Principles of Data Science (5530)-Assignment 2

Name: Akhila Reddyrajula
ID: 16346391

Used Car Dataset Analysis

The provided dataset contains information about used cars, including attributes such as make and model, location, year of manufacture, mileage, fuel type, transmission type, and price. The objective of this analysis is to preprocess the data and perform various operations to gain insights into the dataset.

```
[67]  1  import pandas as pd
      2  from datetime import datetime
      3
      4  # Load the dataset
      5  data = pd.read_csv("/content/sample_data/train.csv")
```

```
[68]  1  # Check the column names
      2  print(data.columns)
```

```
Index(['Unnamed: 0', 'Name', 'Location', 'Year', 'Kilometers_Driven',
       'Fuel_Type', 'Transmission', 'Owner_Type', 'Mileage', 'Engine', 'Power',
       'Seats', 'New_Price', 'Price'],
      dtype='object')
```

Preprocessing Steps

a) Handling Missing Values

Missing values are identified in all columns, and a decision is made whether to impute or drop them. For columns where missing values constitute less than 5% of the total entries, they are imputed using mean, median, or mode, depending on the data type. Otherwise, the entire column is dropped.

```python
# a) Handling missing values
# Look for missing values in all columns
missing_values = data.isnull().sum()

# Decide whether to impute or drop missing values based on the proportion of missing values
# If missing values are less than 5% of the total entries in a column, impute with mean, median, or mode
# Otherwise, drop the column
for col in data.columns:
    if missing_values[col] < 0.05 * len(data):
        if data[col].dtype == 'object':
            mode_val = data[col].mode()[0]
            data[col].fillna(mode_val, inplace=True)
        else:
            mean_val = data[col].mean()
            data[col].fillna(mean_val, inplace=True)
    else:
        data.drop(columns=[col], inplace=True)
```

b) Removing Units from Attributes

Units are removed from certain attributes to convert them to numerical values. For instance, units such as 'kmpl' from 'Mileage', 'CC' from 'Engine', and 'bhp' from 'Power' are removed, leaving only the numerical values.

```python
# b) Removing units from attributes
data['Mileage'] = data['Mileage'].str.replace(' kmpl', '').str.replace(' km/kg', '')
data['Engine'] = data['Engine'].str.replace(' CC', '')
data['Power'] = data['Power'].str.replace(' bhp', '')
```

c) Changing Categorical Variables to One-Hot Encoded Values

Categorical variables like 'Fuel_Type' and 'Transmission' are converted into numerical one-hot encoded values using the pandas get_dummies() function. This process allows for the inclusion of categorical data in machine learning models.

```python
# c) Changing categorical variables into numerical one hot encoded value
data = pd.get_dummies(data, columns=['Fuel_Type', 'Transmission'])
```

d) Creating a New Feature - Current Age of the Car

A new feature named 'Current_Age' is created by subtracting the 'Year' of manufacture from the current year. This feature provides information about the age of the car at the time of analysis.

```python
# d) Creating a new feature - Current Age of the car
current_year = datetime.now().year
data['Current_Age'] = current_year - data['Year']
```

e) Performing Select, Filter, Rename, Mutate, Arrange, and Summarize Operations

Several operations are performed on the dataset:

- **Select:** Specific columns are selected for analysis, such as 'Name', 'Location', 'Year', 'Mileage', etc.

- **Filter:** Cars with a price greater than 50 are filtered out for further analysis.

- **Rename:** Columns are renamed to provide more descriptive names, such as renaming 'Year' to 'Manufacture_Year' and 'Price' to 'Price_in_Lakhs'.

- **Mutate:** A new column named 'Engine_Power' is added by combining the values of 'Engine' and 'Power'.

- **Arrange:** Data is sorted based on 'Current_Age' in descending order to understand the distribution of car ages.

- **Summarize:** Summary statistics are calculated, such as the mean price for each fuel type ('Fuel_Type_Petrol').

```
30
31  # e) Performing select, filter, rename, mutate, arrange, and summarize operations
32  # Select operation - selecting specific columns
33  selected_data = data[['Name', 'Location', 'Year', 'Mileage', 'Fuel_Type_Diesel', 'Fuel_Type_Petrol', 'Transmission_Automatic', 'Transmission_Manual', 'Price']]
34
35  # Filter operation - filtering cars with price greater than 50
36  filtered_data = data[data['Price'] > 50]
37
38  # Rename operation - renaming columns
39  renamed_data = data.rename(columns={'Year': 'Manufacture_Year', 'Price': 'Price_in_Lakhs'})
40
41  # Mutate operation - adding a new column for the value of 'Engine + Power'
42  data['Engine_Power'] = data['Engine'] + data['Power']
43
44  # Arrange operation - sorting data based on 'Current_Age' in descending order
45  arranged_data = data.sort_values(by='Current_Age', ascending=False)
46
47  # Summarize with group by operation - calculating mean price for each fuel type
48  summary_data = data.groupby('Fuel_Type_Petrol')['Price'].mean().reset_index()
49
50  # Displaying the summary statistics
51  print("Summary Statistics:")
52  print(summary_data)
53
54  # Displaying the modified dataset
55  print("Modified Dataset:")
56  print(data.head(15))
```

This analysis provides valuable insights into the used car dataset, including preprocessing steps to handle missing values, converting categorical variables, and creating new features.

```python
# a) Handling missing values
# Look for missing values in all columns
missing_values = data.isnull().sum()

# Decide whether to impute or drop missing values based on the proportion of missing values
# If missing values are less than 5% of the total entries in a column, impute with mean, median, or mode
# Otherwise, drop the column
for col in data.columns:
    if missing_values[col] < 0.05 * len(data):
        if data[col].dtype == 'object':
            mode_val = data[col].mode()[0]
            data[col].fillna(mode_val, inplace=True)
        else:
            mean_val = data[col].mean()
            data[col].fillna(mean_val, inplace=True)
    else:
        data.drop(columns=[col], inplace=True)

# b) Removing units from attributes
data['Mileage'] = data['Mileage'].str.replace(' kmpl', '').str.replace(' km/kg', '')
data['Engine'] = data['Engine'].str.replace(' CC', '')
data['Power'] = data['Power'].str.replace(' bhp', '')

# c) Changing categorical variables into numerical one hot encoded value
data = pd.get_dummies(data, columns=['Fuel_Type', 'Transmission'])

# d) Creating a new feature - Current Age of the car
current_year = datetime.now().year
data['Current_Age'] = current_year - data['Year']

# e) Performing select, filter, rename, mutate, arrange, and summarize operations
# Select operation - selecting specific columns
selected_data = data[['Name', 'Location', 'Year', 'Mileage', 'Fuel_Type_Diesel', 'Fuel_Type_Petrol', 'Transmission_Automatic', 'Transmission_Manual', 'Price']]

# Filter operation - filtering cars with price greater than 50
filtered_data = data[data['Price'] > 50]

# Rename operation - renaming columns
renamed_data = data.rename(columns={'Year': 'Manufacture_Year', 'Price': 'Price_in_Lakhs'})

# Mutate operation - adding a new column for the value of 'Engine + Power'
data['Engine_Power'] = data['Engine'] + data['Power']

# Arrange operation - sorting data based on 'Current_Age' in descending order
arranged_data = data.sort_values(by='Current_Age', ascending=False)

# Summarize with group by operation - calculating mean price for each fuel type
summary_data = data.groupby('Fuel_Type_Petrol')['Price'].mean().reset_index()

# Displaying the summary statistics
print("Summary Statistics:")
print(summary_data)

# Displaying the modified dataset
print("Modified Dataset:")
print(data.head(15))
```

Output:

```
print("Modified Dataset:")
print(data.head(10))
```

```
Summary Statistics:
   Fuel_Type_Petrol      Price
0                 0  12.960632
1                 1   5.756688
Modified Dataset:
   Unnamed: 0                              Name    Location  Year  \
0           1     Hyundai Creta 1.6 CRDi SX Option        Pune  2015
1           2                         Honda Jazz V     Chennai  2011
2           3                    Maruti Ertiga VDI     Chennai  2012
3           4      Audi A4 New 2.0 TDI Multitronic  Coimbatore  2013
4           6                Nissan Micra Diesel XV      Jaipur  2013
5           7    Toyota Innova Crysta 2.8 GX AT 8S      Mumbai  2016
6           8  Volkswagen Vento Diesel Comfortline        Pune  2013
7           9       Tata Indica Vista Quadrajet LS     Chennai  2012
8          10                    Maruti Ciaz Zeta       Kochi  2018
9          11          Honda City 1.5 V AT Sunroof     Kolkata  2012

   Kilometers_Driven Owner_Type Mileage Engine   Power  Seats  Price  \
0              41000      First   19.67   1582   126.2    5.0  12.50
1              46000      First      13   1199    88.7    5.0   4.50
2              87000      First   20.77   1248   88.76    7.0   6.00
3              40670     Second    15.2   1968   140.8    5.0  17.74
4              86999      First   23.08   1461    63.1    5.0   3.50
5              36000      First   11.36   2755   171.5    8.0  17.50
6              64430      First   20.54   1598   103.6    5.0   5.20
7              65932     Second    22.3   1248      74    5.0   1.95
8              25692      First   21.56   1462  103.25    5.0   9.95
9              60000      First    16.8   1497   116.3    5.0   4.49

   Fuel_Type_Diesel  Fuel_Type_Electric  Fuel_Type_Petrol  \
0                 1                   0                 0
1                 0                   0                 1
2                 1                   0                 0
3                 1                   0                 0
4                 1                   0                 0
5                 1                   0                 0
6                 1                   0                 0
7                 1                   0                 0
8                 0                   0                 1
9                 0                   0                 1

   Transmission_Automatic  Transmission_Manual  Current_Age Engine_Power
0                       0                    1           9    1582126.2
1                       0                    1          13     119988.7
2                       0                    1          12    124888.76
3                       1                    0          11    1968140.8
4                       0                    1          11     146163.1
5                       1                    0           8    2755171.5
6                       0                    1          11    1598103.6
7                       0                    1          12       124874
8                       0                    1           6   1462103.25
9                       1                    0          12    1497116.3
```