Data Science Project 5

Predicting Boston Housing Prices

Table of contents

01

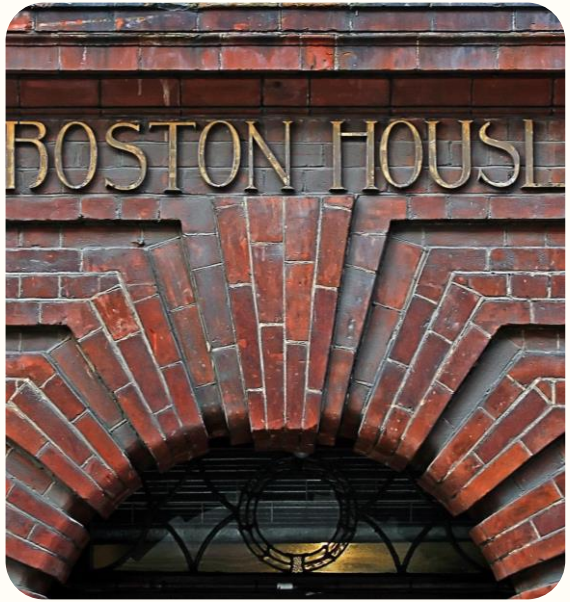
Business Problems

Background

Dalam proyek ini, dilakukan analisis dan pembangunan model prediksi yang kuat berdasarkan data historis untuk akurasi perkirakan harga rumah di Boston.

Model yang telah dilatih dengan data ini kemudian dievaluasi menggunakan metrik seperti R-squared, Mean Absolute Error (MAE), Mean Squared Error (MSE), dan Root Mean Squared Error (RMSE) dan dipilih mana yang memiliki kinerja terbaik, sehingga dapat digunakan untuk memprediksi harga rumah.

Dengan menggunakan model yang telah teruji pemangku kepentingan dapat mengurangi risiko terkait keputusan finansial dan operasional di sektor properti.



Purpose

01

Mengecek korelasi variable yang memiliki pengaruh dengan ‘MEDV’

Mengidentifikasi variabel-variabel yang memiliki pengaruh signifikan terhadap variabel target 'MEDV'.

02

Mengembangkan model prediktif yang dapat memperkirakan nilai ‘MEDV’

Melalui pemilihan dan penyesuaian model-machine learning, seperti XGBoost, Random Forest, atau Linear Regression dan evaluasi model menggunakan metrik seperti R-squared, MAE, MSE, dan RMSE. Model manakah yang baik untuk memprediksi nilai ‘MEDV’.

02

Data Understanding

Dataset Information

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    int64  
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    int64  
 9   TAX      506 non-null    float64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(12), int64(2)
memory usage: 55.5 KB
```

Dataset diperoleh dari:
https://lib.stat.cmu.edu/datasets/boston atau Kaggle.

- Data set terdiri dari 506 baris dan 14 kolom.
- Tidak ada kolom variabel yang memiliki nilai null dan tidak terdapat missing value.
- Dataset terdiri dari tipe data (Dtype) berupa float64 dan int64.

Data Description

MEDV	Nilai median rumah yang ditempati pemiliki dalam $1.000 **(variabel target)**
CRIM	Tingkat kejahatan per kapita di kota.
ZN	Proporsi lahan tinggal yang di-zoning untuk lot lebih dari 25.000 kaki persegi.
INDUS	Proporsi lahan bisnis non-retail per kota.
CHAS	Variabel dummy Charles River (= 1 jika suatu daerah berbatasan dengan sungai; 0 jika tidak).
NOX	Konsentrasi oksida nitrat (bagian per 10 juta).
RM	Jumlah rata-rata kamar per hunian.
AGE	Proporsi unit yang ditempati pemilik yang dibangun sebelum tahun 1940.
DIS	Jarak terbobot ke lima pusat pekerjaan di Boston.

TAX	Tarif pajak properti nilai penuh per $10.000.
PTRATIO	Rasio murid-guru per kota.
B	$1000(Bk - 0{,}63)^2$, di mana Bk adalah proporsi penduduk kulit hitam per kota.
LSTAT	Persentase status sosial rendah dari populasi.

03

Exploratory Data Analysis

Statistik Deskriptif

```
df.describe()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18.455534	356.674032	12.653063	22.532806
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864	7.141062	9.197104
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.730000	5.000000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500	6.950000	17.025000
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000	11.360000	21.200000
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000	16.955000	25.000000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	37.970000	50.000000

- Rata-rata (mean) nilai MEDV adalah sekitar $22.53 ribu, dengan nilai maksimum mencapai $50 ribu.
- Standar deviasi yang relatif tinggi (sekitar $9.20 ribu) menunjukkan variasi yang signifikan dalam nilai MEDV.
- Distribusi nilai MEDV cenderung miring ke kanan, dengan nilai kuartil atas yang lebih tinggi dari kuartil bawah.
- Variabel RM (rata-rata jumlah kamar) dapat menjadi faktor yang signifikan dalam mempengaruhi nilai MEDV, karena rumah dengan lebih banyak kamar umumnya memiliki nilai yang lebih tinggi.

Data Preparation

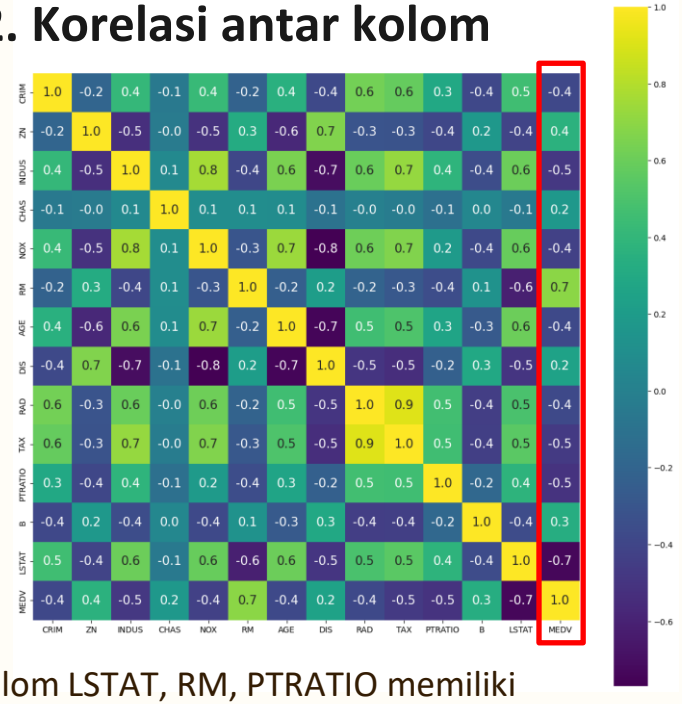
1. Handling Missing Value

```
In [7]: df.isnull().sum()

Out[7]: CRIM       0
        ZN         0
        INDUS      0
        CHAS       0
        NOX        0
        RM         0
        AGE        0
        DIS        0
        RAD        0
        TAX        0
        PTRATIO    0
        B          0
        LSTAT      0
        MEDV       0
        dtype: int64
```

Tidak terdapat missing values pada semua kolom.

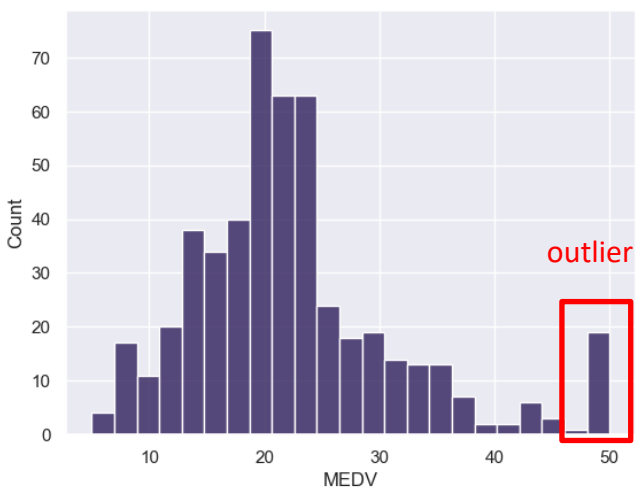
2. Korelasi antar kolom



Kolom LSTAT, RM, PTRATIO memiliki korelasi yang cukup besar terhadap variabel target (MEDV).

Data Visualization

1. Histogram dari kolom target ‘MEDV’ dalam df

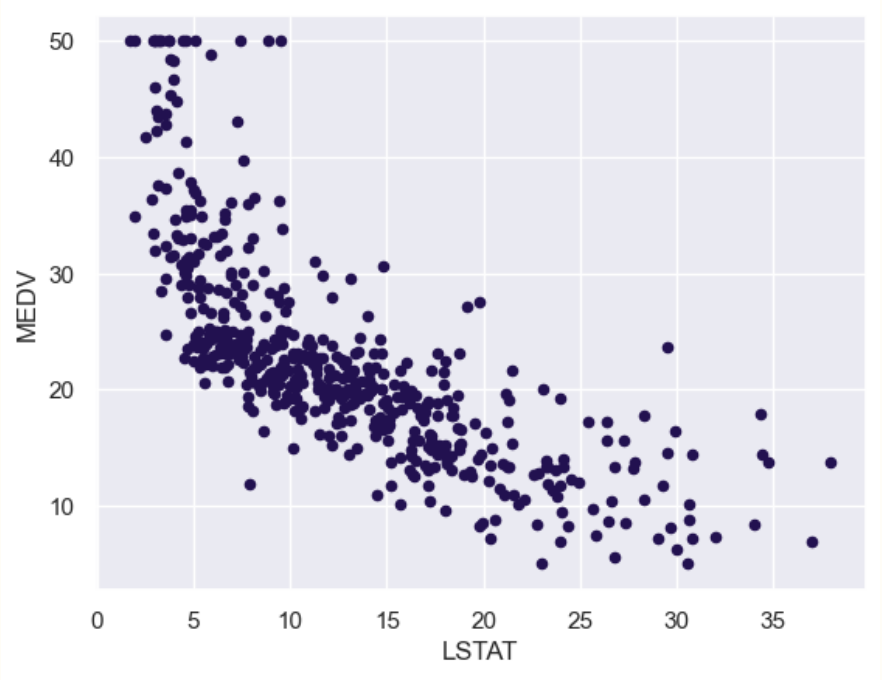



```
#Mengukur skewness dan kurtosis
print(f"Skewness: {df['MEDV'].skew()}")
print(f"Kurtosis: {df['MEDV'].kurt()}")
```

```
Skewness: 1.1080984082549072
Kurtosis: 1.495196944165818
```

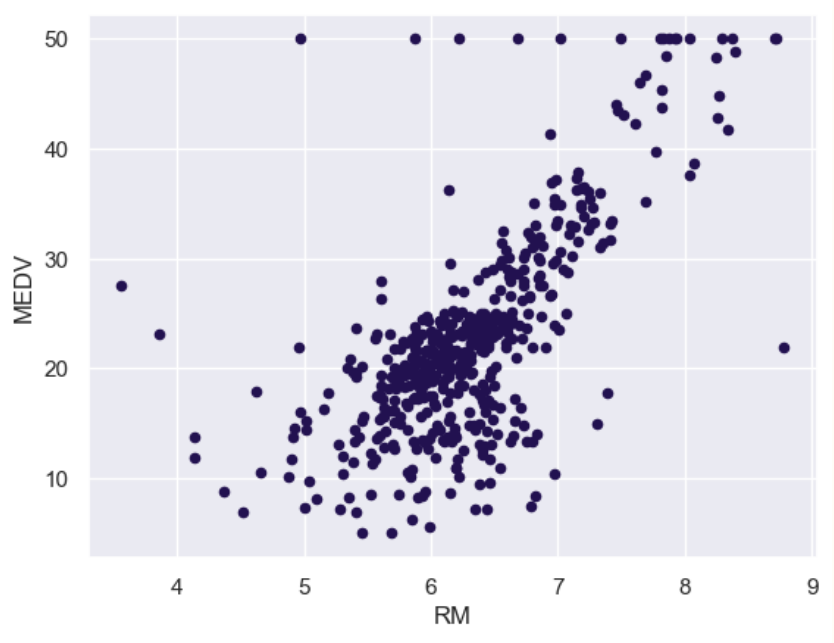
- Nilai 'MEDV' terdistribusi secara normal dengan terdapat sedikit outlier.
- Nilai skewness sekitar 1.108 menunjukkan bahwa distribusi dari variabel 'MEDV’, mengindikasikan bahwa ekor distribusi cenderung lebih panjang di sebelah kanan, dan nilai-nilai yang lebih tinggi dari rata-rata dapat menjadi outliers.
- Nilai kurtosis sekitar 1.495 menunjukkan bahwa distribusi dari variabel 'MEDV' mengindikasikan bahwa distribusi memiliki ekor yang lebih berat dan puncak yang lebih tinggi dibandingkan dengan distribusi normal.

2. Hubungan LSTAT terhadap variabel target ‘MEDV’



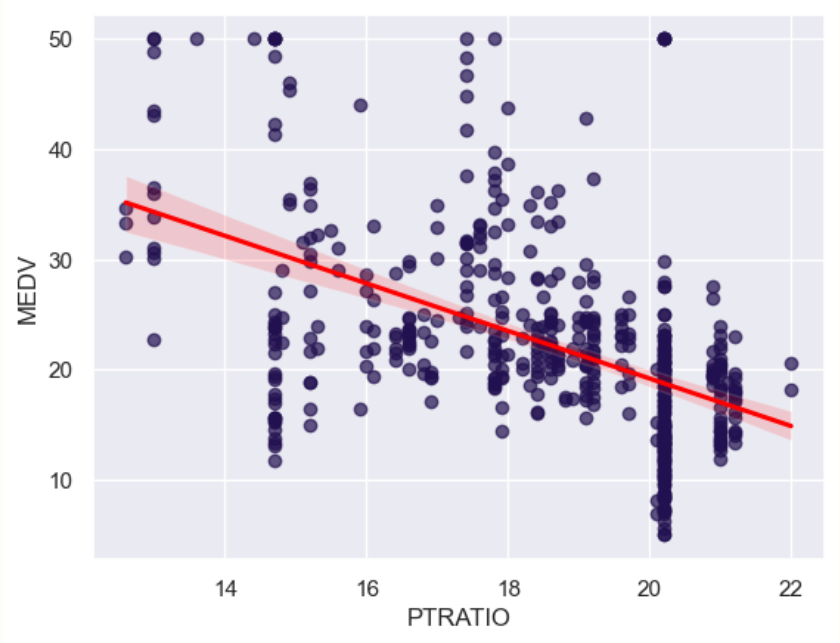
- Variabel LSTAT berkorelasi negatif terhadap variabel target MEDV. Artinya, Ketika semakin rendah LSTAT (persentase status sosial rendah dari populasi) maka maka akan semakin tinggi MEDV (nilai median rumah yang ditempati pemilik dalam $1.000.
- Hal ini mengindikasikan bahwa wilayah-wilayah dengan persentase status sosial rendah yang lebih tinggi cenderung memiliki nilai median rumah yang lebih rendah.

3. Hubungan RM terhadap variabel target ‘MEDV’



- Variabel RM berkorelasi positif dengan variabel target MEDV. Artinya, semakin besar jumlah rata-rata per-hunian (RM) maka akan semakin tinggi nilai median rumah yang ditempati pemilik dalam $1.000 (MEDV).
- Ketika rata-rata jumlah kamar per-hunian meningkat, maka cenderung terjadi peningkatan pada nilai median rumah yang ditempati pemilik. Ini diartikan bahwa rumah dengan lebih banyak kamar cenderung memiliki nilai yang lebih tinggi.

4. Hubungan PTRATIO terhadap variabel target ‘MEDV’

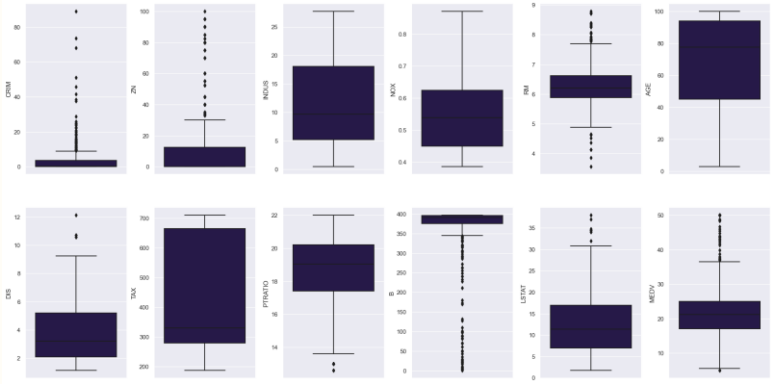


- Variabel PTRATIO berkorelasi negative terhadap variable target MEDV. Artinya, semakin tinggi rasio murid-guru per kota (PTRATIO) maka akan semakin rendah nilai median rumah yang ditempati pemilik dalam $1.000 (MEDV).
- Hal ini dapat terjadi karena rasio murid-guru yang lebih tinggi cenderung memiliki perumahan yang terjangkau. Sehingga nilai rumah MEDV dapat menjadi lebih rendah.

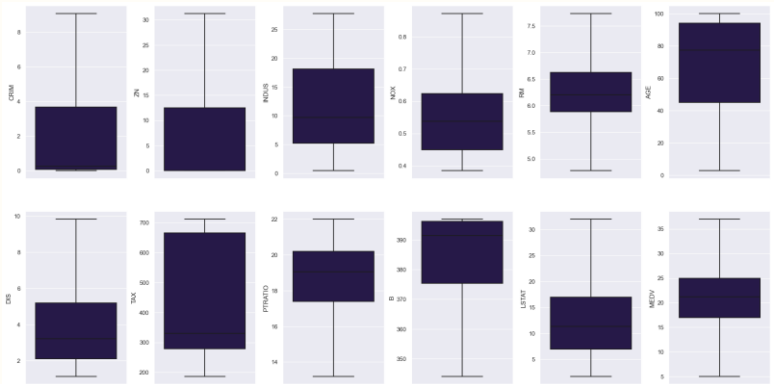
5. Mengelola Outlier pada Data

Karena banyak kolom memiliki outlier, maka pengelolaan outlier dilakukan dengan menerapkan metode IQR pada setiap kolom untuk mengidentifikasi dan mengatasi outlier.

Before:

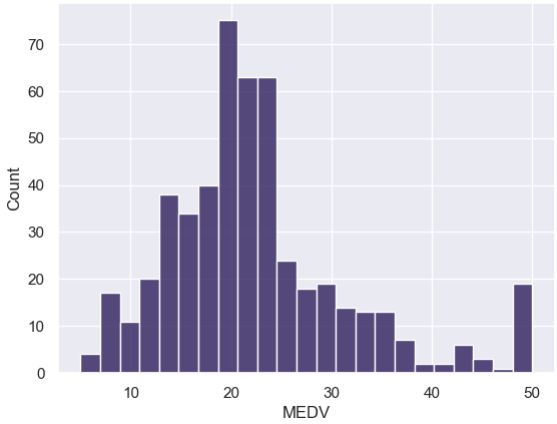


After:



5. Perbandingan setelah pengelolaan outlier

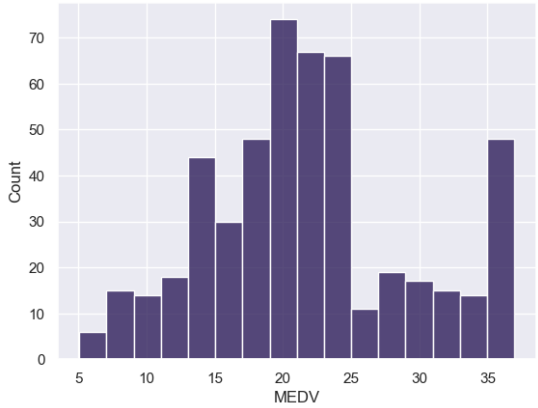
Before:




```
#Mengukur skewness dan kurtosis
print(f"Skewness: {df['MEDV'].skew()}")
print(f"Kurtosis: {df['MEDV'].kurt()}")
```

```
Skewness: 1.1080984082549072
Kurtosis: 1.495196944165818
```

After:




```
#Mengukur skewness dan kurtosis
print(f"Skewness: {data['MEDV'].skew()}")
print(f"Kurtosis: {data['MEDV'].kurt()}")
```

```
Skewness: 0.35361370415925136
Kurtosis: -0.33443614021963786
```

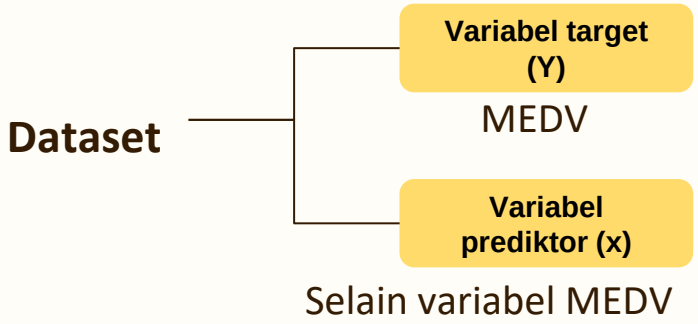
- Pengelolaan outlier telah mengurangi efek kemiringan ke kanan, membuat distribusi yang lebih mendekati simetris.
- Pengelolaan outlier telah mengurangi puncak dan ketebalan ekor distribusi, membuat distribusi lebih mendekati distribusi normal standar.

04

Data Pre-Processing

Splitting Dataset

Variabel dummy ‘CHAS’ dengan nilai 1 dan 0 tetap bertipe data integer, tidak dilakukan pengubahan menjadi tipe data objek (kategorikal) secara eksplisit. Hal ini dikarenakan representasi 1 dan 0 sebagai integer sudah cukup untuk kebutuhan variable dummy biner.




```
X = data.drop(['MEDV'],axis=1)
y = data['MEDV']
```

Variabel Prediktor (X)

```
X
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33
...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.67
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.08
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.64
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.48
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.88

506 rows × 13 columns

Variabel target (Y)

```
y
```

```
0      24.0
1      21.6
2      34.7
3      33.4
4      36.2
       ...
501    22.4
502    20.6
503    23.9
504    22.0
505    11.9
Name: MEDV, Length: 506, dtype: float64
```

- Data train dibagi menjadi 70% dan data set dibagi menjadi 30%

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state=4)
```

X_train

```
X_train
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
84	0.050590	0.00	4.49	0	0.449	6.3890	48.0	4.7794	3	247.0	18.5	396.90000	9.62
354	0.043010	31.25	1.91	0	0.413	5.6630	21.9	9.8208	4	334.0	22.0	382.80000	8.05
221	0.407710	0.00	6.20	1	0.507	6.1640	91.3	3.0480	8	307.0	17.4	395.24000	21.46
34	1.612820	0.00	8.14	0	0.538	6.0960	96.9	3.7598	4	307.0	21.0	344.10625	20.34
267	0.578340	20.00	3.97	0	0.575	7.7305	67.0	2.4216	5	264.0	13.2	384.54000	7.44
...	...	...	...	...	...	...	...	...	...	...	...	...	...
385	9.069639	0.00	18.10	0	0.700	5.2770	98.1	1.4261	24	666.0	20.2	396.90000	30.81
197	0.046660	31.25	1.52	0	0.404	7.1070	36.6	7.3090	2	329.0	13.2	354.31000	8.61
439	9.069639	0.00	18.10	0	0.740	5.6270	93.9	1.8172	24	666.0	20.2	396.90000	22.88
174	0.084470	0.00	4.05	0	0.510	5.8590	68.7	2.7019	5	296.0	16.6	393.23000	9.64
122	0.092990	0.00	25.65	0	0.581	5.9610	92.9	2.0869	2	188.0	19.1	378.09000	17.93

354 rows × 13 columns

Y_train

```
y_train
```

```
84     23.9000
354    18.2000
221    21.7000
34     13.5000
267    36.9625
        ...
385     7.2000
197    30.3000
439    12.8000
174    22.6000
122    20.5000
Name: MEDV, Length: 354, dtype: float64
```

X_test

```
X_test
```

]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
8	0.211240	12.50	7.87	0	0.524	5.6310	100.0	6.0821	5	311.0	15.2	386.63000	29.93
289	0.042970	31.25	5.32	0	0.405	6.5650	22.9	7.3172	6	293.0	16.6	371.72000	9.51
68	0.135540	12.50	6.07	0	0.409	5.5940	36.8	6.4980	4	345.0	18.9	396.90000	13.09
211	0.375780	0.00	10.59	1	0.489	5.4040	88.6	3.6650	4	277.0	18.6	395.24000	23.98
226	0.382140	0.00	6.20	0	0.504	7.7305	86.5	3.2157	8	307.0	17.4	387.38000	3.13
...	...	...	...	...	...	...	...	...	...	...	...	...	...
446	6.288070	0.00	18.10	0	0.740	6.3410	96.4	2.0720	24	666.0	20.2	344.10625	17.79
364	3.474280	0.00	18.10	1	0.718	7.7305	82.9	1.9047	24	666.0	20.2	354.55000	5.29
337	0.030410	0.00	5.19	0	0.515	5.8950	59.6	5.6150	5	224.0	20.2	394.81000	10.56
39	0.027630	31.25	2.95	0	0.428	6.5950	21.8	5.4011	3	252.0	18.3	395.63000	4.32
478	9.069639	0.00	18.10	0	0.614	6.1850	96.7	2.1705	24	666.0	20.2	379.70000	18.03

152 rows × 13 columns

Y_test

```
y_test
```

4]:

```
8       16.5000
289     24.8000
68      17.4000
211     19.3000
226     36.9625
         ...
446     14.9000
364     21.9000
337     18.5000
39      30.8000
478     14.6000
Name: MEDV, Length: 152, dtype: float64
```

05

Data Modeling & Model Evaluation

Metode SPSS

- Statistical package for the Sosial Sciences (SPSS) digunakan untuk melakukan berbagai jenis analisis statistik, termasuk regresi linear.

- Dalam SPSS ini, analisis regresi dilakukan dengan menggunakan metode Ordinary Least Squares (OLS), output yang dihasilkan mencakup berbagai informasi statistik yang berguna untuk mengevaluasi dan menginterpretasi model regresi linear.

```python
#Prediksi secara statistik (menggunakan metode SPSS)
import statsmodels.api as sm

# Split the columns into y and x
y_ = y
X_ = X

# Define the model
X_ = sm.add_constant(X_)
model = sm.OLS(y_, X_)

# Fit the model
result = model.fit()

# Print the model summary
print(result.summary())
```

OLS Regression Results

Dep. Variable:	MEDV	R-squared:	0.788
Model:	OLS	Adj. R-squared:	0.783
Method:	Least Squares	F-statistic:	141.0
Date:	Thu, 04 Jan 2024	Prob (F-statistic):	1.82e-156
Time:	19:48:16	Log-Likelihood:	-1351.1
No. Observations:	506	AIC:	2730.
Df Residuals:	492	BIC:	2789.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	35.0959	5.419	6.476	0.000	24.448	45.744
CRIM	-0.6124	0.146	-4.196	0.000	-0.899	-0.326
ZN	0.0399	0.020	1.953	0.051	-0.000	0.080
INDUS	-0.0191	0.046	-0.415	0.678	-0.110	0.071
CHAS	1.7065	0.644	2.650	0.008	0.441	2.972
NOX	-13.1199	2.876	-4.561	0.000	-18.772	-7.468
RM	2.9924	0.356	8.395	0.000	2.292	3.693
AGE	-0.0089	0.010	-0.897	0.370	-0.028	0.011
DIS	-1.1025	0.154	-7.146	0.000	-1.406	-0.799
RAD	0.3318	0.064	5.155	0.000	0.205	0.458
TAX	-0.0095	0.003	-3.444	0.001	-0.015	-0.004
PTRATIO	-0.7967	0.101	-7.876	0.000	-0.995	-0.598
B	0.0067	0.010	0.700	0.484	-0.012	0.026
LSTAT	-0.4683	0.040	-11.578	0.000	-0.548	-0.389

Omnibus:	76.804	Durbin-Watson:	1.065
Prob(Omnibus):	0.000	Jarque-Bera (JB):	154.152
Skew:	0.851	Prob(JB):	3.36e-34
Kurtosis:	5.102	Cond. No.	2.06e+04

Insight & Informasi:

- **R-squared: M**engukur seberapa baik model dapat menjelaskan variasi dalam data. Nilai 0.788 menunjukkan bahwa sekitar 78.8% variabilitas MEDV dapat dijelaskan oleh model.
- **F-Statistik**: F-statistik (141.0) digunakan untuk menguji signifikansi keseluruhan model.
- **Koefisien regresi**: koefisien positif untuk RM (jumlah kamar) menunjukkan bahwa kenaikan dalam jumlah kamar dikaitkan dengan peningkatan nilai rumah.
- **Uji Heteroskedastisitas**: Durbin-Watson (1.065) dapat digunakan untuk menguji apakah ada autokorelasi dalam residu. Nilai mendekati 2 menunjukkan ketidaksamarataan residual yang lebih rendah.

1. Linear Regression

```
[36]: from sklearn.linear_model import LinearRegression

      # Model Linear Regressor
      lm = LinearRegression()

      # Fit data ke model untuk di Training
      lm.fit(X_train, y_train)
```

```
Out[36]: LinearRegression
         LinearRegression()
```

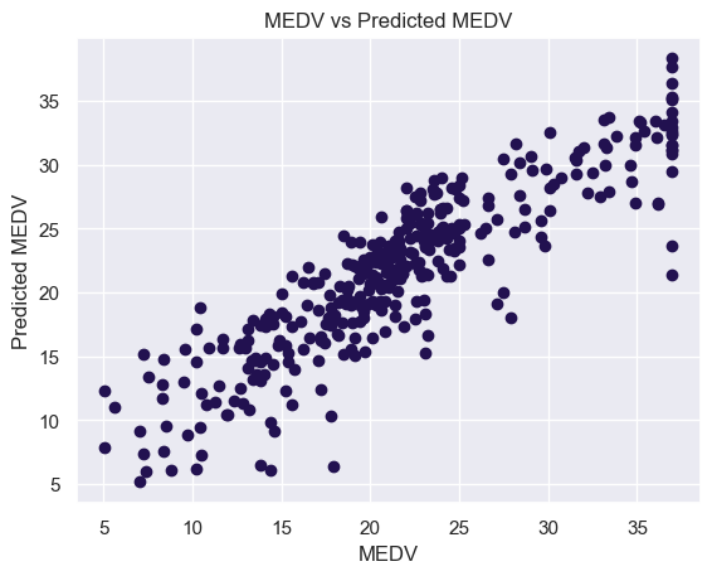
```
[37]: lm.intercept_ #titik potong(intersep atau bias) dari garis regresi.
```

```
Out[37]: 37.72161681582859
```

- Nilai intercept ini menunjukkan prediksi variable dependen ketika semua variable independen setara dengan nol.
- Jika melakukan regressi untuk memprediksi harga rumah berdasarkan beberapa fitur, nilai intercept mungkin mewakili harga dasar rumah ketika semua fitur lainnya tidak ada.

Linear Regression Model Evaluation

- Model evaluation dilakukan untuk mengevaluasi seberapa baik model statistik atau machine learning dapat melakukan prediksi atau klasifikasi pada data uji (test).
- Metode evaluasi yang digunakan yaitu R-Squared, MAE, MSE, dan RMSE.



Model Evaluation Train

```
print("R^2: ", metrics.r2_score(y_train, y_pred))
print("MAE: ", metrics.mean_absolute_error(y_train, y_pred))
print("MSE: ", metrics.mean_squared_error(y_train, y_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_train, y_pred)))
```

```
R^2:  0.7843001397120484
MAE:  2.612682060023276
MSE:  11.804083716350966
RMSE:  3.435707163940339
```

Model Evaluation Test

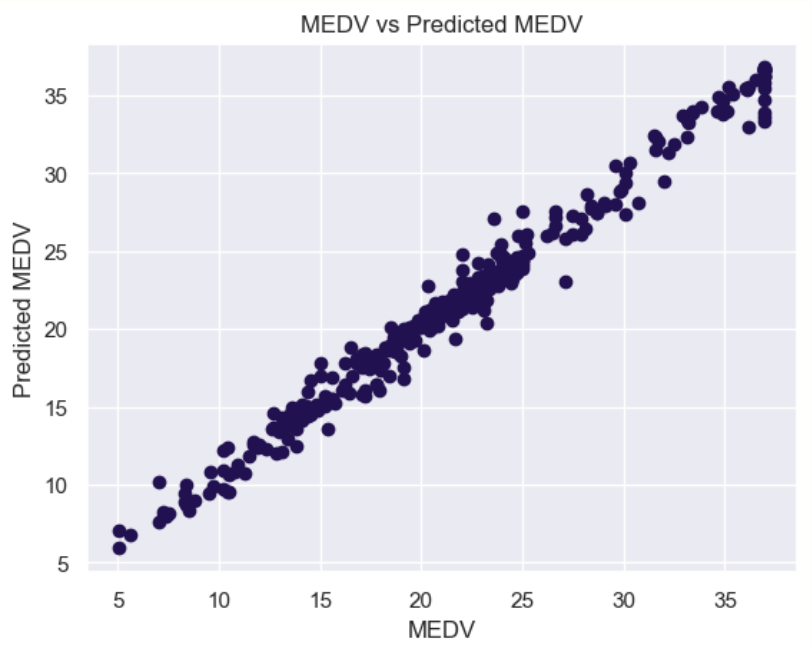
```
# Model Evaluation Test
acc_linreg = metrics.r2_score(y_test, y_pred)
print("R^2: ", acc_linreg)
print("MAE: ", metrics.mean_absolute_error(y_test, y_pred))
print("MSE: ", metrics.mean_squared_error(y_test, y_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
R^2:  0.785864589916547
MAE:  2.775470070332819
MSE:  13.73346448170595
RMSE:  3.705868923978012
```

Perbandingan dari Model Evaluation train dan test:

- Model memiliki kinerja yang serupa baiknya pada data pelatihan dan data uji, karena nilai R-squared pada kedua set datanya hampir sama.
- MAE, MSE, dan RMSE pada data uji cenderung lebih sedikit tinggi dibandingkan dengan data pelatihan, tetapi selisihnya tidak signifikan.
- Model tidak mengalami overfitting dan underfitting yang signifikan, karena kinerjanya relatif stabil pada data pelatihan dan data uji.

Random Forest Regressor



Dengan menggunakan metode evaluasi random forest regressor, titik-titik scatter plot atau sebaran datanya tidak terlalu tersebar seperti pada metode evaluasi dengan linear regression.
Hal ini dikarenakan bahwa metode ini umumnya lebih tahan terhadap outliers karena tidak terlalu dipengaruhi oleh titik-titik ekstrem.

Model Evaluation Train

```
5]: # Model evaluation
print('R^2: ', metrics.r2_score(y_train, y_pred))
print('MAE: ', metrics.mean_absolute_error(y_train, y_pred))
print('MSE: ', metrics.mean_squared_error(y_train, y_pred))
print('RMSE: ',np.sqrt(metrics.mean_squared_error(y_train, y_pred)))
```

```
R^2:  0.9813936464937282
MAE:  0.7363213276836135
MSE:  1.0182248340395461
RMSE:  1.0090712730226474
```

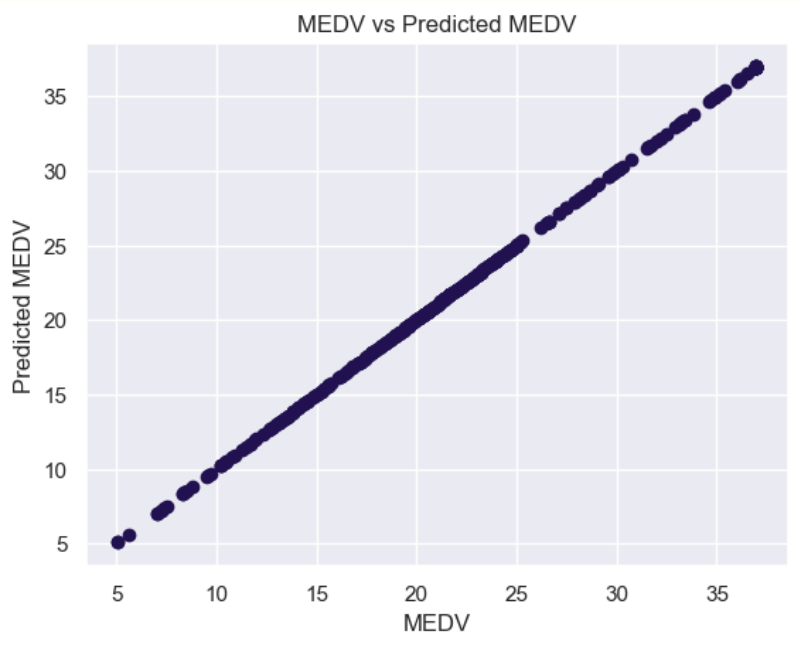
Model Evaluation Test

```
|: #Model evaluation
acc_rf = metrics.r2_score(y_test, y_test_pred)
print('R^2:', acc_rf)
print("MAE: ", metrics.mean_absolute_error(y_test, y_test_pred))
print("MSE: ", metrics. mean_squared_error(y_test, y_test_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_test, y_test_pred)))
```

```
R^2: 0.8585089604507046
MAE:  2.0182006578947327
MSE:  9.074455109374993
RMSE:  3.012383625864241
```

- Model memiliki kinerja yang sangat baik pada data pelatihan, dan kemampuannya untuk menggeneralisasi ke data uji (test) juga cukup baik.
- Meskipun terdapat peningkatan dalam kesalahan prediksi pada data uji, nilai MAE dan RMSE masih dalam rentang yang dapat diterima, terutama jika mempertimbangkan kompleksitas data atau kondisi khusus masalah.

XGBoost Model Evaluation



Titik-titik scatter plot mengikuti garis diagonal (garis 45 derajat) dimana hal ini menunjukkan bahwa prediksi model sangat dekat dengan nilai sebenarnya dan model ini menunjukkan bahwa model dapat membuat prediksi yang baik.

Model Evaluation Train

```
]: print("R^2: ", metrics.r2_score(y_train, y_pred))
   print("MAE: ", metrics.mean_absolute_error(y_train, y_pred))
   print("MSE: ", metrics.mean_squared_error(y_train, y_pred))
   print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_train, y_pred)))
```

```
R^2:  0.9999987170376842
MAE:  0.005757531203792709
MSE:  7.020957065638508e-05
RMSE:  0.008379115147578834
```

Model Evaluation Test

```
acc_xgb = metrics.r2_score(y_test, y_test_pred)
print("R^2: ", acc_xgb)
print("MAE: ", metrics.mean_absolute_error(y_train, y_pred))
print("MSE: ", metrics.mean_squared_error(y_train, y_pred))
print("RMSE: ", np.sqrt(metrics.mean_squared_error(y_train, y_pred)))
```

```
R^2:  0.868618372638429
MAE:  0.005757531203792709
MSE:  7.020957065638508e-05
RMSE:  0.008379115147578834
```

- Model memiliki kinerja yang sangat baik pada data pelatihan dan mampu menggeneralisasi dengan sangat baik ke data uji.
- Kesalahan prediksi yang sangat kecil pada kedua set data menunjukkan bahwa model memiliki tingkat presisi yang sangat tinggi dan dapat diandalkan dalam memprediksi nilai variabel dependen.

DataFrame Models

	Model	R-squared Score
2	XGBoost	86.861837
1	Random Forest	85.850896
0	Linear Regression	78.586459

- XGBoost dan Random Forest menunjukkan model yang baik dibandingkan Linear Regression berdasarkan R-squared score.
- R-squared score yang tinggi pada kedua model tersebut (86.86% untuk XGBoost dan 85.51% untuk Random Forest) menandakan bahwa keduanya memiliki kemampuan yang baik untuk menjelaskan variasi dalam data.

08

Conclusions

01

Mengecek korelasi variable yang memiliki pengaruh dengan ‘MEDV’

Variabel yang cukup bepengaruh terhadap MEDV yaitu persentase status sosial rendah dari populasi (LSTAT), Jumlah rata-rata kamar per hunian (RM), dan Rasio murid-guru per kota (PTRATIO) dengan nilai korelasi sebesar -0.7, 0.7 dan -0.5.

02

Mengembangkan model prediktif yang dapat memperkirakan nilai ‘MEDV’

Setelah dilakukan evaluasi model menggunakan XGBoost, Random Forest, atau Linear Regression dan dilihat berdasarkan R-squared, MAE, MSE, dan RMSE. Model yang baik untuk memprediksi nilai ‘MEDV’ yaitu XGBoost dan Random Forest menunjukkan model yang baik dibandingkan Linear Regression berdasarkan R-squared score (86.86% untuk XGBoost dan 85.51% untuk Random Forest)

Inne Andarini Herdianti S. Si

Data Science Enthusiast

A bachelor of Science degree in Physics was obtained from the Bandung Institute of Technology. I'm eager to dive into the world of data science. Please check out some of the projects I've worked on my GitHub or LinkedIn. I'm excited about the opportunity to bring my skills and enthusiasm for Data Science!

Contact:

https://www.linkedin.com/in/inneandarini/

inneandarinii@gmail.com

https://github.com/inneandarinii

Thanks!