01_cleaning

February 5, 2026

1 Engine Atlas - Cleaning and Feature Engineering

This notebook standardizes columns, coerces numeric types, handles outliers, and creates derived features like displacement and power density.

[1]:
```
from pathlib import Path
import sys

ROOT = Path('..').resolve()
SRC = ROOT / 'src'
sys.path.append(str(SRC))

import pandas as pd
from engine_atlas.data_processing import clean_engine_data, schema_report

DATA_PATH = ROOT / 'data' / 'Car Dataset 1945-2020.csv'
df = clean_engine_data(str(DATA_PATH))
df.head()
```

[1]:
```
   id_trim make model    generation  year_from  year_to     series    trim  \
0        1   AC   ACE  1 generation     1993.0   2000.0  Cabriolet  3.5 MT
1        2   AC   ACE  1 generation     1993.0   2000.0  Cabriolet  4.6 MT
2        3   AC   ACE  1 generation     1993.0   2000.0  Cabriolet  4.9 AT
3        4   AC   ACE  1 generation     1993.0   2000.0   Roadster  2.9 AT
4        5   AC   ACE  1 generation     1993.0   2000.0   Roadster  2.9 MT

   body_type load_height_mm  … battery_capacity_kw_per_h electric_range_km  \
0  Cabriolet            NaN  …                       NaN               NaN
1  Cabriolet            NaN  …                       NaN               NaN
2  Cabriolet            NaN  …                       NaN               NaN
3        NaN            NaN  …                       NaN               NaN
4        NaN            NaN  …                       NaN               NaN

  charging_time_h    year bore_mm stroke_mm displacement_l hp_per_liter  \
0             NaN  1993.0    83.0       NaN          3.505   100.998573
1             NaN  1993.0    90.0       NaN          4.601    70.854162
2             NaN  1993.0   101.0       NaN          4.942    52.610279
3             NaN  1993.0     NaN       NaN            NaN          NaN
```

```
4             NaN  1993.0     NaN       NaN            NaN          NaN

               engine_signature balanced_score
0   AC Gasoline V-type 8.0 3.5L       1.792864
1   AC Gasoline V-type 8.0 4.6L       1.642758
2  AC Gasoline V-type 8.0 4.94L       0.588031
3                       AC    L            NaN
4                       AC    L            NaN

[5 rows x 85 columns]
```

[2]:
```
report = schema_report(df)
report.rows, report.cols
```

[2]:
```
(70823, 85)
```

[3]:
```
report.missing_by_col.head(20)
```

[3]:
```
overhead_camshaft                          70822
bore_stroke_ratio                          70821
steering_type                              70821
cylinder_bore_and_stroke_cycle_mm          70818
stroke_mm                                  70818
charging_time_h                            70816
electric_range_km                          70808
battery_capacity_kw_per_h                  70808
rating_name                                69811
safety_assessment                          69811
co2_emissions_g_km                         68994
cargo_volume_m3                            68588
cargo_compartment_length_width_height_mm   67486
load_height_mm                             67460
front_rear_axle_load_kg                    64466
compression_ratio                          64285
engine_placement                           64198
max_power_kw                               63203
wheel_size_r14                             62587
clearance_mm                               60300
dtype: int64
```

[4]:
```
# Ensure df is from the current cleaning logic before saving.
df = clean_engine_data(str(DATA_PATH))

required_cols = {
    'year',
    'engine_hp',
    'acceleration_0_100_km_h_s',
```

```
    'number_of_cylinders',
    'mixed_fuel_consumption_per_100_km_l',
    'co2_emissions_g_km',
    'engine_signature',
    'hp_per_liter',
}
missing = required_cols.difference(df.columns)
if missing:
    raise ValueError(f'Missing required columns: {sorted(missing)}')

output_path = ROOT / 'data' / 'processed' / 'engine_atlas_cleaned.parquet'
df.to_parquet(output_path, index=False)
output_path
```

[4]:
```
PosixPath('/home/albi/Documents/engine-
atlas/data/processed/engine_atlas_cleaned.parquet')
```