Service Emission Test Analysis Dashboard

Y Q M W D

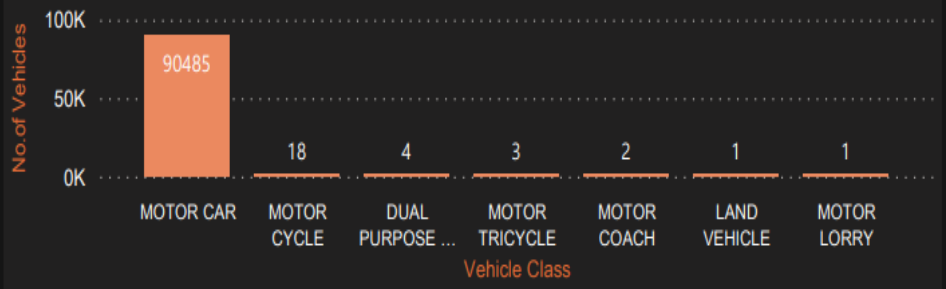
Year

2012 2013 2014 2015 2016

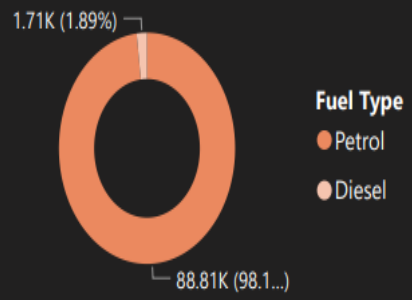
Pass Rate	Fail Rate	Average Mileage	Total Vehicles
97.43%	2.57%	96.53K	91K

Test Location

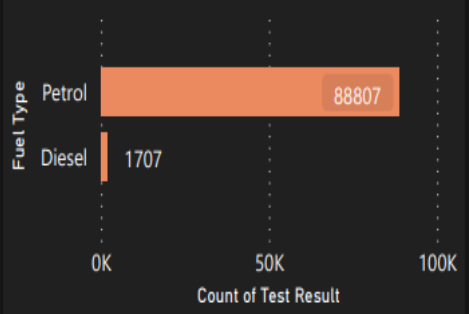
Vehicle Class Numbers



No.of Vehicles by Fuel Type

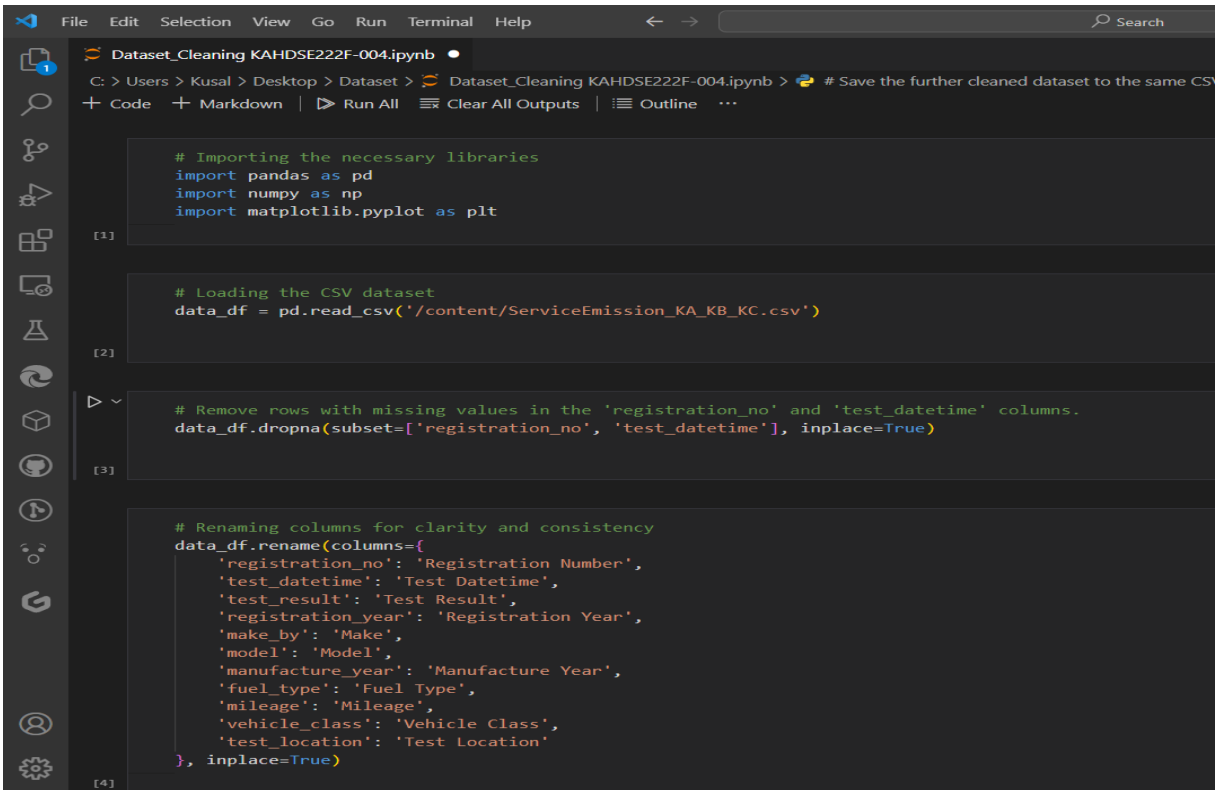


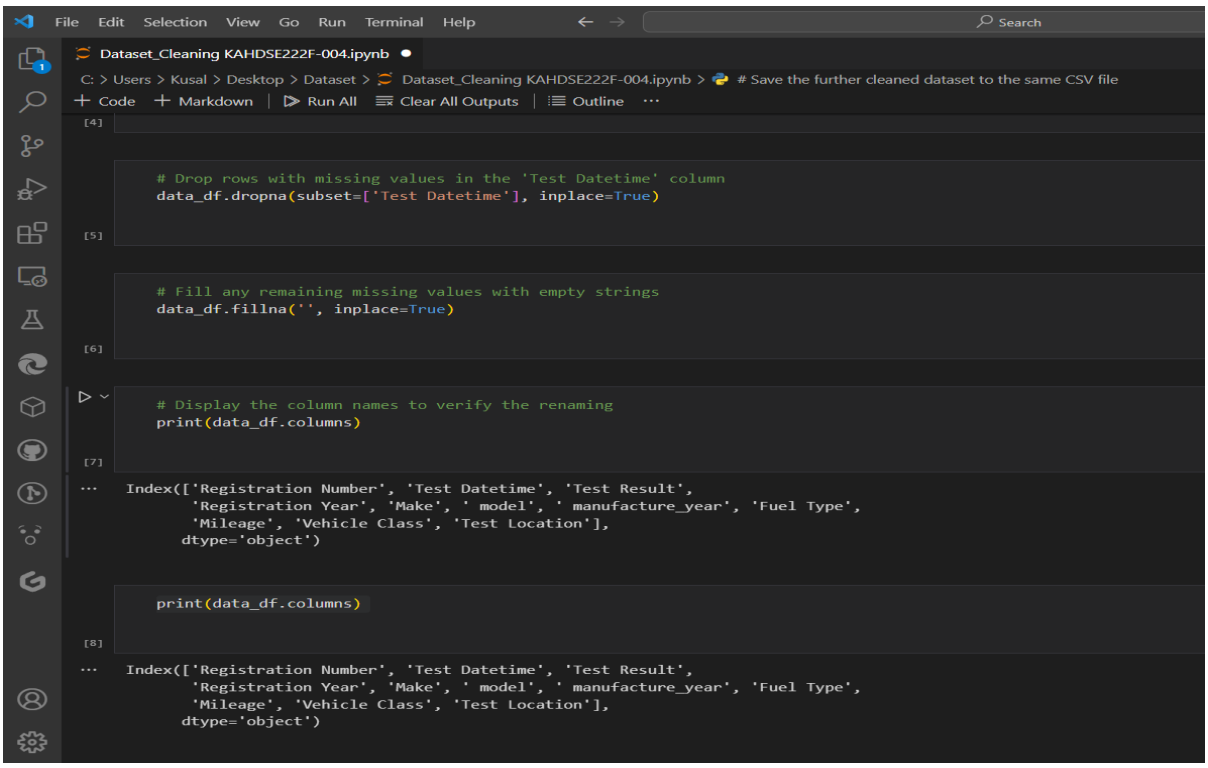
Test Result by Fuel Type

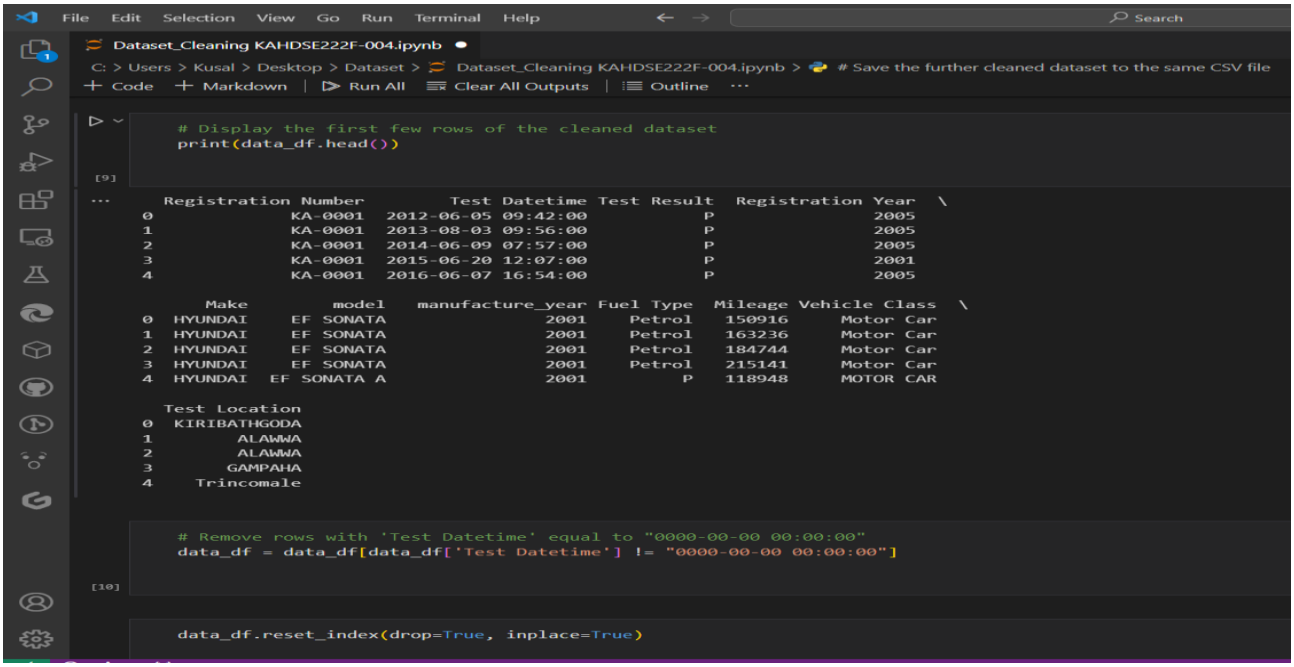


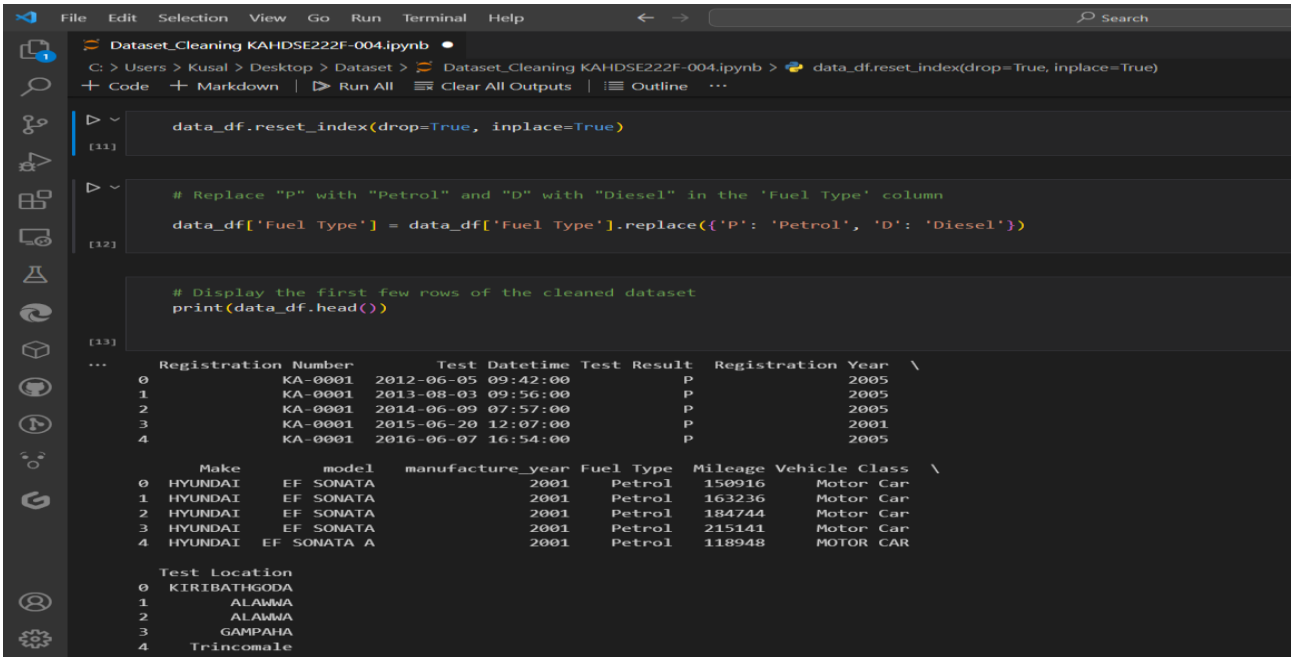
Registration Year	F	P	Total
2016	1	4	5
2015		19	19
2013		3	3
2012		6	6
2011		16	16
2010		175	175
2009	6	230	236
2008	2	159	161
2007	5	83	88
2006	1216	55695	56911
2005	1048	30686	31734
2004	3	204	207
2003	6	370	376
2002	18	368	386
2001	9	126	135
Total	2322	88192	90514

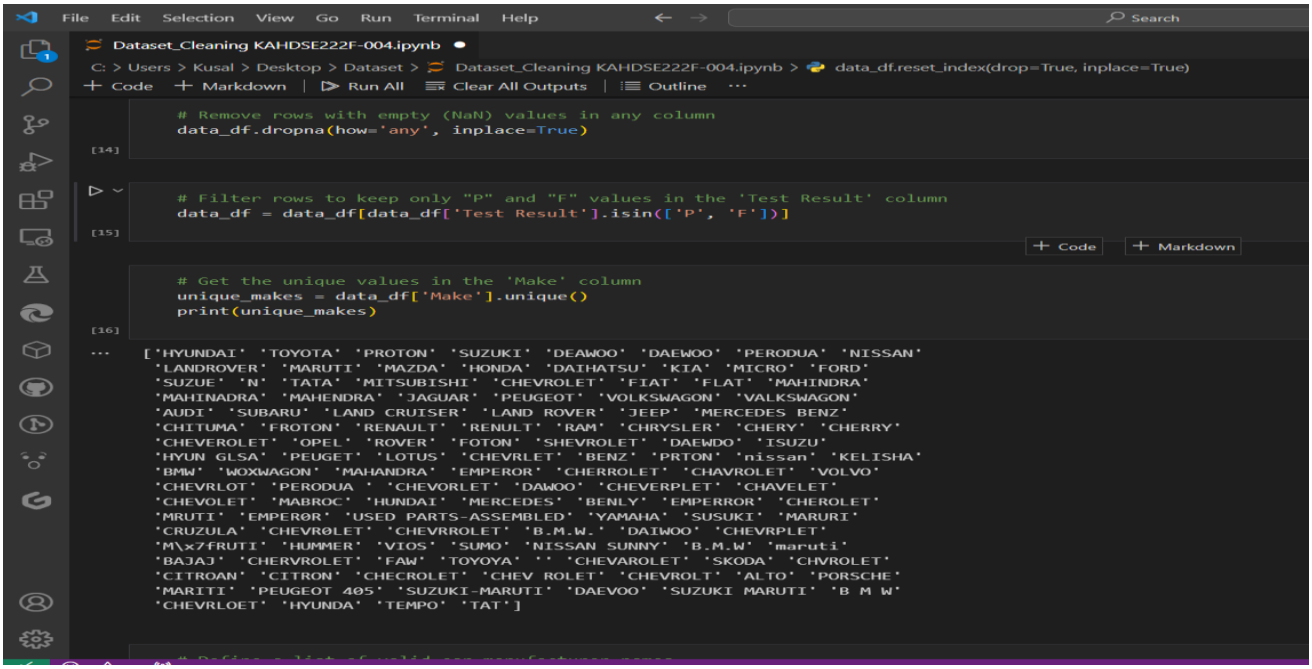
CODE

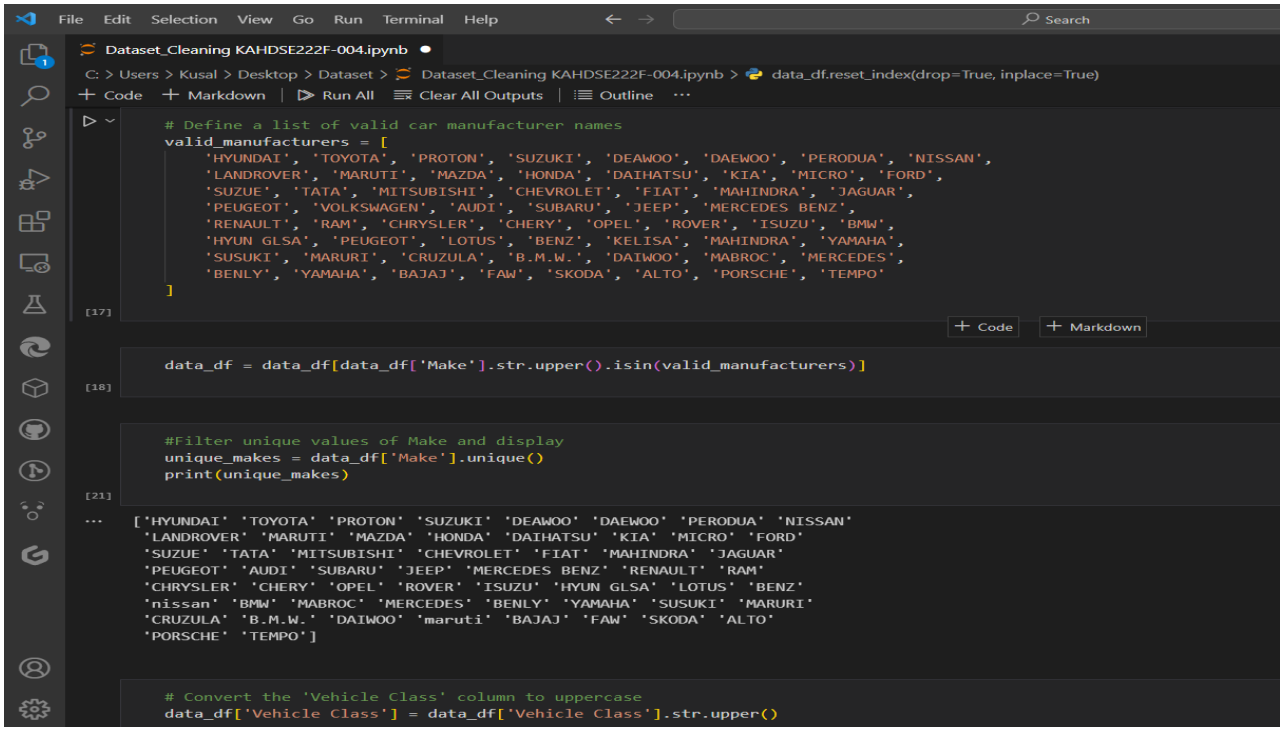

File Edit Selection View Go Run Terminal Help
Search
Dataset_Cleaning KAHDSE222F-004.ipynb
C: > Users > Kusal > Desktop > Dataset > Dataset_Cleaning KAHDSE222F-004.ipynb > # Save the further cleaned dataset to the same CSV
Code Markdown Run All Clear All Outputs Outline
# Importing the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
[1]
# Loading the CSV dataset
data_df = pd.read_csv('/content/ServiceEmission_KA_KB_KC.csv')
[2]
# Remove rows with missing values in the 'registration_no' and 'test_datetime' columns.
data_df.dropna(subset=['registration_no', 'test_datetime'], inplace=True)
[3]
# Renaming columns for clarity and consistency
data_df.rename(columns={
'registration_no': 'Registration Number',
'test_datetime': 'Test Datetime',
'test_result': 'Test Result',
'registration_year': 'Registration Year',
'make_by': 'Make',
'model': 'Model',
'manufacture_year': 'Manufacture Year',
'fuel_type': 'Fuel Type',
'mileage': 'Mileage',
'vehicle_class': 'Vehicle Class',
'test_location': 'Test Location'
}, inplace=True)
[4]

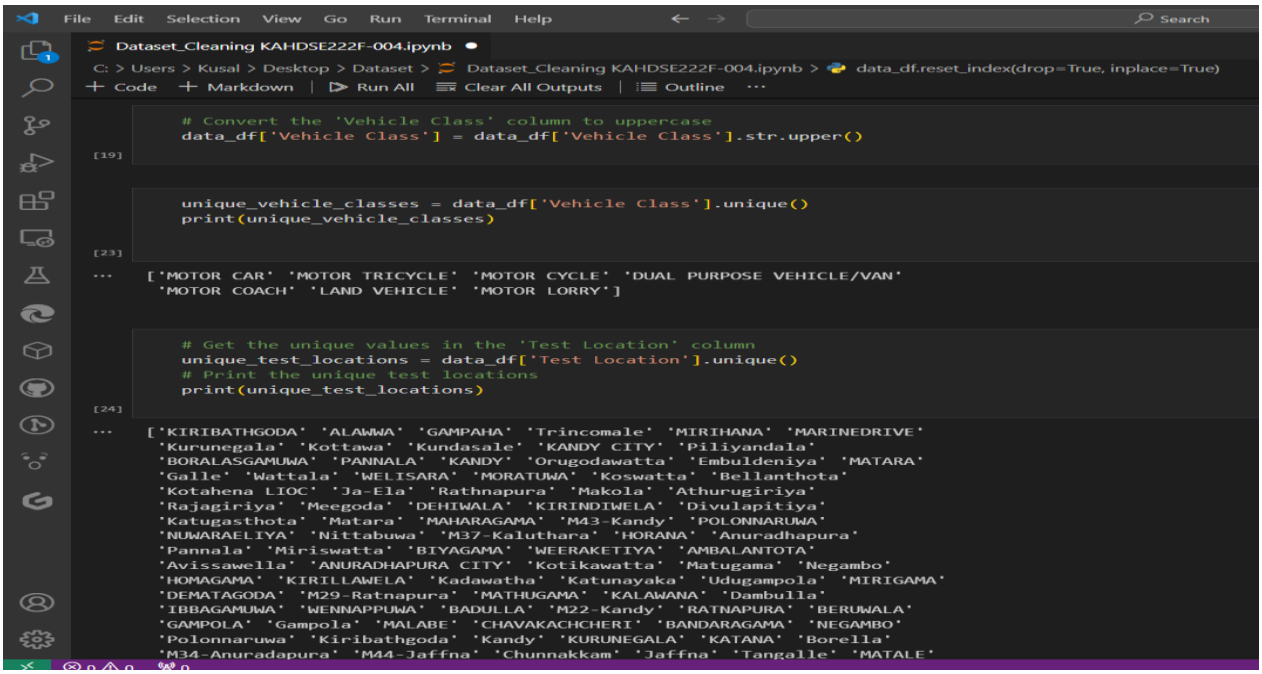

File Edit Selection View Go Run Terminal Help
Search
Dataset_Cleaning KAHDSE222F-004.ipynb
C: > Users > Kusal > Desktop > Dataset > Dataset_Cleaning KAHDSE222F-004.ipynb > # Save the further cleaned dataset to the same CSV file
Code Markdown Run All Clear All Outputs Outline
[4]
# Drop rows with missing values in the 'Test Datetime' column
data_df.dropna(subset=['Test Datetime'], inplace=True)
[5]
# Fill any remaining missing values with empty strings
data_df.fillna('', inplace=True)
[6]
# Display the column names to verify the renaming
print(data_df.columns)
[7]
Index(['Registration Number', 'Test Datetime', 'Test Result',
'Registration Year', 'Make', ' model', ' manufacture_year', 'Fuel Type',
'Mileage', 'Vehicle Class', 'Test Location'],
dtype='object')
print(data_df.columns)
[8]
Index(['Registration Number', 'Test Datetime', 'Test Result',
'Registration Year', 'Make', ' model', ' manufacture_year', 'Fuel Type',
'Mileage', 'Vehicle Class', 'Test Location'],
dtype='object')

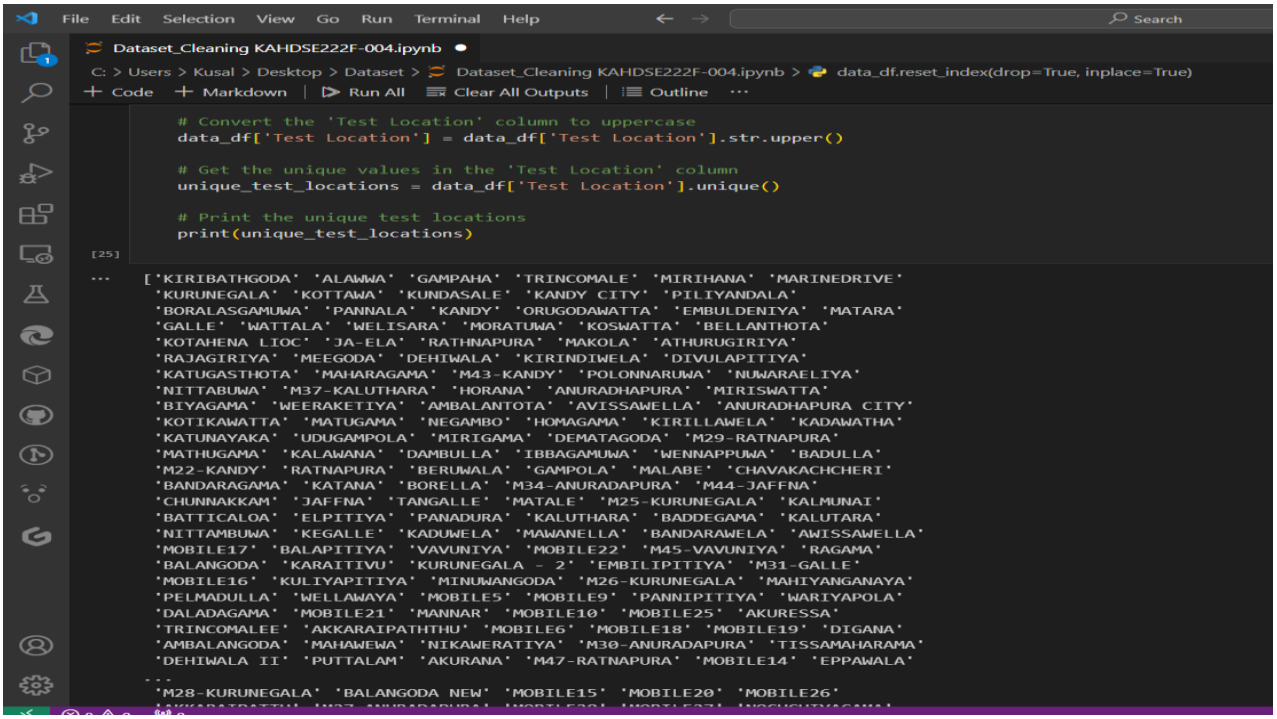

File Edit Selection View Go Run Terminal Help
Search
Dataset_Cleaning KAHDSE222F-004.ipynb
C: > Users > Kusal > Desktop > Dataset > Dataset_Cleaning KAHDSE222F-004.ipynb > # Save the further cleaned dataset to the same CSV file
Code Markdown Run All Clear All Outputs Outline
# Display the first few rows of the cleaned dataset
print(data_df.head())
[9]
Registration Number Test Datetime Test Result Registration Year \
0 KA-0001 2012-06-05 09:42:00 P 2005
1 KA-0001 2013-08-03 09:56:00 P 2005
2 KA-0001 2014-06-09 07:57:00 P 2005
3 KA-0001 2015-06-20 12:07:00 P 2001
4 KA-0001 2016-06-07 16:54:00 P 2005
Make model manufacture_year Fuel Type Mileage Vehicle Class \
0 HYUNDAI EF SONATA 2001 Petrol 150916 Motor Car
1 HYUNDAI EF SONATA 2001 Petrol 163236 Motor Car
2 HYUNDAI EF SONATA 2001 Petrol 184744 Motor Car
3 HYUNDAI EF SONATA 2001 Petrol 215141 Motor Car
4 HYUNDAI EF SONATA A 2001 P 118948 MOTOR CAR
Test Location
0 KIRIBATHGODA
1 ALAWWA
2 ALAWWA
3 GAMPAHA
4 Trincomale
# Remove rows with 'Test Datetime' equal to "0000-00-00 00:00:00"
data_df = data_df[data_df['Test Datetime'] != "0000-00-00 00:00:00"]
[10]
data_df.reset_index(drop=True, inplace=True)

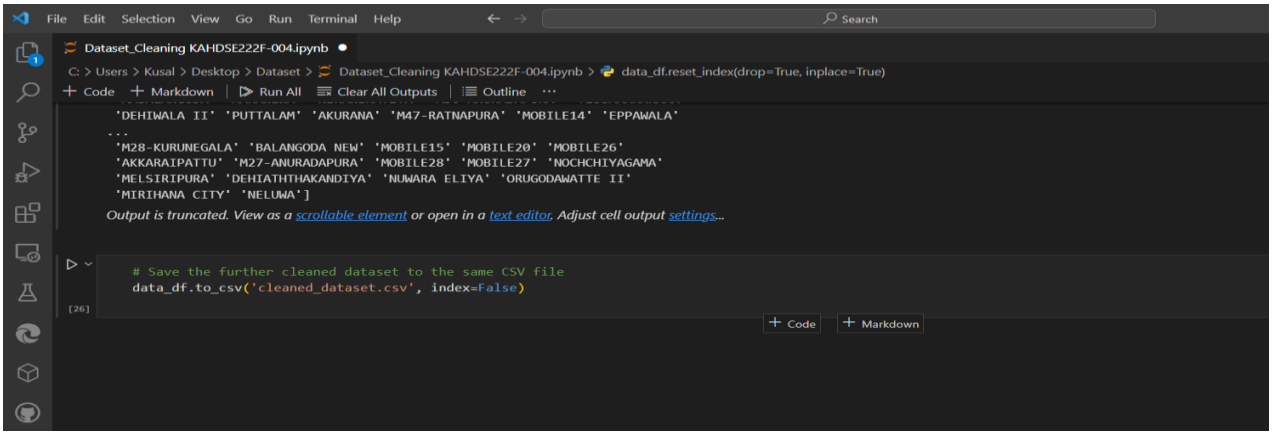

File Edit Selection View Go Run Terminal Help
Search
Dataset_Cleaning KAHDSE222F-004.ipynb
C: > Users > Kusal > Desktop > Dataset > Dataset_Cleaning KAHDSE222F-004.ipynb > data_df.reset_index(drop=True, inplace=True)
Code Markdown Run All Clear All Outputs Outline
data_df.reset_index(drop=True, inplace=True)
[11]
Replace "P" with "Petrol" and "D" with "Diesel" in the 'Fuel Type' column
data_df['Fuel Type'] = data_df['Fuel Type'].replace({'P': 'Petrol', 'D': 'Diesel'})
[12]
# Display the first few rows of the cleaned dataset
print(data_df.head())
[13]
Registration Number Test Datetime Test Result Registration Year \
0 KA-0001 2012-06-05 09:42:00 P 2005
1 KA-0001 2013-08-03 09:56:00 P 2005
2 KA-0001 2014-06-09 07:57:00 P 2005
3 KA-0001 2015-06-20 12:07:00 P 2001
4 KA-0001 2016-06-07 16:54:00 P 2005
Make model manufacture_year Fuel Type Mileage Vehicle Class \
0 HYUNDAI EF SONATA 2001 Petrol 150916 Motor Car
1 HYUNDAI EF SONATA 2001 Petrol 163236 Motor Car
2 HYUNDAI EF SONATA 2001 Petrol 184744 Motor Car
3 HYUNDAI EF SONATA 2001 Petrol 215141 Motor Car
4 HYUNDAI EF SONATA A 2001 Petrol 118948 MOTOR CAR
Test Location
0 KIRIBATHGODA
1 ALAWWA
2 ALAWWA
3 GAMPAHA
4 Trincomale


File Edit Selection View Go Run Terminal Help
Search
Dataset_Cleaning KAHDSE222F-004.ipynb
C: > Users > Kusal > Desktop > Dataset > Dataset_Cleaning KAHDSE222F-004.ipynb > data_df.reset_index(drop=True, inplace=True)
Code Markdown Run All Clear All Outputs Outline
# Remove rows with empty (NaN) values in any column
data_df.dropna(how='any', inplace=True)
[14]
# Filter rows to keep only "P" and "F" values in the 'Test Result' column
data_df = data_df[data_df['Test Result'].isin(['P', 'F'])]
[15]
Code Markdown
# Get the unique values in the 'Make' column
unique_makes = data_df['Make'].unique()
print(unique_makes)
[16]
['HYUNDAI' 'TOYOTA' 'PROTON' 'SUZUKI' 'DEAWOO' 'DAEWOO' 'PERODUA' 'NISSAN'
'LANDROVER' 'MARUTI' 'MAZDA' 'HONDA' 'DAIHATSU' 'KIA' 'MICRO' 'FORD'
'SUZUE' 'N' 'TATA' 'MITSUBISHI' 'CHEVROLET' 'FIAT' 'FLAT' 'MAHINDRA'
'MAHINADRA' 'MAHENDRA' 'JAGUAR' 'PEUGEOT' 'VOLKSWAGON' 'VALKSWAGON'
'AUDI' 'SUBARU' 'LAND CRUISER' 'LAND ROVER' 'JEEP' 'MERCEDES BENZ'
'CHITUMA' 'FROTON' 'RENAULT' 'RENULT' 'RAM' 'CHRYSLER' 'CHERY' 'CHERRY'
'CHEVEROLET' 'OPEL' 'ROVER' 'FOTON' 'SHEVROLET' 'DAEWDO' 'ISUZU'
'HYUN GLSA' 'PEUGET' 'LOTUS' 'CHEVRLET' 'BENZ' 'PRTON' 'nissan' 'KELISHA'
'BMW' 'WOXWAGON' 'MAHANDRA' 'EMPEROR' 'CHERROLET' 'CHAVROLET' 'VOLVO'
'CHEVRLOT' 'PERODUA ' 'CHEVORLET' 'DAWOO' 'CHEVERPLET' 'CHAVELET'
'CHEVOLET' 'MABROC' 'HUNDAI' 'MERCEDES' 'BENLY' 'EMPERROR' 'CHEROLET'
'MRUTI' 'EMPER0R' 'USED PARTS-ASSEMBLED' 'YAMAHA' 'SUSUKI' 'MARURI'
'CRUZULA' 'CHEVR0LET' 'CHEVRROLET' 'B.M.W.' 'DAIWOO' 'CHEVRPLET'
'M\x7fRUTI' 'HUMMER' 'VIOS' 'SUMO' 'NISSAN SUNNY' 'B.M.W' 'maruti'
'BAJAJ' 'CHERVROLET' 'FAW' 'TOYOYA' '' 'CHEVAROLET' 'SKODA' 'CHVROLET'
'CITROAN' 'CITRON' 'CHECROLET' 'CHEV ROLET' 'CHEVROLT' 'ALTO' 'PORSCHE'
'MARITI' 'PEUGEOT 405' 'SUZUKI-MARUTI' 'DAEVOO' 'SUZUKI MARUTI' 'B M W'
'CHEVRLOET' 'HYUNDA' 'TEMPO' 'TAT']


File Edit Selection View Go Run Terminal Help
Search
Dataset_Cleaning KAHDSE222F-004.ipynb
C: > Users > Kusal > Desktop > Dataset > Dataset_Cleaning KAHDSE222F-004.ipynb > data_df.reset_index(drop=True, inplace=True)
Code Markdown Run All Clear All Outputs Outline
# Define a list of valid car manufacturer names
valid_manufacturers = [
'HYUNDAI', 'TOYOTA', 'PROTON', 'SUZUKI', 'DEAWOO', 'DAEWOO', 'PERODUA', 'NISSAN',
'LANDROVER', 'MARUTI', 'MAZDA', 'HONDA', 'DAIHATSU', 'KIA', 'MICRO', 'FORD',
'SUZUE', 'TATA', 'MITSUBISHI', 'CHEVROLET', 'FIAT', 'MAHINDRA', 'JAGUAR',
'PEUGEOT', 'VOLKSWAGEN', 'AUDI', 'SUBARU', 'JEEP', 'MERCEDES BENZ',
'RENAULT', 'RAM', 'CHRYSLER', 'CHERY', 'OPEL', 'ROVER', 'ISUZU', 'BMW',
'HYUN GLSA', 'PEUGEOT', 'LOTUS', 'BENZ', 'KELISA', 'MAHINDRA', 'YAMAHA',
'SUSUKI', 'MARURI', 'CRUZULA', 'B.M.W.', 'DAIWOO', 'MABROC', 'MERCEDES',
'BENLY', 'YAMAHA', 'BAJAJ', 'FAW', 'SKODA', 'ALTO', 'PORSCHE', 'TEMPO'
]
[17]
Code Markdown
data_df = data_df[data_df['Make'].str.upper().isin(valid_manufacturers)]
[18]
#Filter unique values of Make and display
unique_makes = data_df['Make'].unique()
print(unique_makes)
[21]
['HYUNDAI' 'TOYOTA' 'PROTON' 'SUZUKI' 'DEAWOO' 'DAEWOO' 'PERODUA' 'NISSAN'
'LANDROVER' 'MARUTI' 'MAZDA' 'HONDA' 'DAIHATSU' 'KIA' 'MICRO' 'FORD'
'SUZUE' 'TATA' 'MITSUBISHI' 'CHEVROLET' 'FIAT' 'MAHINDRA' 'JAGUAR'
'PEUGEOT' 'AUDI' 'SUBARU' 'JEEP' 'MERCEDES BENZ' 'RENAULT' 'RAM'
'CHRYSLER' 'CHERY' 'OPEL' 'ROVER' 'ISUZU' 'HYUN GLSA' 'LOTUS' 'BENZ'
'nissan' 'BMW' 'MABROC' 'MERCEDES' 'BENLY' 'YAMAHA' 'SUSUKI' 'MARURI'
'CRUZULA' 'B.M.W.' 'DAIWOO' 'maruti' 'BAJAJ' 'FAW' 'SKODA' 'ALTO'
'PORSCHE' 'TEMPO']
Convert the 'Vehicle Class' column to uppercase
data_df['Vehicle Class'] = data_df['Vehicle Class'].str.upper()


File Edit Selection View Go Run Terminal Help
Search
Dataset_Cleaning KAHDSE222F-004.ipynb
C: > Users > Kusal > Desktop > Dataset > Dataset_Cleaning KAHDSE222F-004.ipynb > data_df.reset_index(drop=True, inplace=True)
Code Markdown Run All Clear All Outputs Outline
Convert the 'Vehicle Class' column to uppercase
data_df['Vehicle Class'] = data_df['Vehicle Class'].str.upper()
[19]
unique_vehicle_classes = data_df['Vehicle Class'].unique()
print(unique_vehicle_classes)
[23]
['MOTOR CAR' 'MOTOR TRICYCLE' 'MOTOR CYCLE' 'DUAL PURPOSE VEHICLE/VAN'
'MOTOR COACH' 'LAND VEHICLE' 'MOTOR LORRY']
# Get the unique values in the 'Test Location' column
unique_test_locations = data_df['Test Location'].unique()
# Print the unique test locations
print(unique_test_locations)
[24]
['KIRIBATHGODA' 'ALAWWA' 'GAMPAHA' 'Trincomale' 'MIRIHANA' 'MARINEDRIVE'
'Kurunegala' 'Kottawa' 'Kundasale' 'KANDY CITY' 'Piliyandala'
'BORALASGAMUWA' 'PANNALA' 'KANDY' 'Orugodawatta' 'Embuldeniya' 'MATARA'
'Galle' 'Wattala' 'WELISARA' 'MORATUWA' 'Koswatta' 'Bellanthota'
'Kotahena LIOC' 'Ja-Ela' 'Rathnapura' 'Makola' 'Athurugiriya'
'Rajagiriya' 'Meegoda' 'DEHIWALA' 'KIRINDIWELA' 'Divulapitiya'
'Katugasthota' 'Matara' 'MAHARAGAMA' 'M43-Kandy' 'POLONNARUWA'
'NUWARAELIYA' 'Nittabuwa' 'M37-Kaluthara' 'HORANA' 'Anuradhapura'
'Pannala' 'Miriswatta' 'BIYAGAMA' 'WEERAKETIYA' 'AMBALANTOTA'
'Avissawella' 'ANURADHAPURA CITY' 'Kotikawatta' 'Matugama' 'Negambo'
'HOMAGAMA' 'KIRILLAWELA' 'Kadawatha' 'Katunayaka' 'Udugampola' 'MIRIGAMA'
'DEMATAGODA' 'M29-Ratnapura' 'MATHUGAMA' 'KALAWANA' 'Dambulla'
'IBBAGAMUWA' 'WENNAPPUWA' 'BADULLA' 'M22-Kandy' 'RATNAPURA' 'BERUWALA'
'GAMPOLA' 'Gampola' 'MALABE' 'CHAVAKACHCHERI' 'BANDARAGAMA' 'NEGAMBO'
'Polonnaruwa' 'Kiribathgoda' 'Kandy' 'KURUNEGALA' 'KATANA' 'Borella'
'M34-Anuradapura' 'M44-Jaffna' 'Chunnakkam' 'Jaffna' 'Tangalle' 'MATALE'


File Edit Selection View Go Run Terminal Help
Search
Dataset_Cleaning KAHDSE222F-004.ipynb
C: > Users > Kusal > Desktop > Dataset > Dataset_Cleaning KAHDSE222F-004.ipynb > data_df.reset_index(drop=True, inplace=True)
Code Markdown Run All Clear All Outputs Outline
Convert the 'Test Location' column to uppercase
data_df['Test Location'] = data_df['Test Location'].str.upper()
# Get the unique values in the 'Test Location' column
unique_test_locations = data_df['Test Location'].unique()
# Print the unique test locations
print(unique_test_locations)
[25]
['KIRIBATHGODA' 'ALAWWA' 'GAMPAHA' 'TRINCOMALE' 'MIRIHANA' 'MARINEDRIVE'
'KURUNEGALA' 'KOTTAWA' 'KUNDASALE' 'KANDY CITY' 'PILIYANDALA'
'BORALASGAMUWA' 'PANNALA' 'KANDY' 'ORUGODAWATTA' 'EMBULDENIYA' 'MATARA'
'GALLE' 'WATTALA' 'WELISARA' 'MORATUWA' 'KOSWATTA' 'BELLANTHOTA'
'KOTAHENA LIOC' 'JA-ELA' 'RATHNAPURA' 'MAKOLA' 'ATHURUGIRIYA'
'RAJAGIRIYA' 'MEEGODA' 'DEHIWALA' 'KIRINDIWELA' 'DIVULAPITIYA'
'KATUGASTHOTA' 'MAHARAGAMA' 'M43-KANDY' 'POLONNARUWA' 'NUWARAELIYA'
'NITTABUWA' 'M37-KALUTHARA' 'HORANA' 'ANURADHAPURA' 'MIRISWATTA'
'BIYAGAMA' 'WEERAKETIYA' 'AMBALANTOTA' 'AVISSAWELLA' 'ANURADHAPURA CITY'
'KOTIKAWATTA' 'MATUGAMA' 'NEGAMBO' 'HOMAGAMA' 'KIRILLAWELA' 'KADAWATHA'
'KATUNAYAKA' 'UDUGAMPOLA' 'MIRIGAMA' 'DEMATAGODA' 'M29-RATNAPURA'
'MATHUGAMA' 'KALAWANA' 'DAMBULLA' 'IBBAGAMUWA' 'WENNAPPUWA' 'BADULLA'
'M22-KANDY' 'RATNAPURA' 'BERUWALA' 'GAMPOLA' 'MALABE' 'CHAVAKACHCHERI'
'BANDARAGAMA' 'KATANA' 'BORELLA' 'M34-ANURADAPURA' 'M44-JAFFNA'
'CHUNNAKKAM' 'JAFFNA' 'TANGALLE' 'MATALE' 'M25-KURUNEGALA' 'KALMUNAI'
'BATTICALOA' 'ELPITIYA' 'PANADURA' 'KALUTHARA' 'BADDEGAMA' 'KALUTARA'
'NITTAMBUWA' 'KEGALLE' 'KADUWELA' 'MAWANELLA' 'BANDARAWELA' 'AWISSAWELLA'
'MOBILE17' 'BALAPITIYA' 'VAVUNIYA' 'MOBILE22' 'M45-VAVUNIYA' 'RAGAMA'
'BALANGODA' 'KARAITIVU' 'KURUNEGALA - 2' 'EMBILIPITIYA' 'M31-GALLE'
'MOBILE16' 'KULIYAPITIYA' 'MINUWANGODA' 'M26-KURUNEGALA' 'MAHIYANGANAYA'
'PELMADULLA' 'WELLAWAYA' 'MOBILE5' 'MOBILE9' 'PANNIPITIYA' 'WARIYAPOLA'
'DALADAGAMA' 'MOBILE21' 'MANNAR' 'MOBILE10' 'MOBILE25' 'AKURESSA'
'TRINCOMALEE' 'AKKARAIPATHTHU' 'MOBILE6' 'MOBILE18' 'MOBILE19' 'DIGANA'
'AMBALANGODA' 'MAHAWEWA' 'NIKAWERATIYA' 'M30-ANURADAPURA' 'TISSAMAHARAMA'
'DEHIWALA II' 'PUTTALAM' 'AKURANA' 'M47-RATNAPURA' 'MOBILE14' 'EPPAWALA'
...
'M28-KURUNEGALA' 'BALANGODA NEW' 'MOBILE15' 'MOBILE20' 'MOBILE26'


File Edit Selection View Go Run Terminal Help
Search
Dataset_Cleaning KAHDSE222F-004.ipynb
C: > Users > Kusal > Desktop > Dataset > Dataset_Cleaning KAHDSE222F-004.ipynb > data_df.reset_index(drop=True, inplace=True)
Code Markdown Run All Clear All Outputs Outline
'DEHIWALA II' 'PUTTALAM' 'AKURANA' 'M47-RATNAPURA' 'MOBILE14' 'EPPAWALA'
...
'M28-KURUNEGALA' 'BALANGODA NEW' 'MOBILE15' 'MOBILE20' 'MOBILE26'
'AKKARAIPATTU' 'M27-ANURADAPURA' 'MOBILE28' 'MOBILE27' 'NOCHCHIYAGAMA'
'MELSIRIPURA' 'DEHIATHTHAKANDIYA' 'NUWARA ELIYA' 'ORUGODAWATTE II'
'MIRIHANA CITY' 'NELUWA']
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
# Save the further cleaned dataset to the same CSV file
data_df.to_csv('cleaned_dataset.csv', index=False)
[26]
Code Markdown