Predict Car Prices with Machine Learning

RISTANTO
DigitalSkola

Table of Contents

Machine learning yang digunakan adalah dengan pemrograman Python

Menentukan Label Data

Label / target:
kolom ‘Price’

Fitur:
Kolom selain ‘Price’

Total kolom dataset ada 18 kolom yang termasuk di dalamnya mencakup informasi mengenai Id, price, levy, manufacturer, model, production year, category, leather interior, fuel type, engine volume, mileage, cylinders, gear box type, drive wheels, doors, wheel, color dan airbags.

Back

Mengumpulkan Data

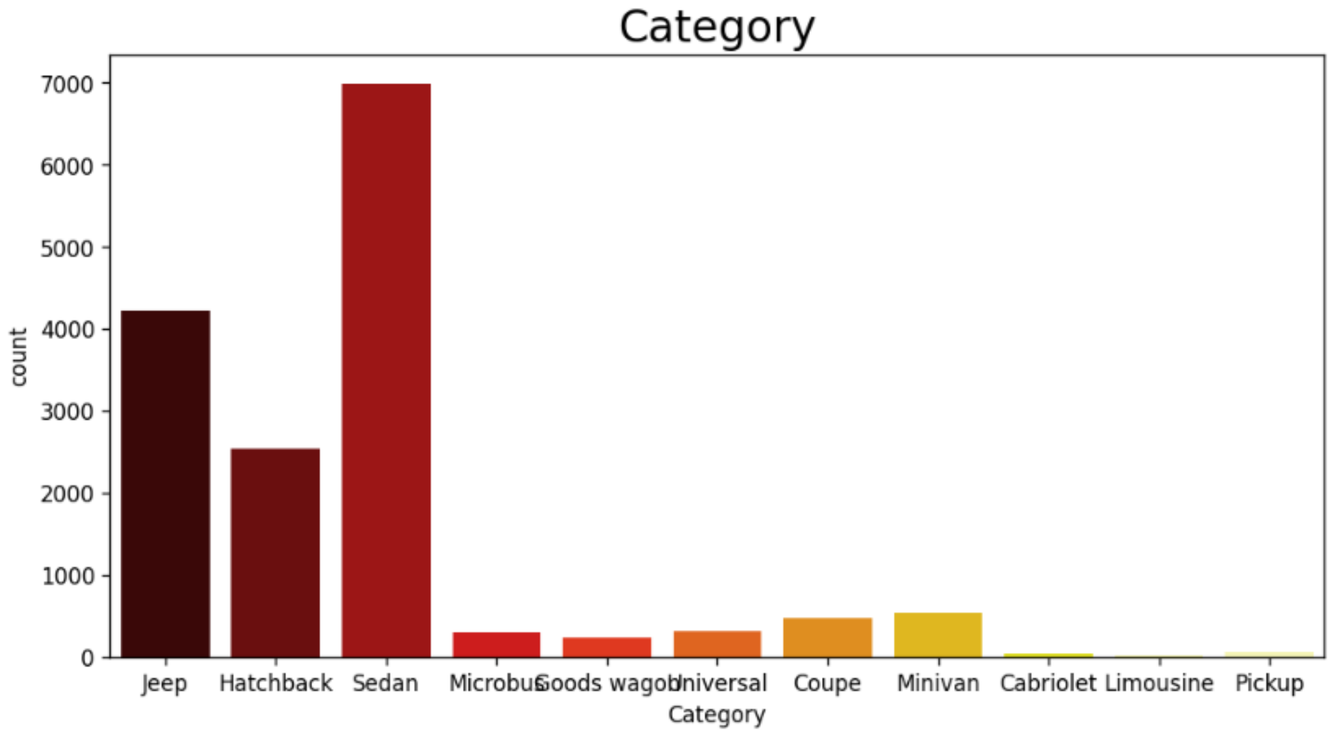
Car price prediction dataset ini diambil dari Kaggle:
https://www.kaggle.com/datasets/deepcontractor/car-price-prediction-challenge

Dan merupakan sekumpulan data yang memiliki 19237 baris x 18 kolom. Tujuan dari dataset adalah untuk menentukan prediksi harga mobil berdasarkan fitur-fitur yang dimiliki oleh dataset.

Back

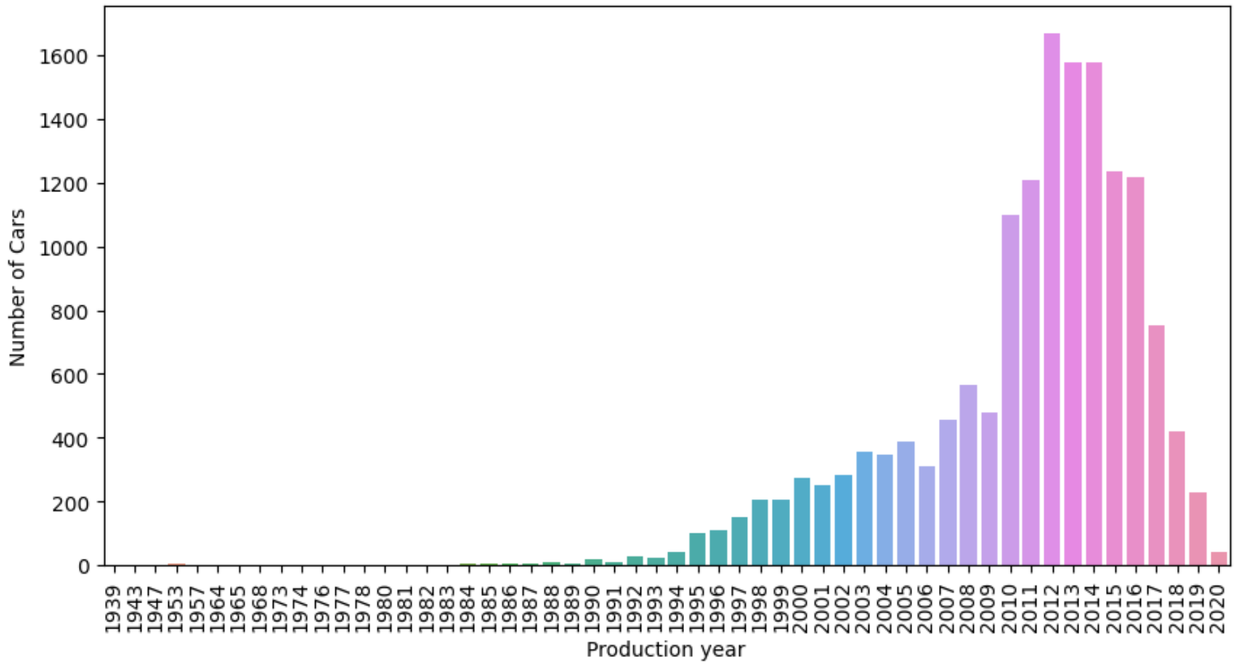
Menelaah
Data

Back



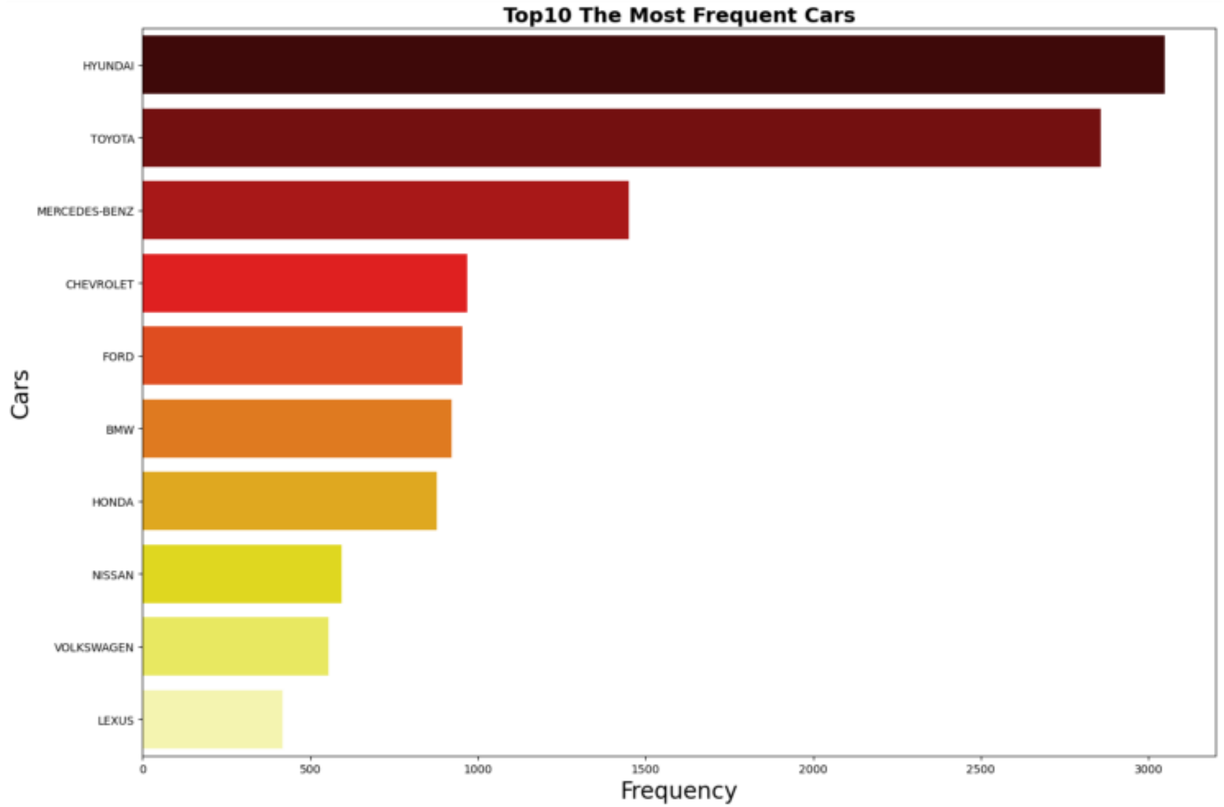
Category sedan merupakan category yang paling banyak

Back



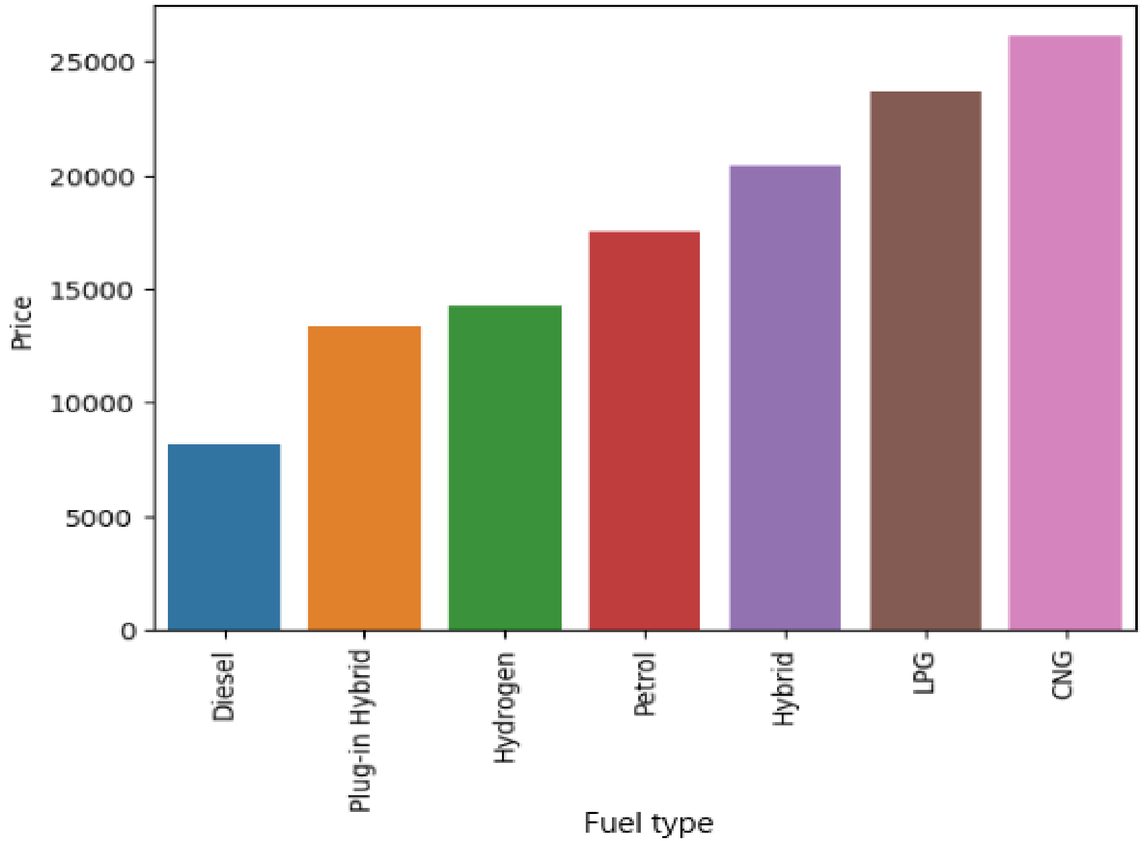
Production year yang memiliki jumlah paling banyak mulai tahun 2010 hingga tahun 2016

Back



Hyundai dan toyota merupakan merupakan jumlah brand manufacturer yang paling dominan

Back



Diesel merupakan fuel type yang memiliki harga paling murah, sedangkan CNG merupakan fuel type dengan harga paling mahal

Back

Memvalidasi Data

```
# Check Duplikasi
dataset.duplicated().sum()
```

```
313
```

Setelah dilakukan pengecekan duplikasi data, didapatkan sebanyak 313 data duplikasi

```
dataset['Levy'].unique()
```

```
array(['1399', '1018', '-', '862', '446', '891',
'761', '751', '394', '1053', '1055', '1079', '810',
'2386', '1850', '531', '586', …
```

Terdapat nilai (-) dan tipe datanya 'object' maka dilakukan penggantian untuk (-) menjadi (0) dan tipe datanya menjadi integer

Back

Menentukan Object Data

	Price	Levy	Manufacturer	Model	Prod. year	Category	Leather interior	Fuel type	Engine volume	Mileage	Cylinders	Gear box type	Drive wheels	Wheel	Color	Airbags
0	13328	1399	LEXUS	RX 450	2010	Jeep	Yes	Hybrid	3.5	186005 km	6.0	Automatic	4x4	Left wheel	Silver	12
1	16621	1018	CHEVROLET	Equinox	2011	Jeep	No	Petrol	3	192000 km	6.0	Tiptronic	4x4	Left wheel	Black	8
2	8467	0	HONDA	FIT	2006	Hatchback	No	Petrol	1.3	200000 km	4.0	Variator	Front	Right-hand drive	Black	2
3	3607	862	FORD	Escape	2011	Jeep	Yes	Hybrid	2.5	168966 km	4.0	Automatic	4x4	Left wheel	White	0
4	11726	446	HONDA	FIT	2014	Hatchback	Yes	Petrol	1.3	91901 km	4.0	Automatic	Front	Left wheel	Silver	4

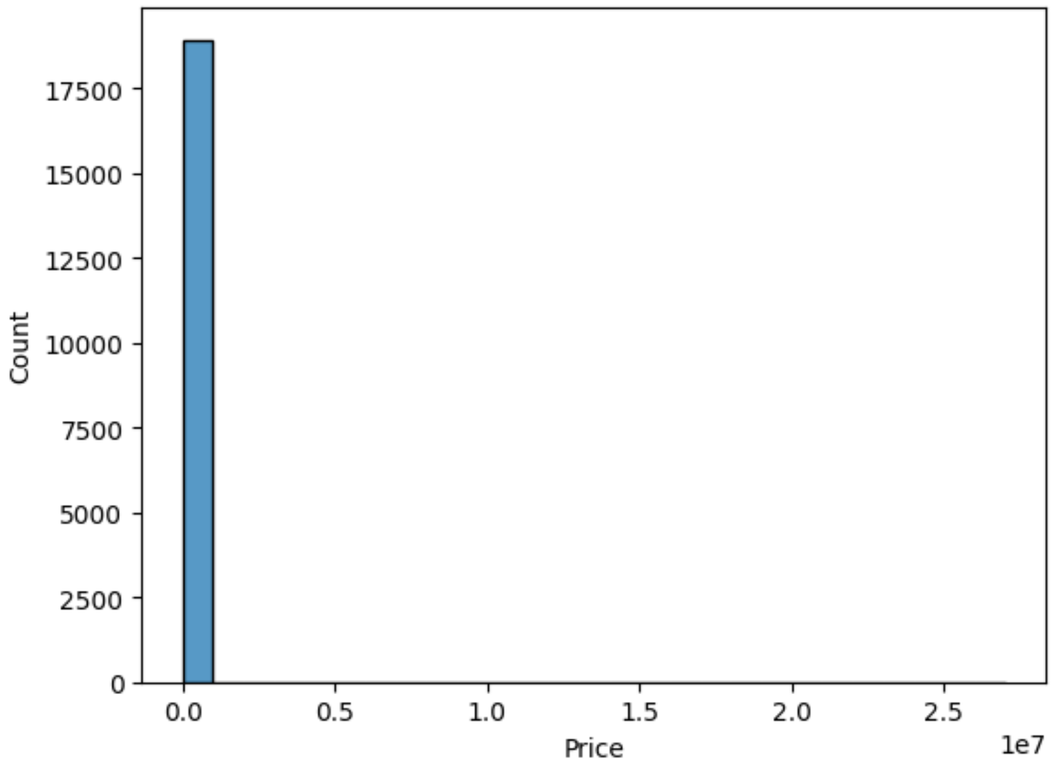
Objek data pada tabel ini yaitu variabel "Price (target)".

Ada pengurangan di fitur yaitu drop kolom 'ID' dan 'Doors' karena tidak dipakai dalam model.

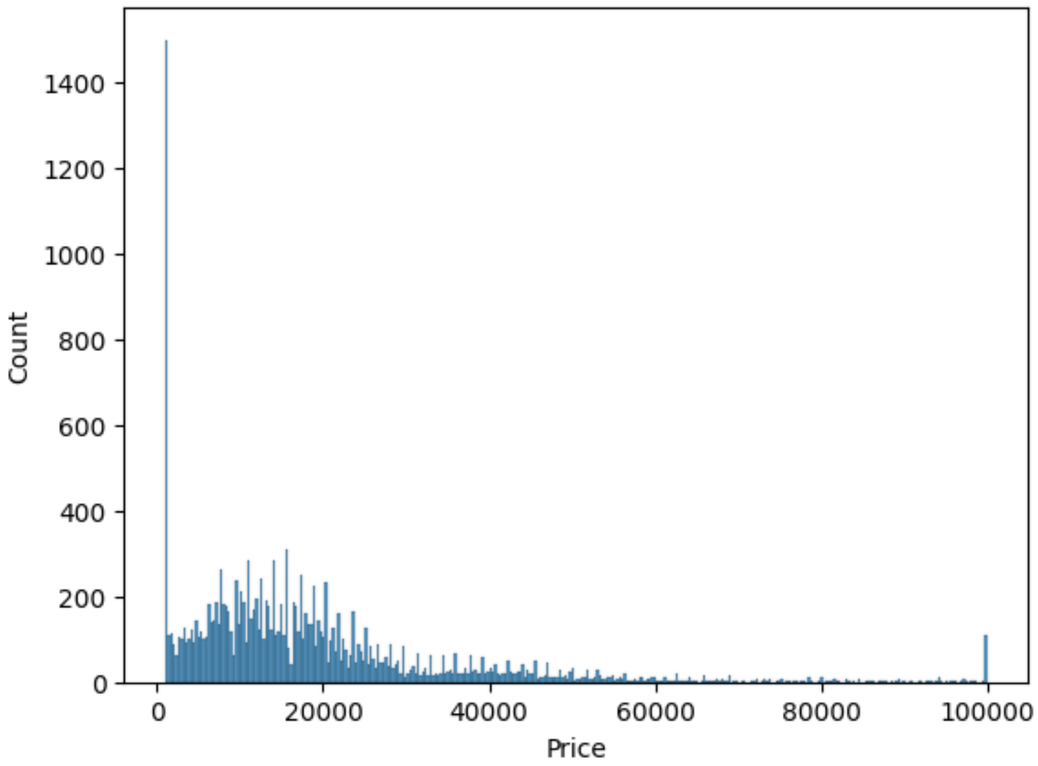
Back

#Histogram:

Before



After



Pembatasan nilai pada batas atas dan batas bawah:
threshold_low = 1000
threshold_high = 100000

Back

Membersihkan Data

1. Duplikasi data berkaitan dengan integritas dan keakuratan analisis data oleh karena itu 313 data duplikasi tersebut di drop

```
# karena ada duplikasi maka 'drop' duplikasi
dataset.drop_duplicates(inplace= True)
```

2. Kolom Levy yang missing value diisi dengan 0 (nol)

```
dataset["Levy"] = np.where(dataset["Levy"] == "-" ,0 , dataset["Levy"]).astype(int)
```

Back

Mengkonstruksi Data

Kolom	Keterangan
Mileage	Menghapus 'km' dan mengubahnya menjadi tipe data integer
Fuel type	Merubah setiap hybrid menjadi Fuel (petrol) + Electric source
Engine volume	Menghapus kata 'turbo' dan mengubah tipe data menjadi float, kemudian merubah setiap nilai (0) dan >10 dengan nilai yang sesuai.
Electric Source	Membuat Kolom baru yang hanya berisi data “Hybrid” dan bertipe data float

Back

Mengkonstruksi Data

```
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Price             18924 non-null  int64
 1   Levy              18924 non-null  object
 2   Manufacturer      18924 non-null  object
 3   Model             18924 non-null  object
 4   Prod. year        18924 non-null  int64
 5   Category          18924 non-null  object
 6   Leather interior  18924 non-null  object
 7   Fuel type         18924 non-null  object
 8   Engine volume     18924 non-null  object
 9   Mileage           18924 non-null  object
 10  Cylinders         18924 non-null  float64
 11  Gear box type     18924 non-null  object
 12  Drive wheels      18924 non-null  object
 13  Wheel             18924 non-null  object
 14  Color             18924 non-null  object
 15  Airbags           18924 non-null  int64
dtypes: float64(1), int64(3), object(12)
```

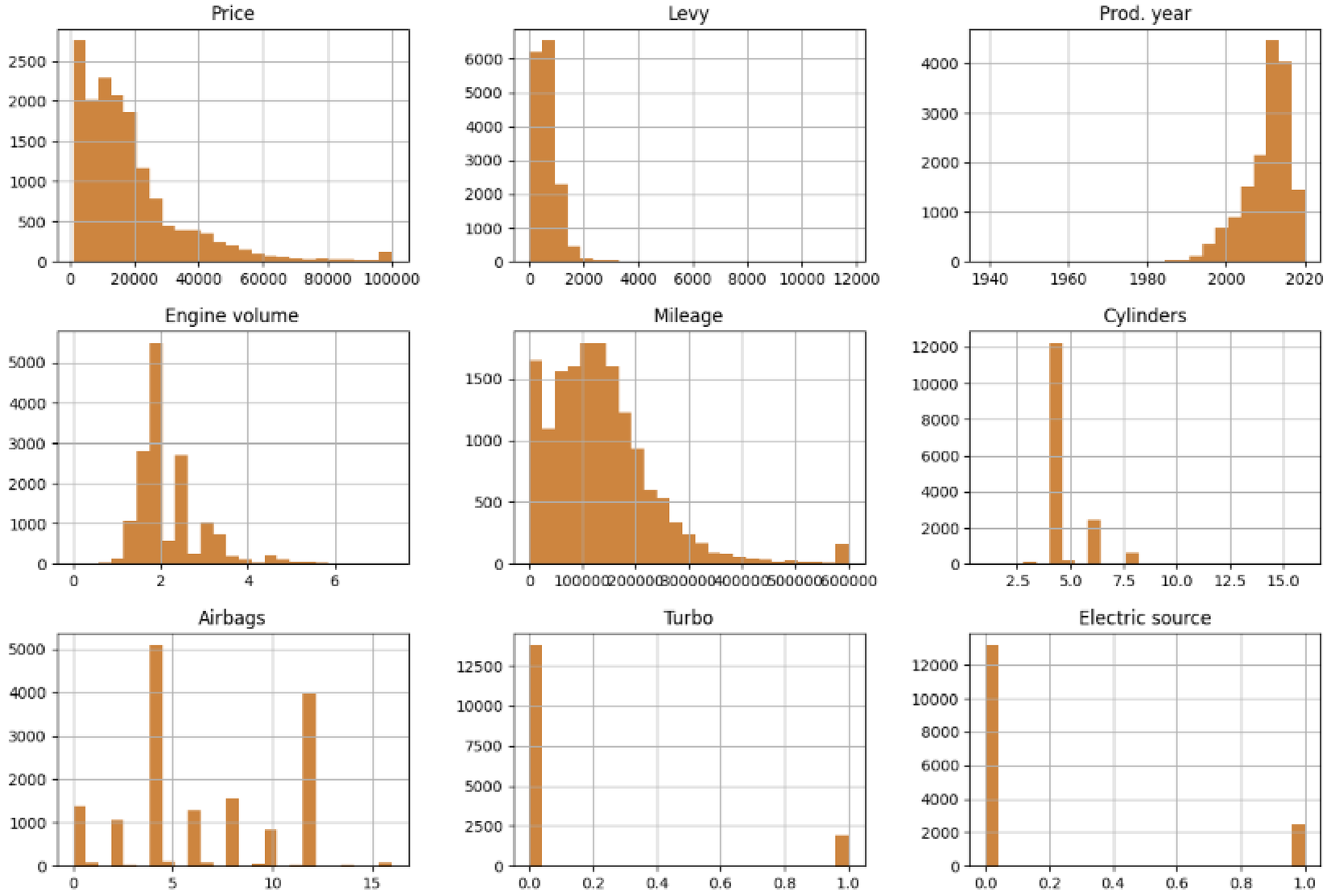
Data Info before construction

```
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Price             18924 non-null  int64
 1   Levy              18924 non-null  int64
 2   Manufacturer      18924 non-null  object
 3   Model             18924 non-null  object
 4   Prod. year        18924 non-null  int64
 5   Category          18924 non-null  object
 6   Leather interior  18924 non-null  object
 7   Fuel type         18924 non-null  object
 8   Engine volume     18924 non-null  object
 9   Mileage           18924 non-null  object
 10  Cylinders         18924 non-null  float64
 11  Gear box type     18924 non-null  object
 12  Drive wheels      18924 non-null  object
 13  Wheel             18924 non-null  object
 14  Color             18924 non-null  object
 15  Airbags           18924 non-null  int64
 16  Electric source   18924 non-null  float64
dtypes: float64(2), int64(4), object(11)
```

Data Info after construction

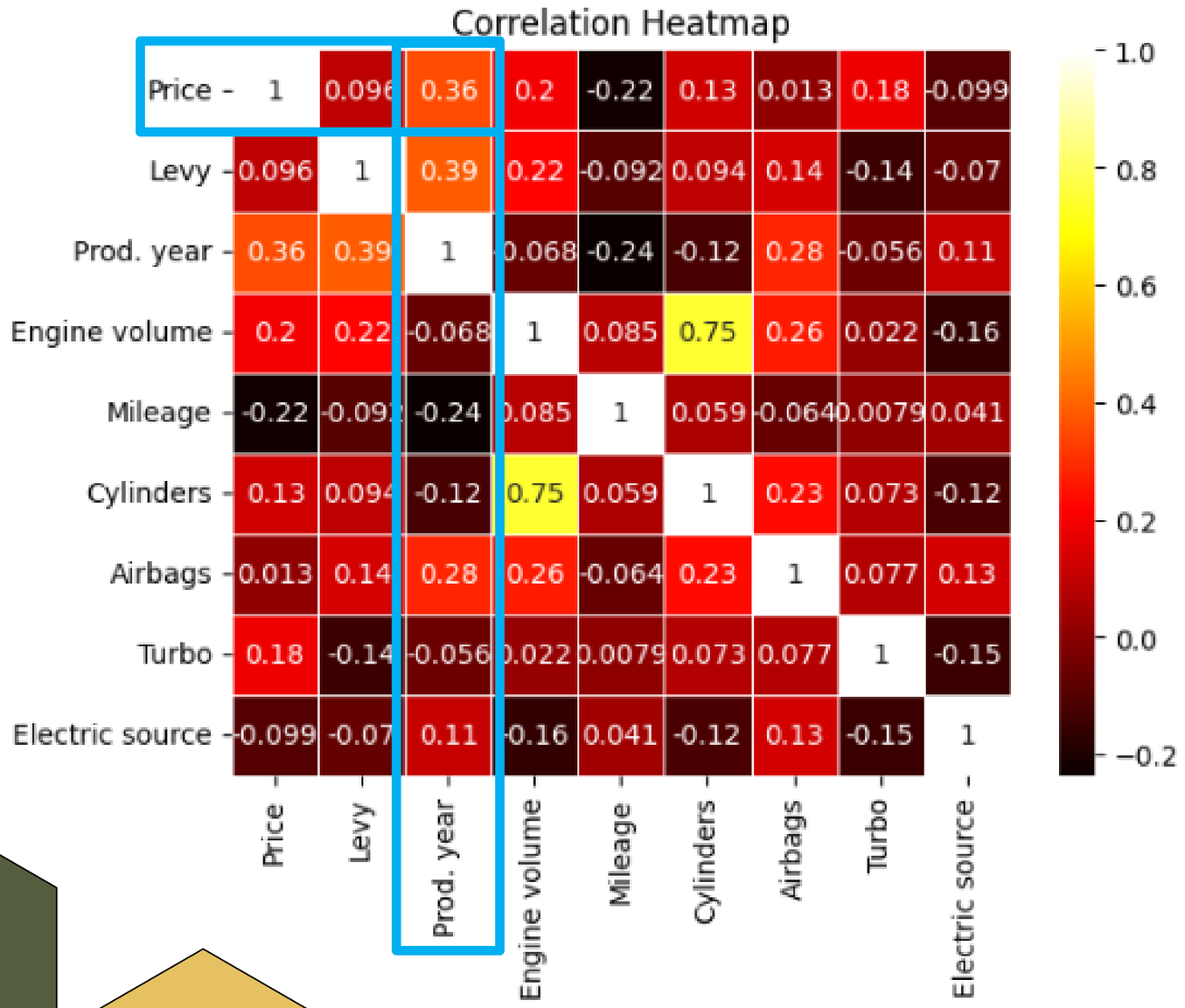
Back

Distribusi Data



Back

Korelasi Data



Dapat dilihat bahwa hubungan atau relasi antara kolom Price dengan Prod. year memiliki relasi dengan nilai 0.36

Back

Mengkonstruksi Data

Label / target:
kolom ‘Price’

Fitur kategorikal:
Dilabeling dengan LabelEncoder

Hasil:
9 fitur ter-encode ke numerik:

Manufacturer	Model	Category	Leather interior	Fuel type	Gear box type	Drive wheels	Wheel	Color
LEXUS	RX 450	Jeep	Yes	Petrol	Automatic	4x4	Left wheel	Silver
CHEVROLET	Equinox	Jeep	No	Petrol	Tiptronic	4x4	Left wheel	Black
HONDA	FIT	Hatchback	No	Petrol	Variator	Front	Right-hand drive	Black

Back

Mengkonstruksi Data

Selanjutnya dilakukan Feature Engineering menggunakan Robust scaler (untuk variabel X):

```
#Bagi datanya menjadi Variabel
independen dan dependen
X = data.drop(['Price'], axis=1)
y = data['Price']
```

```
from sklearn.preprocessing import RobustScaler
scaler = RobustScaler()
scaler.fit(X)
X = scaler.transform(X)
```

```
X.tail(5)
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
15699	0.909091	0.409904	-0.8	0.0	0.0	0.0	0.0	0.0	0.454545	0.006961	-0.166667	-0.25	1.642954	0.0	0.750	0.0	1.0
15700	0.242424	-0.647868	-1.2	0.0	-4.0	1.0	1.0	0.0	0.454545	-0.741299	-2.166667	0.00	1.578365	0.0	-0.125	1.0	0.0
15701	-0.151515	0.657497	0.4	0.0	0.0	2.0	0.0	0.0	0.363636	0.222738	-0.166667	0.50	0.357998	0.0	0.250	0.0	0.0
15702	-0.151515	0.806052	-0.6	0.0	-3.0	0.0	0.0	0.0	0.000000	0.228538	-0.333333	0.00	-0.040870	0.0	-0.250	0.0	0.0
15703	-0.151515	0.657497	0.4	0.0	0.0	0.0	0.0	0.0	0.636364	0.132251	0.000000	0.50	0.581288	0.0	0.750	0.0	1.0

Back

Membangun Model

Model akan ditrain pada 5 regressor, yaitu:

Random forest

Decision Tree

Linear regression

XGboost

LightGBM

```
X_train, X_test, y_train, y_test =
train_test_split(X,y, test_size =
0.1, random_state = 0)
```

Back

Linear Regression

```
lm = LinearRegression()
lm.fit(X_train, y_train)
```

Evaluasi

Hasil Pemodelan

Kategori	Train	Test
R-squared	33.04	36.51
MAE	9492.13	9594.38
RMSE	13702.88	13722.42

Back

Random Forest

```
reg = RandomForestRegressor(n_estimators = 200,
                            criterion= 'squared_error',
                            random_state= 100)
```

```
reg = RandomForestRegressor()
reg.fit(X_train, y_train)
```

Kategori	Train	Test
R-squared	96.11	80.49
MAE	1821.56	4599.36
RMSE	3301.03	7606.61

Kategori	Train	Test
R-squared	96.07	80.13
MAE	1837.68	4631.35
RMSE	3338.60	7676.79

Back

XGBoost

```
reg = XGBRegressor(eval_metric= 'rmse',
                   n_estimators= 200,
                   learning_rate= 0.1)
```

```
reg = XGBRegressor()
reg.fit(X_train, y_train)
```

Kategori	Train	Test
R-squared	88.38	80.48
MAE	3741.63	4800.12
RMSE	5706.21	7608.96

Kategori	Train	Test
R-squared	90.98	79.21
MAE	3389.00	4860.89
RMSE	5029.33	7852.22

Back

LightGBM

```
reg = LGBMRegressor(num_trees= 150,
                    num_leaves= 71,
                    objective= 'regression')
```

```
reg = LGBMRegressor()
reg.fit(X_train, y_train)
```

Kategori	Train	Test
R-squared	90.20	80.30
MAE	3398.98	4715.54
RMSE	5241.28	7642.56

Kategori	Train	Test
R-squared	82.39	78.19
MAE	4472.95	5011.99
RMSE	7026.16	8042.57

Back

Decision Tree

```
reg = DecisionTreeRegressor(random_state = 100,
                            criterion= 'squared_error',
                            splitter= 'best',
                            min_samples_leaf= 5,
                            min_samples_split= 10
                            )
```

```
reg = DecisionTreeRegressor(random_state = 0)
reg.fit(X_train, y_train)
```

Kategori	Train	Test
R-squared	84.44	71.23
MAE	3692.71	5475.73
RMSE	6604.74	9237.61

Kategori	Train	Test
R-squared	99.60	62.07
MAE	82.26	5907.01
RMSE	1049.53	10606.16

Back

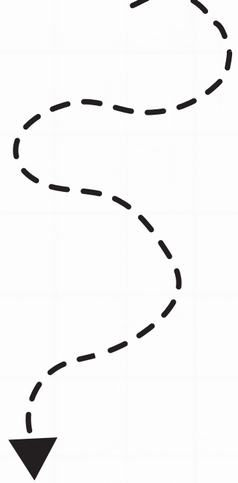
Summary

Summary R-Squared

	Model	R-squared Score
0	Random Forest	80.502944
1	XGBoost	80.480248
2	LightGBM	80.307476
3	Decision Tree	71.229785
4	Linear Regression	36.512955

Dipilih model random forest untuk diteruskan ke model deployment

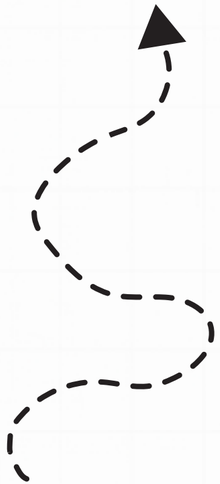
Back



Link Dataset dan Script

Google Colab

Github



Thank
You