
|   |   |   |   |   |   |--- transmission_enc <= 1.00
|   |   |   |   |   |   |   |--- model_enc <= 85.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- model_enc >  85.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- transmission_enc >  1.00
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Manufacturer_enc >  4.00
|   |   |   |   |   |   |--- class: 1
|   |   |   |--- price >  0.34
|   |   |   |   |--- mpg <= 0.53
|   |   |   |   |   |--- mpg <= 0.37
|   |   |   |   |   |   |--- year <= 2015.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- year >  2015.50
```

```
|   |   |   |   |   |   |   |   |--- mpg <= 0.27
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.13
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- mpg >  0.13
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.89
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.16
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- mpg >  0.16
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- price >  0.89
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.16
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- mpg >  0.16
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- mpg >  0.27
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- mpg >  0.37
|   |   |   |   |   |   |   |--- model_enc <= 116.50
|   |   |   |   |   |   |   |   |--- tax <= 0.66
|   |   |   |   |   |   |   |   |   |--- transmission_enc <= 2.50
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.38
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.38
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.48
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- mileage >  0.48
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- transmission_enc >  2.50
|   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |--- tax >  0.66
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- model_enc >  116.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- mpg >  0.53
|   |   |   |   |   |   |--- mileage <= 0.16
|   |   |   |   |   |   |   |--- model_enc <= 112.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- model_enc >  112.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- mileage >  0.16
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- engineSize >  0.16
|   |   |   |   |--- price <= 0.78
|   |   |   |   |   |--- mpg <= 0.74
|   |   |   |   |   |   |--- year <= 2012.50
|   |   |   |   |   |   |   |--- mpg <= 0.61
|   |   |   |   |   |   |   |   |--- mpg <= 0.52
|   |   |   |   |   |   |   |   |   |--- model_enc <= 176.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.02
```

```
|   |   |   |   |   |   |   |   |   |   |--- tax <= 0.31
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- tax >  0.31
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- price >  0.02
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.14
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- price >  0.14
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |--- model_enc >  176.50
|   |   |   |   |   |   |   |   |   |--- price <= 0.12
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- price >  0.12
|   |   |   |   |   |   |   |   |   |   |--- year <= 2009.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |--- year >  2009.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |--- mpg >  0.52
|   |   |   |   |   |   |   |   |--- tax <= 0.19
|   |   |   |   |   |   |   |   |   |--- engineSize <= 0.41
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- engineSize >  0.41
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- tax >  0.19
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- mpg >  0.61
|   |   |   |   |   |   |   |--- engineSize <= 0.34
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- engineSize >  0.34
|   |   |   |   |   |   |   |   |--- year <= 2011.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- year >  2011.50
|   |   |   |   |   |   |   |   |   |--- price <= 0.10
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.09
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- price >  0.09
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- price >  0.10
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- year >  2012.50
|   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |--- price <= 0.07
|   |   |   |   |   |   |   |   |--- price <= 0.06
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- price >  0.06
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- price >  0.07
```

```
|   |   |   |   |   |   |   |   |--- price <= 0.76
|   |   |   |   |   |   |   |   |   |--- price <= 0.10
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.68
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.68
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |--- price >  0.10
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.69
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 15
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.69
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 8
|   |   |   |   |   |   |   |   |--- price >  0.76
|   |   |   |   |   |   |   |   |   |--- tax <= 0.56
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- tax >  0.56
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |--- mpg <= 0.68
|   |   |   |   |   |   |   |   |--- price <= 0.73
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 5.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 114.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  114.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 8
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 30.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  30.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 8
|   |   |   |   |   |   |   |   |--- price >  0.73
|   |   |   |   |   |   |   |   |   |--- engineSize <= 0.42
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- engineSize >  0.42
|   |   |   |   |   |   |   |   |   |   |--- tax <= 0.56
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- tax >  0.56
|   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |--- mpg >  0.68
|   |   |   |   |   |   |   |   |--- price <= 0.22
|   |   |   |   |   |   |   |   |   |--- price <= 0.21
|   |   |   |   |   |   |   |   |   |   |--- class: 0
```

```
|   |   |   |   |   |   |   |   |   |--- price >  0.21
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.32
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.32
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |--- price >  0.22
|   |   |   |   |   |   |   |   |   |--- price <= 0.24
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.26
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.26
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |--- price >  0.24
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.07
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.07
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |--- mpg >  0.74
|   |   |   |   |   |--- model_enc <= 30.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- model_enc >  30.50
|   |   |   |   |   |   |--- model_enc <= 70.00
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- model_enc >  70.00
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |--- price >  0.78
|   |   |   |   |--- tax <= 0.59
|   |   |   |   |   |--- price <= 0.81
|   |   |   |   |   |   |--- mpg <= 0.30
|   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |--- mpg >  0.30
|   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- price >  0.81
|   |   |   |   |   |   |--- class: 2
|   |   |   |   |--- tax >  0.59
|   |   |   |   |   |--- class: 0
|   |--- mpg >  0.76
|   |   |--- mpg <= 0.99
|   |   |   |--- Manufacturer_enc <= 0.50
|   |   |   |   |--- class: 0
|   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- year >  2017.50
```

```
|   |   |   |   |   |   |--- tax <= 0.31
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- tax >  0.31
|   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |--- mileage <= 0.24
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- mileage >  0.24
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- mpg >  0.99
|   |   |   |   |--- price <= 0.25
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- price >  0.25
|   |   |   |   |   |--- price <= 0.99
|   |   |   |   |   |   |--- price <= 0.53
|   |   |   |   |   |   |   |--- price <= 0.45
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.00
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.38
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- price >  0.38
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.40
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |   |--- price >  0.40
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.22
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- engineSize >  0.22
|   |   |   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  3.00
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- price >  0.45
|   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |   |--- engineSize <= 0.09
|   |   |   |   |   |   |   |   |   |   |--- tax <= 0.25
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- tax >  0.25
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.27
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |   |--- mileage >  0.27
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 3
|   |   |   |   |   |   |   |   |   |--- engineSize >  0.09
|   |   |   |   |   |   |   |   |   |   |--- year <= 2015.50
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.47
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 3
```

```
|   |   |   |   |   |   |   |   |   |   |--- price >  0.47
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- year >  2015.50
|   |   |   |   |   |   |   |   |   |   |--- tax <= 0.53
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |--- tax >  0.53
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- price >  0.53
|   |   |   |   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- transmission_enc >  0.50
|   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |--- engineSize <= 0.09
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- engineSize >  0.09
|   |   |   |   |   |   |   |   |   |--- model_enc <= 169.50
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.36
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- mileage >  0.36
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |--- model_enc >  169.50
|   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |--- price >  0.99
|   |   |   |   |   |--- class: 0
|--- engineSize >  0.45
|   |--- mpg <= 0.39
|   |   |--- engineSize <= 0.58
|   |   |   |--- mpg <= 0.34
|   |   |   |   |--- tax <= 0.59
|   |   |   |   |   |--- class: 2
|   |   |   |   |--- tax >  0.59
|   |   |   |   |   |--- model_enc <= 171.50
|   |   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |   |--- engineSize <= 0.52
|   |   |   |   |   |   |   |   |--- price <= 0.75
|   |   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.33
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 8
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.33
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 6
|   |   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 155.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
```

```
depth 7
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  155.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 6
|   |   |   |   |   |   |   |   |   |--- price >  0.75
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 155.00
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.76
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- price >  0.76
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  155.00
|   |   |   |   |   |   |   |   |   |   |   |--- model_enc <= 157.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- model_enc >  157.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- engineSize >  0.52
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.32
|   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.55
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.23
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- mpg >  0.23
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- engineSize >  0.55
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.14
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- mileage >  0.14
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |--- mpg >  0.32
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  3.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |   |--- mpg <= 0.33
|   |   |   |   |   |   |   |   |--- model_enc <= 155.00
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.07
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- mileage >  0.07
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 136.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  136.00
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |--- model_enc >  155.00
|   |   |   |   |   |   |   |   |   |--- price <= 0.78
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 0.50
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- price >  0.78
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.21
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.21
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- mpg >  0.33
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.04
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- mileage >  0.04
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- model_enc >  171.50
|   |   |   |   |   |   |--- model_enc <= 174.00
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- model_enc >  174.00
|   |   |   |   |   |   |   |--- mpg <= 0.31
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- mpg >  0.31
|   |   |   |   |   |   |   |   |--- year <= 2018.00
|   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |--- year >  2018.00
|   |   |   |   |   |   |   |   |   |--- tax <= 0.66
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- tax >  0.66
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |--- mpg >  0.34
|   |   |   |   |--- Manufacturer_enc <= 7.50
|   |   |   |   |   |--- model_enc <= 95.50
|   |   |   |   |   |   |--- model_enc <= 0.50
|   |   |   |   |   |   |   |--- price <= 0.47
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- price >  0.47
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- model_enc >  0.50
|   |   |   |   |   |   |   |--- tax <= 0.94
|   |   |   |   |   |   |   |   |--- year <= 2008.50
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.89
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- mileage >  0.89
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- year >  2008.50
```

```
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.36
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 77.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 6
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  77.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |--- mpg >  0.36
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 8.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |--- model_enc >  8.50
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |--- tax >  0.94
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- model_enc >  95.50
|   |   |   |   |   |   |--- price <= 0.54
|   |   |   |   |   |   |   |--- model_enc <= 109.50
|   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.36
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- mpg >  0.36
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- model_enc >  109.50
|   |   |   |   |   |   |   |   |--- mileage <= 0.87
|   |   |   |   |   |   |   |   |   |--- tax <= 0.59
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 5.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- tax >  0.59
|   |   |   |   |   |   |   |   |   |--- price <= 0.54
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- price >  0.54
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |--- mileage >  0.87
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- price >  0.54
|   |   |   |   |   |   |   |--- engineSize <= 0.48
|   |   |   |   |   |   |   |   |--- model_enc <= 106.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- model_enc >  106.50
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.07
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- mileage >  0.07
```

```
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.07
|   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.07
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- engineSize >  0.48
|   |   |   |   |   |   |   |   |   |--- engineSize <= 0.53
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- engineSize >  0.53
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- Manufacturer_enc >  7.50
|   |   |   |   |   |   |--- mpg <= 0.38
|   |   |   |   |   |   |   |--- year <= 2010.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- year >  2010.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- mpg >  0.38
|   |   |   |   |   |   |   |--- class: 2
|   |   |   |--- engineSize >  0.58
|   |   |   |   |--- model_enc <= 77.00
|   |   |   |   |   |--- price <= 0.85
|   |   |   |   |   |   |--- mpg <= 0.30
|   |   |   |   |   |   |   |--- mileage <= 0.86
|   |   |   |   |   |   |   |   |--- mpg <= 0.22
|   |   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.63
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.21
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.21
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |--- mileage >  0.63
|   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.70
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |--- engineSize >  0.70
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 2.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 64.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  64.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  2.50
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.65
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.65
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
```

```
depth 2
|   |   |   |   |   |   |   |--- mpg >  0.22
|   |   |   |   |   |   |   |   |--- engineSize <= 0.64
|   |   |   |   |   |   |   |   |   |--- model_enc <= 67.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  67.00
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 69.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  69.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |--- engineSize >  0.64
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 61.00
|   |   |   |   |   |   |   |   |   |   |   |--- model_enc <= 13.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- model_enc >  13.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  61.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- mileage >  0.86
|   |   |   |   |   |   |   |   |--- model_enc <= 4.50
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.26
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.15
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- price >  0.15
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.15
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- price >  0.15
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |--- mpg >  0.26
|   |   |   |   |   |   |   |   |   |   |--- year <= 2007.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- year >  2007.50
|   |   |   |   |   |   |   |   |   |   |   |--- model_enc <= 3.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- model_enc >  3.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- model_enc >  4.50
|   |   |   |   |   |   |   |   |   |--- model_enc <= 70.00
|   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.97
|   |   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 7
|   |   |   |   |   |   |   |   |   |--- engineSize >  0.97
```

```
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  70.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- mpg >  0.30
|   |   |   |   |   |   |--- engineSize <= 0.64
|   |   |   |   |   |   |   |--- Manufacturer_enc <= 0.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 4.50
|   |   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |   |--- year <= 2008.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |--- year >  2008.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 9
|   |   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 37.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 6
|   |   |   |   |   |   |   |   |   |--- model_enc >  37.00
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  4.50
|   |   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.47
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |--- price >  0.47
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.32
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- mpg >  0.32
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |--- engineSize >  0.64
|   |   |   |   |   |   |   |--- tax <= 0.56
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- tax >  0.56
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- price >  0.85
|   |   |   |   |   |--- mpg <= 0.21
|   |   |   |   |   |   |--- Manufacturer_enc <= 2.00
|   |   |   |   |   |   |   |--- model_enc <= 61.00
|   |   |   |   |   |   |   |   |--- price <= 0.89
|   |   |   |   |   |   |   |   |   |--- price <= 0.88
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- price >  0.88
```

```
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- price >  0.89
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.00
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.11
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- mpg >  0.11
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.00
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.97
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- price >  0.97
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- model_enc >  61.00
|   |   |   |   |   |   |   |   |   |--- price <= 0.97
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- price >  0.97
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.97
|   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |--- price >  0.97
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Manufacturer_enc >  2.00
|   |   |   |   |   |   |   |--- model_enc <= 68.50
|   |   |   |   |   |   |   |   |--- mpg <= 0.19
|   |   |   |   |   |   |   |   |   |--- tax <= 0.56
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- tax >  0.56
|   |   |   |   |   |   |   |   |   |   |--- year <= 2013.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- year >  2013.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |--- mpg >  0.19
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 13.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  13.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.89
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- price >  0.89
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |--- model_enc >  68.50
|   |   |   |   |   |   |   |   |--- mpg <= 0.14
|   |   |   |   |   |   |   |   |   |--- model_enc <= 69.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  69.50
|   |   |   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
```

```
|   |   |   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- mpg >  0.14
|   |   |   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.16
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- mpg >  0.16
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- mpg >  0.21
|   |   |   |   |   |   |   |--- model_enc <= 38.00
|   |   |   |   |   |   |   |   |--- engineSize <= 0.78
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.33
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 2.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  2.00
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.32
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- mpg >  0.32
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |--- mpg >  0.33
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 8.00
|   |   |   |   |   |   |   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  8.00
|   |   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 2.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  2.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- engineSize >  0.78
|   |   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 13.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  13.50
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.23
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- mpg >  0.23
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 4.50
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.30
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
```

```
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.30
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- model_enc >  4.50
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.23
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.23
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- model_enc >  38.00
|   |   |   |   |   |   |   |   |--- model_enc <= 72.50
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.27
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 55.00
|   |   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.77
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- engineSize >  0.77
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  55.00
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.87
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- price >  0.87
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |--- mpg >  0.27
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.28
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.17
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- mileage >  0.17
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.28
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- model_enc >  72.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- model_enc >  77.00
|   |   |   |   |   |--- mpg <= 0.14
|   |   |   |   |   |   |--- model_enc <= 87.00
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- model_enc >  87.00
|   |   |   |   |   |   |   |--- model_enc <= 101.50
|   |   |   |   |   |   |   |   |--- model_enc <= 99.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- model_enc >  99.50
|   |   |   |   |   |   |   |   |   |--- engineSize <= 0.97
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- engineSize >  0.97
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- model_enc >  101.50
```

```
|   |   |   |   |   |   |   |--- model_enc <= 155.50
|   |   |   |   |   |   |   |   |--- engineSize <= 0.92
|   |   |   |   |   |   |   |   |   |--- model_enc <= 151.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  151.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.89
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- price >  0.89
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- engineSize >  0.92
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.10
|   |   |   |   |   |   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |--- mpg >  0.10
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 4.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  4.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |--- model_enc >  155.50
|   |   |   |   |   |   |   |   |--- mpg <= 0.12
|   |   |   |   |   |   |   |   |   |--- tax <= 0.59
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- tax >  0.59
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 8
|   |   |   |   |   |   |   |   |--- mpg >  0.12
|   |   |   |   |   |   |   |   |   |--- model_enc <= 159.50
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.14
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 8
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.14
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  159.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.44
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- price >  0.44
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- mpg >  0.14
|   |   |   |   |   |--- engineSize <= 0.64
```

```
|   |   |   |   |   |   |--- mpg <= 0.34
|   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |--- price <= 0.31
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.25
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- mpg >  0.25
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 185.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  185.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |--- price >  0.31
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 7.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 10
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  7.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |--- mpg <= 0.18
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.15
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 4.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  4.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- mpg >  0.15
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.18
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.18
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |--- mpg >  0.18
|   |   |   |   |   |   |   |   |   |--- model_enc <= 171.00
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.58
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 7
|   |   |   |   |   |   |   |   |   |   |--- price >  0.58
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 9
|   |   |   |   |   |   |   |   |   |--- model_enc >  171.00
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.27
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.27
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |--- mpg >  0.34
|   |   |   |   |   |   |   |--- tax <= 0.94
|   |   |   |   |   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 7.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  7.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- year >  2017.50
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 2.50
|   |   |   |   |   |   |   |   |   |   |--- tax <= 0.59
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- tax >  0.59
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  2.50
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  3.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |--- tax >  0.94
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- engineSize >  0.64
|   |   |   |   |   |   |--- mpg <= 0.21
|   |   |   |   |   |   |   |--- model_enc <= 120.50
|   |   |   |   |   |   |   |   |--- mpg <= 0.17
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 5.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- mpg >  0.17
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 4.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  4.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- model_enc >  120.50
|   |   |   |   |   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |   |   |   |   |--- model_enc <= 154.50
```

```
|   |   |   |   |   |   |   |   |   |   |--- year <= 2011.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- year >  2011.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  154.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 183.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  183.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- year >  2017.50
|   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.97
|   |   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 5.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 7
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- engineSize >  0.97
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- mpg >  0.21
|   |   |   |   |   |   |   |--- mileage <= 0.03
|   |   |   |   |   |   |   |   |--- model_enc <= 135.50
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.27
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- mpg >  0.27
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.28
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.28
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- model_enc >  135.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- mileage >  0.03
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 0.50
|   |   |   |   |   |   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- year >  2016.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.50
|   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.97
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |--- engineSize >  0.97
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  3.50
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.25
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.25
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |--- mpg >  0.39
|   |   |--- Manufacturer_enc <= 5.50
|   |   |   |--- mpg <= 0.48
|   |   |   |   |--- engineSize <= 0.58
|   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |--- Manufacturer_enc <= 2.50
|   |   |   |   |   |   |   |--- model_enc <= 55.00
|   |   |   |   |   |   |   |   |--- price <= 0.51
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- price >  0.51
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- model_enc >  55.00
|   |   |   |   |   |   |   |   |--- engineSize <= 0.48
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.02
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- mileage >  0.02
|   |   |   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- engineSize >  0.48
|   |   |   |   |   |   |   |   |   |--- engineSize <= 0.55
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 67.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  67.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- engineSize >  0.55
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.23
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- price >  0.23
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Manufacturer_enc >  2.50
|   |   |   |   |   |   |   |--- mileage <= 0.12
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 4.50
|   |   |   |   |   |   |   |   |   |--- price <= 0.43
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.44
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.44
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- price >  0.43
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  4.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- mileage >  0.12
|   |   |   |   |   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
```

```
|   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.48
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- engineSize >  0.48
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- year >  2017.50
|   |   |   |   |   |   |   |   |   |--- model_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.45
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.45
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- model_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |--- engineSize <= 0.48
|   |   |   |   |   |   |   |--- mpg <= 0.47
|   |   |   |   |   |   |   |   |--- mpg <= 0.41
|   |   |   |   |   |   |   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- transmission_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- mpg >  0.41
|   |   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.35
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- price >  0.35
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  3.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- mpg >  0.47
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.00
|   |   |   |   |   |   |   |   |   |--- model_enc <= 116.00
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 1.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  1.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- model_enc >  116.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- Manufacturer_enc >  3.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- engineSize >  0.48
|   |   |   |   |   |   |--- mpg <= 0.41
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- mpg >  0.41
|   |   |   |   |   |   |   |--- mileage <= 0.16
```

```
|   |   |   |   |   |   |   |   |   |--- price <= 0.48
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.42
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.42
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |--- price >  0.48
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.66
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |--- price >  0.66
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |--- mileage >  0.16
|   |   |   |   |   |   |   |   |   |--- price <= 0.62
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.16
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.16
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- price >  0.62
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- engineSize >  0.58
|   |   |   |   |   |--- tax <= 0.44
|   |   |   |   |   |   |--- model_enc <= 43.00
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- model_enc >  43.00
|   |   |   |   |   |   |   |--- model_enc <= 151.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- model_enc >  151.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- tax >  0.44
|   |   |   |   |   |   |--- model_enc <= 33.00
|   |   |   |   |   |   |   |--- mpg <= 0.41
|   |   |   |   |   |   |   |   |--- price <= 0.67
|   |   |   |   |   |   |   |   |   |--- tax <= 0.75
|   |   |   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 6
|   |   |   |   |   |   |   |   |   |--- tax >  0.75
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- price >  0.67
|   |   |   |   |   |   |   |   |   |--- model_enc <= 2.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- model_enc >  2.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 4.50
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  4.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- mpg >  0.41
|   |   |   |   |   |   |   |   |--- engineSize <= 0.64
|   |   |   |   |   |   |   |   |   |--- model_enc <= 31.50
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.43
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 7
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.43
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 7
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  31.50
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.69
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- price >  0.69
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- engineSize >  0.64
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- model_enc >  33.00
|   |   |   |   |   |   |   |   |--- model_enc <= 171.00
|   |   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.45
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.45
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.10
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.10
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.61
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 6
|   |   |   |   |   |   |   |   |   |   |--- price >  0.61
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- model_enc >  171.00
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.41
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 175.50
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.02
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- mileage >  0.02
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  175.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |   |   |   |   |   |--- mpg >  0.41
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- mpg >  0.48
|   |   |   |   |--- mpg <= 0.99
|   |   |   |   |   |--- engineSize <= 0.48
|   |   |   |   |   |   |--- mpg <= 0.53
|   |   |   |   |   |   |   |--- model_enc <= 76.50
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 0.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |--- model_enc >  76.50
|   |   |   |   |   |   |   |   |   |--- model_enc <= 117.00
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.43
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.43
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.44
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- mileage >  0.44
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- model_enc >  117.00
|   |   |   |   |   |   |   |   |   |   |--- year <= 2016.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- year >  2016.00
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.51
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- mpg >  0.51
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- mpg >  0.53
|   |   |   |   |   |   |   |   |--- mpg <= 0.57
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  0.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 77.00
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.59
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- price >  0.59
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |   |   |   |   |   |--- model_enc >  77.00
|   |   |   |   |   |   |   |   |   |   |--- year <= 2017.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- year >  2017.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- mpg >  0.57
|   |   |   |   |   |   |   |   |   |--- year <= 2011.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- year >  2011.00
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.18
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.66
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.66
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- price >  0.18
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.55
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 5
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.55
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 6
|   |   |   |   |   |--- engineSize >  0.48
|   |   |   |   |   |   |--- mpg <= 0.84
|   |   |   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |   |   |--- year <= 2007.50
|   |   |   |   |   |   |   |   |   |--- price <= 0.10
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- price >  0.10
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- year >  2007.50
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.52
|   |   |   |   |   |   |   |   |   |   |--- tax <= 0.16
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |--- tax >  0.16
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 6
|   |   |   |   |   |   |   |   |--- mpg >  0.52
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.79
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 10
|   |   |   |   |   |   |   |   |--- mpg >  0.79
|   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |   |   |--- model_enc <= 6.00
|   |   |   |   |   |   |   |   |   |--- model_enc <= 3.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |   |   |   |   |   |--- model_enc >  3.50
|   |   |   |   |   |   |   |   |   |   |--- engineSize <= 0.78
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |--- engineSize >  0.78
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- model_enc >  6.00
|   |   |   |   |   |   |   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- transmission_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.62
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- mpg >  0.62
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |--- mpg >  0.84
|   |   |   |   |   |   |   |   |--- engineSize <= 0.56
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- engineSize >  0.56
|   |   |   |   |   |   |   |   |   |--- price <= 0.86
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.99
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- mileage >  0.99
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- Manufacturer_enc >  3.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |--- price >  0.86
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.40
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- mileage >  0.40
|   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |--- mpg >  0.99
|   |   |   |   |   |   |--- transmission_enc <= 0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- transmission_enc >  0.50
|   |   |   |   |   |   |   |--- price <= 0.48
|   |   |   |   |   |   |   |   |--- tax <= 0.25
|   |   |   |   |   |   |   |   |   |--- year <= 2016.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.47
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.55
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.55
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 7
|   |   |   |   |   |   |   |   |   |--- price >  0.47
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.47
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |--- price >  0.47
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |--- year >  2016.50
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.90
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.47
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- price >  0.47
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.90
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- tax >  0.25
|   |   |   |   |   |   |   |   |   |--- model_enc <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- model_enc >  1.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 17.00
|   |   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.99
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |   |--- mileage >  0.99
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  17.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- price >  0.48
|   |   |   |   |   |   |   |   |--- engineSize <= 0.86
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.69
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.51
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.51
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- price >  0.51
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |--- price >  0.51
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- mileage >  0.69
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.74
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- price >  0.74
|   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |--- engineSize >  0.86
|   |   |   |   |   |   |   |   |--- tax <= 0.25
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- tax >  0.25
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |--- Manufacturer_enc >  5.50
|   |   |   |   |--- transmission_enc <= 1.50
|   |   |   |   |   |--- engineSize <= 0.48
|   |   |   |   |   |   |--- mpg <= 0.63
```

```
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- mpg >  0.63
|   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |--- engineSize >  0.48
|   |   |   |   |   |--- mpg <= 0.87
|   |   |   |   |   |   |--- price <= 0.55
|   |   |   |   |   |   |   |--- mpg <= 0.44
|   |   |   |   |   |   |   |   |--- model_enc <= 26.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- model_enc >  26.50
|   |   |   |   |   |   |   |   |   |--- tax <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- tax >  0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- mpg >  0.44
|   |   |   |   |   |   |   |   |--- mpg <= 0.71
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- mpg >  0.71
|   |   |   |   |   |   |   |   |   |--- price <= 0.39
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- price >  0.39
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.39
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- price >  0.39
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- price >  0.55
|   |   |   |   |   |   |   |   |--- Manufacturer_enc <= 7.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- Manufacturer_enc >  7.50
|   |   |   |   |   |   |   |   |   |--- model_enc <= 87.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- model_enc >  87.00
|   |   |   |   |   |   |   |   |   |   |--- mpg <= 0.41
|   |   |   |   |   |   |   |   |   |   |   |--- year <= 2019.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- year >  2019.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- mpg >  0.41
|   |   |   |   |   |   |   |   |   |   |   |--- year <= 2018.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- year >  2018.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- mpg >  0.87
|   |   |   |   |   |   |--- Manufacturer_enc <= 7.50
|   |   |   |   |   |   |   |--- engineSize <= 0.53
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- engineSize >  0.53
|   |   |   |   |   |   |   |   |--- class: 0
```

```
|   |   |   |   |   |   |   |--- Manufacturer_enc >  7.50
|   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |--- transmission_enc >  1.50
|   |   |   |   |--- tax <= 0.59
|   |   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |   |--- price <= 0.67
|   |   |   |   |   |   |   |--- price <= 0.14
|   |   |   |   |   |   |   |   |--- mpg <= 0.71
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- mpg >  0.71
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- price >  0.14
|   |   |   |   |   |   |   |   |--- mileage <= 0.02
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.02
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- mileage >  0.02
|   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |--- mileage >  0.02
|   |   |   |   |   |   |   |   |   |--- model_enc <= 83.50
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.26
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |   |   |--- price >  0.26
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 7
|   |   |   |   |   |   |   |   |   |--- model_enc >  83.50
|   |   |   |   |   |   |   |   |   |   |--- model_enc <= 120.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 7
|   |   |   |   |   |   |   |   |   |   |--- model_enc >  120.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 11
|   |   |   |   |   |   |--- price >  0.67
|   |   |   |   |   |   |   |--- mpg <= 0.99
|   |   |   |   |   |   |   |   |--- model_enc <= 39.50
|   |   |   |   |   |   |   |   |   |--- mpg <= 0.55
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.69
|   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |--- price >  0.69
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 6
|   |   |   |   |   |   |   |   |   |--- mpg >  0.55
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- model_enc >  39.50
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.02
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.02
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |--- mileage >  0.02
|   |   |   |   |   |   |   |   |   |   |--- class: 4
```

```
|   |   |   |   |   |   |   |   |   |--- mileage >  0.02
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.19
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.19
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 2
|   |   |   |   |   |   |   |   |--- mpg >  0.99
|   |   |   |   |   |   |   |   |   |--- price <= 0.73
|   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |--- price >  0.73
|   |   |   |   |   |   |   |   |   |   |--- price <= 0.76
|   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |   |   |--- price >  0.76
|   |   |   |   |   |   |   |   |   |   |   |--- price <= 0.79
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |   |   |--- price >  0.79
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |   |   |--- model_enc <= 105.00
|   |   |   |   |   |   |   |--- model_enc <= 87.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- model_enc >  87.00
|   |   |   |   |   |   |   |   |--- mileage <= 0.76
|   |   |   |   |   |   |   |   |   |--- mileage <= 0.07
|   |   |   |   |   |   |   |   |   |   |--- mileage <= 0.07
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of
depth 3
|   |   |   |   |   |   |   |   |   |   |--- mileage >  0.07
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- mileage >  0.07
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- mileage >  0.76
|   |   |   |   |   |   |   |   |   |--- price <= 0.36
|   |   |   |   |   |   |   |   |   |   |--- class: 4
|   |   |   |   |   |   |   |   |   |--- price >  0.36
|   |   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |--- model_enc >  105.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- tax >  0.59
|   |   |   |   |   |--- engineSize <= 0.48
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- engineSize >  0.48
|   |   |   |   |   |   |--- Manufacturer_enc <= 6.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- Manufacturer_enc >  6.50
|   |   |   |   |   |   |   |--- model_enc <= 157.00
|   |   |   |   |   |   |   |   |--- mileage <= 0.00
|   |   |   |   |   |   |   |   |   |--- price <= 0.57
|   |   |   |   |   |   |   |   |   |   |--- class: 1
```

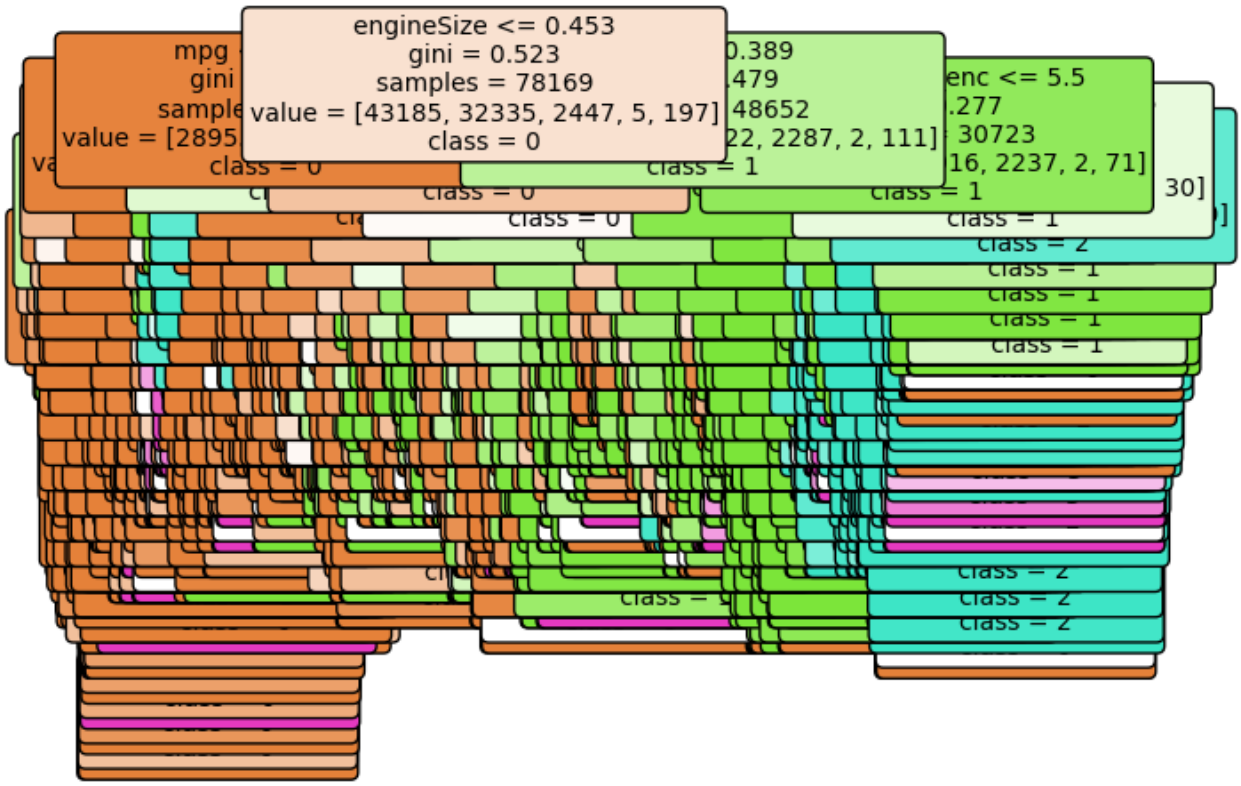
```
|   |   |   |   |   |   |   |   |   |--- price >  0.57
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- mileage >  0.00
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- model_enc >  157.00
|   |   |   |   |   |   |   |   |--- mileage <= 0.13
|   |   |   |   |   |   |   |   |   |--- class: 2
|   |   |   |   |   |   |   |   |--- mileage >  0.13
|   |   |   |   |   |   |   |   |   |--- class: 1
```

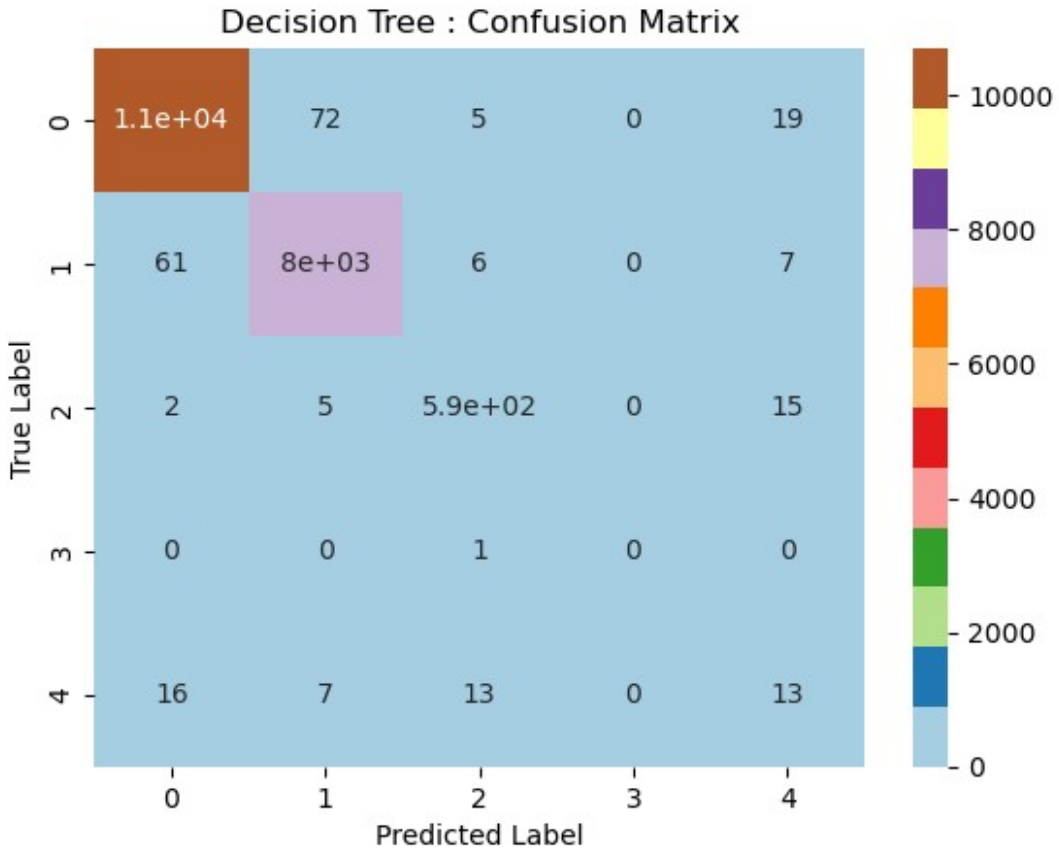
```
             feature  importance
8         engineSize       0.503
7                mpg       0.267
1          model_enc       0.066
2  Manufacturer_enc       0.053
0  transmission_enc       0.040
4              price       0.022
3               year       0.021
6                tax       0.018
5            mileage       0.009
```

```
# Decision Tree : Model Prediction (Training Subset)
dtc_model_predict = dtc_model.predict(train_cars_inputs);
dtc_model_predict

# Decision Tree : Prediction (Testing Subset)
dtc_predict = dtc_model.predict(test_cars_inputs); dtc_predict

# Decision Tree : Model Evaluation (Training Subset)
dtc_model_conf_mat = pd.DataFrame(confusion_matrix(train_cars_output,
dtc_model_predict)); dtc_model_conf_mat
dtc_model_perf = classification_report(train_cars_output,
dtc_model_predict); print(dtc_model_perf)

# Decision Tree : Prediction Evaluation (Testing Subset)
dtc_predict_conf_mat = pd.DataFrame(confusion_matrix(test_cars_output,
dtc_predict)); dtc_predict_conf_mat
dtc_predict_perf = classification_report(test_cars_output,
dtc_predict); print(dtc_predict_perf)

# Decision Tree : Plot [Training Subset]
train_subset_dtc_plot = plot_tree(dtc_model,
feature_names=numeric_encoded_df_input_names,
class_names=numeric_encoded_df_output_labels, rounded=True,
filled=True, fontsize=10)
plt.show()

# Confusion Matrix : Plot [Testing Subset]
ax = plt.axes()
```

```
sns.heatmap(dtc_predict_conf_mat, annot=True, cmap='Paired')
ax.set_xlabel('Predicted Label')
ax.set_ylabel('True Label')
ax.set_title('Decision Tree : Confusion Matrix')
plt.show()
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     43185
           1       1.00      1.00      1.00     32335
           2       1.00      1.00      1.00      2447
           3       1.00      1.00      1.00         5
           4       1.00      1.00      1.00       197

    accuracy                           1.00     78169
   macro avg       1.00      1.00      1.00     78169
weighted avg       1.00      1.00      1.00     78169

              precision    recall  f1-score   support

           0       0.99      0.99      0.99     10797
           1       0.99      0.99      0.99      8084
           2       0.96      0.96      0.96       612
           3       0.00      0.00      0.00         1
           4       0.24      0.27      0.25        49

    accuracy                           0.99     19543
   macro avg       0.64      0.64      0.64     19543
weighted avg       0.99      0.99      0.99     19543


/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
metrics/_classification.py:1344: UndefinedMetricWarning: Precision and
F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
metrics/_classification.py:1344: UndefinedMetricWarning: Precision and
F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
metrics/_classification.py:1344: UndefinedMetricWarning: Precision and
F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```


engineSize <= 0.453
gini = 0.523
samples = 78169
value = [43185, 32335, 2447, 5, 197]
class = 0

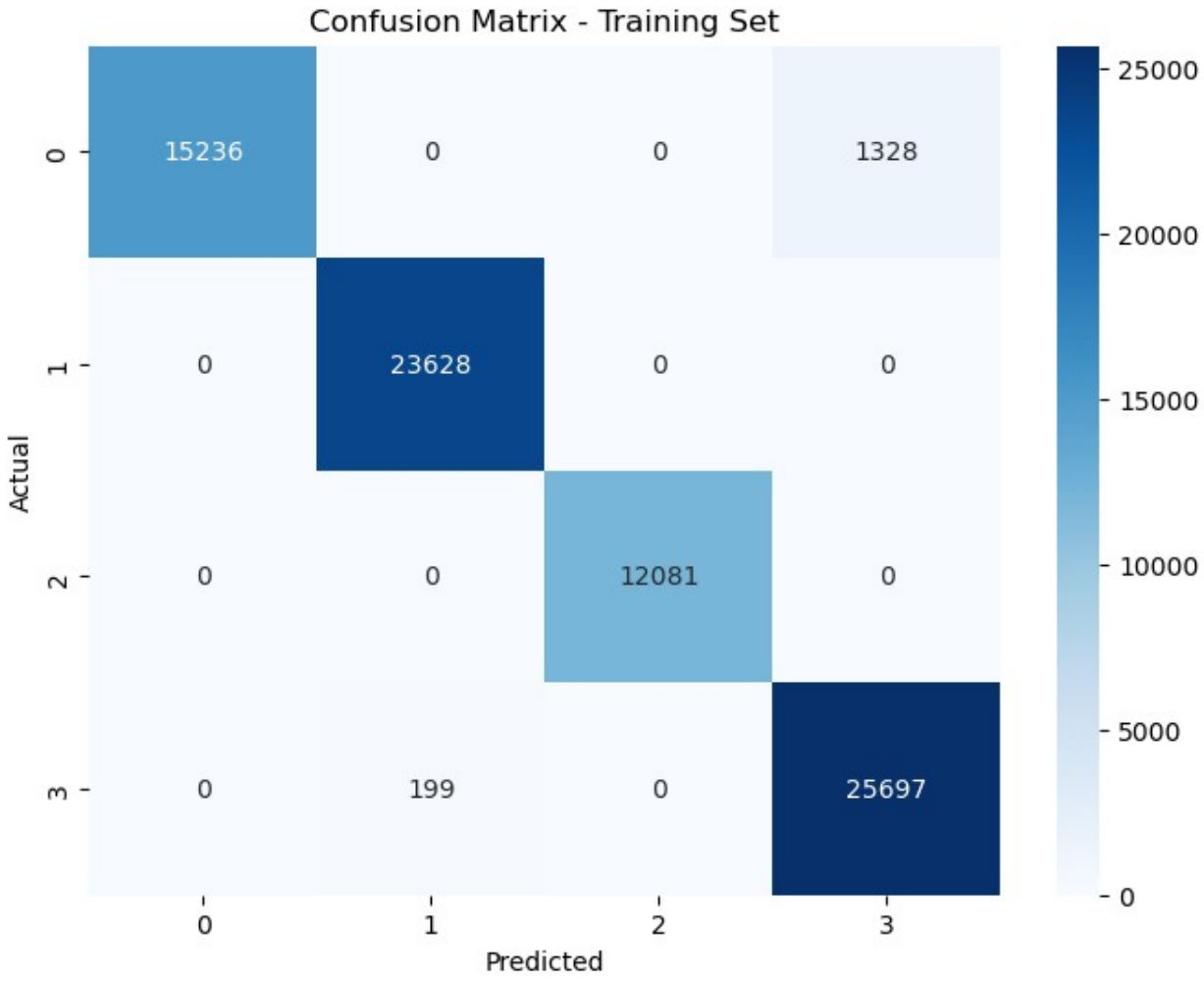

Decision Tree : Confusion Matrix
True Label
Predicted Label
0
1
2
3
4
1.1e+04
72
5
0
19
61
8e+03
6
0
7
2
5
5.9e+02
0
15
0
0
1
0
0
16
7
13
0
13
10000
8000
6000
4000
2000
0

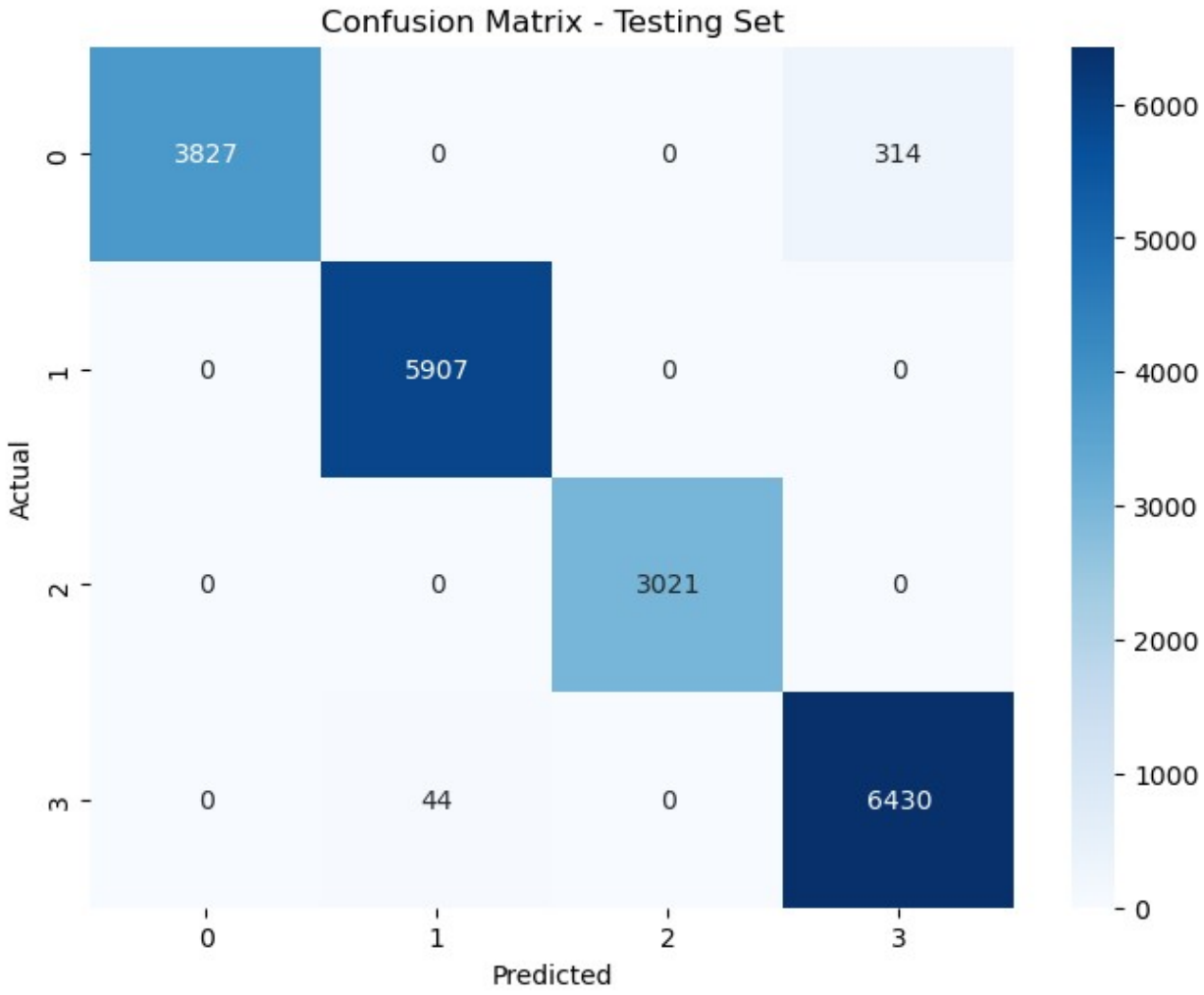
```
# Create a DataFrame to store feature importances
dtc_imp_features = pd.DataFrame({'feature': X.columns, 'importance':
np.round(dt_model_gini.feature_importances_, 3)})

# Sort the DataFrame by importance in descending order
dtc_imp_features.sort_values('importance', ascending=False,
inplace=True)

# Print the DataFrame
print(dtc_imp_features)
```

```
            feature  importance
0         model_enc         1.0
1  transmission_enc         0.0
2      fuelType_enc         0.0
3  Manufacturer_enc         0.0
4              year         0.0
5             price         0.0
6           mileage         0.0
7               tax         0.0
8               mpg         0.0
9        engineSize         0.0
```

```
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Initialize the SVM classifier
svm_classifier = SVC()

# Train the classifier
svm_classifier.fit(X_train, y_train)

# Make predictions on the training and testing sets
train_predictions = svm_classifier.predict(X_train)
test_predictions = svm_classifier.predict(X_test)

# Generate confusion matrices
train_conf_matrix = confusion_matrix(y_train, train_predictions)
test_conf_matrix = confusion_matrix(y_test, test_predictions)

# Plot confusion matrix for training set
plt.figure(figsize=(8, 6))
sns.heatmap(train_conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Training Set')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

```
# Plot confusion matrix for testing set
plt.figure(figsize=(8, 6))
sns.heatmap(test_conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Testing Set')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Generate classification report
train_classification_report = classification_report(y_train,
train_predictions)
test_classification_report = classification_report(y_test,
test_predictions)

# Print classification report for training set
print("Classification Report - Training Set:")
print(train_classification_report)

# Print classification report for testing set
print("\nClassification Report - Testing Set:")
print(test_classification_report)
```


Confusion Matrix - Training Set
15236
0
0
1328
0
23628
0
0
0
0
12081
0
0
199
0
25697
Actual
Predicted
0
1
2
3
25000
20000
15000
10000
5000
0




```
Classification Report - Training Set:
              precision    recall  f1-score   support

           0       1.00      0.92      0.96     16564
           1       0.99      1.00      1.00     23628
           2       1.00      1.00      1.00     12081
           3       0.95      0.99      0.97     25896

    accuracy                           0.98     78169
   macro avg       0.99      0.98      0.98     78169
weighted avg       0.98      0.98      0.98     78169


Classification Report - Testing Set:
              precision    recall  f1-score   support

           0       1.00      0.92      0.96      4141
           1       0.99      1.00      1.00      5907
           2       1.00      1.00      1.00      3021
```

```
           3       0.95      0.99      0.97      6474

    accuracy                           0.98     19543
   macro avg       0.99      0.98      0.98     19543
weighted avg       0.98      0.98      0.98     19543
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_curve, auc

import time
import psutil

# Record start time
start_time = time.time()

# Initialize and train Logistic Regression model
log_reg = LogisticRegression(random_state=42)
log_reg.fit(X_train, y_train)

# Record end time
end_time = time.time()

# Calculate memory usage
memory_usage = psutil.Process().memory_info().rss / 1024 / 1024  # in
MB

# Print memory and time statistics
print("Memory Usage (MB):", memory_usage)
print("Execution Time (seconds):", end_time - start_time)
```

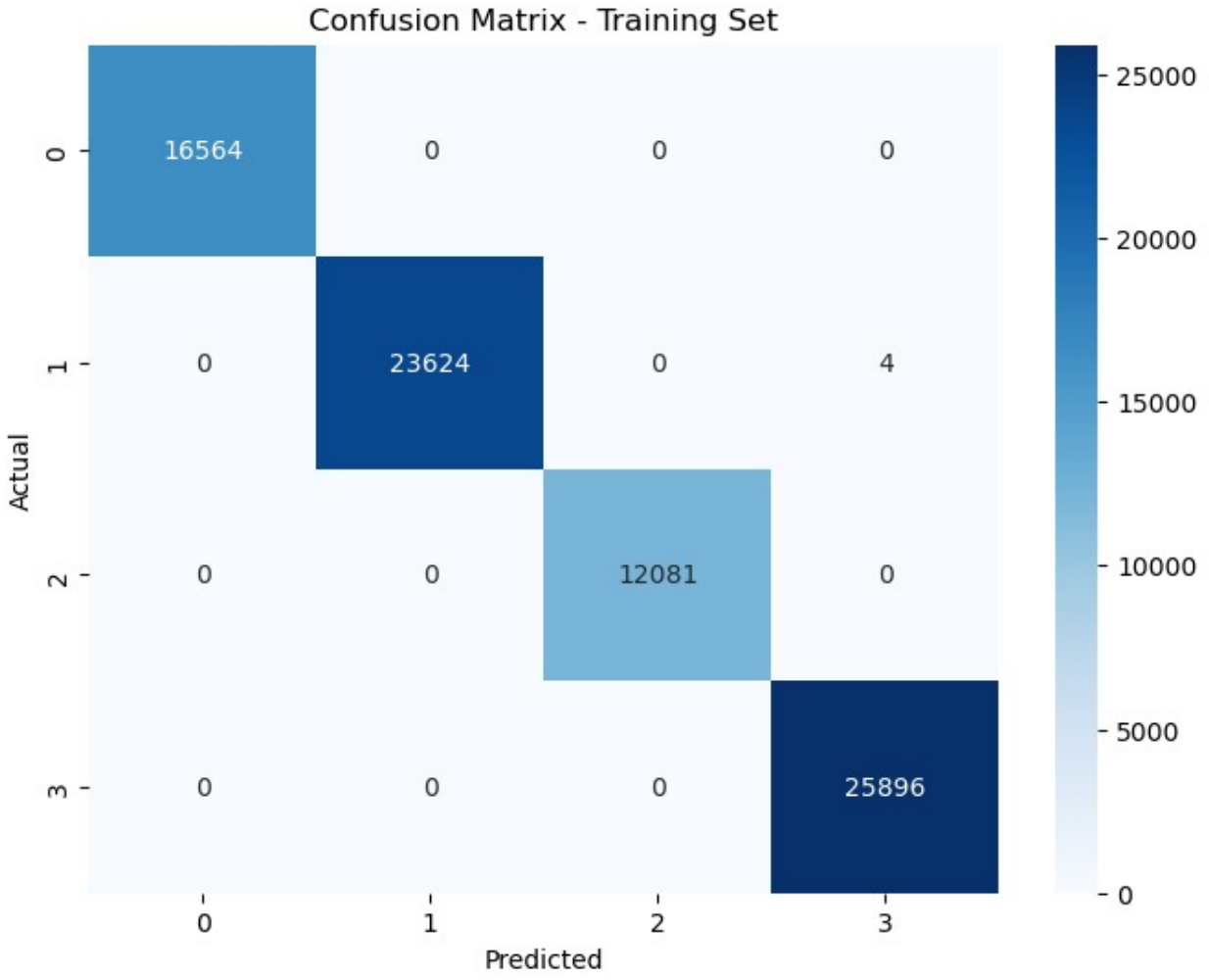
```
Memory Usage (MB): 128.609375
Execution Time (seconds): 1.310236930847168

/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to
converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

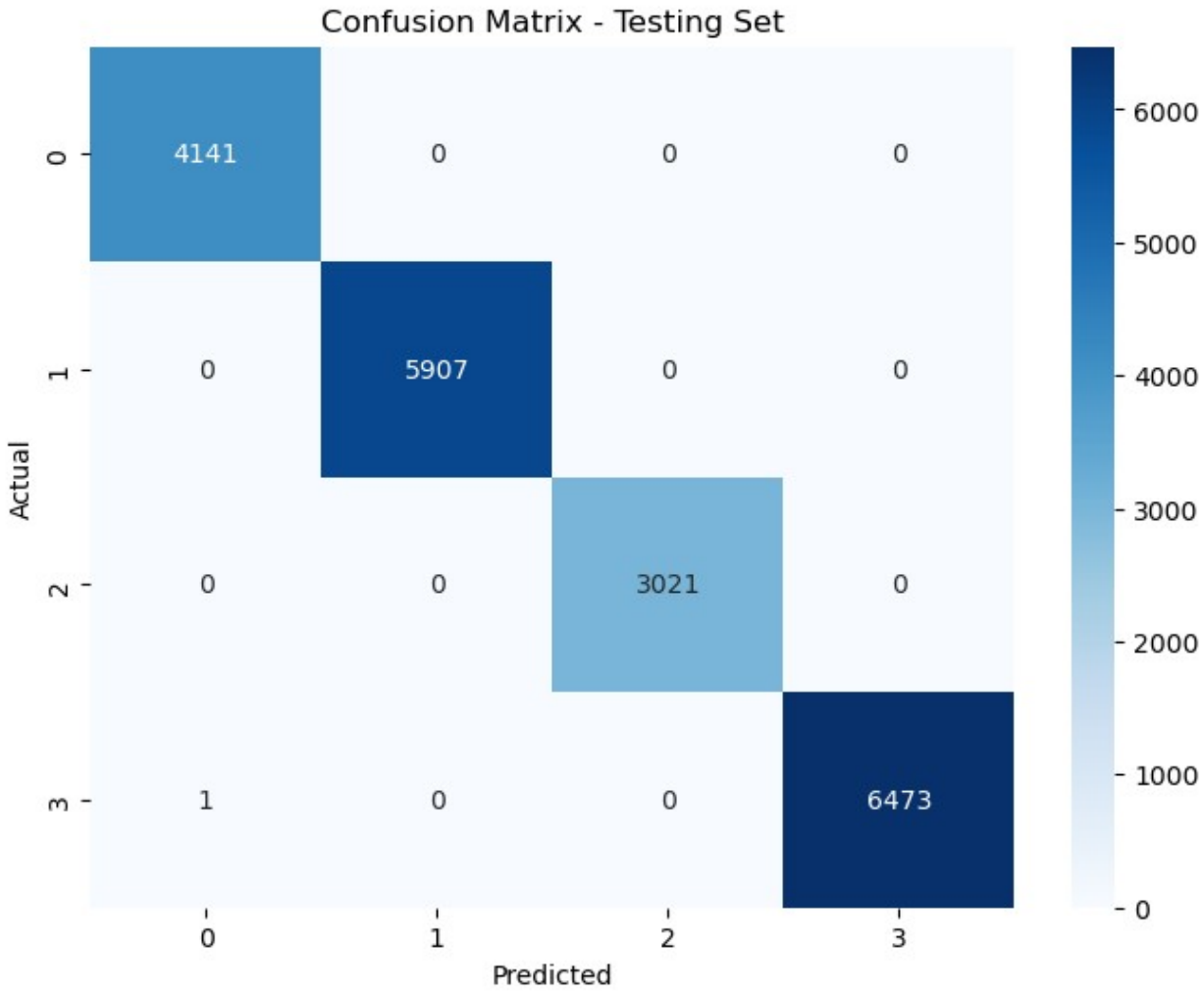
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
```

```
import matplotlib.pyplot as plt

# Initialize the Logistic Regression classifier
logistic_classifier = LogisticRegression()

# Train the classifier
logistic_classifier.fit(X_train, y_train)

# Make predictions on the training and testing sets
train_predictions = logistic_classifier.predict(X_train)
test_predictions = logistic_classifier.predict(X_test)

# Generate confusion matrices
train_conf_matrix = confusion_matrix(y_train, train_predictions)
test_conf_matrix = confusion_matrix(y_test, test_predictions)

# Plot confusion matrix for training set
plt.figure(figsize=(8, 6))
sns.heatmap(train_conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Training Set')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Plot confusion matrix for testing set
plt.figure(figsize=(8, 6))
sns.heatmap(test_conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Testing Set')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Generate classification report
train_classification_report = classification_report(y_train,
train_predictions)
test_classification_report = classification_report(y_test,
test_predictions)

# Print classification report for training set
print("Classification Report - Training Set:")
print(train_classification_report)

# Print classification report for testing set
print("\nClassification Report - Testing Set:")
print(test_classification_report)
```

```
/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to
converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```



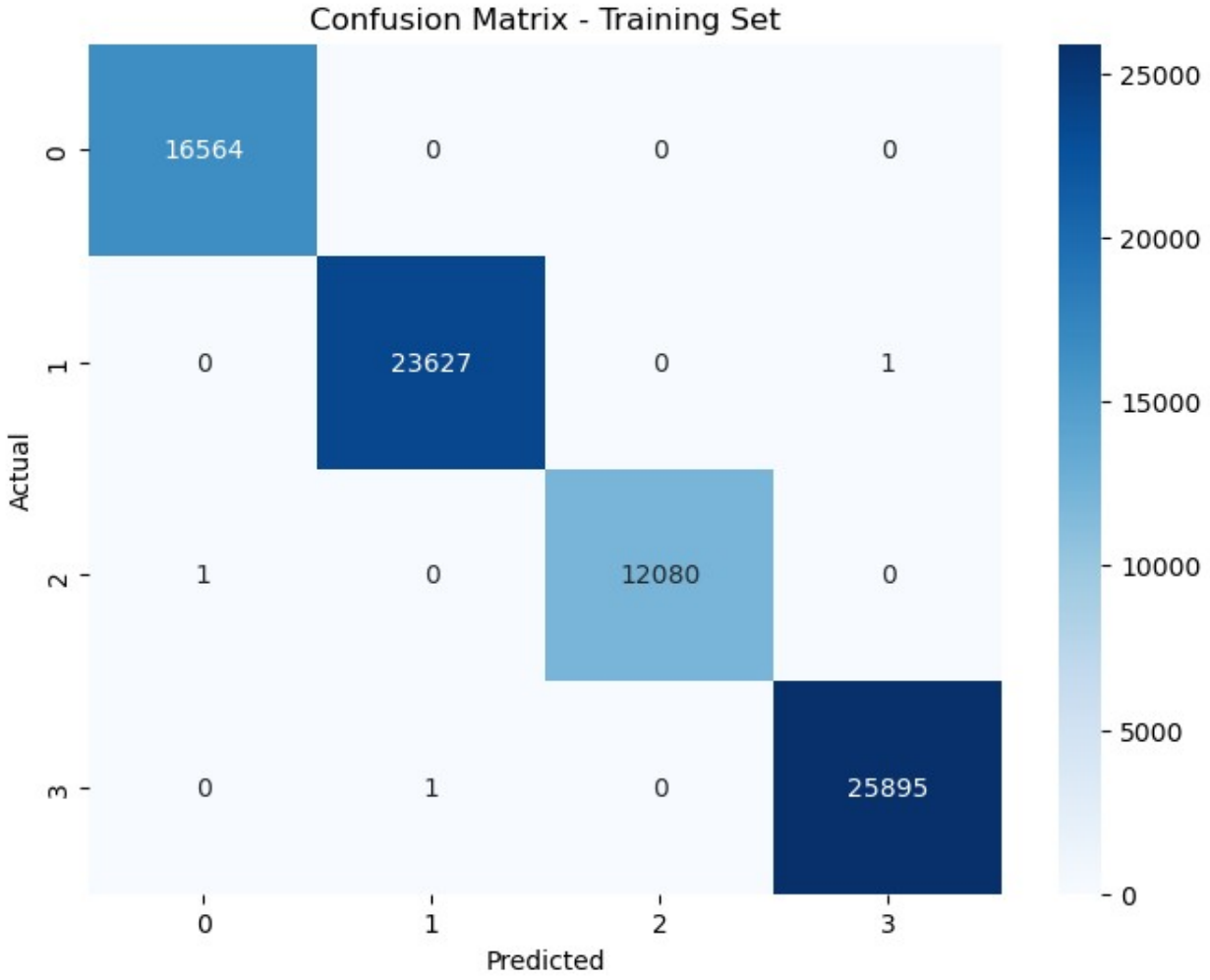


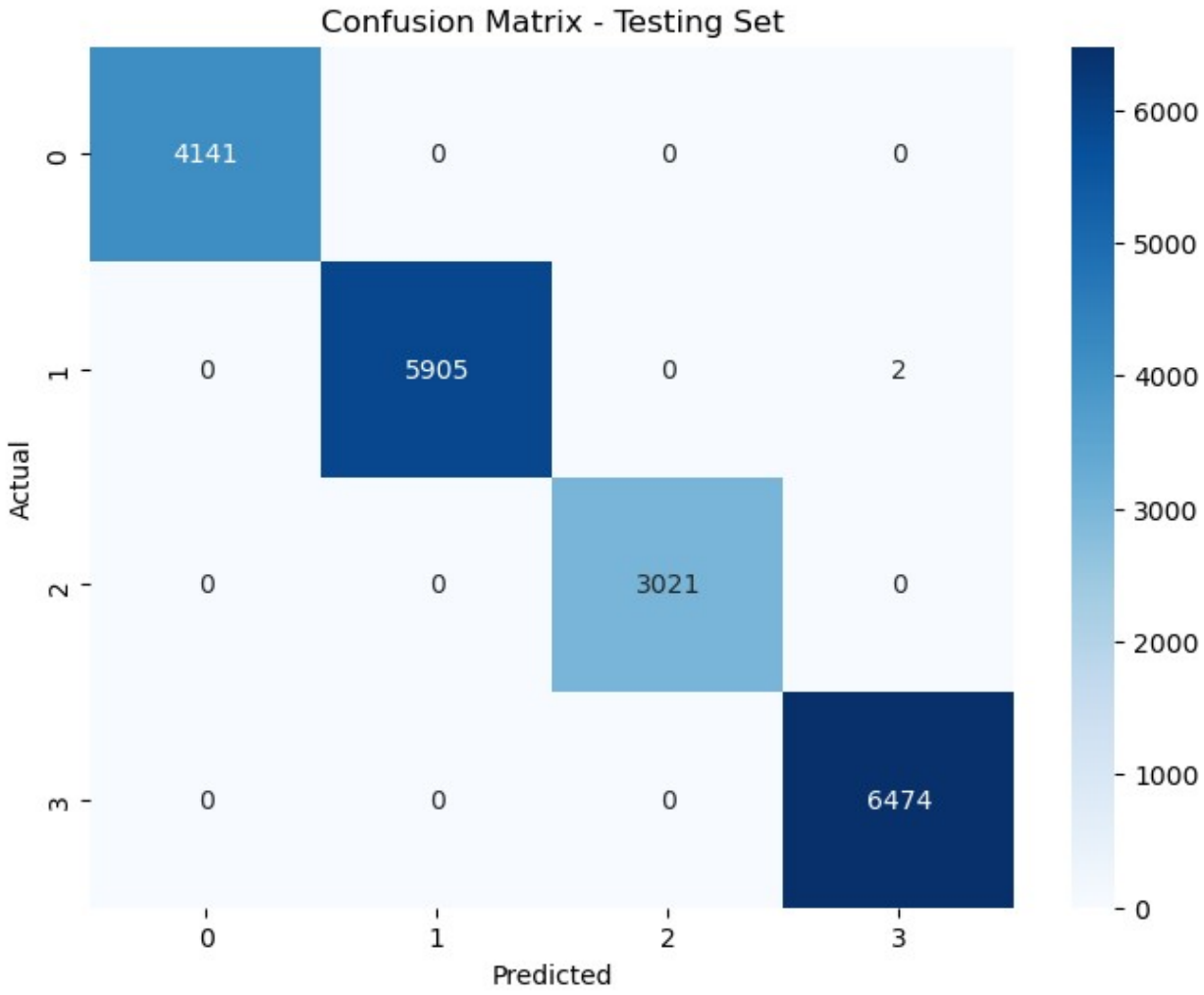

```
Classification Report - Training Set:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     16564
           1       1.00      1.00      1.00     23628
           2       1.00      1.00      1.00     12081
           3       1.00      1.00      1.00     25896

    accuracy                           1.00     78169
   macro avg       1.00      1.00      1.00     78169
weighted avg       1.00      1.00      1.00     78169


Classification Report - Testing Set:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4141
           1       1.00      1.00      1.00      5907
           2       1.00      1.00      1.00      3021
```

```
           3       1.00      1.00      1.00      6474

    accuracy                           1.00     19543
   macro avg       1.00      1.00      1.00     19543
weighted avg       1.00      1.00      1.00     19543
```

```python
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Initialize the KNN classifier
knn_classifier = KNeighborsClassifier()

# Train the classifier
knn_classifier.fit(X_train, y_train)

# Make predictions on the training and testing sets
train_predictions = knn_classifier.predict(X_train)
test_predictions = knn_classifier.predict(X_test)

# Generate confusion matrices
train_conf_matrix = confusion_matrix(y_train, train_predictions)
test_conf_matrix = confusion_matrix(y_test, test_predictions)

# Plot confusion matrix for training set
plt.figure(figsize=(8, 6))
sns.heatmap(train_conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Training Set')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Plot confusion matrix for testing set
plt.figure(figsize=(8, 6))
sns.heatmap(test_conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix - Testing Set')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Generate classification report
train_classification_report = classification_report(y_train,
train_predictions)
test_classification_report = classification_report(y_test,
test_predictions)

# Print classification report for training set
print("Classification Report - Training Set:")
```

```
print(train_classification_report)

# Print classification report for testing set
print("\nClassification Report - Testing Set:")
print(test_classification_report)
```






```
Classification Report - Training Set:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     16564
           1       1.00      1.00      1.00     23628
           2       1.00      1.00      1.00     12081
           3       1.00      1.00      1.00     25896

    accuracy                           1.00     78169
   macro avg       1.00      1.00      1.00     78169
weighted avg       1.00      1.00      1.00     78169


Classification Report - Testing Set:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4141
           1       1.00      1.00      1.00      5907
           2       1.00      1.00      1.00      3021
```

```
           3       1.00      1.00      1.00      6474

    accuracy                           1.00     19543
   macro avg       1.00      1.00      1.00     19543
weighted avg       1.00      1.00      1.00     19543
```

```
## 3.2.2.1.2. PO2 | PS2:: Classification Model Performance Evaluation: Time Statistics | (CPU | GPU) Memory Statistics (Base Model: Decision Tree)

from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
import time
from memory_profiler import memory_usage

# Function to train and evaluate a model
def train_and_evaluate_model(classifier, X_train, y_train, X_test, y_test):
    # Training time
    start_time = time.time()
    classifier.fit(X_train, y_train)
    end_time = time.time()
    training_time = end_time - start_time

    # Memory usage during training
    memory_usage_training = max(memory_usage((classifier.fit, (X_train, y_train))))

    # Testing time
    start_time = time.time()
    classifier.predict(X_test)
    end_time = time.time()
    testing_time = end_time - start_time

    # Memory usage during testing
    memory_usage_testing = max(memory_usage((classifier.predict, (X_test,))))

    return training_time, testing_time, memory_usage_training, memory_usage_testing

# Initialize classifiers
decision_tree_classifier = DecisionTreeClassifier()
svm_classifier = SVC()
knn_classifier = KNeighborsClassifier()

# Train and evaluate decision tree classifier
decision_tree_results =
```

```
train_and_evaluate_model(decision_tree_classifier, X_train, y_train,
X_test, y_test)

# Train and evaluate SVM classifier
svm_results = train_and_evaluate_model(svm_classifier, X_train,
y_train, X_test, y_test)

# Train and evaluate KNN classifier
knn_results = train_and_evaluate_model(knn_classifier, X_train,
y_train, X_test, y_test)

# Display results
print("Decision Tree Classifier:")
print("Training Time:", decision_tree_results[0], "seconds")
print("Testing Time:", decision_tree_results[1], "seconds")
print("Memory Usage (Training):", decision_tree_results[2], "MiB")
print("Memory Usage (Testing):", decision_tree_results[3], "MiB")
print()

print("SVM Classifier:")
print("Training Time:", svm_results[0], "seconds")
print("Testing Time:", svm_results[1], "seconds")
print("Memory Usage (Training):", svm_results[2], "MiB")
print("Memory Usage (Testing):", svm_results[3], "MiB")
print()

print("KNN Classifier:")
print("Training Time:", knn_results[0], "seconds")
print("Testing Time:", knn_results[1], "seconds")
print("Memory Usage (Training):", knn_results[2], "MiB")
print("Memory Usage (Testing):", knn_results[3], "MiB")
```

```
Decision Tree Classifier:
Training Time: 0.08125734329223633 seconds
Testing Time: 0.003000020980834961 seconds
Memory Usage (Training): 94.515625 MiB
Memory Usage (Testing): 96.265625 MiB

SVM Classifier:
Training Time: 24.109742164611816 seconds
Testing Time: 18.749216079711914 seconds
Memory Usage (Training): 531.8125 MiB
Memory Usage (Testing): 85.8125 MiB

KNN Classifier:
Training Time: 0.03143191337585449 seconds
Testing Time: 0.2598700523376465 seconds
Memory Usage (Training): 81.8125 MiB
Memory Usage (Testing): 88.453125 MiB
```

```
##3.2.2.1.1. PO2 | PS2:: Classification Model Performance Evaluation: Confusion Matrix {Accuray, Recall, Precision, F1-Score} (Base Model: Decision Tree)

from sklearn.metrics import classification_report, confusion_matrix

# Function to evaluate model performance
def evaluate_model_performance(classifier, X_train, y_train, X_test, y_test):
    # Train the classifier
    classifier.fit(X_train, y_train)

    # Make predictions
    y_pred = classifier.predict(X_test)

    # Generate confusion matrix
    conf_matrix = confusion_matrix(y_test, y_pred)

    # Generate classification report
    report = classification_report(y_test, y_pred)

    return conf_matrix, report

# Base Model: Decision Tree
dt_conf_matrix, dt_report = evaluate_model_performance(decision_tree_classifier, X_train, y_train, X_test, y_test)
print("Decision Tree Confusion Matrix:")
print(dt_conf_matrix)
print("\nDecision Tree Classification Report:")
print(dt_report)

# Comparison Model: Support Vector Machine
svm_conf_matrix, svm_report = evaluate_model_performance(svm_classifier, X_train, y_train, X_test, y_test)
print("\nSupport Vector Machine Confusion Matrix:")
print(svm_conf_matrix)
print("\nSupport Vector Machine Classification Report:")
print(svm_report)

# Comparison Model: K Nearest Neighbors
knn_conf_matrix, knn_report = evaluate_model_performance(knn_classifier, X_train, y_train, X_test, y_test)
print("\nK Nearest Neighbors Confusion Matrix:")
print(knn_conf_matrix)
print("\nK Nearest Neighbors Classification Report:")
print(knn_report)
```

```
Decision Tree Confusion Matrix:
[[4141    0    0    0]
 [   0 5907    0    0]
 [   0    0 3021    0]
 [   0    0    0 6474]]

Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4141
           1       1.00      1.00      1.00      5907
           2       1.00      1.00      1.00      3021
           3       1.00      1.00      1.00      6474

    accuracy                           1.00     19543
   macro avg       1.00      1.00      1.00     19543
weighted avg       1.00      1.00      1.00     19543


Support Vector Machine Confusion Matrix:
[[3827    0    0  314]
 [   0 5907    0    0]
 [   0    0 3021    0]
 [   0   44    0 6430]]

Support Vector Machine Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.92      0.96      4141
           1       0.99      1.00      1.00      5907
           2       1.00      1.00      1.00      3021
           3       0.95      0.99      0.97      6474

    accuracy                           0.98     19543
   macro avg       0.99      0.98      0.98     19543
weighted avg       0.98      0.98      0.98     19543


K Nearest Neighbors Confusion Matrix:
[[4141    0    0    0]
 [   0 5905    0    2]
 [   0    0 3021    0]
 [   0    0    0 6474]]

K Nearest Neighbors Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4141
           1       1.00      1.00      1.00      5907
           2       1.00      1.00      1.00      3021
           3       1.00      1.00      1.00      6474
```

```
    accuracy                           1.00     19543
   macro avg       1.00      1.00      1.00     19543
weighted avg       1.00      1.00      1.00     19543
```

```
#3.2.2.2.1. PO2 | PS2:: Classification Model Performance Evaluation: Confusion Matrix {Accuray, Recall, Precision, F1-Score} (Comparison Models: Logistic Regression | Support Vector Machine | K Nearest Neighbour)

from sklearn.metrics import classification_report, confusion_matrix

# Function to evaluate model performance
def evaluate_model_performance(classifier, X_train, y_train, X_test, y_test):
    # Train the classifier
    classifier.fit(X_train, y_train)

    # Make predictions
    y_pred = classifier.predict(X_test)

    # Generate confusion matrix
    conf_matrix = confusion_matrix(y_test, y_pred)

    # Generate classification report
    report = classification_report(y_test, y_pred)

    return conf_matrix, report

# Comparison Model: Logistic Regression
logistic_conf_matrix, logistic_report = evaluate_model_performance(logistic_classifier, X_train, y_train, X_test, y_test)
print("Logistic Regression Confusion Matrix:")
print(logistic_conf_matrix)
print("\nLogistic Regression Classification Report:")
print(logistic_report)

# Comparison Model: Support Vector Machine
svm_conf_matrix, svm_report = evaluate_model_performance(svm_classifier, X_train, y_train, X_test, y_test)
print("\nSupport Vector Machine Confusion Matrix:")
print(svm_conf_matrix)
print("\nSupport Vector Machine Classification Report:")
print(svm_report)

# Comparison Model: K Nearest Neighbors
knn_conf_matrix, knn_report =
```

```
evaluate_model_performance(knn_classifier, X_train, y_train, X_test,
y_test)
print("\nK Nearest Neighbors Confusion Matrix:")
print(knn_conf_matrix)
print("\nK Nearest Neighbors Classification Report:")
print(knn_report)
```

```
/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to
converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(

Logistic Regression Confusion Matrix:
[[4141    0    0    0]
 [   0 5907    0    0]
 [   0    0 3021    0]
 [   1    0    0 6473]]

Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4141
           1       1.00      1.00      1.00      5907
           2       1.00      1.00      1.00      3021
           3       1.00      1.00      1.00      6474

    accuracy                           1.00     19543
   macro avg       1.00      1.00      1.00     19543
weighted avg       1.00      1.00      1.00     19543


Support Vector Machine Confusion Matrix:
[[3827    0    0  314]
 [   0 5907    0    0]
 [   0    0 3021    0]
 [   0   44    0 6430]]

Support Vector Machine Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.92      0.96      4141
           1       0.99      1.00      1.00      5907
```

```
           2       1.00      1.00      1.00      3021
           3       0.95      0.99      0.97      6474

    accuracy                           0.98     19543
   macro avg       0.99      0.98      0.98     19543
weighted avg       0.98      0.98      0.98     19543


K Nearest Neighbors Confusion Matrix:
[[4141    0    0    0]
 [   0 5905    0    2]
 [   0    0 3021    0]
 [   0    0    0 6474]]

K Nearest Neighbors Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      4141
           1       1.00      1.00      1.00      5907
           2       1.00      1.00      1.00      3021
           3       1.00      1.00      1.00      6474

    accuracy                           1.00     19543
   macro avg       1.00      1.00      1.00     19543
weighted avg       1.00      1.00      1.00     19543
```

```
#3.2.2.2.2. PO2 | PS2:: Classification Model Performance Evaluation: Time Statistics | (CPU | GPU) Memory Statistics (Comparison Models: Logistic Regression | Support Vector Machine | K Nearest Neighbour)


import time
from memory_profiler import memory_usage

# Function to train and evaluate a model
def train_and_evaluate_model(classifier, X_train, y_train, X_test, y_test):
    # Training time
    start_time = time.time()
    classifier.fit(X_train, y_train)
    end_time = time.time()
    training_time = end_time - start_time

    # Memory usage during training
    memory_usage_training = max(memory_usage((classifier.fit, (X_train, y_train))))

    # Testing time
    start_time = time.time()
    classifier.predict(X_test)
```

```
    end_time = time.time()
    testing_time = end_time - start_time

    # Memory usage during testing
    memory_usage_testing = max(memory_usage((classifier.predict,
(X_test,))))

    return training_time, testing_time, memory_usage_training,
memory_usage_testing

# Initialize classifiers
logistic_classifier = LogisticRegression()
svm_classifier = SVC()
knn_classifier = KNeighborsClassifier()

# Train and evaluate Logistic Regression classifier
logistic_results = train_and_evaluate_model(logistic_classifier,
X_train, y_train, X_test, y_test)
print("Logistic Regression:")
print("Training Time:", logistic_results[0], "seconds")
print("Testing Time:", logistic_results[1], "seconds")
print("Memory Usage (Training):", logistic_results[2], "MiB")
print("Memory Usage (Testing):", logistic_results[3], "MiB")
print()

# Train and evaluate SVM classifier
svm_results = train_and_evaluate_model(svm_classifier, X_train,
y_train, X_test, y_test)
print("Support Vector Machine:")
print("Training Time:", svm_results[0], "seconds")
print("Testing Time:", svm_results[1], "seconds")
print("Memory Usage (Training):", svm_results[2], "MiB")
print("Memory Usage (Testing):", svm_results[3], "MiB")
print()

# Train and evaluate KNN classifier
knn_results = train_and_evaluate_model(knn_classifier, X_train,
y_train, X_test, y_test)
print("K Nearest Neighbors:")
print("Training Time:", knn_results[0], "seconds")
print("Testing Time:", knn_results[1], "seconds")
print("Memory Usage (Training):", knn_results[2], "MiB")
print("Memory Usage (Testing):", knn_results[3], "MiB")
```

```
/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to
converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
```

```
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to
converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(

Logistic Regression:
Training Time: 1.2410788536071777 seconds
Testing Time: 0.006331920623779297 seconds
Memory Usage (Training): 149.59375 MiB
Memory Usage (Testing): 142.875 MiB

Support Vector Machine:
Training Time: 23.700400829315186 seconds
Testing Time: 18.813068866729736 seconds
Memory Usage (Training): 499.703125 MiB
Memory Usage (Testing): 53.28125 MiB

K Nearest Neighbors:
Training Time: 0.031110286712646484 seconds
Testing Time: 0.25697898864746094 seconds
Memory Usage (Training): 96.140625 MiB
Memory Usage (Testing): 102.5625 MiB
```

```python
# 3.2.3.1. PO3 | PS3:: Variable or Feature Analysis: Base Model
(Decision Tree)
#3.2.3.1.1. List of Relevant or Important Variables or Features and
their Thresholds

from sklearn.tree import DecisionTreeClassifier

# Initialize the Decision Tree classifier
decision_tree_classifier = DecisionTreeClassifier()

# Train the classifier
```

```
decision_tree_classifier.fit(X_train, y_train)

# Get feature importances
feature_importances = decision_tree_classifier.feature_importances_

# Get the list of feature names
feature_names = X_train.columns

# Create a dictionary to store feature importance with their names
feature_importance_dict = dict(zip(feature_names,
feature_importances))

# Sort the dictionary by feature importance
sorted_feature_importance = sorted(feature_importance_dict.items(),
key=lambda x: x[1], reverse=True)

# Print the list of relevant variables or features and their
thresholds
print("List of Relevant Variables or Features and their Thresholds:")
for feature, importance in sorted_feature_importance:
    print(f"Feature: {feature}, Importance: {importance}")
```

```
List of Relevant Variables or Features and their Thresholds:
Feature: model_enc, Importance: 1.0
Feature: transmission_enc, Importance: 0.0
Feature: fuelType_enc, Importance: 0.0
Feature: Manufacturer_enc, Importance: 0.0
Feature: year, Importance: 0.0
Feature: price, Importance: 0.0
Feature: mileage, Importance: 0.0
Feature: tax, Importance: 0.0
Feature: mpg, Importance: 0.0
Feature: engineSize, Importance: 0.0
```

```
# Assuming you already have a trained Decision Tree classifier
(decision_tree_classifier)

# Get feature importances
feature_importances = decision_tree_classifier.feature_importances_

# Get the list of feature names
feature_names = X_train.columns

# Create a dictionary to store feature importance with their names
feature_importance_dict = dict(zip(feature_names,
feature_importances))

# Sort the dictionary by feature importance in descending order
sorted_feature_importance = sorted(feature_importance_dict.items(),
key=lambda x: x[1], reverse=True)
```

```python
# Assuming a threshold for importance score to determine non-relevance
threshold = 0.01  # You can adjust this threshold based on your
preference

# Print the list of non-relevant variables or features based on the
threshold
print("List of Non-Relevant Variables or Features:")
for feature, importance in sorted_feature_importance:
    if importance < threshold:
        print(f"Feature: {feature}, Importance: {importance}")
```

```
List of Non-Relevant Variables or Features:
Feature: transmission_enc, Importance: 0.0
Feature: fuelType_enc, Importance: 0.0
Feature: Manufacturer_enc, Importance: 0.0
Feature: year, Importance: 0.0
Feature: price, Importance: 0.0
Feature: mileage, Importance: 0.0
Feature: tax, Importance: 0.0
Feature: mpg, Importance: 0.0
Feature: engineSize, Importance: 0.0
```

```python
#
# Get a list of non-relevant or non-important features for the
Decision Tree model
non_relevant_features_decision_tree = [feature for feature in
feature_names if feature not in feature_importance_dict]
print("List of Non-Relevant Variables or Features for Decision Tree:")
print(non_relevant_features_decision_tree)
```

```
List of Non-Relevant Variables or Features for Decision Tree:
[]
```

```python
from sklearn.linear_model import LogisticRegression

# Initialize the Logistic Regression classifier
logistic_classifier = LogisticRegression()

# Train the classifier
logistic_classifier.fit(X_train, y_train)

# Get coefficients
coefficients = logistic_classifier.coef_[0]

# Get feature names
feature_names = X_train.columns

# Create a dictionary to store feature coefficients with their names
feature_coefficient_dict = dict(zip(feature_names, coefficients))

# Sort the dictionary by absolute coefficient values
```

```
sorted_feature_coefficient = sorted(feature_coefficient_dict.items(),
key=lambda x: abs(x[1]), reverse=True)

# Print the list of relevant variables or features and their
coefficients
print("List of Relevant Variables or Features and their Coefficients
for Logistic Regression:")
for feature, coefficient in sorted_feature_coefficient:
    print(f"Feature: {feature}, Coefficient: {coefficient}")
```

```
List of Relevant Variables or Features and their Coefficients for
Logistic Regression:
Feature: model_enc, Coefficient: 2.0160207121130873
Feature: transmission_enc, Coefficient: -0.46444989129248176
Feature: tax, Coefficient: 0.3158408992084213
Feature: Manufacturer_enc, Coefficient: 0.304645875095408
Feature: fuelType_enc, Coefficient: -0.1006337827381201
Feature: price, Coefficient: -0.09487635464045102
Feature: mileage, Coefficient: -0.0865212892010832
Feature: mpg, Coefficient: -0.059954036719245
Feature: year, Coefficient: -0.05432771937864214
Feature: engineSize, Coefficient: -0.03430266169177579

/Users/sanskritibahl/anaconda3/lib/python3.11/site-packages/sklearn/
linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to
converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```