Python For Data Science Assignment - 1

Name: Sahil S Prabhu
SRN: PES2UG22EC115

MCQ:

Q1:

You have s = "12.0" in Python. Which of the following conversions will succeed without raising an exception and produce an integer value 12?

A. int(s)
B. int(float(s))
C. float(int(s))
D. str(int(s))

Q2:

Given a pandas DataFrame df whose KM column contains the string "50,000" and some numeric rows, which of the following is the best immediate diagnosis and action?

A. df['KM'].dtype will already be int64, so no action needed.
B. df['KM'] is likely read as an object due to the comma; remove commas and convert to numeric using df['KM'].str.replace(',', '').astype(int).
C. Use df.select_dtypes(include=['number']) to convert the column.
D. Replace missing values with df['KM'].fillna(df['KM'].mode()) then apply astype(int)

Descriptive Type:

Q1:
The price column in df has some NaN and is right-skewed. Give 2–3 lines of code to impute missing values and 2–3 line explanation why you chose that strategy.

Sol:
For skewed distributions the median is robust to outliers whereas the mean would be distorted; filling NaN with the median preserves a typical central value without being pulled by extreme prices.
Code:

```
med = df['price'].median(skipna=True)
df['price'].fillna(med, inplace=True)
```

Q2:
Column fuel contains values like 'Petrol', 'Diesel', 'CNG' and some typos. Show 2–3 lines of code to map known labels to integers and create a category-coded column; explain why category dtype can help.

Sol:
Mapping normalizes labels and converts them to integers for models. Using category saves memory and makes categorical operations (grouping, encoding) efficient,

unmapped or messy values become NaN for further handling.

Code:

```
mapping = {'petrol':0, 'diesel':1, 'cng':2}
df['fuel_code']=df['fuel'].str.lower().str.strip().map(mapping)
df['fuel_code']=df['fuel_code'].astype('category').cat.codes
```