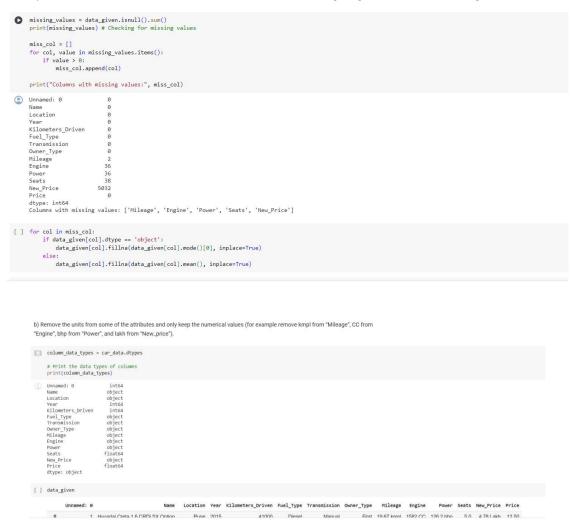
## Principles of Data Science - 5530

## Assignment-2

Name: Nikhil Jagadeesh Sriram

Student ID: 16352573

(A) In this context, I am addressing the identification and handling of missing values across all columns. For categorical columns, the missing values are being imputed with the mode. This approach is chosen to ensure the preservation of the existing data within that categorical field, thereby minimizing the potential for bias in the imputed values. As for numerical columns, missing values are being imputed with the mean. This method is adopted to maintain the distribution and central tendency of the available data in the numerical column, aligning the imputed values with the overall dataset distribution and mitigating the risk of introducing noticeable bias.



(B) Strip the units from select attributes, retaining only the numerical values. For instance, eliminate "kmpl" from the "Mileage" column, "CC" from the "Engine" column, "bhp" from the "Power" column, and "lakh" from the "New\_price" column.



(C) Convert the categorical variables, namely "Fuel\_Type" and "Transmission," into numerical one-hot encoded values.

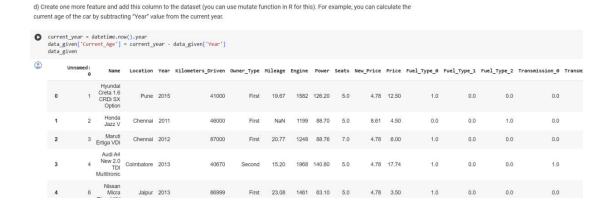
```
C) Change the categorical variables ("Fuel_Type" and 'Transmission') into numerical one hot encoded value.

[] label_encoder = LabelEncoder()
data_given['Fuel_Type_Label'] = label_encoder.fit_transform(data_given['Fuel_Type'])
data_given['Fransmission_Label'] = label_encoder.fit_transform(data_given['Fuel_Type'])
onehot_encoder = OneHotEncoder(sparsesFalse)

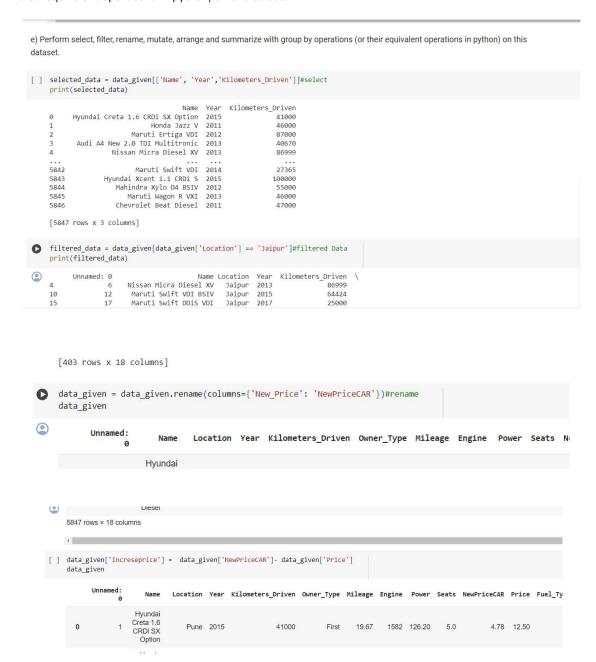
fuel_type_encoded = onehot_encoder.fit_transform(data_given['Fuel_Type_Label']))
transmission_encoded = onehot_encoder.fit_transform(data_given['Fuel_Type_Label']))

fuel_type_encoded f = pd.DataFrame(fuel_type_encoded, columns=['Fuel_Type_' + str(i) for i in range(fuel_type_encoded.shape[1])))
transmission_encoded_df = pd.DataFrame(fuel_type_encoded, columns=['Transmission_' + str(i) for i in range(transmission_encoded.shape[1])))
data_given = pd.concat([data_given, fuel_type_encoded_df, transmission_encoded_df], axis=1)
data_given.drop(['Fuel_Type', 'Transmission', 'Fuel_Type_Label', 'Transmission_Label'], axis=1, inplace=True)
```

(D) Introduce a new feature by calculating the current age of each car and add this column to the dataset. One way to achieve this is by subtracting the "Year" value from the current year.



(E) Perform select, filter, rename, mutate, arrange and summarize with group by operations (or their equivalent operations in python) on this dataset



4



(2) Unnamed: Location Year Kilometers\_Driven Owner\_Type Mileage Engine Po 0 Tata Nano 65000 1660 1713 Pune 2011 26.00 Second 624 Lx

5847 rows × 19 columns

average\_price\_by\_fuel\_type = data\_given.groupby('Kilometers\_Driven')['Mileage'].mean()#summarize average\_price\_by\_fuel\_type

Kilometers\_Driven 171 24.700000 600 1000 21.500000 18.493333 22.415000 1001 1011 17.180000 20.360000 20.540000 19.300000 17.180000 62000 20.360000 720000 20.540000 775000 19.30000 6500000 15.970000 Name: Mileage, Length: 3019, dtype: float64

[ ] data\_given.to\_csv('clean\_data.csv', index=False)