

PDS ASSIGNMENT-2

BOMMA VISHAL REDDY

16340457

TASK A: Look for the missing values in all the columns and either impute them (replace with mean, median, or mode) or drop them. Justify your action for this task.

Step 1: Reading The data from the train.csv which was given
<https://app.box.com/s/jm6pw202asu4xd3uypwtry2rqk691y1i>

Step 2:

- First we import pandas as pd
- Then we read the train.csv file using `pd.read_csv(train.csv)` into data variable
- Then we print the data variable, if we get the output then we successfully loaded and saved the train.csv into data.

Jupyter PDS_Assignment2_16340457 Last Checkpoint: 29 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

Task a) Look for the missing values in all the columns and either impute them (replace with mean, median, or mode) or drop them. Justify your action for this task.

```
[1]: import pandas as pd
```

```
[2]: # Reading the provided dataset
data = pd.read_csv('train.csv')
```

```
[3]: data.head(10)
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
1	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	13 km/kg	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
2	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
3	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74
4	6	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN	3.50
5	7	Toyota Innova Crysta 2.8 GX AT 8S	Mumbai	2016	36000	Diesel	Automatic	First	11.36 kmpl	2755 CC	171.5 bhp	8.0	21 Lakh	17.50
6	8	Volkswagen Vento Diesel Comfortline	Pune	2013	64430	Diesel	Manual	First	20.54 kmpl	1598 CC	103.6 bhp	5.0	NaN	5.20
7	9	Tata Indica Vista Quadrajet LS	Chennai	2012	65932	Diesel	Manual	Second	22.3 kmpl	1248 CC	74 bhp	5.0	NaN	1.95
8	10	Maruti Ciaz Zeta	Kochi	2018	25692	Petrol	Manual	First	21.56 kmpl	1462 CC	103.25 bhp	5.0	10.65 Lakh	9.95
9	11	Honda City 1.5 V AT Sunroof	Kolkata	2012	60000	Petrol	Automatic	First	16.8 kmpl	1497 CC	116.3 bhp	5.0	NaN	4.49

Step3:

- Now we are going to check for the missing values
- This code inspects missing values in the dataset and processes specific columns using regular expressions to extract numeric data ('Mileage', 'Engine', 'Power', 'New_Price', 'Seats') from string representations. It converts these values to floating-point numbers and fills missing data with the mean or median of the respective column.

JupyterPDS_Assignment2_16340457-Copy1Last Checkpoint: 22 minutes ago

FileEditViewRunKernelSettingsHelp

Trusted

JupyterLabPython 3 (ipykernel)

[10]:

Checking for missing values in the dataset
missing_values = data.isnull().sum()
print("Missing Values:")
print(missing_values)

Missing Values:
Unnamed: 00
Name0
Location0
Year0
Kilometers_Driven0
Fuel_Type0
Transmission0
Owner_Type0
Mileage2
Engine36
Power36
Seats38
New_Price5032
Price0
dtype: int64

[11]:

import re

data['Mileage'] = data['Mileage'].apply(lambda x: re.findall(r'\d+\.\d*', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Mileage'].fillna(data['Mileage'].mean(), inplace=True)

data['Engine'] = data['Engine'].apply(lambda x: re.findall(r'\d+', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Engine'].fillna(data['Engine'].median(), inplace=True)

data['Power'] = data['Power'].apply(lambda x: re.findall(r'\d+\.\d*', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Power'].fillna(data['Power'].median(), inplace=True)

data['Seats'] = data['Seats'].apply(lambda x: re.findall(r'\d+\.\d*', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Seats'].fillna(data['Seats'].mean(), inplace=True)

#dropping new_price column because it contains high number of null values
data.drop(['New_Price', 'Unnamed: 0'], axis=1, inplace=True)

[12]:

data

JupyterPDS_Assignment2_16340457-Copy1Last Checkpoint: 22 minutes ago

FileEditViewRunKernelSettingsHelp

Trusted

JupyterLabPython 3 (ipykernel)

[12]:

data

[12]:

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50
1	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	13.00	1199.0	88.70	5.0	4.50
2	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74
4	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08	1461.0	63.10	5.0	3.50
...
5842	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.40	1248.0	74.00	5.0	4.75
5843	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.40	1120.0	71.00	5.0	4.00
5844	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second	14.00	2498.0	112.00	8.0	2.90
5845	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First	18.90	998.0	67.10	5.0	2.65
5846	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First	25.44	936.0	57.60	5.0	2.50

5847 rows x 12 columns

Justification:

- Dropping the 'New_Price' column due to a high number of missing values is justified to maintain data quality. With over 5000 missing values, imputing them might introduce potential biases in the dataset, hence the decision to remove the column ensures data integrity and reliability in subsequent analysis.

Task B Remove the units from some of the attributes and only keep the numerical values (for example remove kmpl from “Mileage”, CC from “Engine”, bhp from “Power”, and lakh from “New_price”).

Jupyter PDS_Assignment2_16340457-Copy1 Last Checkpoint: 24 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

Task B b) Remove the units from some of the attributes and only keep the numerical values (for example remove kmpl from “Mileage”, CC from “Engine”, bhp from “Power”, and lakh from “New_price”).

```
[14]: import re

data['Mileage'] = data['Mileage'].apply(lambda x: re.findall(r'\d+\.\d*', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Engine'] = data['Engine'].apply(lambda x: re.findall(r'\d+', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Power'] = data['Power'].apply(lambda x: re.findall(r'\d+\.\d*', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Seats'] = data['Seats'].apply(lambda x: re.findall(r'\d+', str(x))[0] if pd.notnull(x) else x).astype(float)
```

[15]: data

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50
1	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	13.00	1199.0	88.70	5.0	4.50
2	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74
4	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08	1461.0	63.10	5.0	3.50
...
5842	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.40	1248.0	74.00	5.0	4.75
5843	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.40	1120.0	71.00	5.0	4.00
5844	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second	14.00	2498.0	112.00	8.0	2.90
5845	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First	18.90	998.0	67.10	5.0	2.65
5846	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First	25.44	936.0	57.60	5.0	2.50

5847 rows x 12 columns

Task C) Change the categorical variables (“Fuel_Type” and “Transmission”) into numerical one hot encoded value.

- The code `pd.get_dummies()` in Pandas is used to convert categorical variables into dummy/indicator variables. By applying this function to the 'Fuel_Type' and 'Transmission' columns and specifying prefixes for the new columns, it creates new columns for each category in those columns, marking the presence of each category with 1 (or 0 for absence).

Jupyter PDS_Assignment2_16340457-Copy1 Last Checkpoint: 26 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

Task C) Change the categorical variables (“Fuel_Type” and “Transmission”) into numerical one hot encoded value.

```
[16]: data = pd.get_dummies(data, columns=['Fuel_Type', 'Transmission'], prefix=['Fuel', 'Transmission'])
```

```
[17]: data
```

	Name	Location	Year	Kilometers_Driven	Owner_Type	Mileage	Engine	Power	Seats	Price	Fuel_Diesel	Fuel_Electric	Fuel_Petrol	Transmission_Automat
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	First	19.67	1582.0	126.20	5.0	12.50	True	False	False	Fal
1	Honda Jazz V	Chennai	2011	46000	First	13.00	1199.0	88.70	5.0	4.50	False	False	True	Fal
2	Maruti Ertiga VDI	Chennai	2012	87000	First	20.77	1248.0	88.76	7.0	6.00	True	False	False	Fal
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Second	15.20	1968.0	140.80	5.0	17.74	True	False	False	Tr
4	Nissan Micra Diesel XV	Jaipur	2013	86999	First	23.08	1461.0	63.10	5.0	3.50	True	False	False	Fal
...
5842	Maruti Swift VDI	Delhi	2014	27365	First	28.40	1248.0	74.00	5.0	4.75	True	False	False	Fal
5843	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	First	24.40	1120.0	71.00	5.0	4.00	True	False	False	Fal
5844	Mahindra Xylo D4	Jaipur	2013	55000	Second	14.00	2400.0	113.00	9.0	3.00	True	False	False	Fal
...
5845	Maruti Wagon R VXI	Kolkata	2013	46000	First	18.90	998.0	67.10	5.0	2.65	False	False	True	Fal
5846	Chevrolet Beat Diesel	Hyderabad	2011	47000	First	25.44	936.0	57.60	5.0	2.50	True	False	False	Fal

5847 rows x 15 columns

JupyterLab interface showing the execution of code to convert boolean columns to integers. The code defines a list of columns and uses the `astype(int)` method to convert them. A red circle with the number 1 indicates the first cell's output.

```
[18]: columns = ['Fuel_Diesel', 'Fuel_Electric', 'Fuel_Petrol', 'Transmission_Automatic', 'Transmission_Manual']

# Convert specified boolean columns to integers
data[columns] = data[columns].astype(int)

[19]: data
```

JupyterLab interface showing the resulting data table after conversion. The table has 15 columns: Name, Location, Year, Kilometers_Driven, Owner_Type, Mileage, Engine, Power, Seats, Price, Fuel_Diesel, Fuel_Electric, Fuel_Petrol, and Transmission_Automatic. The data is displayed in a table format with alternating light and dark gray rows.

	Name	Location	Year	Kilometers_Driven	Owner_Type	Mileage	Engine	Power	Seats	Price	Fuel_Diesel	Fuel_Electric	Fuel_Petrol	Transmission_Automatic
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	First	19.67	1582.0	126.20	5.0	12.50	1	0	0	
1	Honda Jazz V	Chennai	2011	46000	First	13.00	1199.0	88.70	5.0	4.50	0	0	1	
2	Maruti Ertiga VDI	Chennai	2012	87000	First	20.77	1248.0	88.76	7.0	6.00	1	0	0	
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Second	15.20	1968.0	140.80	5.0	17.74	1	0	0	
4	Nissan Micra Diesel XV	Jaipur	2013	86999	First	23.08	1461.0	63.10	5.0	3.50	1	0	0	
...
5842	Maruti Swift VDI	Delhi	2014	27365	First	28.40	1248.0	74.00	5.0	4.75	1	0	0	
5843	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	First	24.40	1120.0	71.00	5.0	4.00	1	0	0	
5844	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Second	14.00	2498.0	112.00	8.0	2.90	1	0	0	
5845	Maruti Wagon R VXI	Kolkata	2013	46000	First	18.90	998.0	67.10	5.0	2.65	0	0	1	

- This code is transforming certain columns (Fuel and Transmission types) that contain Boolean (True/False) values into integer values, where True becomes 1 and False becomes 0. The specified columns are converted to a numerical representation for further analysis or machine learning models.

Task d) Create one more feature and add this column to the dataset (you can use mutate function in R for this). For example, you can calculate the current age of the car by subtracting “Year” value from the current year.

Jupyter PDS_Assignment2_16340457-Copy1 Last Checkpoint: 36 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

Task d) Create one more feature and add this column to the dataset (you can use mutate function in R for this). For example, you can calculate the current age of the car by subtracting “Year” value from the current year.

```
[20]: # Import the datetime module
from datetime import datetime

# Assuming 'data' is your DataFrame and 'Year' column exists
# Calculate the current year
current_year = datetime.now().year

# Create a new column 'Car_Age' by subtracting 'Year' from the current year
data['Car_Age'] = current_year - data['Year']
```

[21]: data

Jupyter PDS_Assignment2_16340457-Copy1 Last Checkpoint: 36 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

[21]: data

	Name	Location	Year	Kilometers_Driven	Owner_Type	Mileage	Engine	Power	Seats	Price	Fuel_Diesel	Fuel_Electric	Fuel_Petrol	Transmission_Automat
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	First	19.67	1582.0	126.20	5.0	12.50	1	0	0	
1	Honda Jazz V	Chennai	2011	46000	First	13.00	1199.0	88.70	5.0	4.50	0	0	1	
2	Maruti Ertiga VDI	Chennai	2012	87000	First	20.77	1248.0	88.76	7.0	6.00	1	0	0	
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Second	15.20	1968.0	140.80	5.0	17.74	1	0	0	
4	Nissan Micra Diesel XV	Jaipur	2013	86999	First	23.08	1461.0	63.10	5.0	3.50	1	0	0	
...	
5842	Maruti Swift VDI	Delhi	2014	27365	First	28.40	1248.0	74.00	5.0	4.75	1	0	0	
5843	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	First	24.40	1120.0	71.00	5.0	4.00	1	0	0	
5844	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Second	14.00	2498.0	112.00	8.0	2.90	1	0	0	
5845	Maruti Wagon R VXI	Kolkata	2013	46000	First	18.90	998.0	67.10	5.0	2.65	0	0	1	

- This code utilizes the datetime module in Python to calculate the current year. Subsequently, it computes the age of the car by subtracting the 'Year' column from the current year, storing the result in a new column named 'Car_Age'.

