Data Science Project 4

Predicting Housing prices using Advanced Regression Techniques

Table of contents

01 Business Problems

02 Data Understanding

03 Exploratory Data Analysis

04 Data Preprocessing

05 Data Modelling

06 Conclusions

01

Business Problems

Background

Penetapan harga rumah menjadi faktor utama yang memengaruhi keputusan pembelian, penjualan, dan investasi.

Oleh karena itu dilakukan pengembangan model prediksi harga rumah, dengan tujuan meningkatkan akurasi sehingga dapat memberikan pandangan yang lebih akurat bagi para pemangku kepentingan.

Prediksi harga yang akurat memungkinkan para pemangku kepentingan membuat keputusan yang informasional, mengoptimalkan investasi, dan merencanakan strategi pemasaran yang efektif.

Purpose

01

Variabel apa yang memiliki pengaruh terhadap ‘SalePrice’

Bagaimana cara mengecek variabel yang memiliki pengaruh yang cukup besar terhadap variable target ‘SalePrice’.

02

Mengembangkan model prediktif yang dapat memprediksi variable ‘SalePrice’

Dari beberapa model machine learning yang digunakan, model manakah yang memperoleh score tertinggi untuk memprediksi ‘SalePrice’?

03

Memprediksi data rumah baru

Berapa nilai prediksi 'SalePrice' untuk data terbaru rumah dengan menggunakan model prediksi yang telah diterapkan?

02

Data Understanding

Dataset Information

Dataset diperoleh dari: Kaggle

Terdapat 2 dataset yang digunakan untuk memprediksi harga jual rumah di Ames Iowa.

- Data Training yang terdiri dari 1480 baris data dan 81 kolom yang mencakup segala aspek rumah. Data training ini digunakan untuk pelatihan model.
- Data Testing, terdiri dari 1459 baris dengan 80 kolom (tidak menyediakan informasi harga jual, karena variabel ini yang akan diprediksi). Data pengujian ini digunakan untuk mengevaluasi seberapa baik model yang telah dilatih sebelumnya.

03

Exploratory Data Analysis

Data Preparation

1. Pemilihan kolom

```
sorted_correlations = df_train.select_dtypes(include=['float64', 'int64']).corr()['SalePrice'].abs().sort_values(ascending=False)
print(sorted_correlations)
```

```
SalePrice        1.000000
OverallQual      0.790982
GrLivArea        0.708624
GarageCars       0.640409
GarageArea       0.623431
TotalBsmtSF      0.613581
1stFlrSF         0.605852
FullBath         0.560664
TotRmsAbvGrd     0.533723
YearBuilt        0.522897
YearRemodAdd     0.507101
GarageYrBlt      0.486362
```

Kolom yang dipilih adalah kolom yang memiliki pengaruh peluang yang cukup besar terhadap 'SalePrice'.

- Kolom dengan tipe data integer: SalePrice, OverallQual, OverallCond, GrLiv Area, GarageArea, TotalBsmtSF, 1stFlrSF, YearBuilt, dan Lot Area.
- Kolom dengan tipe data object: KitchenQual dan MiscFeature.

Data Description

SalePrice	Harga jual properti dalam dollar **(variabel target)**
OverallQual	Kualitas bahan dan penyelesaian secara keseluruhan (dari 1 – 9).
OverallCond	Rating kondisi rumah secara keseluruhan (dari 1 – 10).
GrLivArea	Luas area tinggal di atas permukaan tanah (sq ft).
TotalBsmtSF	Total luas basement dalam (sq ft).
1stFlrSF	Luas lantai pertama dalam (sq ft).
YearBuilt	Tahun kontruksi dari 1872 hingga 2010.
LotArea	Luas tanah (sq ft).
KitchenQual	Kualitas dapur.
MiscFeature	Fitur lain-lain yang tidak tercakup dalam kategori lain.

Rating OverallQual dan OverallCond

10	Very Excellent
9	Excellent
8	Very Good
7	Good
6	Above Average
5	Average
4	Below Average
3	Fair
2	Poor
1	Very Poor

Rating KitchenQual

Ex	Excellent
Gd	Good
TA	Average/Typical
Fa	Fair
Po	Poor

2. Statistik Deskriptif

```
#Menampilkan statistik deskriptif untuk df_train
df_train.describe()
```

]:

	SalePrice	OverallQual	OverallCond	GrLivArea	GarageArea	TotalBsmtSF	1stFlrSF	YearBuilt	LotArea
count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000
mean	180921.195890	6.099315	5.575342	1515.463699	472.980137	1057.429452	1162.626712	1971.267808	10516.828082
std	79442.502883	1.382997	1.112799	525.480383	213.804841	438.705324	386.587738	30.202904	9981.264932
min	34900.000000	1.000000	1.000000	334.000000	0.000000	0.000000	334.000000	1872.000000	1300.000000
25%	129975.000000	5.000000	5.000000	1129.500000	334.500000	795.750000	882.000000	1954.000000	7553.500000
50%	163000.000000	6.000000	5.000000	1464.000000	480.000000	991.500000	1087.000000	1973.000000	9478.500000
75%	214000.000000	7.000000	6.000000	1776.750000	576.000000	1298.250000	1391.250000	2000.000000	11601.500000
max	755000.000000	10.000000	9.000000	5642.000000	1418.000000	6110.000000	4692.000000	2010.000000	215245.000000

- Rata-rata harga jual rumah adalah sekitar $180,921 dengan variasi yang signifikan dari $34,900 hingga $755,000. Standar deviasi yang relative tinggi ($79,442) menunjukkan variasi yang cukup besar dalam harga jual.
- Rata-rata kualitas keseluruhan rumah (OverallQual) 6,1 dan rata-rata kondisi keseluruhan (OverallCond) sekitar 5,6.

3. Penjelasan setiap kolom

OverallQual

```
df_train.OverallQual.value_counts()
```

```
2]: OverallQual
5      397
6      374
7      319
8      168
4      116
9       43
3       20
10      18
2        3
1        2
Name: count, dtype: int64
```

- Kualitas keseluruhan paling umum adalah 5, dengan 397 rumah memiliki tingkat kualitas tersebut.
- Secara umum, Sebagian rumah memiliki kualitas keseluruhan di kisaran 5 hingga 7, karena jumlahnya cukup signifikan pada kategori-kategori tersebut.

OverallCond

```
df_train.OverallCond.value_counts()
```

```
]: OverallCond
5    821
6    252
7    205
8     72
4     57
3     25
9     22
2      5
1      1
Name: count, dtype: int64
```

- Jumlah rumah dengan kondisi keseluruhan 5 lebih dominan dibandingkan dengan kategori lainnya.
- Mayoritas rumah memiliki kondisi keseluruhan yang dianggap cukup baik atau standar.

YearBuilt

```
df_train.YearBuilt.value_counts()
```

```
YearBuilt
2006    67
2005    64
2004    54
2007    49
2003    45
       ..
1875     1
1911     1
1917     1
1872     1
1905     1
Name: count, Length: 112, dtype: int64
```

- Jumlah rumah yang dibangun pada tahun 2000-an (2000 hingga 2010) cenderung tinggi, menunjukkan adanya pembangunan baru dan modern pada periode tersebut.

- Sebaliknya, rumah yang dibangun pada tahun 1800-an memiliki jumlah yang lebih kecil. Hal ini mungkin mencerminkan karakteristik klasis dan bersejarah dari rumah-rumah tersebut.

KitchenQual

```
df_train.KitchenQual.value_counts()
```

```
KitchenQual
TA    735
Gd    586
Ex    100
Fa     39
Name: count, dtype: int64
```

- Kualitas dapur paling umum adalah "TA" (Average/Typical), dengan 735 rumah memiliki tingkat kualitas tersebut.

- Kualitas dapur paling rendah adalah “Fa” (fair) dengan 39 rumah dan tidak ada rumah yang memiliki kualitas “Po” (poor) dalam penjualan tersebut.

4. Pengecekan Missing Values

```
In [17]: df_train.isna().sum()

Out[17]: SalePrice          0
         OverallQual        0
         OverallCond        0
         KitchenQual        0
         GrLivArea          0
         GarageArea         0
         TotalBsmtSF        0
         1stFlrSF           0
         YearBuilt          0
         MiscFeature     1406
         LotArea            0
         dtype: int64
```

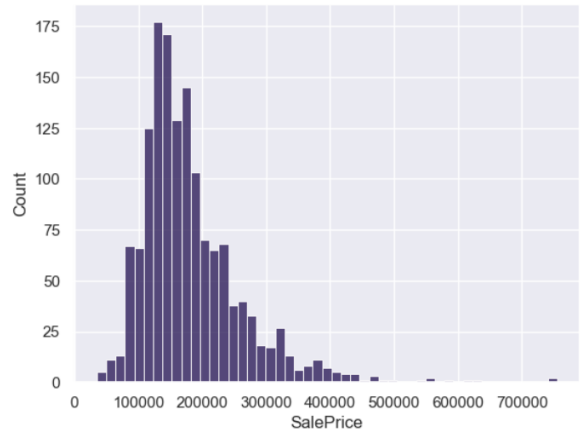
- Tidak terdapat missing value pada beberapa kolom, kecuali kolom ‘MiscFeature’ dengan jumlah missing value 1406.

Data Visualization

1. Histogram dari variabel target ‘SalePrice’ dalam df_train

```
sns.set_theme(palette='magma')
sns.histplot(df_train['SalePrice'])
```

```
8]: <Axes: xlabel='SalePrice', ylabel='Count'>
```

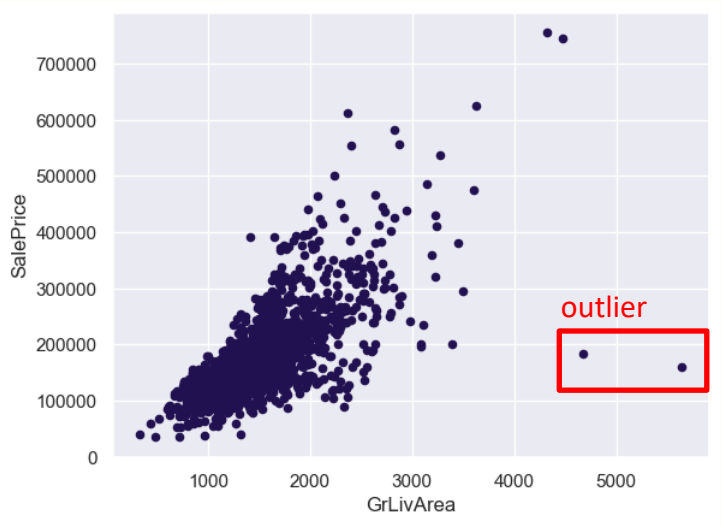



```
#Mengukur skewness dan kurtosis
print(f"Skewness: {df_train['SalePrice'].skew()}")
print(f"Kurtosis: {df_train['SalePrice'].kurt()}")
```

```
Skewness: 1.8828757597682129
Kurtosis: 6.536281860064529
```

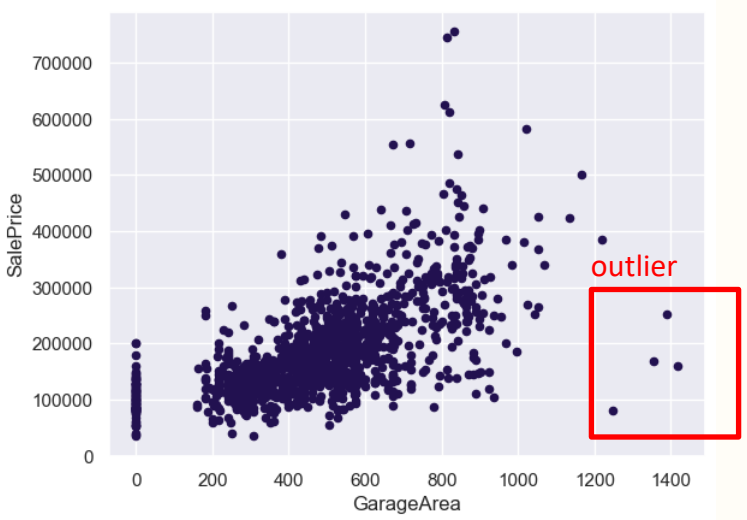
- Nilai skewness positif menunjukkan bahwa Distribusi ‘SalePrice’ memiliki ekor panjang ke kanan. Distribusi ini tidak simetris dan memiliki tail yang lebih Panjang di sisi kanan.
- Nilai kurtosis positif menunjukkan bahwa distribusi ‘SalePrice’ memiliki puncak distribusi (nilai modus) lebih tinggi dibandingkan dengan distribusi normal.

2. Hubungan ‘GrLivArea’ terhadap variabel target ‘SalePrice’



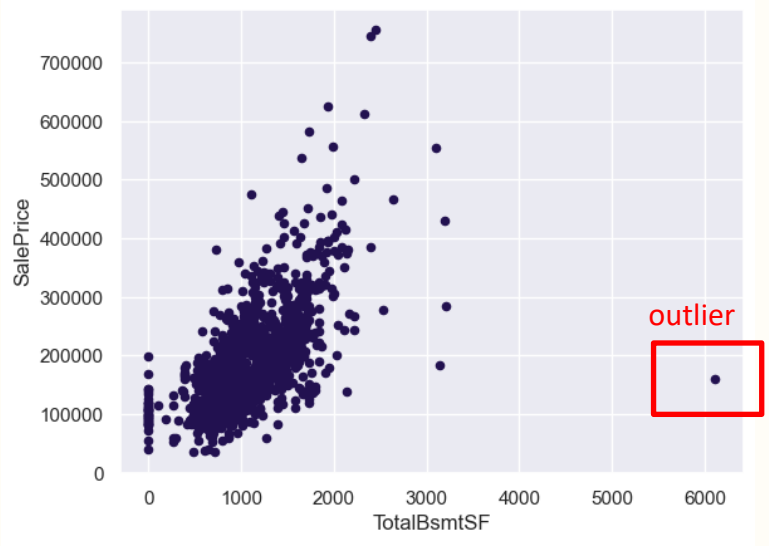
- **Korelasi positif**: seiring dengan meningkatnya luas bangunan di atas permukaan tanah, harga jual properti cenderung meningkat.
- Terdapat outlier, yang mana nilai dari ‘GrLivArea’ yang tinggi, tapi memiliki nilai ‘SalePrice’ yang cukup rendah.

3. Hubungan ‘GarageArea’ terhadap variabel target ‘SalePrice’



- **Korelasi positif**: seiring dengan meningkatnya luas bangunan garasi (GarageArea), harga jual properti cenderung meningkat.
- Luas bangunan garasi tinggi, tetapi harga jualnya relatif rendah, menandakan adanya anomali.

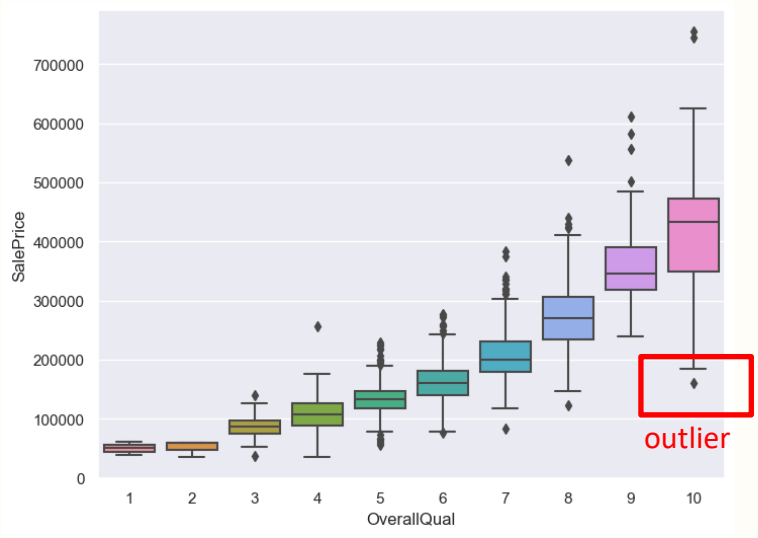
4. Hubungan ‘TotalBsmtSF’ terhadap variabel target ‘SalePrice’



5. Hubungan ‘1stFlrSF’ terhadap variabel target ‘SalePrice’

- **Dari kedua grafik tersebut,** terlihat bahwa total luas basement (TotalBsmtSF) dan total luas lantai pertama (1stFlrSF) memiliki korelasi positif terhadap ‘SalePrice’. Artinya, seiring meningkatnya luas dari kedua kedua variabel tersebut maka maka harga jual rumah juga cenderung meningkat.
- Namun terdapat outlier dimana Ketika total luas basement (TotalBsmrSF) dan total luas lantai pertama (1stFlrSF) tinggi, namun memiliki harga jual rumah yang cukup rendah.

6. Hubungan ‘OverallQual’ terhadap variabel target ‘SalePrice’



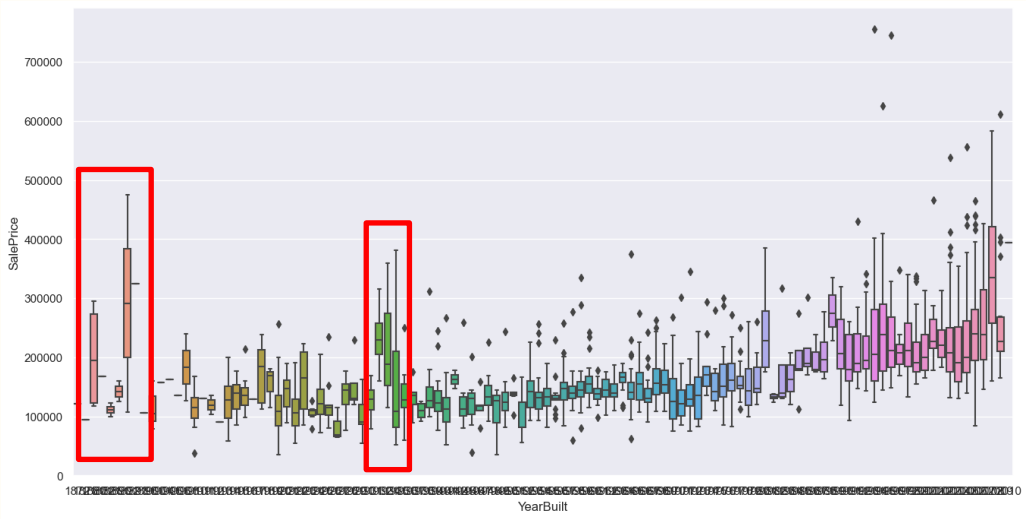
- Semakin tinggi rating OverallQual (kualitas keseluruhan), distribusi harga jual rumah cenderung naik.
- Terdapat dua rumah dengan kualitas keseluruhan (‘OverallQual’) sebesar 10, namun memiliki harga jual (‘SalePrice’) relatif rendah.

```
df_train[((df_train['OverallQual']==10)&(df_train['SalePrice']<200000))]
```

:

	SalePrice	OverallQual	OverallCond	KitchenQual	GrLivArea	GarageArea	TotalBsmtSF	1stFlrSF	YearBuilt	MiscFeature	LotArea
523	184750	10	5	Ex	4676	884	3138	3138	2007	NaN	40094
1298	160000	10	5	Ex	5642	1418	6110	4692	2008	NaN	63887

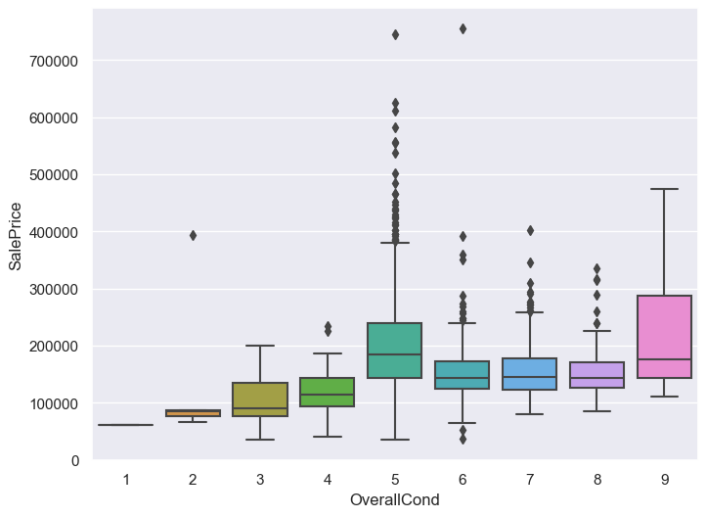
7. Hubungan ‘YearBuilt’ terhadap variabel target ‘SalePrice’



- Tahun awal rumah dibangun adalah tahun 1872.
- Tahun rumah yang paling baru dibandung adalah 2010.

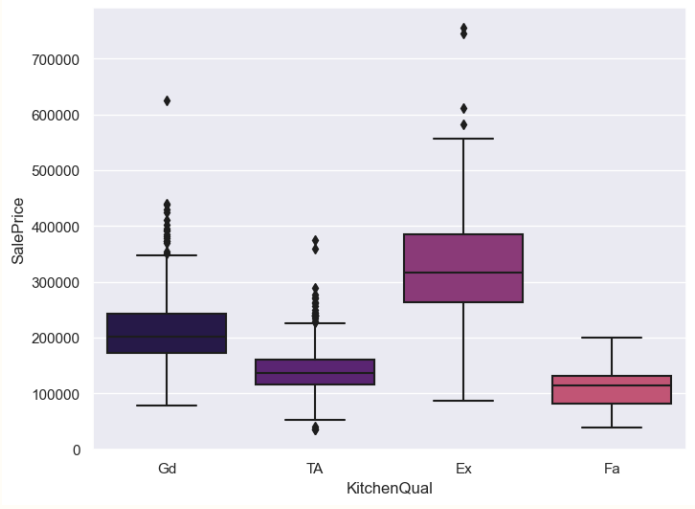
- Umumnya, rumah-rumah yang dibangun pada tahun yang lebih baru cenderung memiliki ‘SalePrice’ yang relatif tinggi. Hal ini sesuai karena properti yang baru atau renovasi biasanya memiliki harga yang lebih tinggi.
- Terdapat juga kasus dimana rumah yang sudah dibangun cukup lama memiliki ‘SalePrice’ yang tinggi. Hal ini terjadi karena rumah-rumah dengan sejarah atau karakteristik khusus dapat memiliki harga jual tinggi meskipun usinya suadah tua.

8. Hubungan ‘OverallCond’ terhadap variabel target ‘SalePrice’



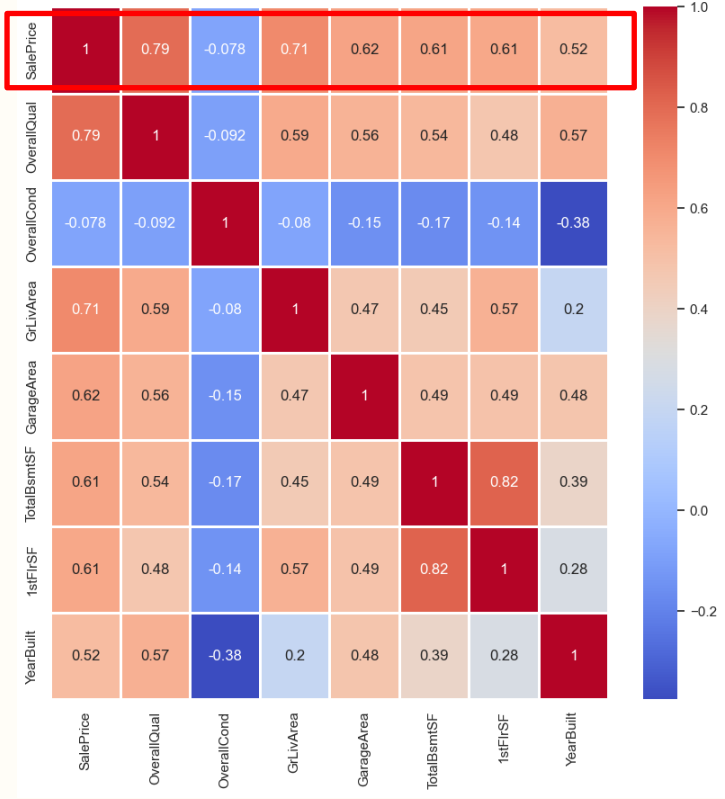
- Rentang kondisi keseluruhan memiliki rentang nilai dari 1 hingga 9, dan sebaliknya.
- Rumah dengan rating 9 cenderung memiliki ‘SalePrice yang tinggi.
- Rumah dengan rating 6, 7, 8 memiliki Distribusi ‘SalePrice’ yang hampir serupa.

9. Hubungan ‘KitchenQual’ terhadap variabel target ‘SalePrice’



- Terdapat hubungan positif antara penilaian kualitas ‘KitchenQual’ terhadap harga jual rumah.
- Rumah dengan penilaian kualitas yang lebih tinggi cenderung memiliki harga jual yang lebih tinggi.

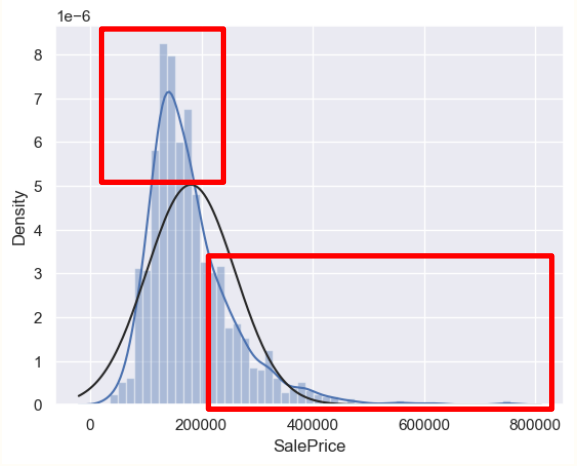
10.Correlation map antar variabel



- Semua kolom berkorelasi positif dengan variable target ‘SalePrice’ kecuali kolom ‘OverallCond’.
- Artinya, semua kolom kecuali kolom ‘OverallCond’ memiliki hubungan yang kuat dengan ‘SalePrice’. Artinya, semakin meningkat nilai variable prediktor maka semakin tinggi harga jual rumah.

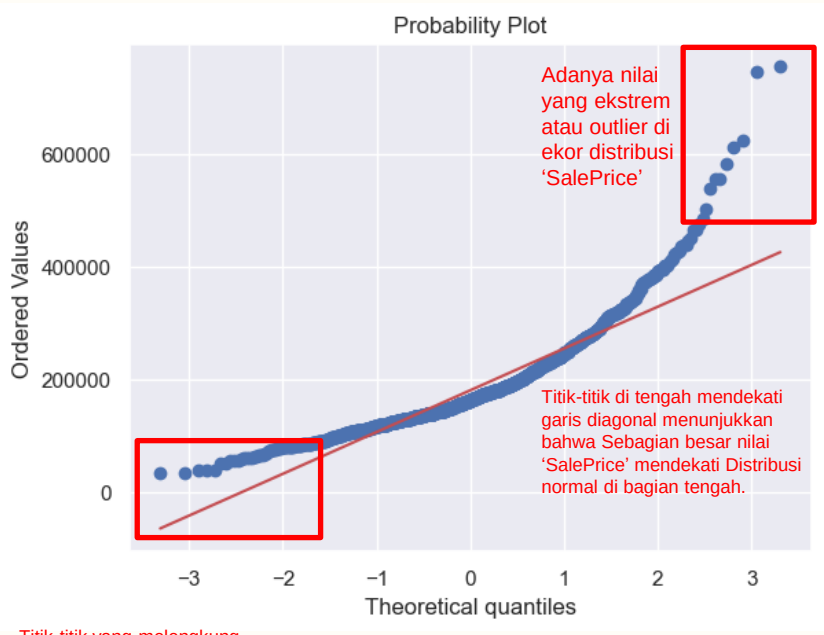
11. Variabel Target ‘SalePrice’

Plot Distribusi ‘SalePrice’ dan membandingkannya dengan distribusi normal.



- Distribusi ‘SalePrice’ tidak mengikuti Distribusi normal secara sempurna. Distribusi ‘SalePrice’ menunjukkan bahwa ekor Distribusi berada di sebelah kanan dan Sebagian besar nilai terletak di sebelah kiri puncak Distribusi. Hal ini dapat diartikan bahwa ada beberapa rumah dengan harga yang sangat tinggi.

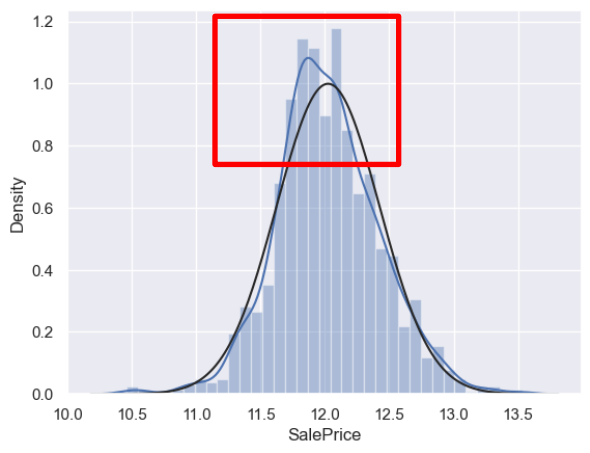
Membuat QQ-Plot dari Distribusi ‘SalePrice’ untuk memeriksa sejauh mana distribusi tersebut sesuai dengan distribusi normal.

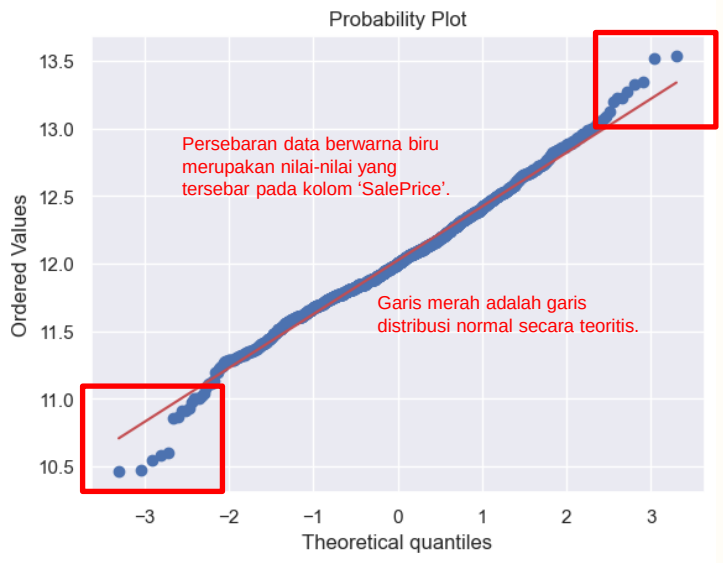


Titik-titik yang melengkung diujung mengindikasikan bahwa ada beberapa rumah denga harga yang sangat tinggi dan rendah.

12. Log Transformation for Variable Target ‘SalePrice’

Log transformation digunakan untuk tujuan mengurangi skewness atau ketidaksimetrisan distribusi dan membuat distribusi menjadi lebih mendekati distribusi normal.





- Garis hitam merupakan garis distribusi normal.
- Garis biru merupakan garis distribusi ‘SalePrice’ yang sudah dilakukan fungsi logaritma natural. Dengan menggunakan logaritma natural, perubahan yang signifikan pada distribusi ‘SalePrice’ dapat terlihat.
- Persebaran data tetap terjaga setelah dilakukan log transformation. Artinya, nilai-nilai ekstrem atau outlier masih tetap ada, tetapi distribusinya menjadi lebih stabil atu mendekati normal.

04

Data Pre-Processing

Data Cleaning

Menggabungkan Train dan Test Sets

```
ntrain = df_train_tf.shape[0]
ntest = df_test.shape[0]
y_train = df_train_tf.SalePrice.values
all_data = pd.concat((df_train_tf, df_test)).reset_index(drop=True)
all_data.drop(['SalePrice'], axis=1, inplace=True)
print("all_data size is : {}".format(all_data.shape))
```

```
all_data size is : (2919, 10)
```

```
print(ntrain, ntest)
```

```
1460 1459
```

- Data yang dihasilkan dari penggabungan kedua set (training dan test) memiliki 2919 baris dan 10 kolom.
- Dengan jumlah data awalnya terdiri dari 1460 baris untuk data training dan 1459 baris untuk data uji (test).

```
total = all_data.isnull().sum().sort_values(ascending=False)
percent = (all_data.isnull().sum()/all_data.isnull().count() * 100).sort_values(ascending=False)
missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
missing_data
```

]:

	Total	Percent
MiscFeature	2814	96.402878
KitchenQual	1	0.034258
GarageArea	1	0.034258
TotalBsmtSF	1	0.034258
OverallQual	0	0.000000
OverallCond	0	0.000000
GrLivArea	0	0.000000
1stFlrSF	0	0.000000
YearBuilt	0	0.000000
LotArea	0	0.000000

- Terdapat missing values pada kolom ‘MiscFeature’ sebanyak 2814.
- Kolom ‘KitchenQual’, ‘TotalBsmtSF’, dan ‘GarageArea’ memiliki missing value masing-masing sebanyak 1.

MiscFeature

- Missing value diisi dengan ‘None’ dengan asumsi bahwa suatu rumah tidak memiliki fitur tambahan.

```
all_data['MiscFeature'] = all_data['MiscFeature'].fillna('None')
```

```
all_data['MiscFeature'].value_counts()
```

```
2]: MiscFeature
    None    2814
    Shed      95
    Gar2       5
    Othr       4
    TenC       1
    Name: count, dtype: int64
```

GarageArea dan TotalBsmtSF

- Missing value diisi dengan ‘0’ dengan asumsi bahwa suatu rumah tidak memiliki garasi dan basement.

```
all_data['GarageArea'] = all_data['GarageArea'].fillna(0)
```

```
all_data['TotalBsmtSF'] = all_data['TotalBsmtSF'].fillna(0)
```

KitchenQual

- Missing value diisi dengan kualitas dapur (variable kategorikal) yang paling sering muncul. Hal ini dilakukan karena asumsi bahwa setiap rumah pasti memiliki dapur.

```
all_data['KitchenQual'].value_counts()
```

```
]: KitchenQual
   TA    1492
   Gd    1151
   Ex     205
   Fa      70
   Name: count, dtype: int64
```

```
#Mengisikan missing value dengan kualitas dapur yang paling sering muncul
all_data['KitchenQual'] = all_data['KitchenQual'].fillna(all_data['KitchenQual'].mode()[0])
```

OverallQual dan OveralCond

- Karena tipe data ‘OverallQual’ sebenarnya bukan integer tetapi lebih tepatnya adalah kategorikal ordinal. Sehingga perlu dilakukan perubahan sebagai berikut.

```
#Changing OverallQual and OverallCond into a categorical variable
#tipe data dari interger to categorical ordinal
all_data['OverallQual'] = all_data['OverallQual'].astype(str)
all_data['OverallCond'] = all_data['OverallCond'].astype(str)
```

Data Transformation

1. Feature Engineering

	OverallQual	OverallCond	KitchenQual
0	7	4	2
1	6	7	3
2	7	4	2
3	7	4	2
4	8	4	2
...	...	...	...
2914	4	6	3
2915	4	4	3
2916	5	6	3
2917	5	4	3
2918	7	4	3

Label Encoding

Label encoding dilakukan untuk mengubah kolom kategorikal (‘OverallQual’, ‘OverallCond’, ‘KitchenQual’) menjadi kolom numerik sehingga dapat disesuaikan dengan model machine learning.

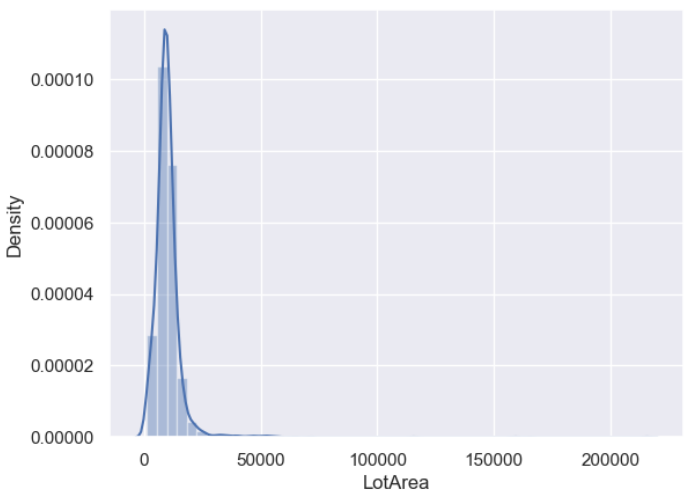
Terdapat 2919 baris data dan 10 kolom.

Skewness Treatment untuk Features

Hal ini dilakukan untuk memahami sejauh mana distribusi fitur-fitur numerik tersebut asimetris.

Skewness in numerical features:

	Skew
LotArea	12.822431
1stFlrSF	1.469604
GrLivArea	1.269358
TotalBsmtSF	1.156894
OverallCond	0.570312
GarageArea	0.239257
OverallQual	-0.326653
YearBuilt	-0.599806
KitchenQual	-1.448023



- LotArea, 1stFlrSF, GrLivArea, dan TotalBsmtSF menunjukkan kecenderungan terhadap nilai-nilai ekstrem yang lebih tinggi, mungkin karena beberapa rumah memiliki luas lot, lantai pertama, dan area basement yang lebih besar.
- Fitur-fitur seperti OverallCond, GarageArea, OverallQual, YearBuilt, dan KitchenQual menunjukkan sedikit asimetri di sebelah kanan atau kiri pada distribusinya.

Terdapat skewness positif yang tingi (12.82) menunjukkan bahwa distribusi data LotArea cenderung memiliki ekor panjang di sebelah kanan. Hal ini dapat disebabkan terdapat beberapa rumah dengan ukuran lot yang sangat besar.

Box Cox Transformation

Ada 5 fitur numerik yang cenderung memiliki skewness (ketidaksimetrian) dalam distribusinya, sehingga dilakukan transformasi Box-Cox pada fitur [‘LotArea’, ‘1stFlrSF’, ‘GrLivArea’, ‘TotalBsmtSF’, ‘KitchenQual’].

```
from scipy.special import boxcox1p
skewed_features = skewness.index
lam = 0.20 # lambda
for feat in skewed_features:
    tmp_data[feat] = boxcox1p(tmp_data[feat], lam)
```

```
tmp_data
```

0]:

	OverallQual	OverallCond	KitchenQual	GrLivArea	GarageArea	TotalBsmtSF	1stFlrSF	YearBuilt	MiscFeature	LotArea
0	7	4	1.228655	17.162564	548.0	14.300394	14.300394	2003	None	25.503637
1	6	7	1.597540	15.856944	460.0	15.856944	15.856944	1976	None	26.291998
2	7	4	1.228655	17.356042	608.0	14.580417	14.580417	2001	None	27.300424
3	7	4	1.228655	17.180669	642.0	13.827349	14.751724	1915	None	26.259338
4	8	4	1.228655	18.303173	836.0	15.455351	15.455351	2000	None	28.868815
...	...	...	...	...	...	...	...	...	...	...
2914	4	6	1.597540	15.262547	0.0	12.642798	12.642798	1970	None	17.719351
2915	4	4	1.597540	15.262547	286.0	12.642798	12.642798	1970	None	17.619961
2916	5	6	1.597540	15.729901	576.0	15.729901	15.729901	1960	None	31.239346
2917	5	4	1.597540	14.788544	0.0	14.546282	14.788544	1992	Shed	26.821947
2918	7	4	1.597540	17.867539	650.0	14.893401	14.893401	1993	None	26.309578

2919 rows × 10 columns

Transformasi Box-Cox digunakan untuk merubah distribusi data menjadi lebih normal atau simetris, sehingga dapat meningkatkan performa model dengan mengurangi skewness dan membuat distribusi data menjadi sesuai dengan asumsi normalitas.

Skewness after Box Cox Transformation

Setelah dilakukan transformasi Box-Cox pada data, terlihat bahwa skewness pada fitur-fitur numerik mengalami perubahan.

Before:

Skewness in numerical features:

	Skew
LotArea	12.822431
1stFlrSF	1.469604
GrLivArea	1.269358
TotalBsmtSF	1.156894
OverallCond	0.570312
GarageArea	0.239257
OverallQual	-0.326653
YearBuilt	-0.599806
KitchenQual	-1.448023

After:

Skew in numerical features:

	Skew
OverallCond	0.570312
LotArea	0.496692
1stFlrSF	0.278546
GarageArea	0.239257
GrLivArea	0.230000
OverallQual	-0.326653
YearBuilt	-0.599806
KitchenQual	-2.156088
TotalBsmtSF	-3.555842

- Transformasi Box-Cox berhasil mengurangi skewness pada kolom ‘LotArea’, distribusi data lebih simetris dan mendekati distribusi normal.
- Nilai skewness pada fitur KitchenQual semakin meningkat setelah dilakukan transformasi Box-Cox.
- Beberapa fitur seperti OverallCond, GarageArea, ‘YearBuilt’ dan OverallQual memiliki skewness yang relatif rendah sebelum transformasi, dan perubahan setelah transformasi tidak terlalu signifikan.

One-Hot Encoding

One-Hot encoding digunakan untuk mengatasi fitur-fitur kategorikal dalam model machine learning yang membutuhkan imput numerik.

MiscFeature_None	MiscFeature_Othr	MiscFeature_Shed	MiscFeature_TenC
True	False	False	False
True	False	False	False
True	False	False	False
True	False	False	False
True	False	False	False
...	...	...	...
True	False	False	False
True	False	False	False
True	False	False	False
False	False	True	False
True	False	False	False

Robust Scaling

Robust scaler diterapkan pada fitur-fitur numerik di dataset sebelum dimasukkan ke dalam model machine learning. Hal ini dilakukan untuk membantu model menjadi lebih tahan terhadap fluktuasi ekstrem dalam data dan meningkatkan kestabilan dan kinerja model.

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.5	0.0	-1.0	0.395595	0.265625	-0.285421	-0.495618	0.631579	-0.254874	0.0	0.0	0.0	0.0
1	0.0	3.0	0.0	-0.305745	-0.078125	0.500909	0.338441	0.063158	0.035514	0.0	0.0	0.0	0.0
2	0.5	0.0	-1.0	0.499525	0.500000	-0.143960	-0.345571	0.589474	0.406961	0.0	0.0	0.0	0.0
3	0.5	0.0	-1.0	0.405320	0.632812	-0.524391	-0.253778	-1.221053	0.023483	0.0	0.0	0.0	0.0
4	1.0	0.0	-1.0	1.008295	1.390625	0.298034	0.123252	0.568421	0.984668	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2914	-1.0	2.0	0.0	-0.625037	-1.875000	-1.122796	-1.383820	-0.063158	-3.122167	0.0	0.0	0.0	0.0
2915	-1.0	0.0	0.0	-0.625037	-0.757812	-1.122796	-1.383820	-0.063158	-3.158777	0.0	0.0	0.0	0.0
2916	-0.5	2.0	0.0	-0.373989	0.375000	0.436730	0.270366	-0.273684	1.857838	0.0	0.0	0.0	0.0
2917	-0.5	0.0	0.0	-0.879657	-1.875000	-0.161204	-0.234048	0.400000	0.230717	-1.0	0.0	1.0	0.0
2918	0.5	0.0	0.0	0.774286	0.664062	0.014152	-0.177862	0.421053	0.041989	0.0	0.0	0.0	0.0

2919 rows × 13 columns

05

Data Modelling

Kfold and Evaluation Score

- Dataset dibagi menjadi 4 lipatan (n_splits=4) dan melakukan evaluasi pada setiap lipatannya.
- Pengacakan data diaktifkan untuk memastikan bahwa setiap lipatan memiliki distribusi yang seragam dan mengurangi potensi bias yang mungkin muncul akibat urutan data yang spesifik.
- Skor evaluasi R-squared digunakan sebagai metrik untuk mengukur sejauh mana model dapat menjelaskan variabilitas dalam data target.

```
from sklearn.model_selection import KFold, cross_val_score
from sklearn.metrics import make_scorer, r2_score

# fungsi untuk melakukan cross validation
def test_model(model, X_train=X_train, y_train=y_train):
    cv = KFold(n_splits = 4, shuffle=True, random_state = 45)
    r2 = make_scorer(r2_score)

    r2_val_score = cross_val_score(model, X_train, y_train, cv=cv, scoring = r2)
    score = [r2_val_score.mean()]
    return score
```

Linear Regression


```
from sklearn.model_selection import KFold, cross_val_score
from sklearn.metrics import make_scorer, r2_score

# fungsi untuk melakukan cross validation
def test_model(model, X_train=X_train, y_train=y_train):
    cv = KFold(n_splits = 4, shuffle=True, random_state = 45)
    r2 = make_scorer(r2_score)

    r2_val_score = cross_val_score(model, X_train, y_train, cv=cv, scoring = r2)
    score = [r2_val_score.mean()]
    return score
```


- Skor evaluasi R-squared yang diperoleh dari pengujian model Linear Regression adalah sekitar 0.844.
- Nilai R-squared tersebut mengindikasikan bahwa model Linear Regression mampu menjelaskan sekitar 84.4% variabilitas dalam data target (SalePrice) pada dataset yang digunakan untuk cross-validation.

Lasso Regression

```
lasso = linear_model.Lasso(alpha=1e-4)
test_model(lasso)
```

```
[0.8445830384045463]
```

Support Vector Machine

- Skor evaluasi R-squared yang diperoleh dari pengujian model Support Vector Regression (SVR) dengan kernel radial basis function (RBF) adalah sekitar 0.844.
- Nilai R-squared ini mengindikasikan bahwa model SVR dengan kernel RBF mampu menjelaskan sekitar 84.4% variabilitas dalam data target (SalePrice) pada dataset yang digunakan untuk cross-validation.

```
from sklearn.svm import SVR
svr_reg = SVR(kernel='rbf')
test_model(svr_reg)
```

```
]: [0.8440642654699704]
```

XGBoost

- Skor evaluasi R-squared yang diperoleh dari pengujian model XGBoost Regressor adalah sekitar 0.841.
- Nilai R-squared ini mengindikasikan bahwa model XGBoost mampu menjelaskan sekitar 84.1% variabilitas dalam data target (SalePrice) pada dataset yang digunakan untuk cross-validation.

```
import xgboost

xgb_reg = xgboost.XGBRegressor()
test_model(xgb_reg)
```

```
5]: [0.8408284419979712]
```

Prediksi data baru dengan SVR

Karena Lasso regression memiliki kekurangan, salah satunya yaitu “Sensitivitas terhadap skala variable: lasso sangat sensitif terhadap skala variabel. Penting untuk melakukan normalisasi atau penskalaan variabel sebelum menerapkan lasso untuk memastikan bahwa semua variabel memiliki dampak yang seimbang pada model.“
Oleh karena itu, data baru untuk memprediksi ‘SalePrice’ diprediksi dengan metode SVR.

Input data baru:

```
5]:  # Input data baru
     data_baru = {'LotArea': [8000],
                  'YearBuilt':[2010],
                  'OverallQual':['8'],
                  'OverallCond':['7'],
                  'GrLivArea':[2000],
                 'MiscFeature':['None'],
                  'GarageArea': [500],
                  'KitchenQual': ['TA'],
                 '1stFlrSF' : [856],
                 'TotalBsmtSF': [856]}
```

Harga rumah ‘SalePrice’ hasil prediksi

```
In [107]:  #Harga rumah hasil prediksi
           y = np.expm1(model.predict(tmp_scaled))
           print(y)

           [166948.04240376]
```

Ketika beberapa fitur-fitur input data baru dimasukkan, hasil prediksi harga rumah berdasarkan model machine learning menunjukkan skor tertinggi (SVM), yakni sebesar $166,948.

06

Conclusions

01

Variabel yang memiliki pengaruh terhadap ‘SalePrice’

Variabel yang memiliki pengaruh yang cukup besar terhadap variable target ‘SalePrice’ adalah GrLivArea, GarageArea, TotalBsmtSF, 1stFlrSF, OverallQual, YearBuilt, OveralCond, dan KitchenQual. Semua kolom berkorelasi positif dengan variable target ‘SalePrice’ kecuali kolom ‘OverallCond’.

02

Mengembangkan model prediktif yang dapat memprediksi variable ‘SalePrice’

Model prediktif yang digunakan yaitu linear regression, lasso regression, support vector machine dan XGBoost. Model dengan mengahasilkan skor tertinggi yaitu Lasso Regression dan Support Vector Machine dengan skor keduanya sebesar 0.844.

03

Memprediksi data rumah baru

Data rumah baru diprediksi dengan metode SVR dimana Ketika beberapa fitur diinputkan ke dalam data-baru menghasilkan ‘SalePrice’ sebesar $166,948.

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!