

PRESENTASI UTS

DATA MINING

Analisis dan Prediksi Harga Mobil Bekas Menggunakan Regresi Linear

HAIDAR PUTRA ATMAJAYA - 5A - 2310631170019

PENDAHULUAN

Pasar mobil bekas adalah sektor yang kompleks dan dinamis. Menentukan harga yang akurat memerlukan pemahaman mendalam terhadap banyak faktor, seperti kondisi mesin, usia, dan riwayat pemakaian.

Proyek ini bertujuan untuk membangun model Regresi Linear yang kuat untuk memprediksi harga jual mobil bekas secara akurat setelah melalui proses pembersihan dan rekayasa fitur data yang ketat.

METODE YANG DIGUNAKAN

1

PYTHON

2

PREPROCESSING
DATA

3

METODE REGRESI
LINIER

4

EVALUASI MODEL

INISIALISASI DAN PEMUATAN DATA

```
[1] ✓ 0 d
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns

[2] ✓ 0 d
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler, MinMaxScaler
    from sklearn.linear_model import LinearRegression
    from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
    import missingno as msno

[4] ✓ 0 d
    url = 'https://raw.githubusercontent.com/FarrellAdityaaa/dataset-uts-datamining/refs/heads/main/used_cars_price_fiks.csv'
    df = pd.read_csv(url)
    df.info()
    df.head()

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 6019 entries, 0 to 6018
    Data columns (total 13 columns):
    #   Column           Non-Null Count  Dtype  
    ---  -- 
    0   Unnamed: 0       6019 non-null   int64  
    1   Name             6019 non-null   object  
    2   Location          6019 non-null   object  
    3   Year              6019 non-null   int64  
    4   Kilometers_Driven 5719 non-null   float64
    5   Fuel_Type          6019 non-null   object  
    6   Transmission        6019 non-null   object  
    7   Owner_Type         6019 non-null   object  
    8   Mileage            6017 non-null   float64
    9   Engine              5983 non-null   float64
    10  Power              5876 non-null   float64
    11  Seats               5977 non-null   float64
    12  Price              6019 non-null   float64
    dtypes: float64(6), int64(2), object(5)
    memory usage: 611.4+ KB
```

- Import Library: Mengimpor semua pustaka utama.
- Pemuatan Data: Memuat dataset dari URL menjadi DATAFRAME.
- Inspeksi Awal: Memeriksa jumlah total entry dan kolom.

ANALISIS DATA EKSPLORATIF AWAL

[5]
✓ 1d

```
▶ # Menampilkan ringkasan statistik dari data numerik dan menghitung jumlah nilai yang hilang per kolom
print('\nDescribe (numerical)')
print(df.describe())
print('\nMissing values per column')
print(df.isna().sum())
```

```
→ Describe (numerical)
      Unnamed: 0      Year Kilometers_Driven   Mileage     Engine \
count  6019.000000  6019.000000      5719.000000  6017.000000  5983.000000
mean   3009.000000  2013.358199    57545.592586  18.134961  1621.276450
std    1737.679967  3.269742     37988.496154  4.582289  601.355233
min    0.000000    1998.000000     171.000000  0.000000  72.000000
25%   1504.500000  2011.000000    33923.000000  15.170000  1198.000000
50%   3009.000000  2014.000000    53000.000000  18.150000  1493.000000
75%   4513.500000  2016.000000    72998.000000  21.100000  1984.000000
max   6018.000000  2019.000000   775000.000000  33.540000  5998.000000
```

- Statistik Deskriptif: Menampilkan ringkasa statistik untuk kolom numerik guna memahami sebaran dan potensi outlier.
- Missing Values: Mengkonfirmasi jumlah nilai yang hilang pada setiap kolom sebelum proses pembersihan dimulai.
- Visualisasi Korelasi: membuat Heatmap Korelasi awal untuk fitur numerik guna melihat hubungan antar variabel, korelasi negatif antara year dan kilometers_Driven dengan harga.

	Power	Seats	Price
count	5876.000000	5977.000000	6019.000000
mean	113.253050	5.278735	9.479468
std	53.874957	0.808840	11.187917
min	34.200000	0.000000	0.440000
25%	75.000000	5.000000	3.500000
50%	97.700000	5.000000	5.640000
75%	138.100000	5.000000	9.950000
max	560.000000	10.000000	160.000000
Missing values per column			
Unnamed: 0	0		
Name	0		
Location	0		
Year	0		
Kilometers_Driven	300		
Fuel_Type	0		
Transmission	0		
Owner_Type	0		
Mileage	2		
Engine	36		
Power	143		
Seats	42		
Price	0		
dtype: int64			

PEMBERSIHAN DATA & TRANSFORMASI

```
Duplicates: 0
```

```
Missing counts:
```

```
Kilometers_Driven      300
Mileage                  2
Engine                   36
Power                    143
Seats                     42
```

```
dtype: int64
```

```
Columns with >40% missing (akan dihapus): []
```

```
Imputed numeric Kilometers_Driven dengan median=53000.0
```

```
Imputed numeric Mileage dengan median=18.15
```

```
Imputed numeric Engine dengan median=1493.0
```

```
Imputed numeric Power dengan median=97.7
```

```
Imputed numeric Seats dengan median=5.0
```

- Penghapusan Duplikat: Mengidentifikasi dan menghapus baris yang sama persis untuk memastikan setiap sampel unik.
- Pengahpusan Kolom tidak Relevan/Mayoritas Hilang: Kolom yang tidak berguna dan kolom dengan data hilang lebih dari 40%.
- Konversi Tipe Data: Membersihkan dan mengkonversi kolom yang berisi satuan teks menjadi tipe data numerik yang benar.
- Feature Engineering: Membuat fitur baru age dengan mengurangi tahun saat ini dengan kolom year.
- Imputasi Nilai Hilang: Numerik dan Kategorikal.

ENCODING DAN FINALISASI PREPROCESSING

```
cat_cols = df.select_dtypes(include=["object", "category"]).columns.tolist()
print("Kolom kategorikal:", cat_cols)

card_threshold = 10
ohe_cols = [c for c in cat_cols if df[c].nunique() <= card_threshold]
high_card_cols = [c for c in cat_cols if df[c].nunique() > card_threshold]
print("One-hot encoding untuk:", ohe_cols)
print("Kolom high-cardinality:", high_card_cols)

df = pd.get_dummies(df, columns=ohe_cols, drop_first=True)

for c in high_card_cols:
    try:
        df[c] = df[c].astype("category").cat.codes
    except Exception:
        df = df.drop(columns=[c])
        print("Kolom high-card dihapus:", c)

Kolom kategorikal: ['Name', 'Location', 'Fuel_Type', 'Transmission', 'Owner_Type']
One-hot encoding untuk: ['Fuel_Type', 'Transmission', 'Owner_Type']
Kolom high-cardinality: ['Name', 'Location']
```

```
# Verifikasi dan analisis korelasi
print("Shape setelah preprocessing:", df.shape)
print("Missing values tersisa?:")
print(df.isna().sum().sum())

if target_col in df.columns:
    corrs = df.corr()[target_col].sort_values(ascending=False)
    print("\nTop korelasi dengan target:")
    print(corrs.head(10))

Shape setelah preprocessing: (6019, 18)
Missing values tersisa?:
0

Top korelasi dengan target:
Price           1.000000
Power          0.769711
Engine          0.657347
Fuel_Type_Diesel 0.320645
Year            0.305327
Seats            0.052811
Fuel_Type_Electric 0.005534
Unnamed: 0       -0.020275
Owner_Type_Fourth & Above -0.021445
Fuel_Type_LPG      -0.025499
Name: Price, dtype: float64
```

- Identifikasi Kolom Kategori: Mengidentifikasi semua kolom kategorikal.
- One-Hot Encoding: Menerapkan One-Hot Encoding (pd.get_dummies) pada kolom kategorikal dengan kardinalitas rendah (jumlah nilai unik ≤ 10), mengubahnya menjadi kolom biner (0 atau 1) agar bisa diproses model.
- Verifikasi Akhir: Memastikan tidak ada lagi nilai yang hilang (missing values: 0) dan mencetak shape akhir data (misalnya, 6019 baris dan 18 kolom).
- Korelasi Target Final: Mencetak Top Korelasi fitur dengan kolom target (Price), menunjukkan Power dan Engine sebagai prediktor terkuat.

PEMODELAN DAN EVALUASI

```
scaler = StandardScaler()
numeric_cols = X_train.select_dtypes(include=[np.number]).columns.tolist()

X_train_num = pd.DataFrame(scaler.fit_transform(X_train[numeric_cols]), columns=numeric_cols, index=X_train.index)
X_test_num = pd.DataFrame(scaler.transform(X_test[numeric_cols]), columns=numeric_cols, index=X_test.index)

X_train_scaled = X_train.copy()
X_test_scaled = X_test.copy()
X_train_scaled[numeric_cols] = X_train_num
X_test_scaled[numeric_cols] = X_test_num
```

```
lr = LinearRegression()
lr.fit(X_train_scaled, y_train)

coefs = pd.Series(lr.coef_, index=X_train_scaled.columns).sort_values(key=abs, ascending=False)
print("\nKoefisien terbesar:")
print(coefs.head(10))
```

```
Koefisien terbesar:
Power          6.554441
Fuel_Type_Electric  5.820072
Fuel_Type_Petrol   -3.879579
Year           2.997305
Owner_Type_Fourth & Above 2.912887
Transmission_Manual -2.626219
Owner_Type_Third    1.349422
Fuel_Type_LPG      1.148557
Mileage          -1.038822
Fuel_Type_Diesel   -0.902905
dtype: float64
```

```
y_pred = lr.predict(X_test_scaled)

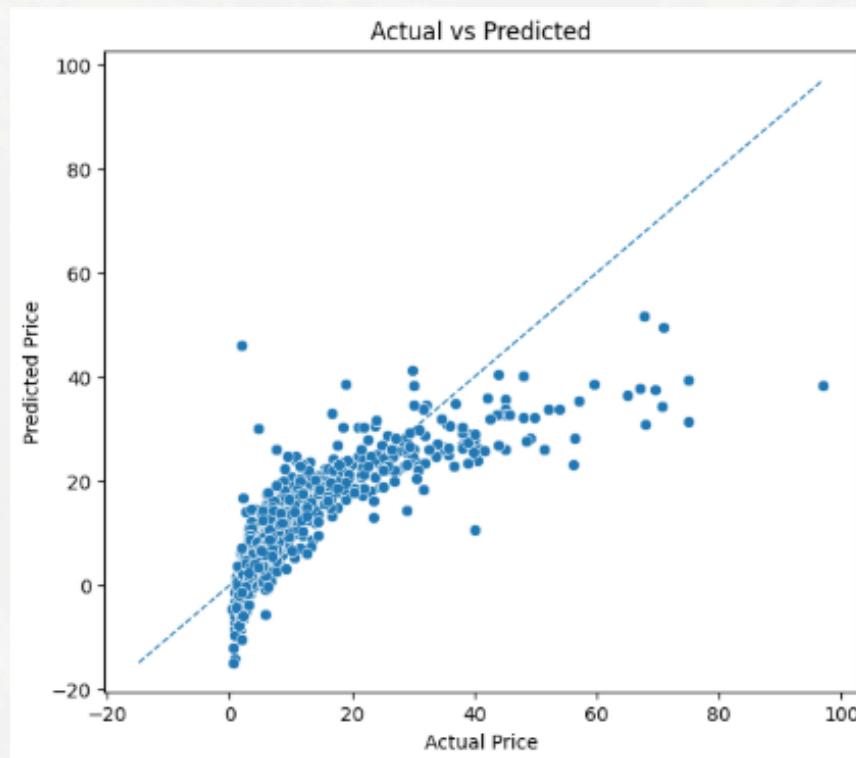
r2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print(f"R2: {r2:.4f}\nMAE: {mae:.4f}\nMSE: {mse:.4f}\nRMSE: {rmse:.4f}")

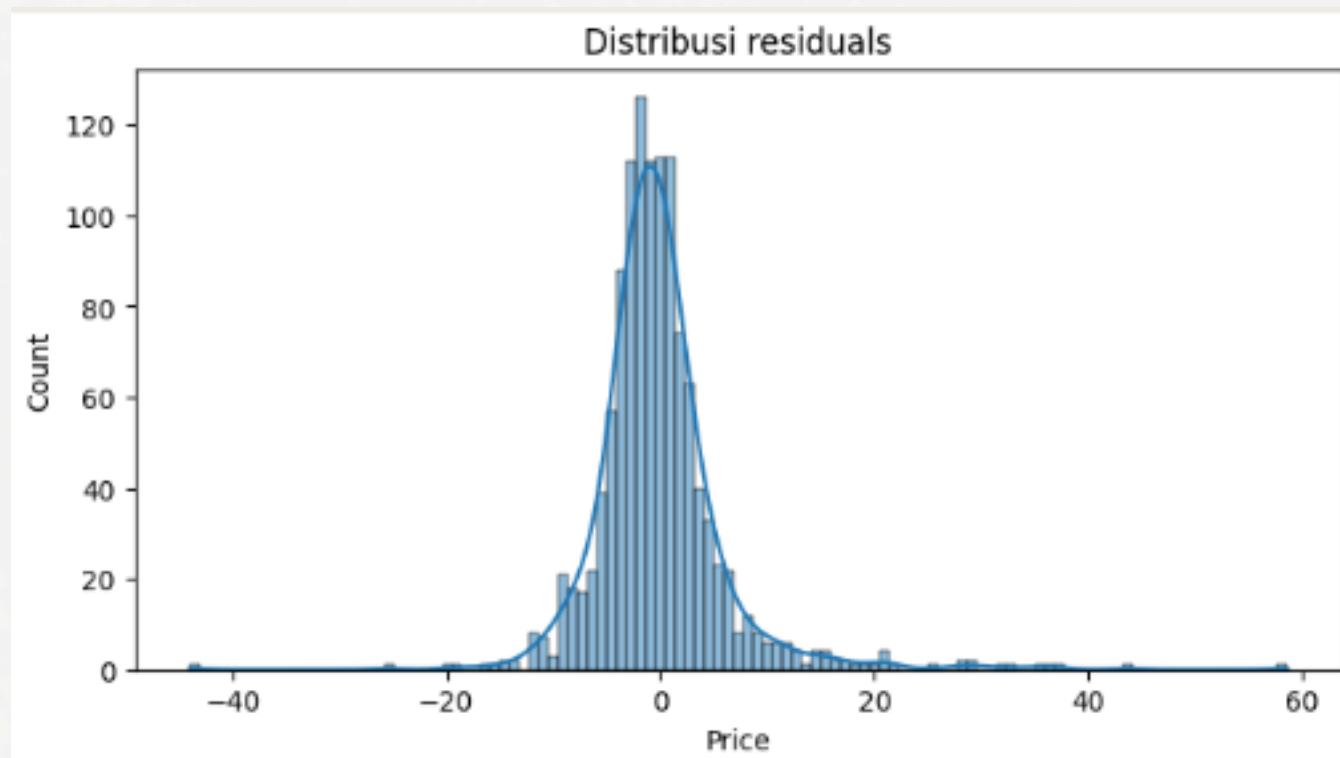
R2: 0.6866
MAE: 3.8584
MSE: 38.5634
RMSE: 6.2099
```

- Penskalalan Fitur: Menggunakan StandardScaler untuk menormalkan fitur numerik, memastikan tidak ada fitur yang mendominasi model karena skalanya.
- Pelatihan Model: Melatih model Regresi Linear (LinearRegression) pada data yang telah disakalakan.
- Koefisien Model: Menampilkan koefisien regresi. Power memiliki koefisien terbesar (6.55441), menunjukkan bahwa setiap peningkatan nilai Power akan memberikan dampak terbesar terhadap harga jual.
- Hasil Metrik Evaluasi: Mencetak kinerja model pada data uji:
R2 (R-squared): 0.6866 → Model dapat menjelaskan 68.66% variasi harga.
RMSE: 6.2099 → Rata-rata kesalahan prediksi harga adalah sekitar 6.21 satuan.

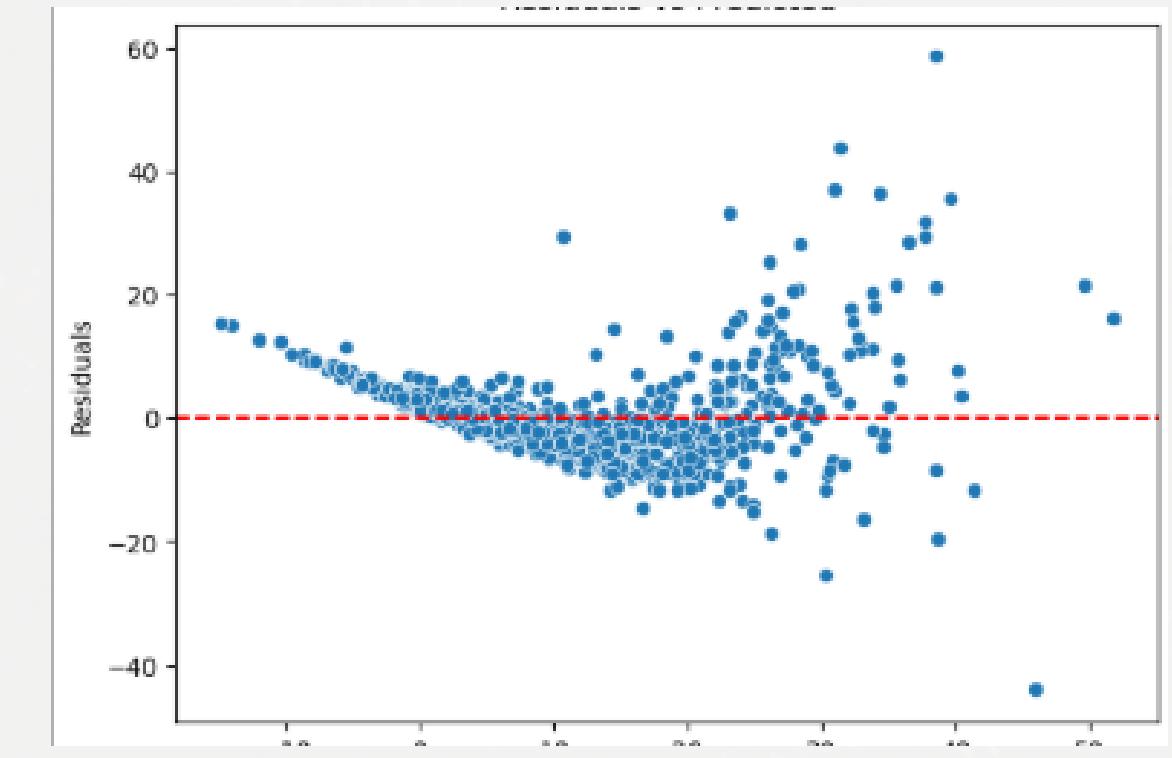
VISUALISASI HASIL



Grafik sebar yang membandingkan Harga Aktual (y_{test}) dengan Harga Prediksi (y_{pred}). Titik yang dekat dengan garis diagonal menunjukkan prediksi yang akurat. Secara visual, model baik pada harga rendah tetapi kurang akurat pada harga sangat tinggi.



Histogram yang menunjukkan bahwa kesalahan prediksi (residual) mendekati distribusi normal berpusat di nol, yang merupakan indikasi yang baik untuk model regresi.



Residuals vs Predicted: Grafik sebar yang menunjukkan penyebaran sisaan terhadap nilai prediksi. Sebaran titik terlihat acak di sekitar garis nol, menunjukkan tidak ada pola sistematis yang terlewat oleh model.

KESIMPULAN

BERDASARKAN HASIL ANALISIS DAN PEMODELAN MENGGUNAKAN REGRESI LINEAR, DAPAT DISIMPULKAN BAHWA FAKTOR-FAKTOR SEPERTI POWER DAN ENGINE MEMILIKI PENGARUH PALING SIGNIFIKAN TERHADAP HARGA MOBIL BEKAS. MODEL YANG DIBANGUN MAMPU MENJELASKAN SEKITAR 68,66% VARIASI HARGA DENGAN NILAI R-SQUARED SEBESAR 0,6866, MENUNJUKKAN KINERJA PREDIKSI YANG CUKUP BAIK. PROSES DATA PREPROCESSING DAN FEATURE ENGINEERING BERPERAN PENTING DALAM MENINGKATKAN AKURASI MODEL DENGAN MEMASTIKAN DATA BERSIH, RELEVAN, DAN SIAP DIGUNAKAN. SECARA KESELURUHAN, MODEL REGRESI LINEAR INI DAPAT MENJADI ALAT BANTU YANG EFEKTIF UNTUK MEMPERKIRAKAN HARGA JUAL MOBIL BEKAS, MESKIPUN PERLU PENGEMBANGAN LEBIH LANJUT DENGAN MENAMBAHKAN VARIABEL LAIN DAN METODE YANG LEBIH KOMPLEKS AGAR HASIL PREDIKSI MENJADI LEBIH AKURAT.

TERIMA KASIH