

Multiple Linear Regression

- Kali ini Ucup mencoba untuk membantu temenya yaitu Joko untuk memprediksi harga rumah untuknya.
 - Joko sendiri tinggal di US tepatnya di King County dan sekarang sedang mencari rumah karena dia baru saja menikah.
 - Data diambil dari kaggle dengan sedikit modifikasi.
 - Joko sendiri ingin membeli rumah dengan jumlah kamar tidur itu 3, jumlah kamar mandinya itu 2, luas rumahnya itu 1800sqft, grade rumahnya 7 dan tahun pembangunannya pada tahun 1990.
 - Yuk bantu Ucup membangun model machine learning untuk membantu joko!
- Langkah Pengerjaan hampir sama dengan yang Simple Linear Regression hanya saja Multivariate Linear Regression memiliki lebih > 1 independent variable (x)

```
In [1]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

Load library

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

Load datasets

```
In [3]: #Nama dataframe kita adalah df yang berisi data dari kc_house_data.csv.
#Features yang digunakan adalah 'bedrooms', 'bathrooms', 'sqft_living', 'grade', 'price' dan 'yr_built'
df = pd.read_csv('/content/drive/MyDrive/MATERI/Pembelajaran Mesin/Praktikum Genap 20212022/kc_house_data.csv',
                usecols=['bedrooms', 'bathrooms', 'sqft_living', 'grade', 'price', 'yr_built'])
```

Sneak peak data

```
In [4]: #Melihat 5 baris teratas dari data
#Independent variabel(x) adalah bedrooms, bathrooms, sqft_living, grade, yr_built
#Dependent variabel(y) adalah price
df.head()
```

```
Out[4]:
```

	price	bedrooms	bathrooms	sqft_living	grade	yr_built
0	221900.0	3	1.00	1180	7	1955
1	538000.0	3	2.25	2570	7	1951
2	180000.0	2	1.00	770	6	1933
3	604000.0	4	3.00	1960	7	1965
4	510000.0	3	2.00	1680	8	1987

- Penjelasan setiap kolom:
 1. bedrooms = Jumlah kamar tidur
 2. bathrooms = Jumlah kamar mandi
 3. sqft_living = Luas rumah dalam satuan sqft
 4. grade = Grading system dari pemerintah King County US
 5. yr_built = Tahun dimana rumah dibangun
 6. price = Harga dari rumah (US\$)

```
In [5]: #Mengetahui jumlah kolom dan baris dari data
#Data kita mempunyai 6 kolom (features) dengan 21613 baris
df.shape
```

```
Out[5]: (21613, 6)
```

```
In [6]: #Melihat informasi data kita mulai dari jumlah data, tipe data, memory yang digunakan dll.
#Dapat dilihat bahwa seluruh data sudah di dalam bentuk numerik
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
# 0  price           21613 non-null  float64
# 1  bedrooms        21613 non-null  int64
# 2  bathrooms        21613 non-null  float64
# 3  sqft_living      21613 non-null  float64
# 4  grade           21613 non-null  int64
# 5  yr_built        21613 non-null  int64
```

```

---
0  price      21613 non-null float64
1  bedrooms  21613 non-null int64
2  bathrooms 21613 non-null float64
3  sqft_living 21613 non-null int64
4  grade      21613 non-null int64
5  yr_built   21613 non-null int64
dtypes: float64(2), int64(4)
memory usage: 1013.2 KB

```

```
In [7]: #Melihat statistical description dari data mulai dari mean, kuartil, standard deviation dll
df.describe()
```

```
Out[7]:
```

	price	bedrooms	bathrooms	sqft_living	grade	yr_built
count	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000
mean	5.400881e+05	3.370842	2.114757	2079.899736	7.656873	1971.005136
std	3.671272e+05	0.930062	0.770163	918.440897	1.175459	29.373411
min	7.500000e+04	0.000000	0.000000	290.000000	1.000000	1900.000000
25%	3.219500e+05	3.000000	1.750000	1427.000000	7.000000	1951.000000
50%	4.500000e+05	3.000000	2.250000	1910.000000	7.000000	1975.000000
75%	6.450000e+05	4.000000	2.500000	2550.000000	8.000000	1997.000000
max	7.700000e+06	33.000000	8.000000	13540.000000	13.000000	2015.000000

- Pada feature bathrooms terdapat nilai pecahan, aneh kan yak kalo ada nilai jumlah kamar mandi pecahan gitu. Maka kita ubah dulu jenis data yang semula float menjadi int.
- Pada feature bedrooms terdapat nilai 33, ini sangat aneh karena masak rumah ada yang punya kamar 33 apalagi ini rumah pribadi. jadi kemungkinan itu typo dan akan saya ganti menjadi 3

```
In [8]: #Mrubah tipe data dari bathrooms yang semula float menjadi int
df['bathrooms'] = df['bathrooms'].astype('int')
```

```
In [9]: #Mengganti nilai 33 menjadi 3
df['bedrooms'] = df['bedrooms'].replace(33,3)
```

Handling Missing Values

```
In [10]: #Mencari dan menangani missing values
#Ternyata data kita tidak ada missing values
df.isnull().sum()
```

```
Out[10]: price      0
bedrooms    0
bathrooms   0
sqft_living 0
grade       0
yr_built    0
dtype: int64
```

Exploratory Data Analysis (EDA)

```
In [11]: df.head()
```

```
Out[11]:
```

	price	bedrooms	bathrooms	sqft_living	grade	yr_built
0	221900.0	3	1	1180	7	1955
1	538000.0	3	2	2570	7	1951
2	180000.0	2	1	770	6	1933
3	604000.0	4	3	1960	7	1965
4	510000.0	3	2	1680	8	1987

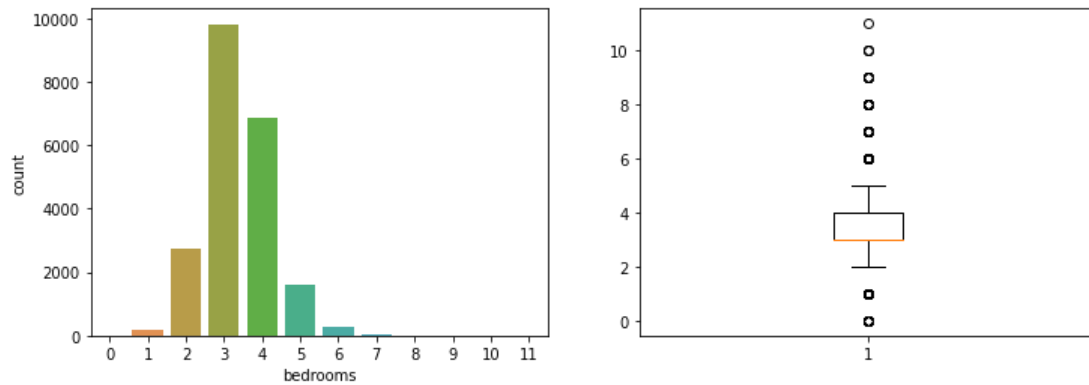
```
In [12]: #Univariate analysis bedrooms
#Melihat distribusi dari bedrooms
f = plt.figure(figsize=(12,4))

f.add_subplot(1,2,1)
sns.countplot(df['bedrooms'])

f.add_subplot(1,2,2)
plt.boxplot(df['bedrooms'])
plt.show()
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning



- Dapat dilihat bahwa sebagian besar jumlah kamar tidur itu di angka 3 dan 4.
- Data memiliki banyak outliers.

```
In [13]: #Univariate analysis bathrooms
#Melihat distribusi dari bathrooms
f = plt.figure(figsize=(12,4))
```

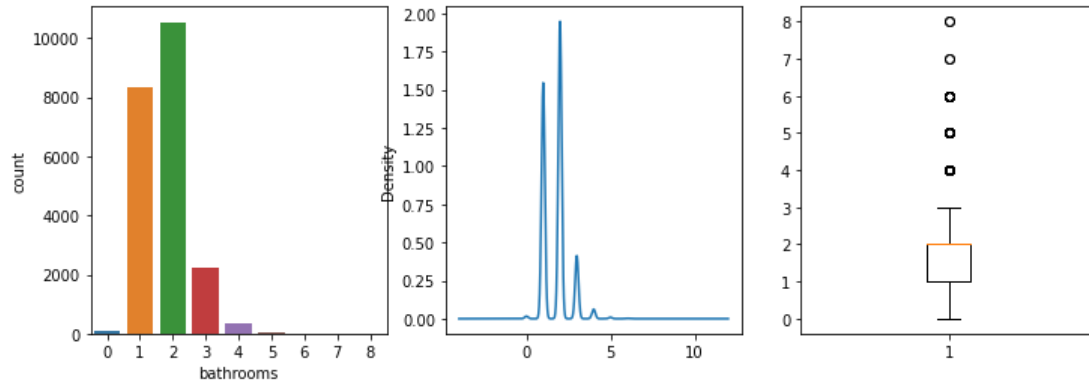
```
f.add_subplot(1,3,1)
sns.countplot(df['bathrooms'])

f.add_subplot(1,3,2)
df['bathrooms'].plot(kind='kde')

f.add_subplot(1,3,3)
plt.boxplot(df['bathrooms'])
plt.show()
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

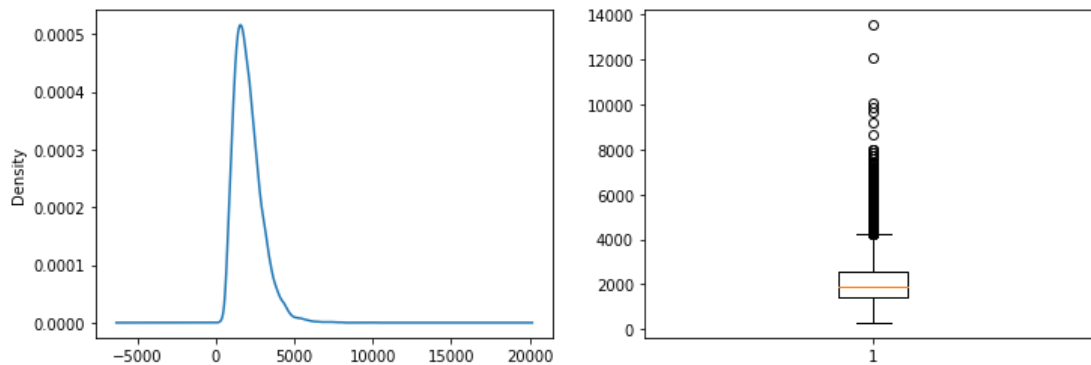


- Jumlah kamar mandi paling banyak berada pada angka 1 dan 2.
- Yang menarik disini adalah dimana ada rumah yang tidak ada kamar mandinya atau jumlahnya 0
- Nilai outlier sendiri lumayan banyak.

```
In [14]: #Univariate analysis sqft_living
#Melihat distribusi dari sqft_living
f = plt.figure(figsize=(12,4))
```

```
f.add_subplot(1,2,1)
df['sqft_living'].plot(kind='kde')

f.add_subplot(1,2,2)
plt.boxplot(df['sqft_living'])
plt.show()
```



- Density dari distribusi luas rumah berada di sekitar angka 2000an.
- Banyak terdapat outliers.

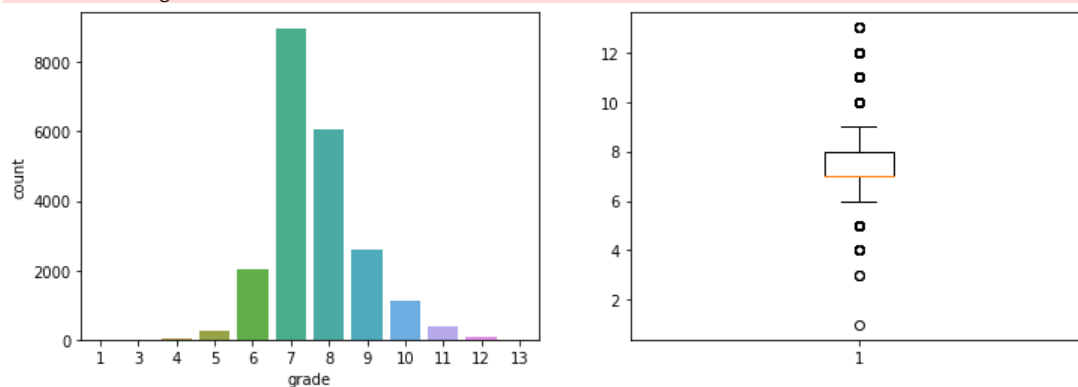
```
In [15]: #Univariate analysis grade
#Melihat distribusi dari grade
f = plt.figure(figsize=(12,4))

f.add_subplot(1,2,1)
sns.countplot(df['grade'])

f.add_subplot(1,2,2)
plt.boxplot(df['grade'])
plt.show()
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning



- Sebagian besar rumah di County King US memiliki grade 7 dan 8.
- Dilihat dari boxplot, data memiliki beberapa outliers.

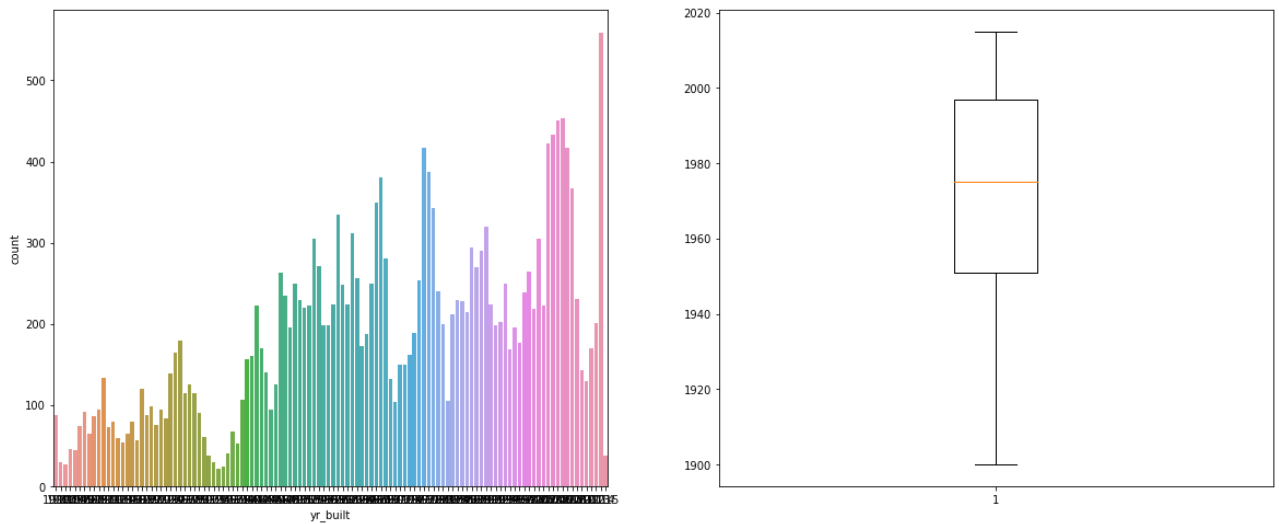
```
In [16]: #Univariate analysis yr_built
#Melihat distribusi dari yr_built
f = plt.figure(figsize=(20,8))

f.add_subplot(1,2,1)
sns.countplot(df['yr_built'])

f.add_subplot(1,2,2)
plt.boxplot(df['yr_built'])
plt.show()
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning



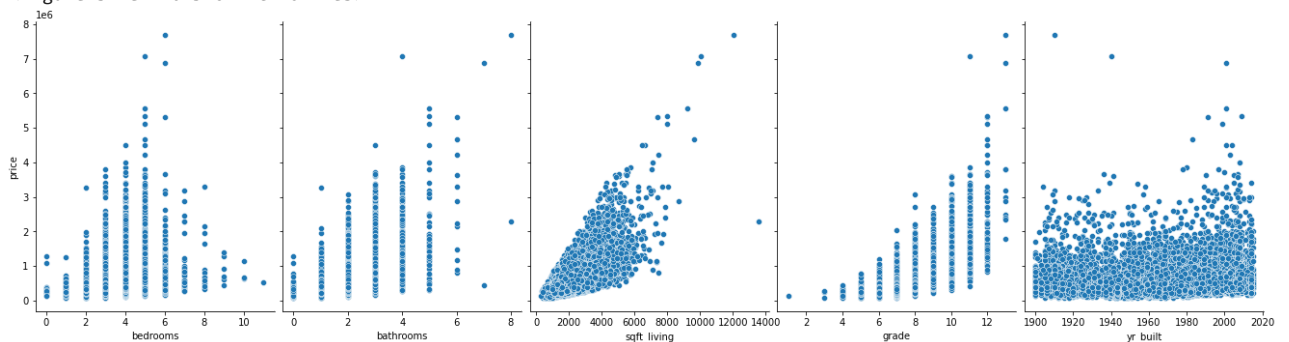
- Dapat dilihat bahwa semakin tua umur dari rumah, maka semakin sedikit orang yang menjual rumahnya tersebut.
- Density terdapat di sekitar tahun 1980an.
- Data tidak memiliki outliers.

```
In [17]: #Bivariate analysis antara independent variable dan dependent variable
#Melihat hubungan antara independent dan dependent
#Menggunakan pairplot
plt.figure(figsize=(10,8))
sns.pairplot(data=df, x_vars=['bedrooms', 'bathrooms', 'sqft_living', 'grade', 'yr_built'], y_vars=['price'], size=5,
plt.show())
```

/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:2076: UserWarning: The `size` parameter has been renamed to `height`; please update your code.

warnings.warn(msg, UserWarning)

<Figure size 720x576 with 0 Axes>



```
In [18]: #Mengetahui nilai korelasi dari independent variable dan dependent variable
df.corr().style.background_gradient().set_precision(2)
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: this method is deprecated in favour of `Styler.format(precision=..)`

```
Out[18]:
```

	price	bedrooms	bathrooms	sqft_living	grade	yr_built
price	1.00	0.32	0.51	0.70	0.67	0.05
bedrooms	0.32	1.00	0.48	0.59	0.37	0.16
bathrooms	0.51	0.48	1.00	0.70	0.61	0.43
sqft_living	0.70	0.59	0.70	1.00	0.76	0.32
grade	0.67	0.37	0.61	0.76	1.00	0.45
yr_built	0.05	0.16	0.43	0.32	0.45	1.00

- Dari tabel korelasi diatas, dapat dilihat bahwa sqft_living mempunyai hubungan linear positif yang sangat kuat dengan price jika dibandingkan yang lain.
- Nilai korelasi yr_built hampir mendekati nol yang menandakan bahwa usia rumah tidak mempengaruhi pada harga rumah.

Modelling

```
In [19]: #Recall data kita
df.head()
```

```
Out[19]:
```

	price	bedrooms	bathrooms	sqft_living	grade	yr_built
0	221900.0	3	1	1180	7	1955
1	538000.0	3	2	2570	7	1951
2	180000.0	2	1	770	6	1933
3	604000.0	4	3	1960	7	1965
4	510000.0	3	2	1680	8	1987

```
In [20]: #Pertama, buat variabel x dan y
x = df.drop(columns='price')
y = df['price']
```

```
In [21]: #Kedua, kita split data kita menjadi training and testing dengan porsi 80:20
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=4)
```

```
In [22]: #Cek shape dari data training dan testing
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)

(17290, 5)
(17290,)
(4323, 5)
(4323,)
```

```
In [23]: #Ketiga, kita bikin object linear regresi
lin_reg = LinearRegression()
```

```
In [24]: #Keempat, train the model menggunakan training data yang sudah displit
lin_reg.fit(x_train, y_train)
```

```
Out[24]: LinearRegression()
```

```
In [25]: #Kelima, cari tau nilai slope/koefisien (m) dan intercept (b)
print(lin_reg.coef_)
print(lin_reg.intercept_)

[-53061.75464279  64658.55790617  188.90926343 131290.89536823
 -3969.55831454]
7031568.245717696
```

```
In [26]: #Kita coba buat kedalam dataframe agar lebih rapi
coef_dict = {
    'features': x.columns,
    'coef_value': lin_reg.coef_
}
coef = pd.DataFrame(coef_dict, columns=['features', 'coef_value'])
coef
```

```
Out[26]:
```

	features	coef_value
0	bedrooms	-53061.754643
1	bathrooms	64658.557906
2	sqft_living	188.909263
3	grade	131290.895368
4	yr_built	-3969.558315

- dari nilai m dan b diatas, kalau dimasukan ke dalam rumus menjadi: $Y = -53061.75x_1 + 64658.56x_2 + 188.91x_3 + 131290.89x_4 - 3969.56x_5 + 7031568$

```
In [27]: y_pred = lin_reg.predict(x_test)
```

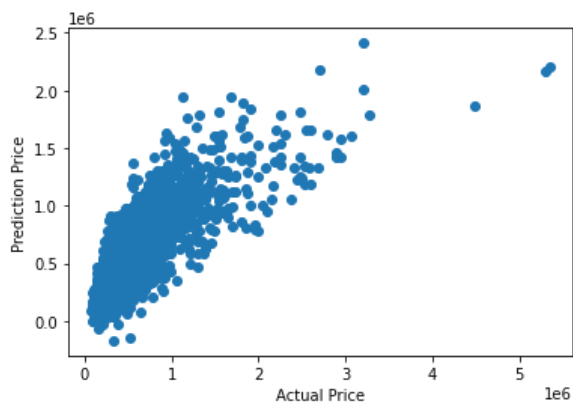
```
In [28]: #Keenam, kita cari tahu accuracy score dari model kita menggunakan testing data yang sudah displit
lin_reg.score(x_test, y_test)
```

```
Out[28]: 0.61251132869411
```

- Model kita mendapatkan accuracy score sebesar 61.13%

```
In [29]: y_prediksi = lin_reg.predict(x_test)
plt.scatter(y_test, y_prediksi)
plt.xlabel('Actual Price')
```

```
plt.ylabel('Prediction Price')
plt.show()
```



Prediction

- Yuk kita prediksi harga rumah sesuai dengan permintaan Joko dengan kriteria sebagai berikut:
 1. bedrooms = 3
 2. bathrooms = 2
 3. sqft_living = 1800 sqft
 4. grade = 7
 5. yr_built = 1990

```
In [30]: #Prediksi harga rumah idaman Joko
lin_reg.predict([[3,2,1800,7,1990]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  "X does not have valid feature names, but"
```

```
Out[30]: array([361351.99342265])
```

- Yeay! Harga rumah idaman Joko dan istrinya adalah sekitar 361351 US\$