Proyek Akhir KASDD

APARTMENT LISTINGS - KASMOM

The team

The team

Dafi Nafidz Radhiyya
2106701564

Ivan Rabbani C.
2106701892

M Fariz Eda A
2106653546

Taqiya Zayin Hanafie
2106751335

Table Of Contents

Latar Belakang & Tujuan

Latar Belakang

Apartment Listing adalah deskripsi rinci mengenai unit apartemen yang tersedia. Hal ini berfungsi sebagai alat penting dalam industri properti, membantu pemilik properti, pemilik tanah, atau agen properti menarik penyewa atau pembeli potensial.

Tujuan

Membantu stakeholder dalam peningkatan pemahaman terkait macam-macam apartment yang terdaftar

Deskripsi Dataset

Dataset Description

Dataset ini berisikan berbagai informasi suatu apartemen. Dataset ini mencakup kondisi sekitar apartemen dan apartemen sendiri serta tawaran pada apartemen.

19 Kolom

12.2k baris

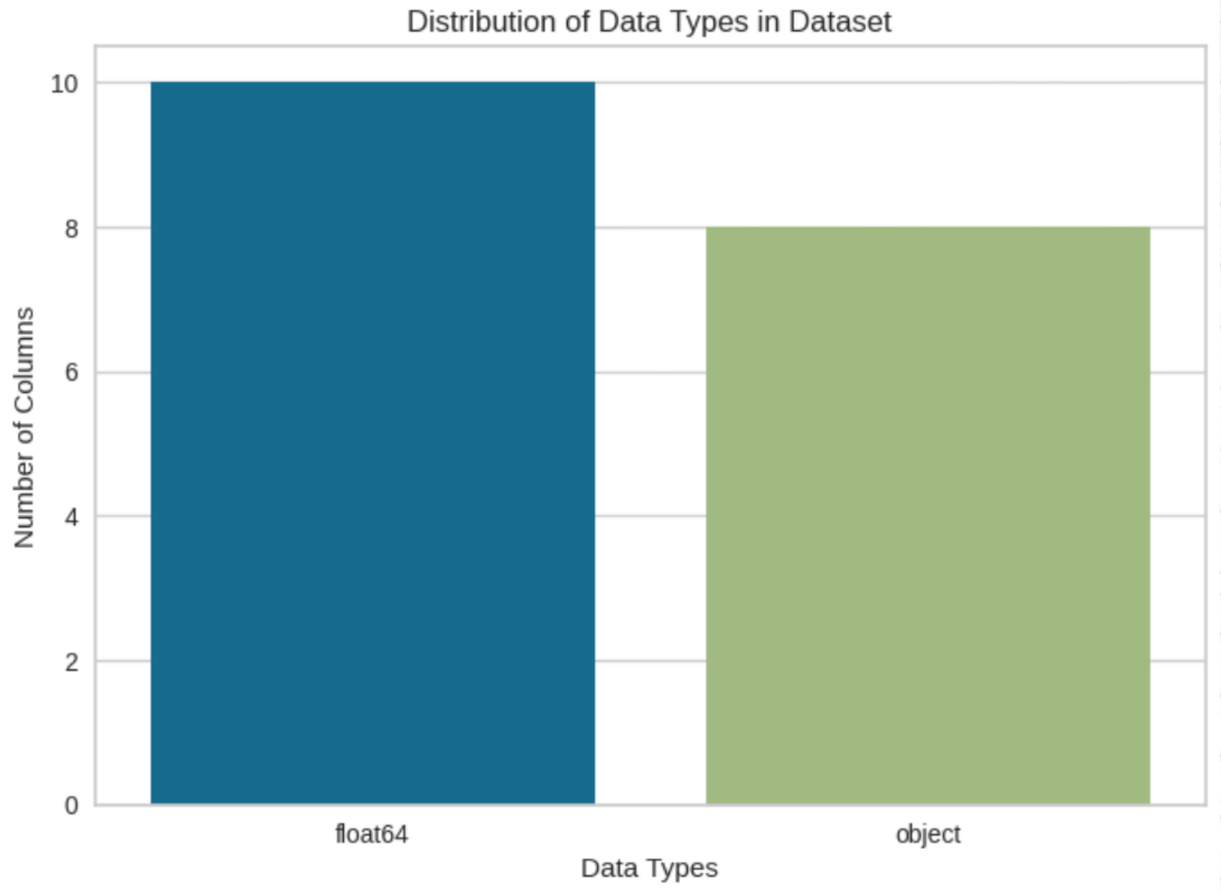
Column Description

ID	ID yang unik untuk setiap apartemen. • latitude: Nilai latitude (garis lintang) untuk apartemen yang sesuai.
longitude	Nilai longitude (garis bujur) untuk apartemen yang sesuai.
district	Distrik atau wilayah di mana apartemen berada. • address_offers: Jumlah tawaran atau penawaran yang terkait dengan alamat apartemen.
agent_offers	Jumlah tawaran atau penawaran yang terkait dengan agen properti yang berhubungan dengan apartemen.
subway_offers	Jumlah tawaran atau penawaran yang terkait dengan stasiun kereta yang berdekatan dengan apartemen
closest_subway	Stasiun kereta terdekat dengan apartemen

dist_to_subway	Jarak antara apartemen dan stasiun kereta terdekat.
subway_grade	Kelas stasiun kereta yang berdekatan den gan apartemen.
subway_dist_to_center	Jarak antara stasiun kereta terdekat dan pusat kota.
rooms	Jumlah kamar di apartemen
floor	Lantai tempat apartment berada
max_floor	Jumlah lantai maksimum dalam bangunan di mana apartemen berada.
footage	Luas dari apartemen.
material	Informasi tentang bahan bangunan apartemen
repair	Kondisi perbaikan apartemen.
category_age	Kategori usia apartemen.
price	Harga apartemen.

Data Type Information

```
RangeIndex: 10353 entries, 0 to 10352
Data columns (total 19 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     10353 non-null  object
 1   latitude               8979 non-null   float64
 2   longitude              9951 non-null   float64
 3   district               9173 non-null   object
 4   address_offers         9884 non-null   float64
 5   agent_offers           8430 non-null   float64
 6   subway_offers          7772 non-null   float64
 7   closest_subway         9319 non-null   object
 8   dist_to_subway         7792 non-null   float64
 9   subway_grade           9556 non-null   object
 10  subway_dist_to_center  9215 non-null   float64
 11  rooms                  8999 non-null   object
 12  floor                  8515 non-null   float64
 13  max_floor              9114 non-null   float64
 14  footage                8618 non-null   float64
 15  material               10310 non-null  object
 16  repair                 8805 non-null   object
 17  price                  10265 non-null  float64
 18  category_age           10353 non-null  object
dtypes: float64(11), object(8)
memory usage: 1.5+ MB
```



Dataset Overview

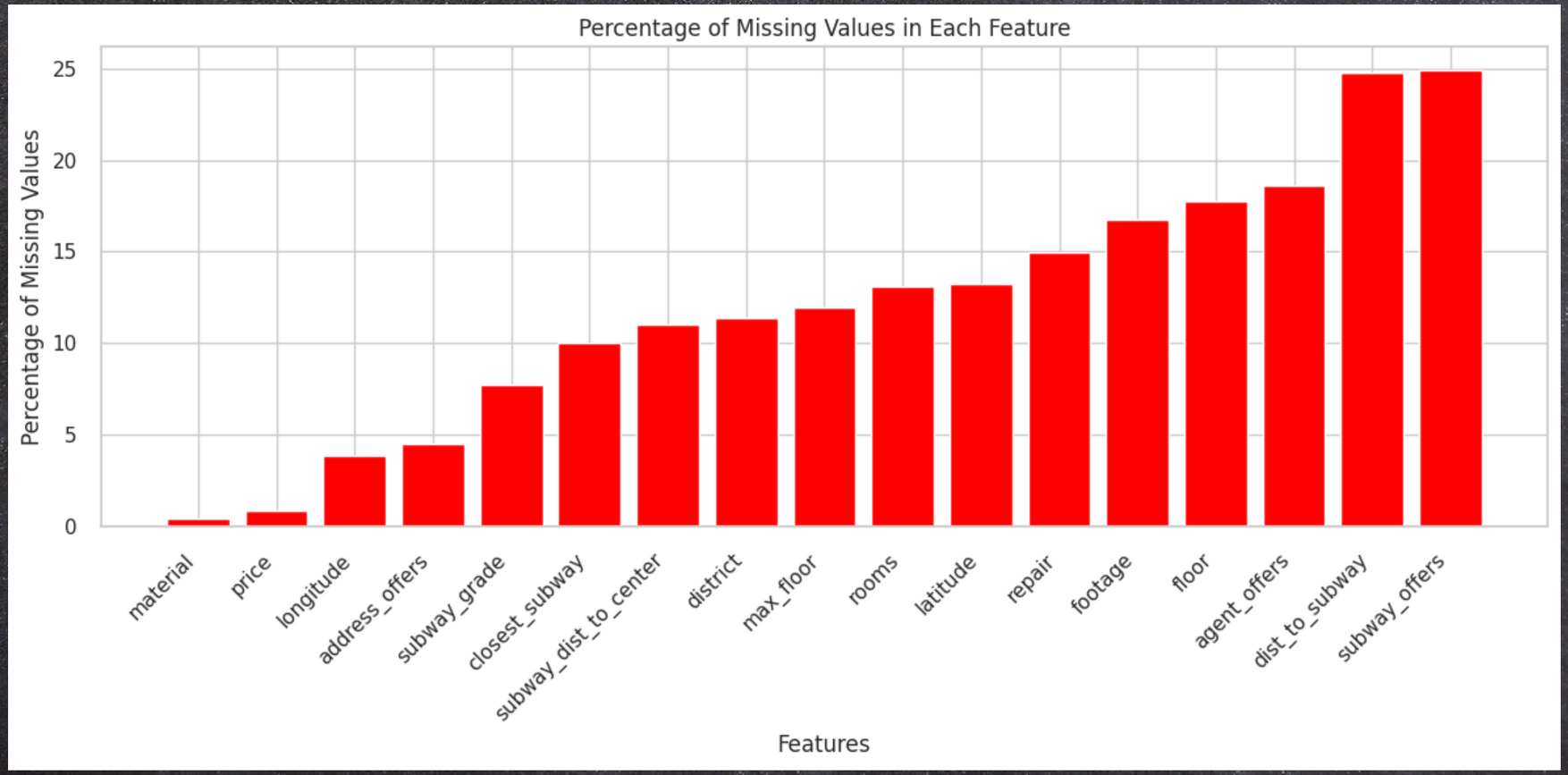
	latitude	longitude	address_offers	agent_offers	subway_offers	dist_to_subway	subway_dist_to_center	floor	max_floor	footage	price
count	8979.000000	9951.000000	9884.000000	8430.000000	7772.000000	7792.000000	9215.000000	8515.000000	9114.000000	8618.000000	1.026500e+04
mean	55.738404	37.618576	1.290065	25.044009	10.210628	928.072767	11918.266739	6.810922	13.445578	46.084252	1.088461e+07
std	0.087579	0.133899	0.729496	61.242519	7.095922	536.066525	4723.460876	5.239973	6.377801	10.478897	3.049273e+06
min	55.527631	37.306055	1.000000	0.000000	1.000000	11.000000	1133.000000	1.000000	2.000000	30.100000	4.900000e+06
25%	55.674936	37.520720	1.000000	1.000000	5.000000	543.000000	8359.000000	3.000000	9.000000	38.000000	8.500000e+06
50%	55.730593	37.604592	1.000000	1.000000	8.000000	815.000000	12418.000000	5.000000	12.000000	44.500000	1.020000e+07
75%	55.806176	37.720569	1.000000	11.000000	13.000000	1177.000000	14800.000000	9.000000	17.000000	53.200000	1.270000e+07
max	55.941577	37.950668	9.000000	292.000000	40.000000	2988.000000	24636.000000	38.000000	39.000000	89.900000	2.000000e+07

Preprocessing

```
Jumlah duplikasi data: 0
```

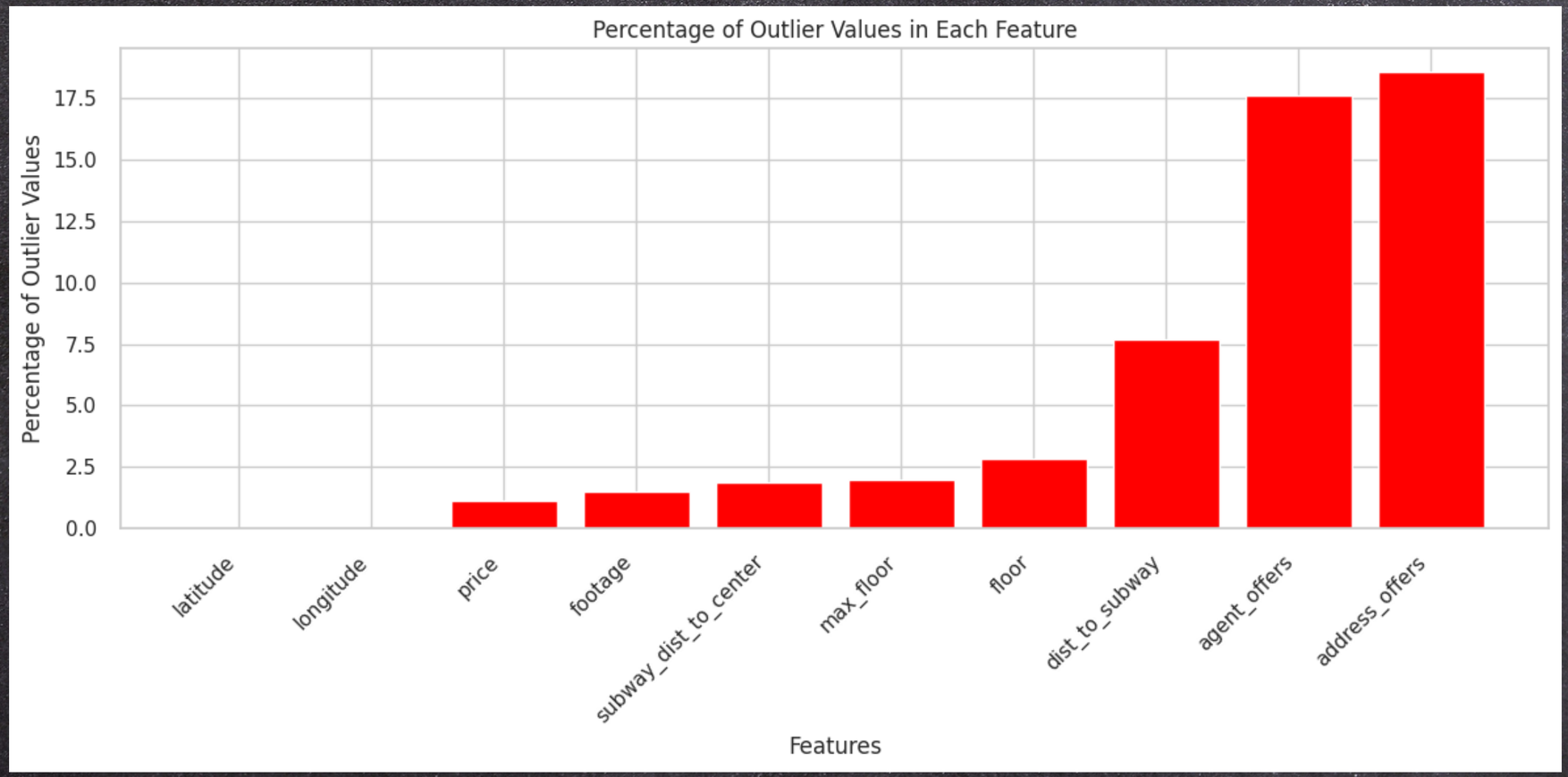
Duplikat tidak ditemukan.

Preprocessing



Menghilangkan atau mengimputasi data missing value dengan presentase <5% dan dianggap tidak berpengaruh karena dianggap kurang berpengaruh terhadap data.

Preprocessing



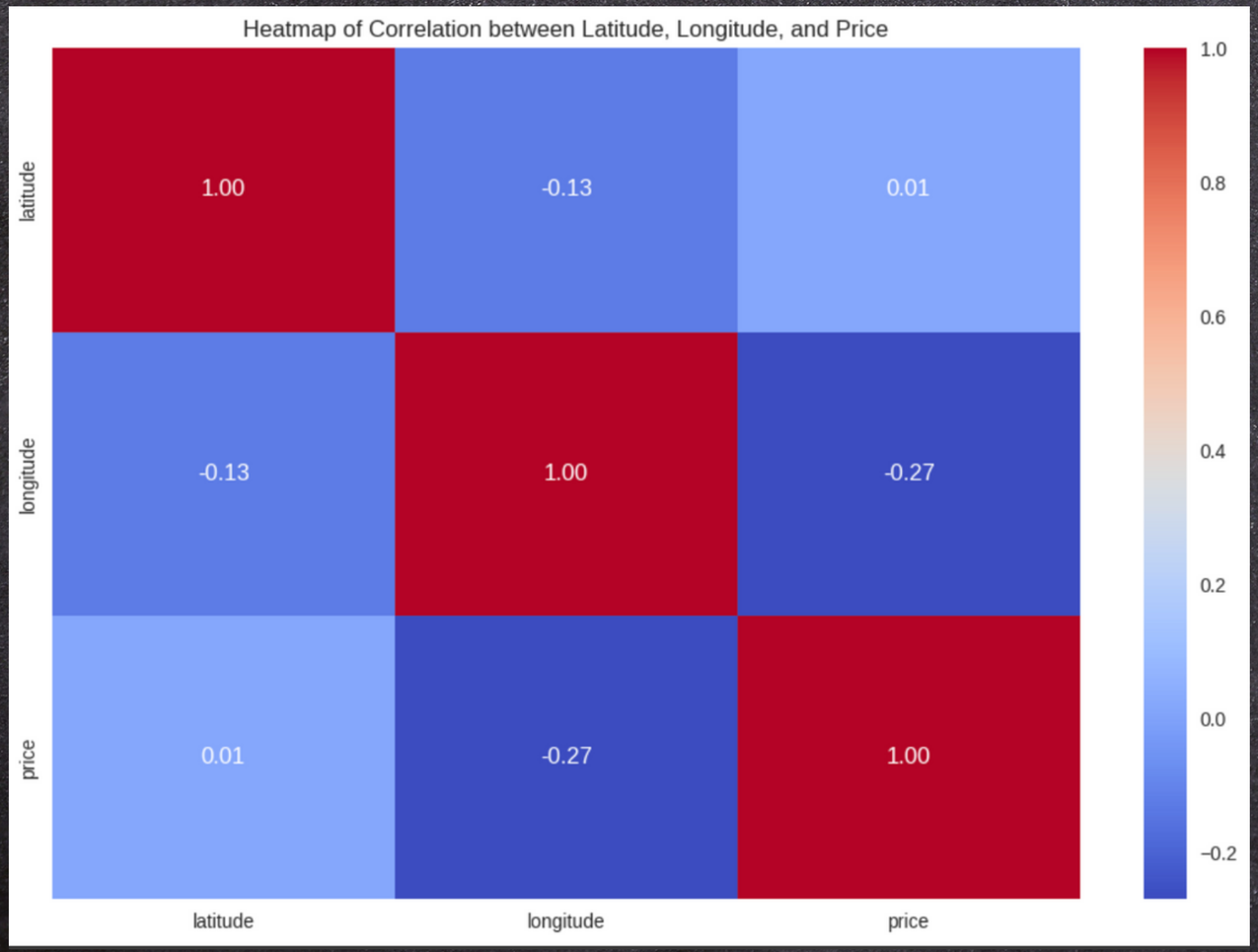
Menghilangkan atau mengimputasi data outlier dengan presentase <5% dan dianggap tidak berpengaruh karena dianggap kurang berpengaruh terhadap data.

EDA TASK

Task: EDA

1. Apakah terdapat hubungan antara lokasi bangunan dengan harga bangunan?
2. Apa ciri-ciri bangunan yang dekat dengan subway dengan jenis kereta yang bukan bawah tanah?
3. Apakah kondisi perbaikan bangunan yang buruk merupakan bangunan tua?
4. Apakah distrik dengan total tawaran tertinggi?
5. Berdasarkan rata-rata harga bangunan, berikan urutan dari tiap kategori material!
6. Berdasarkan rata-rata harga bangunan, berikan urutan dari tiap kategori usia bangunan!

Apakah terdapat hubungan antara lokasi bangunan dengan harga bangunan?



Dapat dilihat informasi bahwa lokasi bangunan (Latitude dan Longitude) dengan harga bangunan memiliki korelasi yang yang sangat rendah. Dengan demikian, dapat disimpulkan bahwa latitude dan longitude memiliki pengaruh yang sangat kecil terhadap harga suatu bangunan.

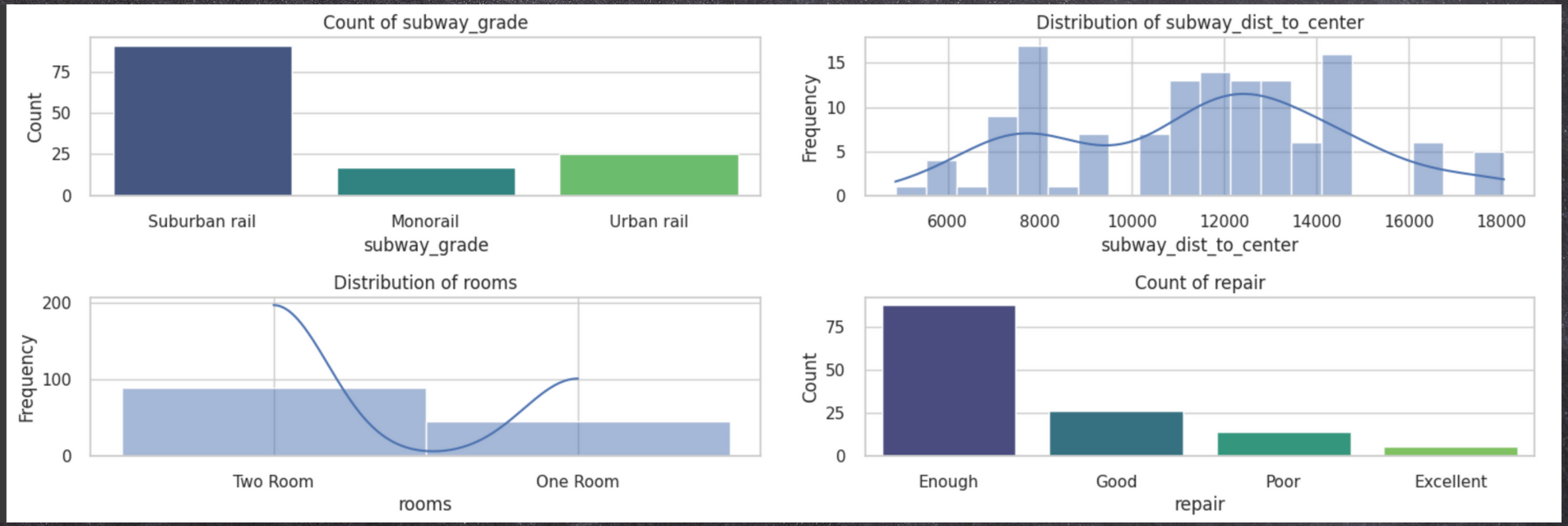
Apa ciri-ciri bangunan yang dekat dengan subway dengan jenis kereta yang bukan bawah tanah?

Kriteria

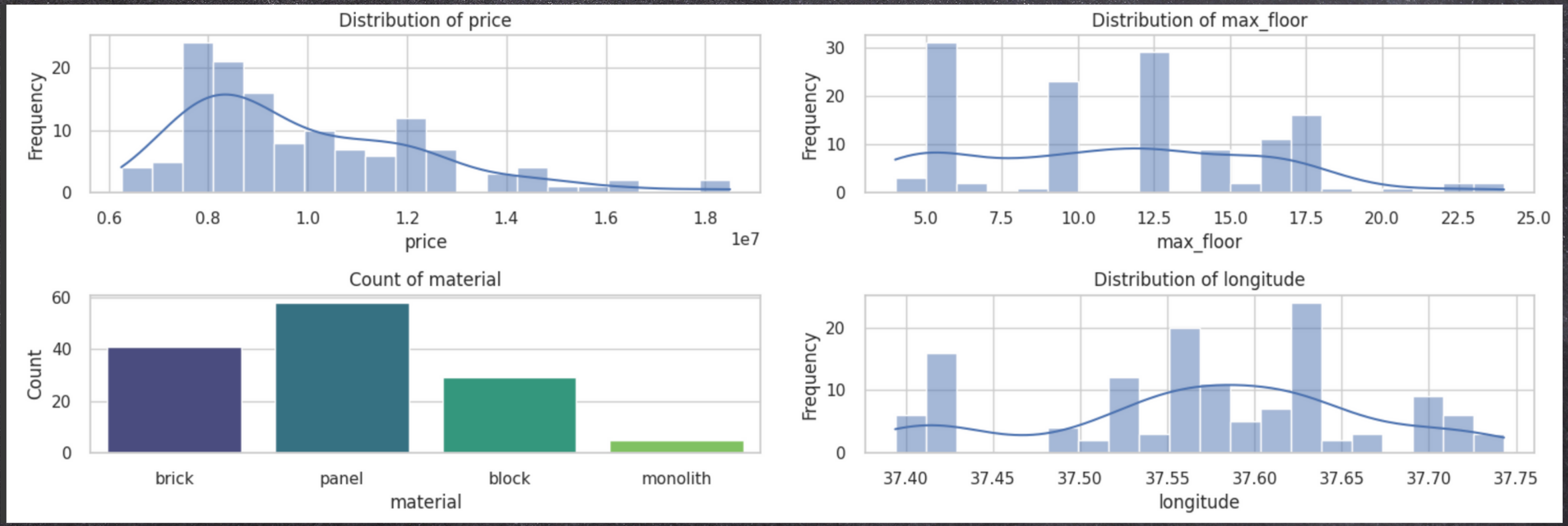
```
data_non_underground_close_to_subway = data_clean[(data_clean['subway_grade'] != 'Underground') & (data_clean['dist_to_subway'] < 500)]

# Characteristics
characteristics = ['latitude', 'longitude', 'district', 'address_offers', 'agent_offers', 'subway_grade', 'subway_dist_to_center',
                   'rooms', 'floor', 'max_floor', 'footage', 'material', 'repair',
                   'category_age', 'price']
```

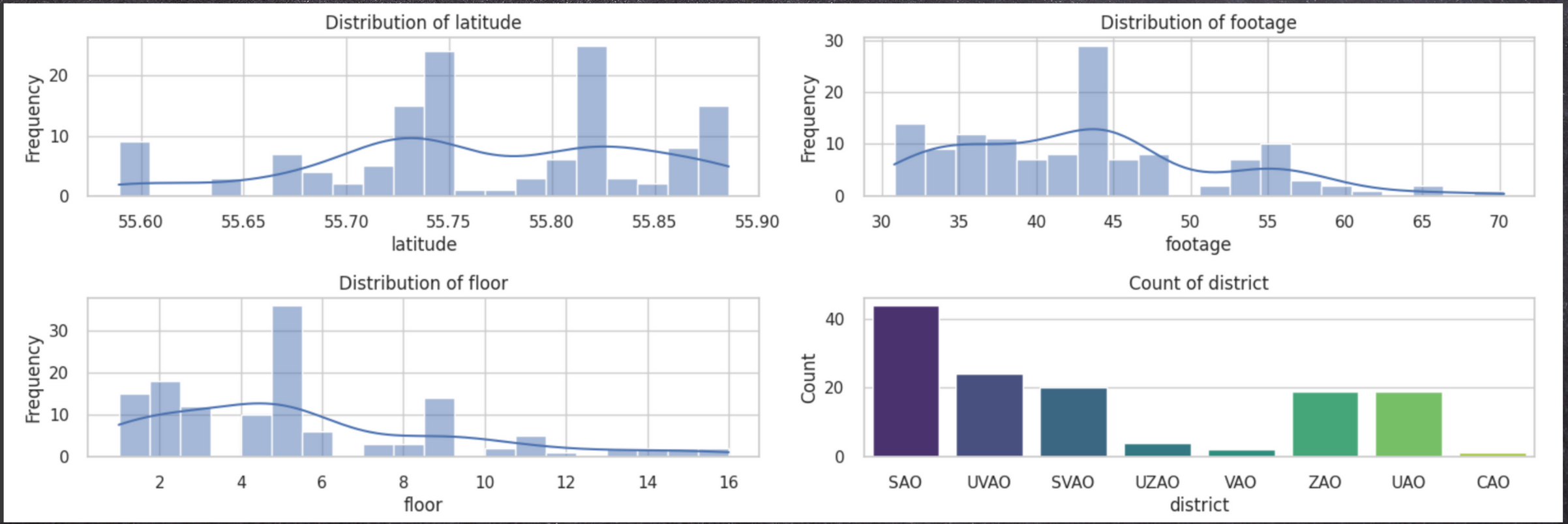
Analisis


Count of subway_grade
Count
75
50
25
0
Suburban rail
Monorail
Urban rail
subway_grade
Distribution of subway_dist_to_center
Frequency
15
10
5
0
6000
8000
10000
12000
14000
16000
18000
subway_dist_to_center
Distribution of rooms
Frequency
200
100
0
Two Room
One Room
rooms
Count of repair
Count
75
50
25
0
Enough
Good
Poor
Excellent
repair

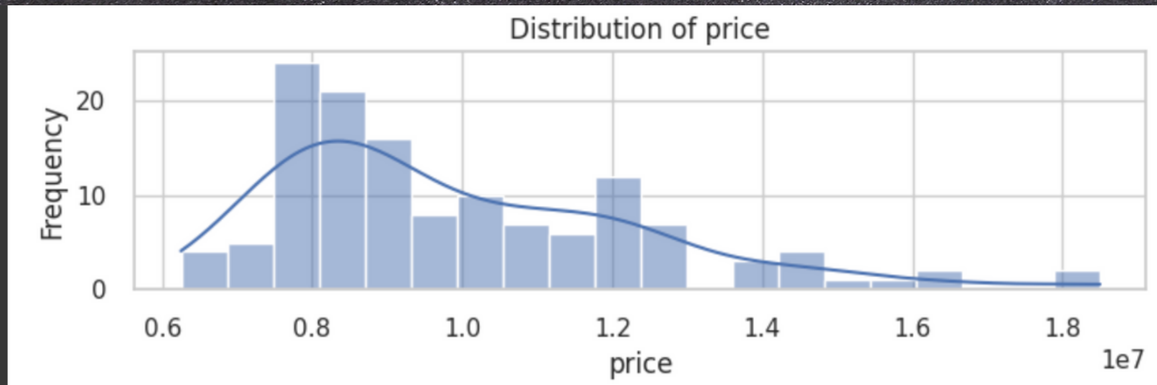
Analisis

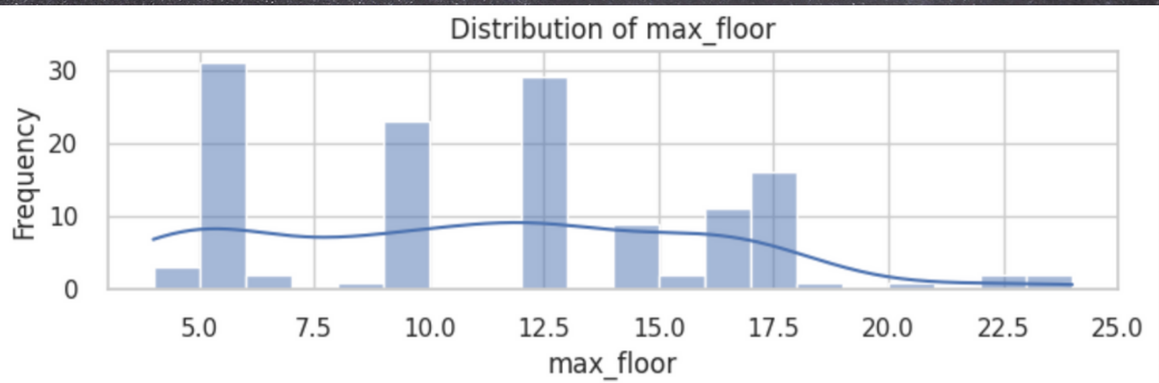

Distribution of price
Frequency
price
1e7
Distribution of max_floor
Frequency
max_floor
Count of material
Count
material
brick
panel
block
monolith
Distribution of longitude
Frequency
longitude

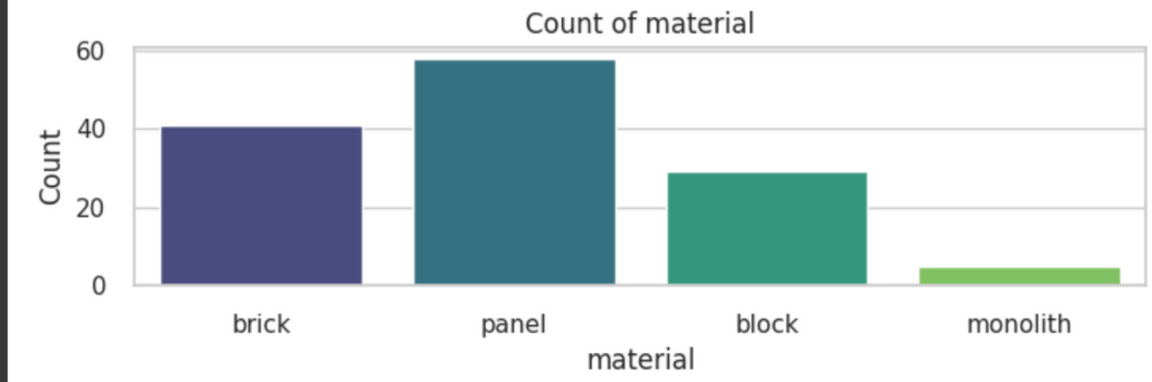
Analisis

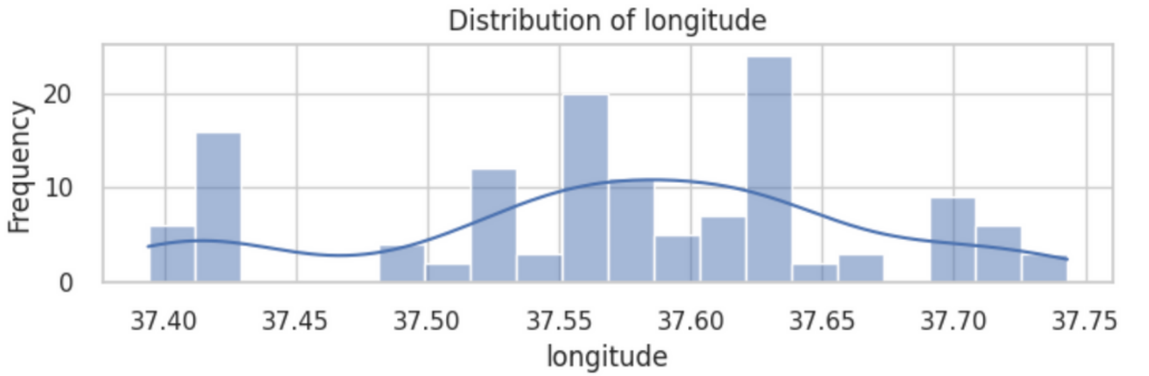

Distribution of latitude
Frequency
0
10
20
55.60
55.65
55.70
55.75
55.80
55.85
55.90
latitude
Distribution of footage
Frequency
0
10
20
30
30
35
40
45
50
55
60
65
70
footage
Distribution of floor
Frequency
0
10
20
30
2
4
6
8
10
12
14
16
floor
Count of district
Count
0
20
40
SAO
UVAO
SVAO
UZAO
VAO
ZAO
UAO
CAO
district

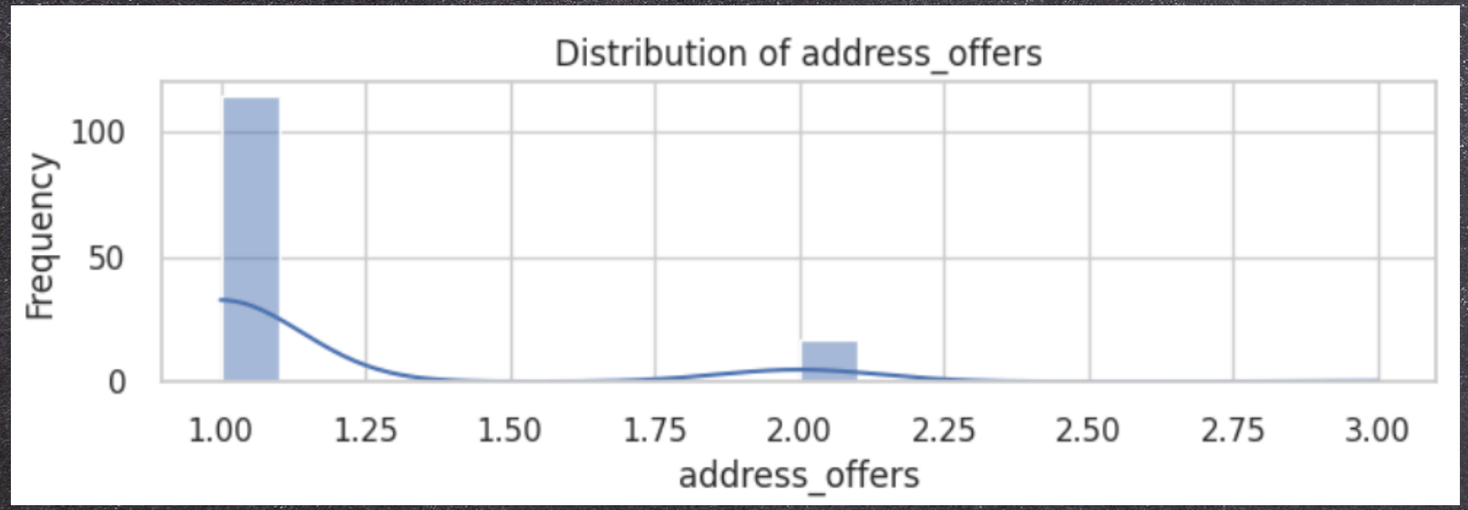
Analisis


Distribution of price
Frequency
0
10
20
0.6
0.8
1.0
1.2
1.4
1.6
1.8
price
1e7


Distribution of max_floor
Frequency
0
10
20
30
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0
max_floor


Count of material
Count
0
20
40
60
brick
panel
block
monolith
material

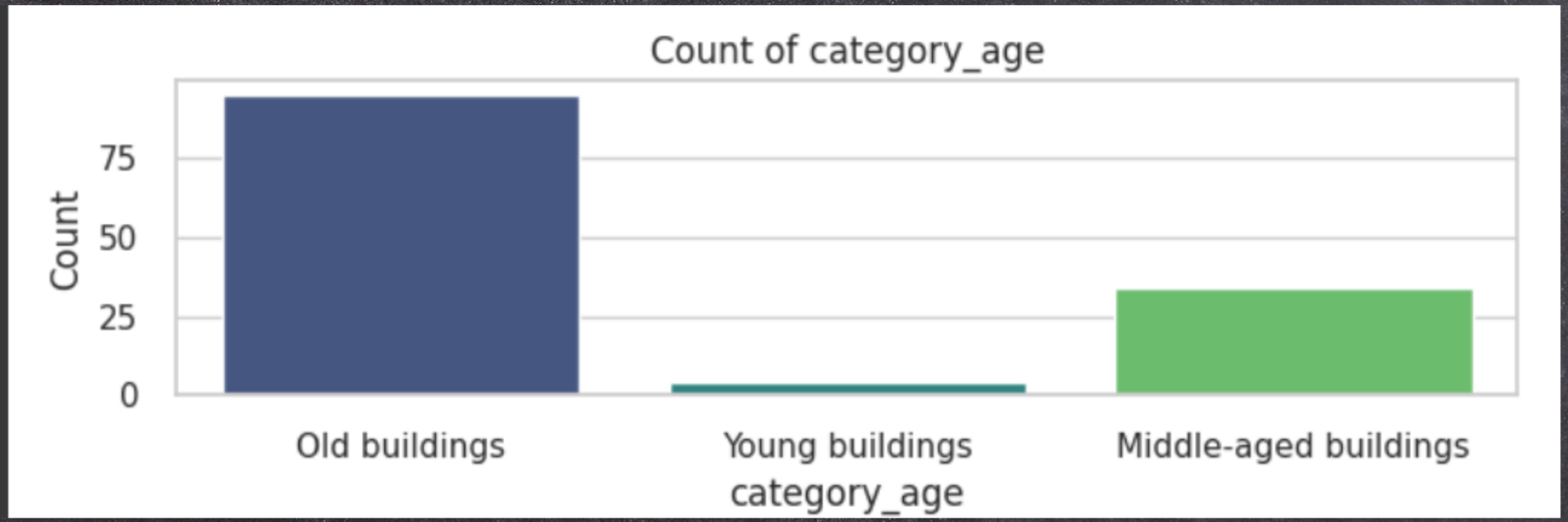

Distribution of longitude
Frequency
0
10
20
37.40
37.45
37.50
37.55
37.60
37.65
37.70
37.75
longitude


Distribution of address_offers
Frequency
100
50
0
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
address_offers


Distribution of agent_offers
Frequency
100
50
0
0
10
20
30
40
50
60
70
agent_offers

Analisis

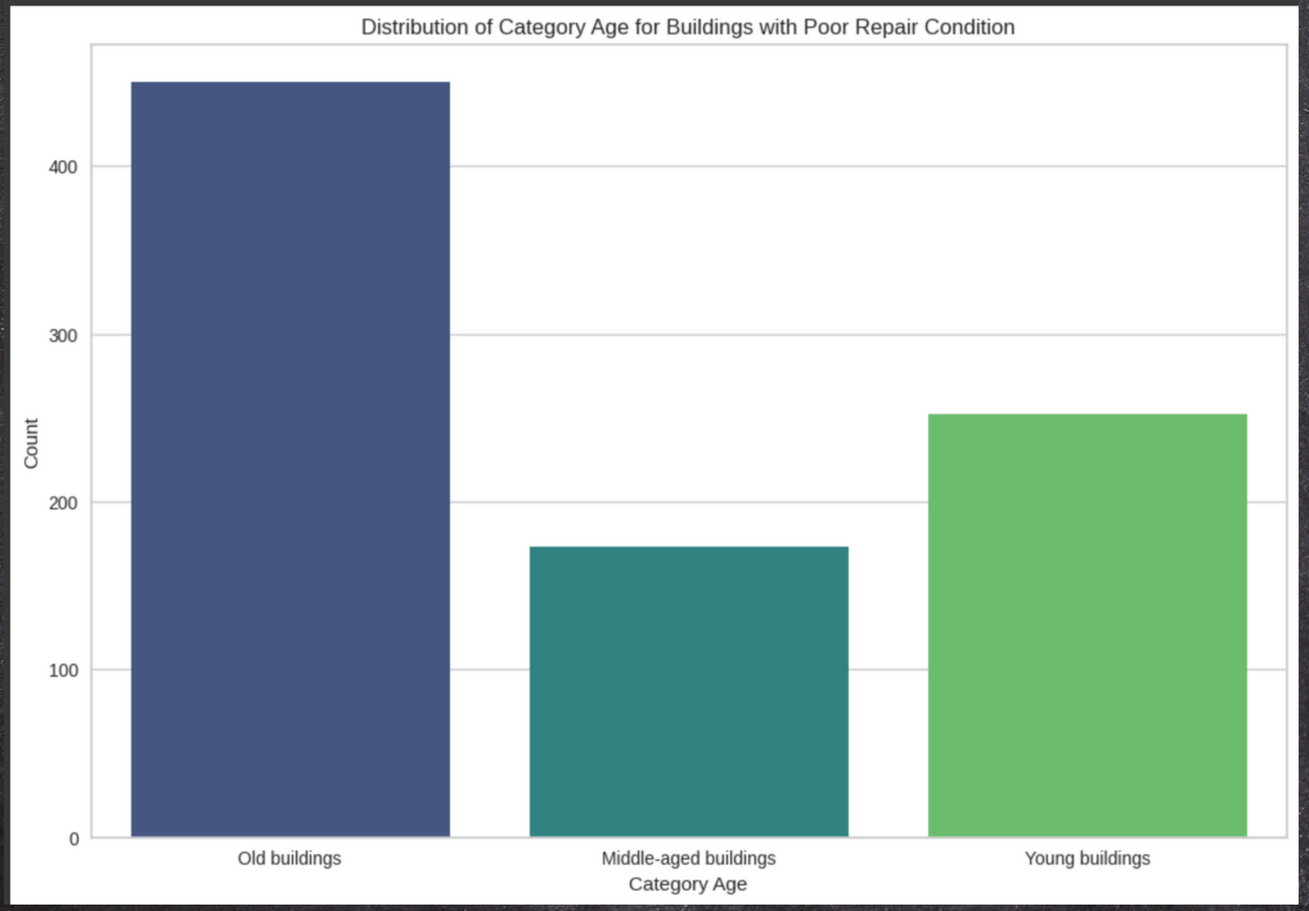
Analisis


Count of category_age
Count
75
50
25
0
Old buildings
Young buildings
Middle-aged buildings
category_age

	Characteristic	Mode
0	latitude	55.738255
1	longitude	37.624858
2	district	SAO
3	address_offers	1.0
4	agent_offers	1.0
5	subway_grade	Suburban rail
6	subway_dist_to_center	14761.0
7	rooms	Two Room
8	floor	5.0
9	max_floor	5.0
10	footage	44.4
11	material	panel
12	repair	Enough
13	category_age	Old buildings
14	price	7500000.0

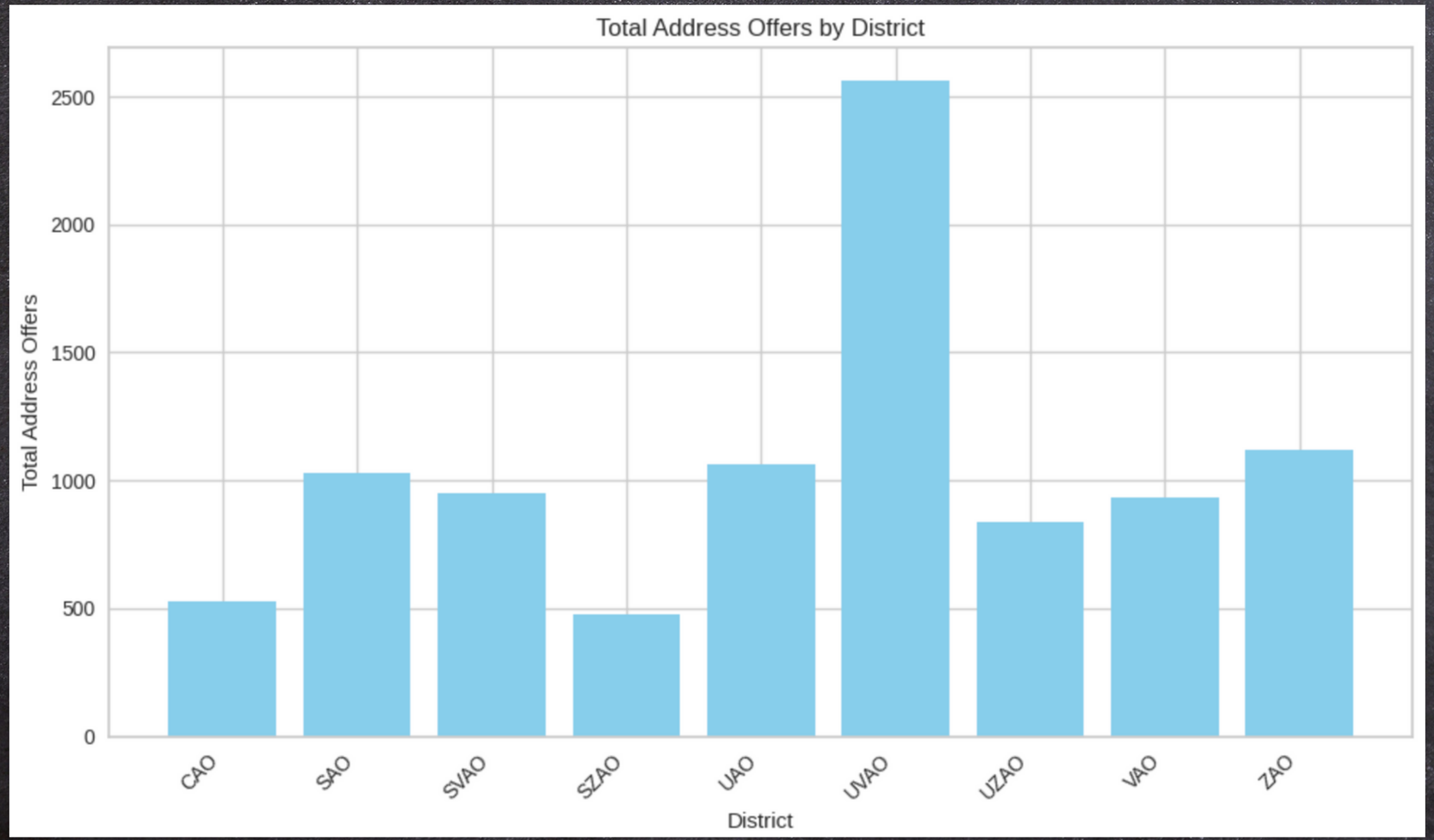
Kami melakukan analisis terhadap visualisasi karakteristik bangunan. Kami melihat bahwa bangunan dengan syarat **bangunan dekat dengan subway** dan **jenis kereta** subway_grade adalah **underground** seperti tabel samping.

Apakah kondisi perbaikan bangunan yang buruk merupakan bangunan tua?



Dilihat dari gambar, distribusi usia bangunan dengan kategori kondisi poor adalah menunjukkan bahwa mayoritas dari persebaran tersebut adalah bangunan tua. Meskipun begitu, hal tersebut tidak dapat menjustifikasi bahwa bangunan buruk = bangunan tua karena terdapat persebaran lain, seperti bangunan semi tua dan bangunan muda.

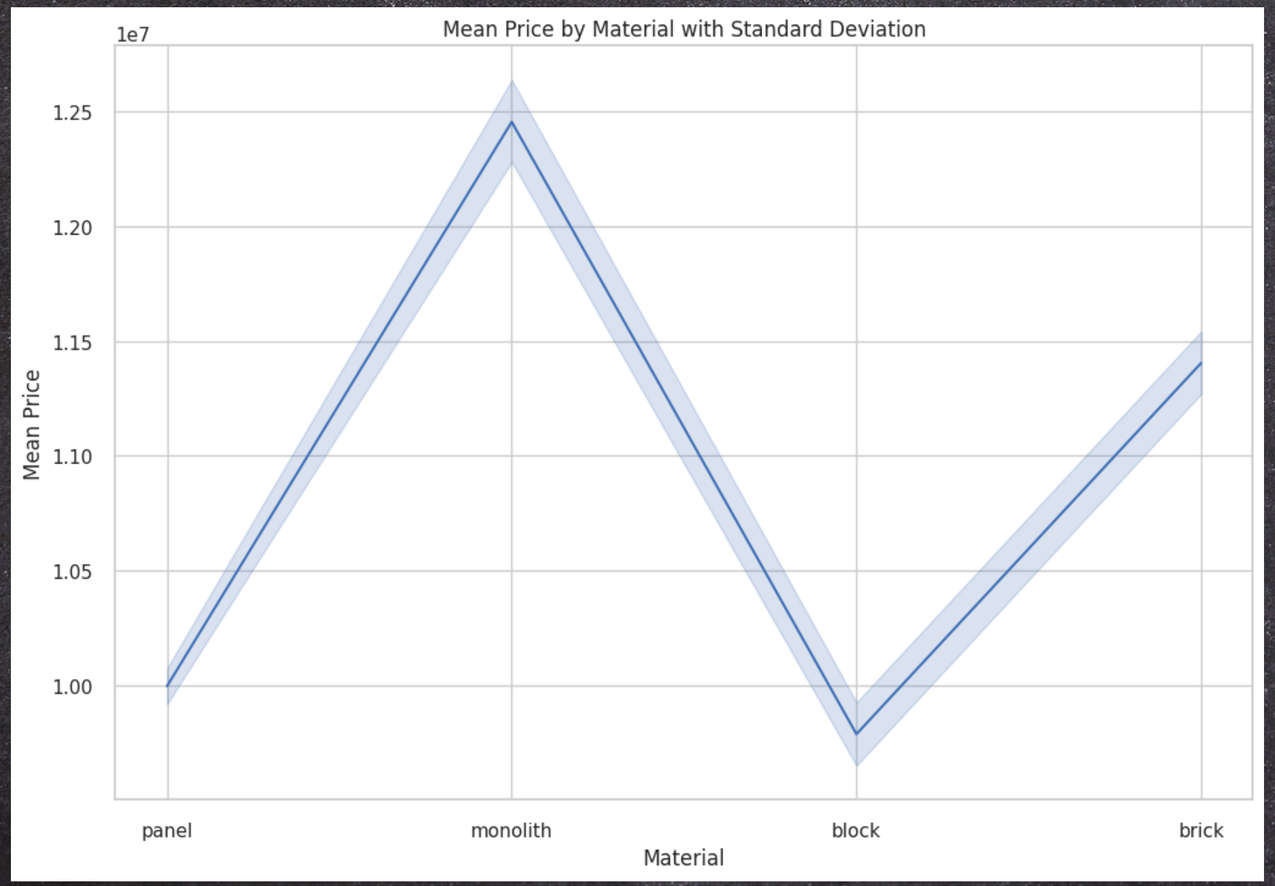
Apakah distrik dengan total tawaran tertinggi?



Dapat terlihat bahwa distrik UVAO memiliki tawaran terbanyak.

Eksplorasi Kelompok

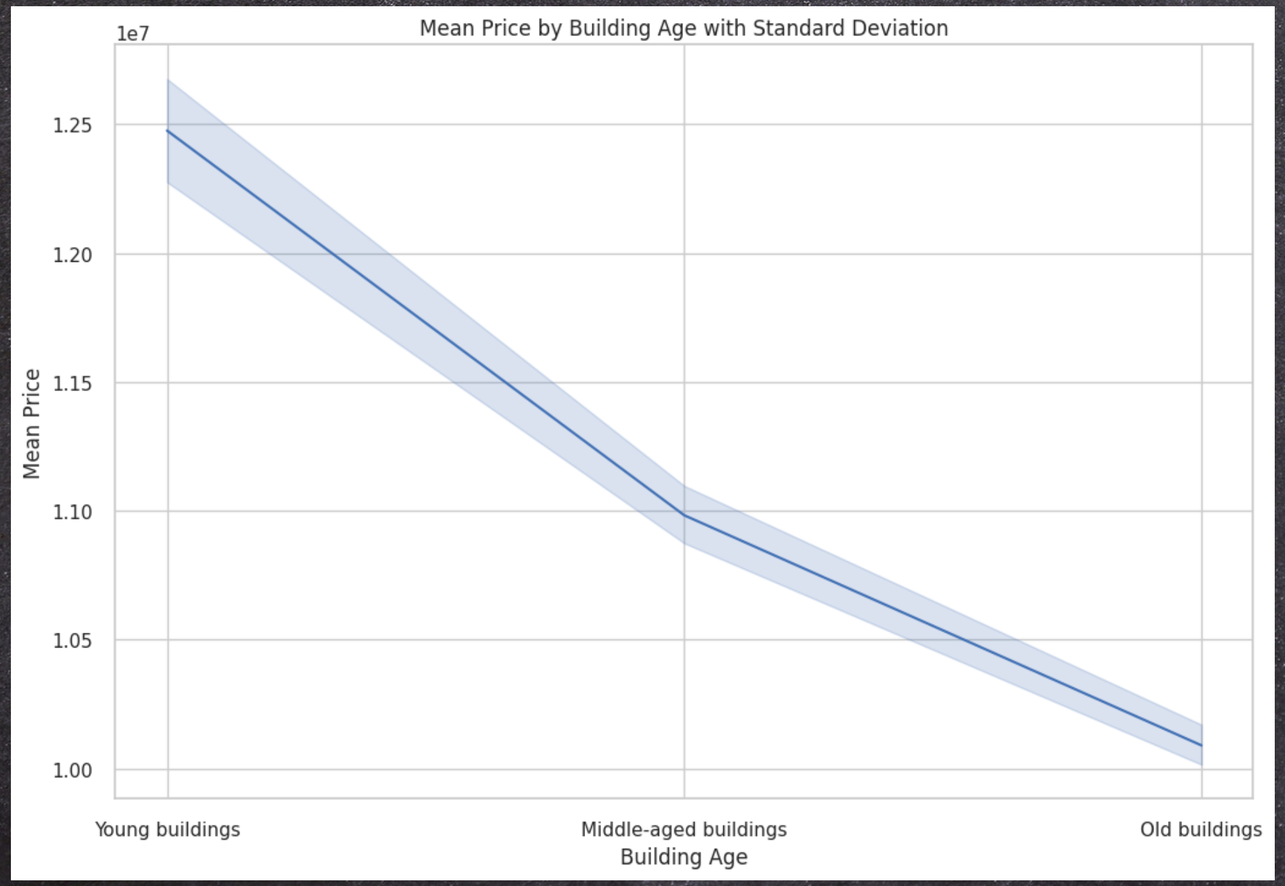
Berdasarkan rata-rata harga bangunan, berikan urutan dari tiap kategori material!



Berdasarkan gambar, dapat disimpulkan bahwa material dengan urutan termahal hingga termurah adalah monolith, brick, panel, dan block.

Eksplorasi Kelompok

Berdasarkan rata-rata harga bangunan, berikan urutan dari tiap kategori usia bangunan!



Berdasarkan gambar, dapat disimpulkan bahwa usia bangunan dengan urutan termahal hingga termurah adalah bangunan muda, pertengahan, dan tua.

CLASIFICATION TASK

Preprocessing

Duplicate Values

Tidak ada data duplikat

Missing Values

- Melakukan **drop row** pada kolom dengan persentase data yang hilang sebesar < 5%
- Melakukan **drop column subway_offers** karena dianggap terlalu banyak data yang hilang dan informasi atribut tersebut cukup digantikan dengan closest_subway dan dist_to_subway
- Melalukan **imputasi data** untuk sisanya

Feature Engineering

Imbalance Handling

Melakukan resampling dengan **SMOTE pada training data** karena dataset imbalance

Encoding

Melakukan label **encoding pada fitur kategorikal**

'geo_clustering'

Membuat fitur baru dari 'longitute' & 'latitude' menggunakan **Kmeans** guna mengidentifikasi dan menganalisis pola atau kelompok geografis dalam data

Standardization

Standarisasi fitur numerikal dengan **StandardScaler**

Model

1. Random Forest
2. K-Nearest Neighbour
3. Neural Network

Evaluasi Random Forest

Parameter Tuning With GridSearch:

- max_depth: 30
- min_samples_leaf: 1
- min_samples_split: 2
- n_estimators: 300

Hasil Evaluasi berdasarkan classification report

	precision	recall	f1-score	support
0	0.81	0.81	0.81	573
1	0.94	0.93	0.93	1010
2	0.82	0.84	0.83	293
accuracy			0.88	1876
macro avg	0.86	0.86	0.86	1876
weighted avg	0.88	0.88	0.88	1876

continue...

Confusion Matrix

prediction	0	1	2
actual			
0	464	58	51
1	66	941	3
2	40	7	246

```
Butuh informasi lebih lengkap? silakan simak di bawah ini :
Accuracy Average: 0.8800639658848614
F1 Macro Average: 0.8583699513319688
F1 Micro Average: 0.8800639658848614
Precision Macro Average: 0.8564742538918536
Precision Micro Average: 0.8800639658848614
Recall Macro Average: 0.8603489119706961
Recall Micro Average: 0.8800639658848614
```

Evaluasi KNN

Parameter Tuning With GridSearch:

- metric: manhattan
- n_neighbors: 1

Hasil Evaluasi berdasarkan classification report

	precision	recall	f1-score	support
0	0.76	0.74	0.75	573
1	0.88	0.88	0.88	1010
2	0.78	0.81	0.79	293
accuracy			0.83	1876
macro avg	0.81	0.81	0.81	1876
weighted avg	0.83	0.83	0.83	1876

continue...

Confusion Matrix

prediction	0	1	2
actual			
0	424	104	45
1	96	892	22
2	36	21	236

Butuh informasi lebih lengkap? silakan simak di bawah ini :

```
Accuracy Average: 0.8272921108742004
F1 Macro Average: 0.8077239615980732
F1 Micro Average: 0.8272921108742004
Precision Macro Average: 0.8061857649019077
Precision Micro Average: 0.8272921108742004
Recall Macro Average: 0.8095313878903213
Recall Micro Average: 0.8272921108742004
```

Evaluasi Neural Network

Parameter Tuning With RandomSearch:

- hidden_layer_sizes=(100,200,20)
- activation: relu
- alpha: 0.01841809216468458 6,
- solver: adam
- random_state=42

Hasil Evaluasi berdasarkan classification report

	precision	recall	f1-score	support
0	0.74	0.75	0.75	573
1	0.90	0.91	0.91	1010
2	0.77	0.73	0.75	293
accuracy			0.83	1876
macro avg	0.80	0.80	0.80	1876
weighted avg	0.83	0.83	0.83	1876

Confusion Matrix

prediction	0	1	2
actual			
0	432	85	56
1	81	920	9
2	68	12	213

continue...

```
Butuh informasi lebih lengkap? silakan simak di bawah ini :
Accuracy Average: 0.8342217484008528
F1 Macro Average: 0.8008350515248468
F1 Micro Average: 0.8342217484008528
Precision Macro Average: 0.8047846989900299
Precision Micro Average: 0.8342217484008528
Recall Macro Average: 0.7972600826724919
Recall Micro Average: 0.8342217484008528
```

Kesimpulan Klasifikasi

1

Random Forest

Winner!

Dari ketiga model yang diuji, **Random Forest menunjukkan performa yang paling unggul** dalam hampir semua aspek metrik yang dinilai, menjadikannya pilihan model terbaik dari kelompok ini.

REGRESSION TASK

Preprocessing

Duplicate Values

Tidak ada data duplikat

Missing Values

- Melakukan **drop row** pada kolom dengan persentase data yang hilang sebesar < 5%
- Melakukan **drop column subway_offers** karena dianggap terlalu banyak data yang hilang.
- Melakukan **drop column subway_offers** karena dua alasan:
  - Bersifat kategorikal dan terlalu banyak nilai unik, sehingga akan merepotkan proses encoding.
  - Dapat diwakilkan dengan atribut lain, yaitu dist_to_subway.
- Melalukan **imputasi data** untuk sisanya

Feature Engineering

Normalization

Melakukan normalisasi **pada data numerikal** menggunakan z-score, kecuali pada atribut longitude dan latitude.

Encoding

Melakukan encoding pada data kategorikal menggunakan metode **Label Encoding.** Nilai label ditentukan dari rata-rata harga apartemen untuk sebuah jenis nilai pada suatu atribut

Model Selection

Pemilihan model dilakukan dengan melihat nilai metriknya seperti R-squared, root mean squared error, mean squared error, dan mean absolute error. Model yang diuji adalah:

```
linear_reg = LinearRegression()
lasso_reg = Lasso()
ridge_reg = Ridge()
tree_reg = DecisionTreeRegressor(random_state=69)
rf_reg = RandomForestRegressor(n_estimators=100, random_state=69)
xgb_reg = XGBRegressor(random_state=69)
mlp_reg = MLPRegressor(random_state=42)
gboost_reg = GradientBoostingRegressor(n_estimators=100, random_state=69)
lgbm_reg = LGBMRegressor(objective='regression', n_estimators=100)
```

Model Selection

Selanjutnya, diambil tiga model dengan nilai metrik terbaik:

```
Random Forest Regression
========================================
MAE: 940530.1315
MSE: 1676104616186.4866
RMSE: 1294644.5907
R_squared: 0.8231
```

```
XGBoost Regressor
========================================
MAE: 928503.9536
MSE: 1661046135876.402
RMSE: 1288815.7882
R_squared: 0.8247
```

```
Light Gradient Boosting Regressor
========================================
[LightGBM] [Info] Auto-choosing col-wi
You can set `force_col_wise=true` to r
[LightGBM] [Info] Total Bins 1369
[LightGBM] [Info] Number of data point
[LightGBM] [Info] Start training from
MAE: 915402.8216
MSE: 1571641054744.855
RMSE: 1253651.0897
R_squared: 0.8341
```

Model

1. Random Forest
2. XGBoost Regressor
3. LGBM Regressor

Evaluasi Random Forest

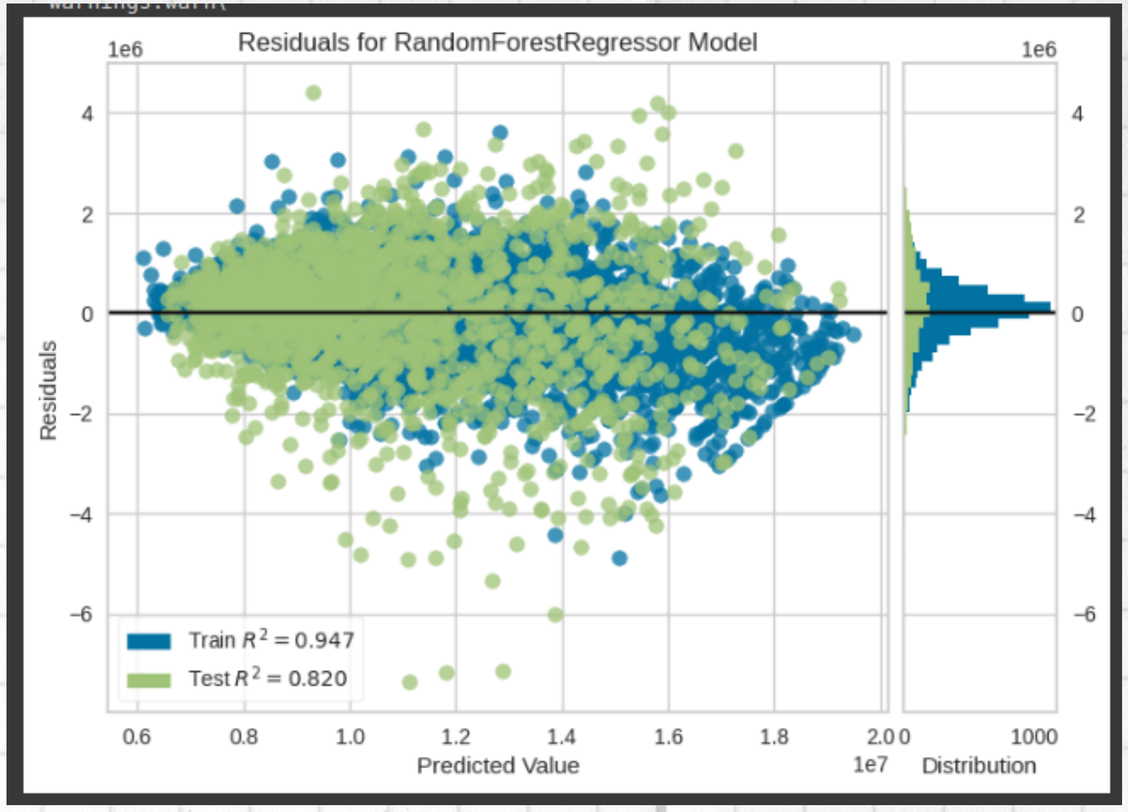
Parameter Tuning With GridSearch:

- random_state: 69,
- n_estimators: 1000,
- learning_rate: 0.312,
- max_depth: 5

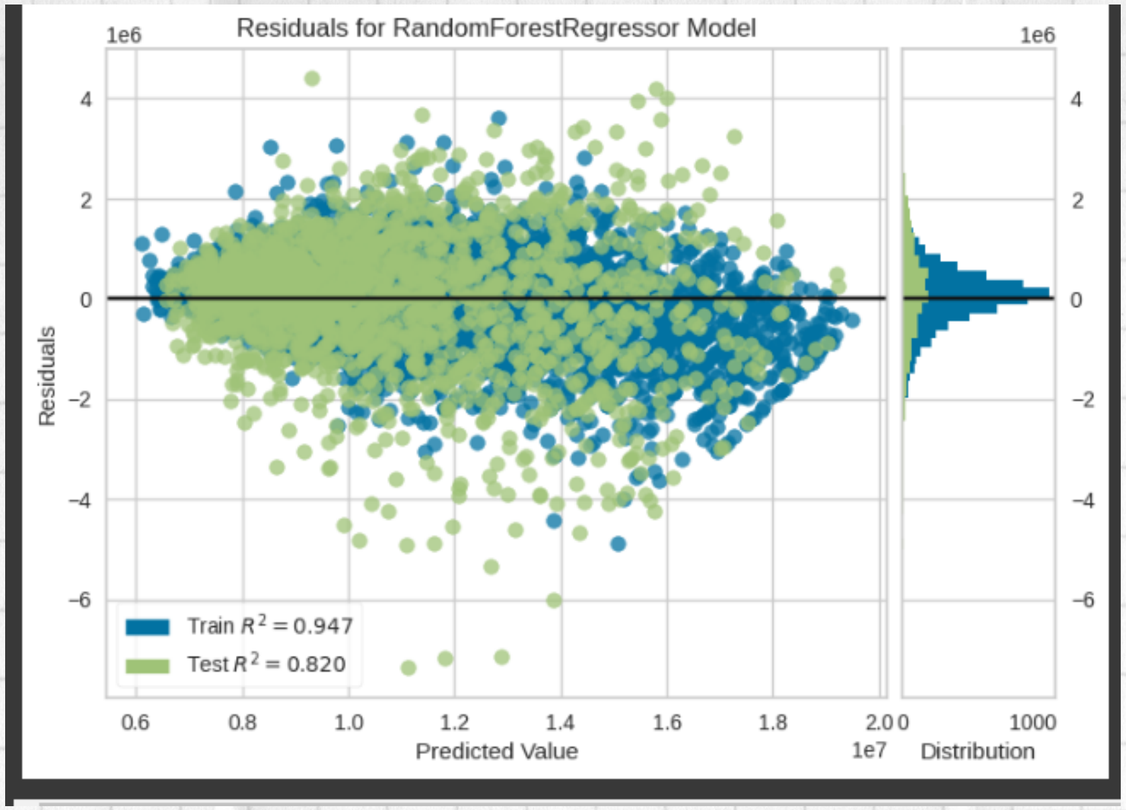
```
Random Fast Regression
==========================================
MAE: 951870.8264
MSE: 1701214224763.3608
RMSE: 1304306.0319
R_squared: 0.8204
```

continue...

Before Hyperparameter Tuning



After Hyperparameter Tuning



Evaluasi XGBoost Regressor

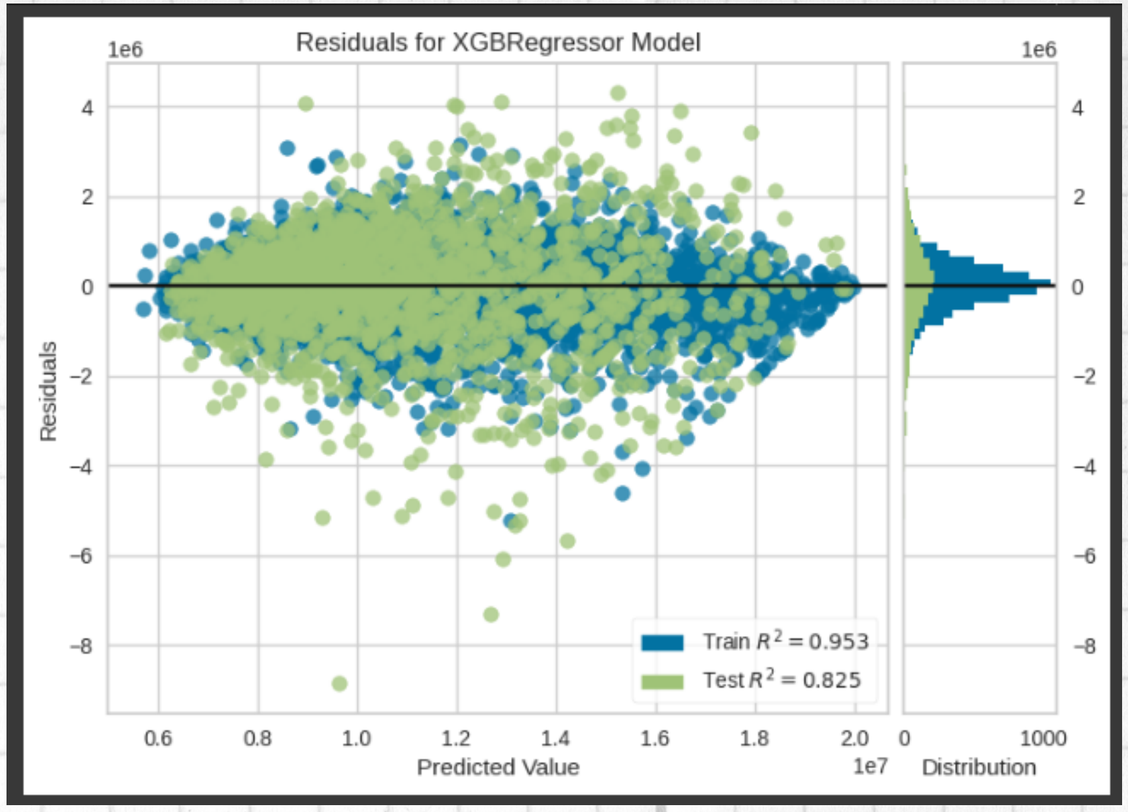
Parameter Tuning With RandomSearch:

- random_state: 69,
- n_estimators: 1000,
- learning_rate: 0.312,
- max_depth: 5

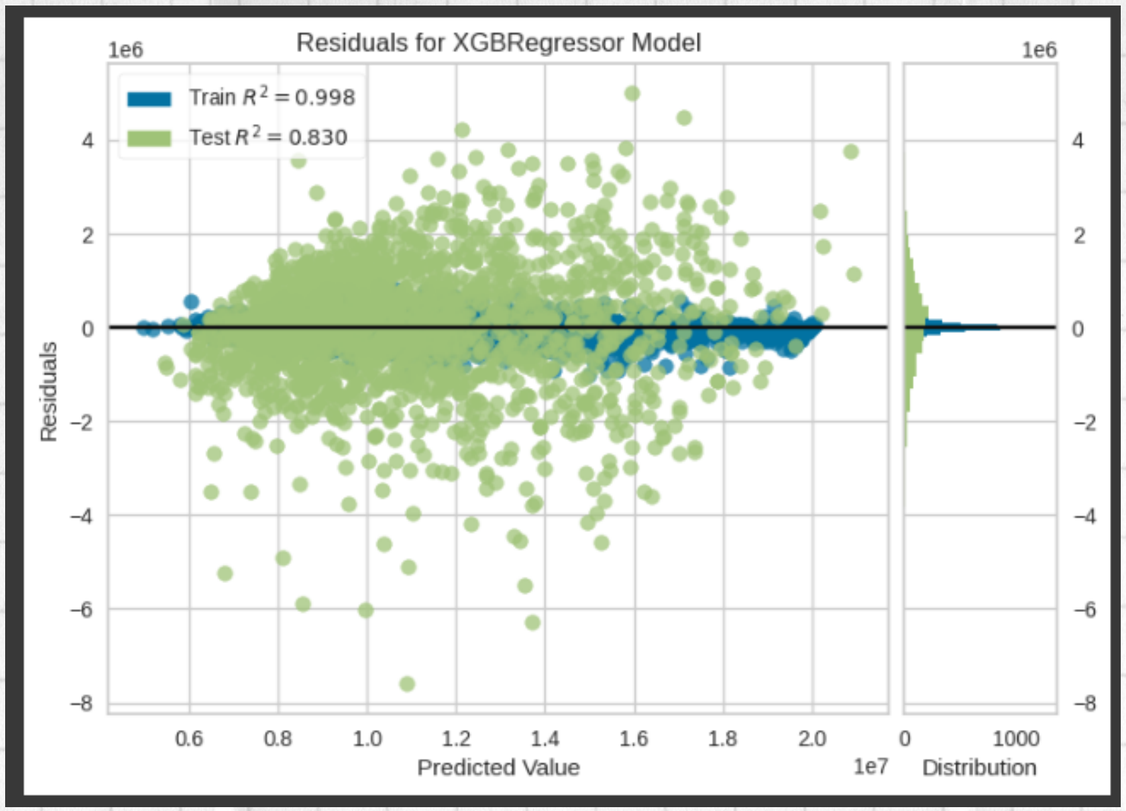
```
XGBRegressor
==========================================
MAE: 923914.2655
MSE: 1609483511633.76
RMSE: 1268654.2128
R_squared: 0.8301
```

continue...

Before Hyperparameter Tuning



After Hyperparameter Tuning



Evaluasi LGBM Regressor

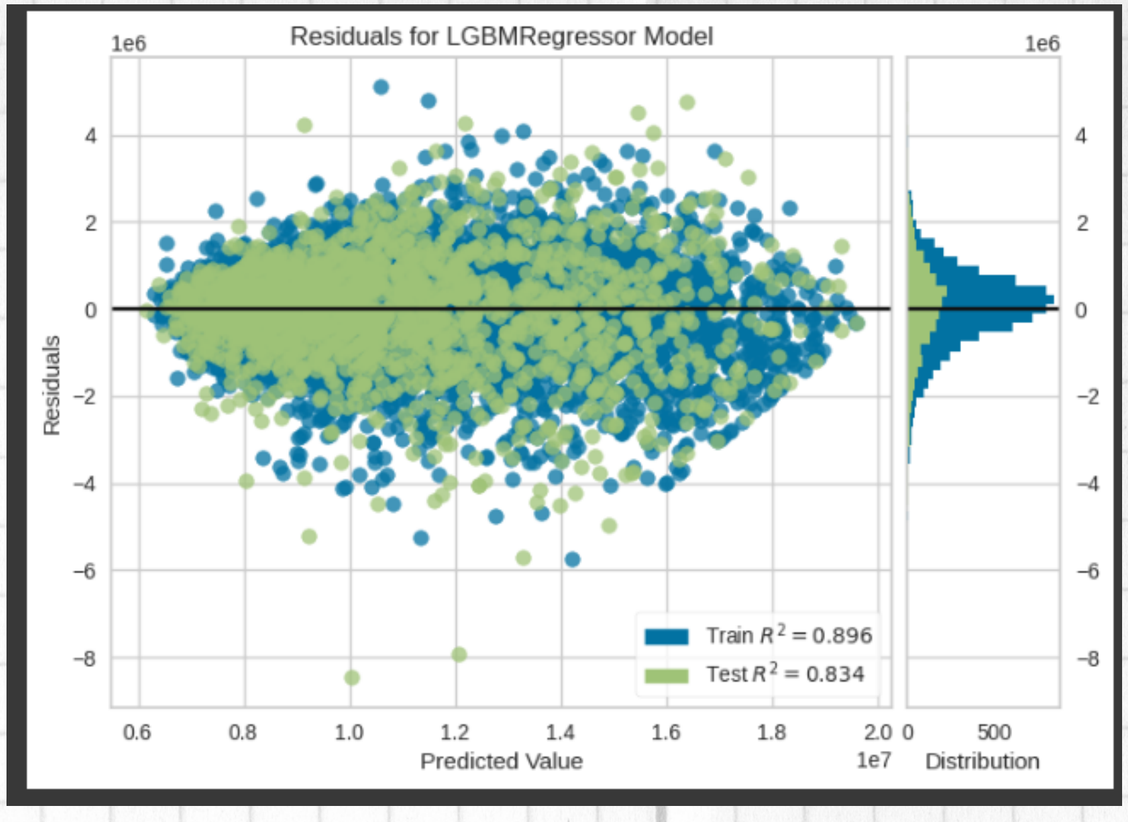
Parameter Tuning With GridSearch:

- colsample_bytree: 0.7
- learning_rate: 0.05
- n_estimators: 1000

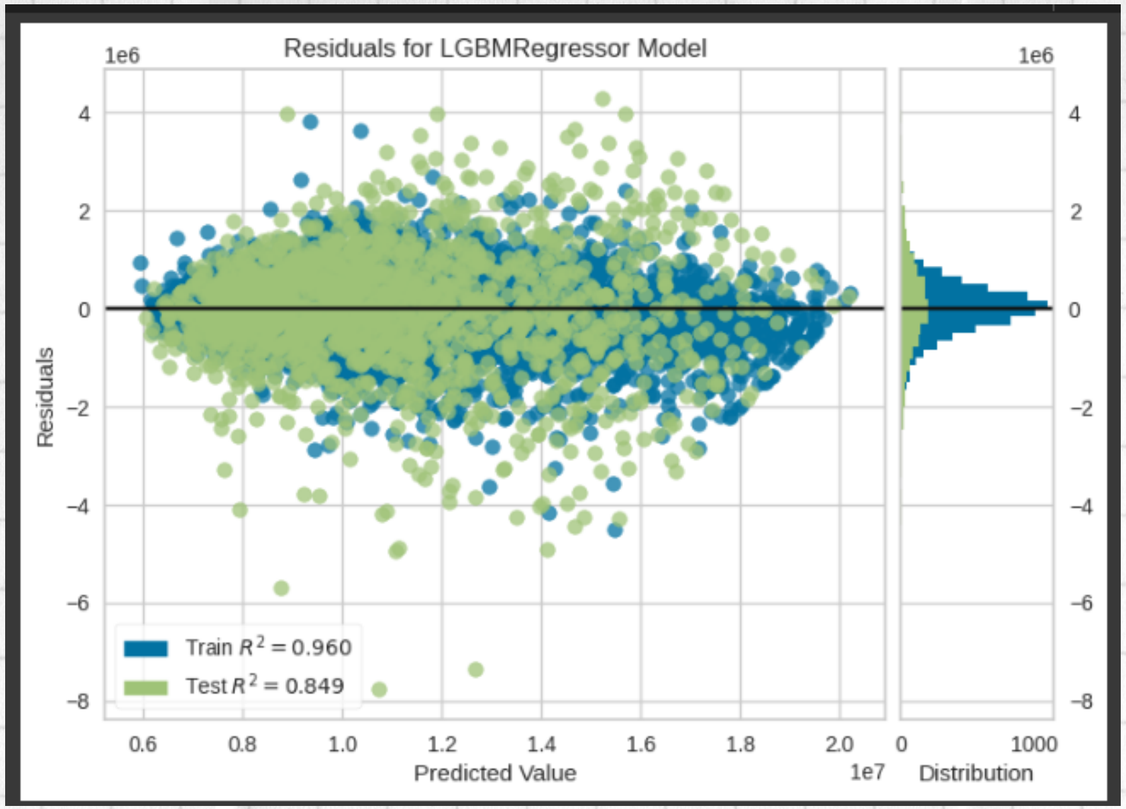
```
LGBMRegressor
========================================
[LightGBM] [Info] Auto-choosing col-wise
You can set `force_col_wise=true` to remo
[LightGBM] [Info] Total Bins 1369
[LightGBM] [Info] Number of data points i
[LightGBM] [Info] Start training from sco
MAE: 858796.3184
MSE: 1432345274273.2698
RMSE: 1196806.281
R_squared: 0.8488
```

continue...

Before Hyperparameter Tuning



After Hyperparameter Tuning



Kesimpulan Regresi

1

LGBM Regressor

Winner!

Dari ketiga model yang diuji, **LGBM Regressor menunjukkan performa yang paling unggul** dalam hampir semua aspek metrik yang dinilai, menjadikannya pilihan model terbaik dari kelompok ini.

CLUSTERING TASK

Konfigurasi Clustering

Banyak Cluster

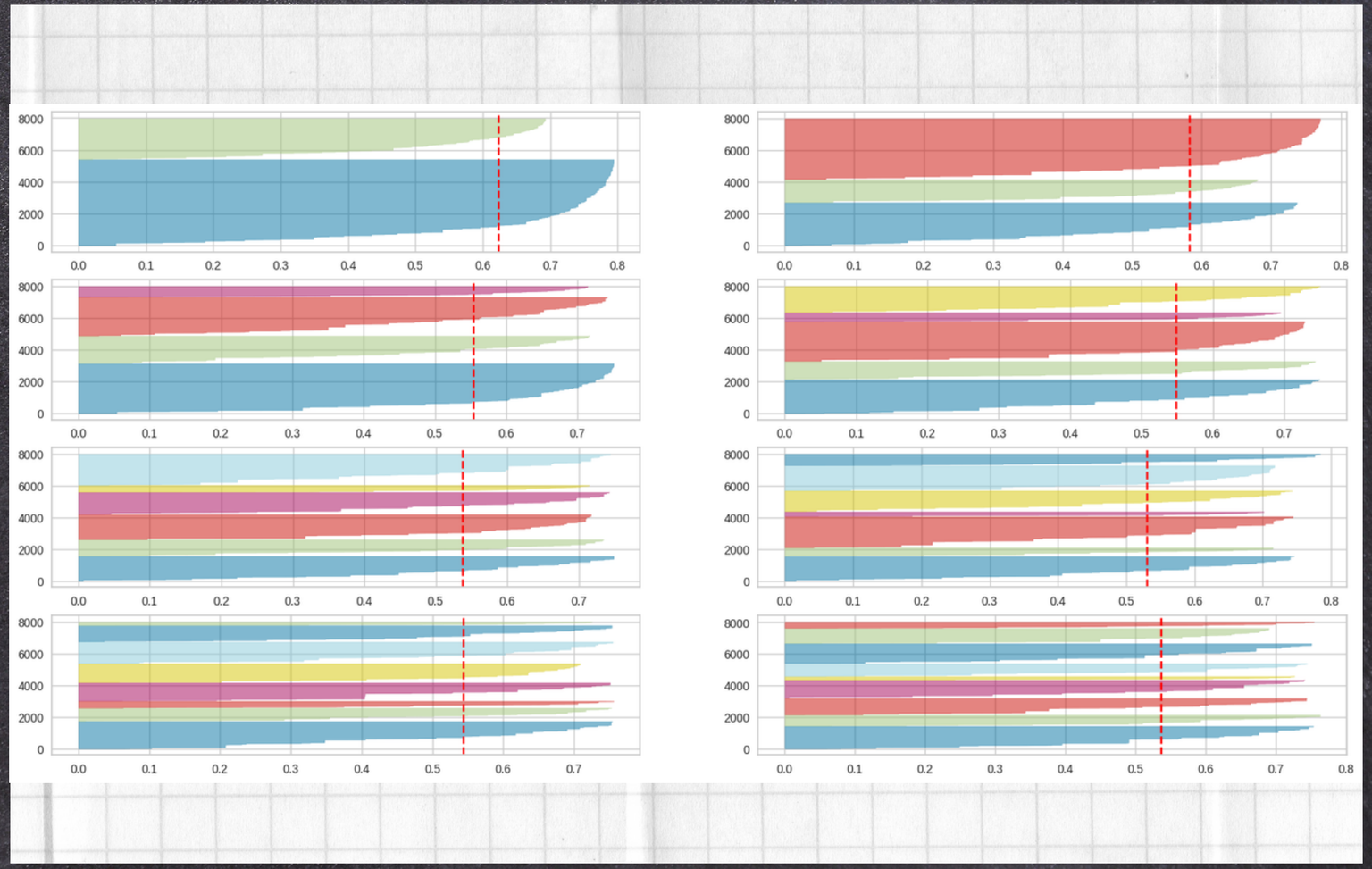
Mencari probabilitas cluster dengan silhouette coefficient tertinggi.

Fitur yang dijadikan acuan

Berdasarkan subjektivitas kelompok, kami menggunakan:

1. District
2. Jarak ke subway terdekat
3. Umur bangunan
4. Kebutuhan perbaikan bangunan
5. Material bangunan
6. Harga bangunan

Percobaan Jumlah Cluster

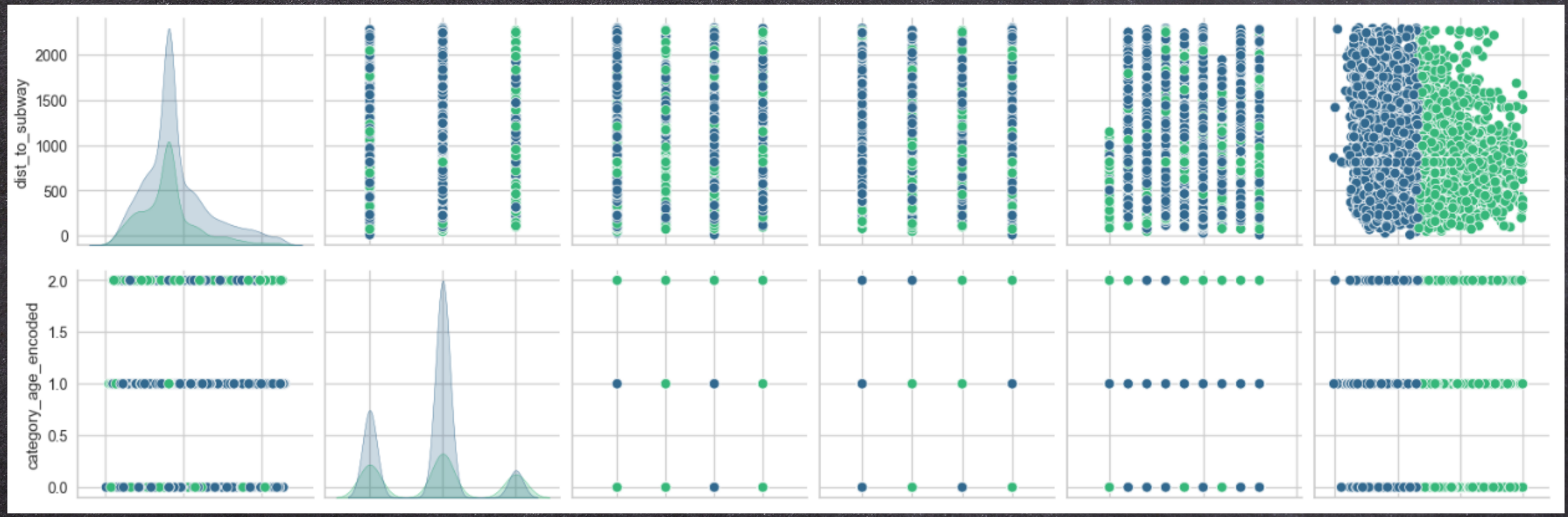


Percobaan Jumlah Cluster

n_clusters	average_silhouette_coefficient
2	0.624718
3	0.585109
4	0.553345
5	0.552312
6	0.535986
7	0.538625
8	0.544021
9	0.541289

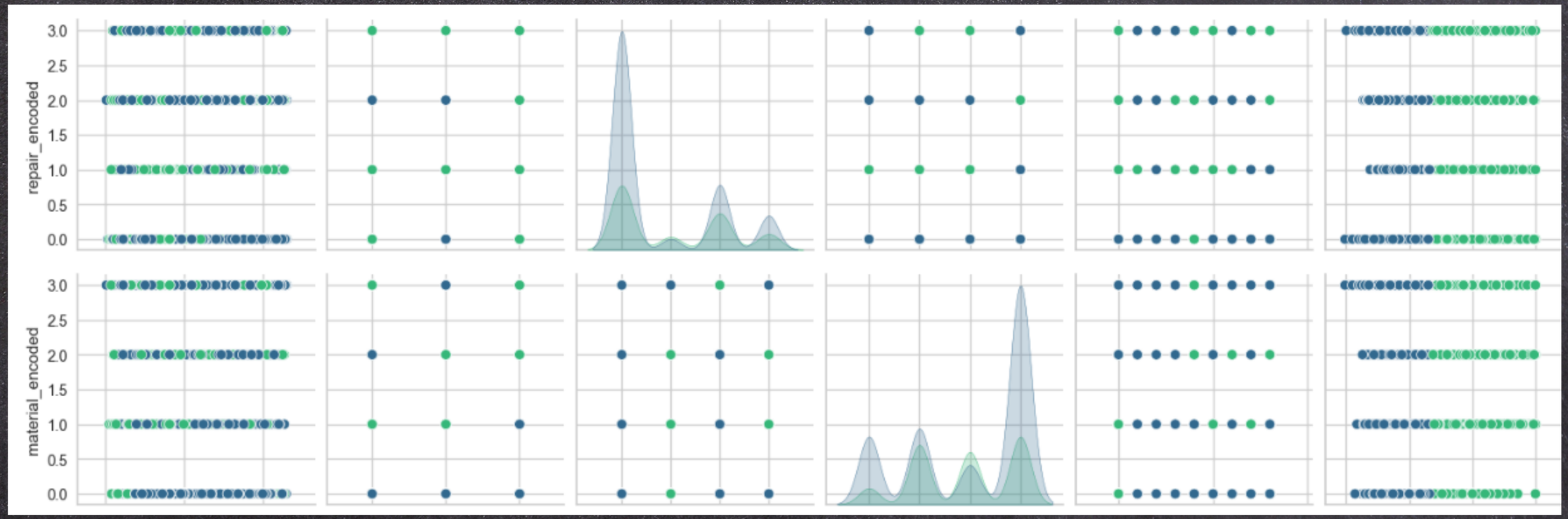
Hasil Clustering

Terhadap jarak ke subway dan usia bangunan



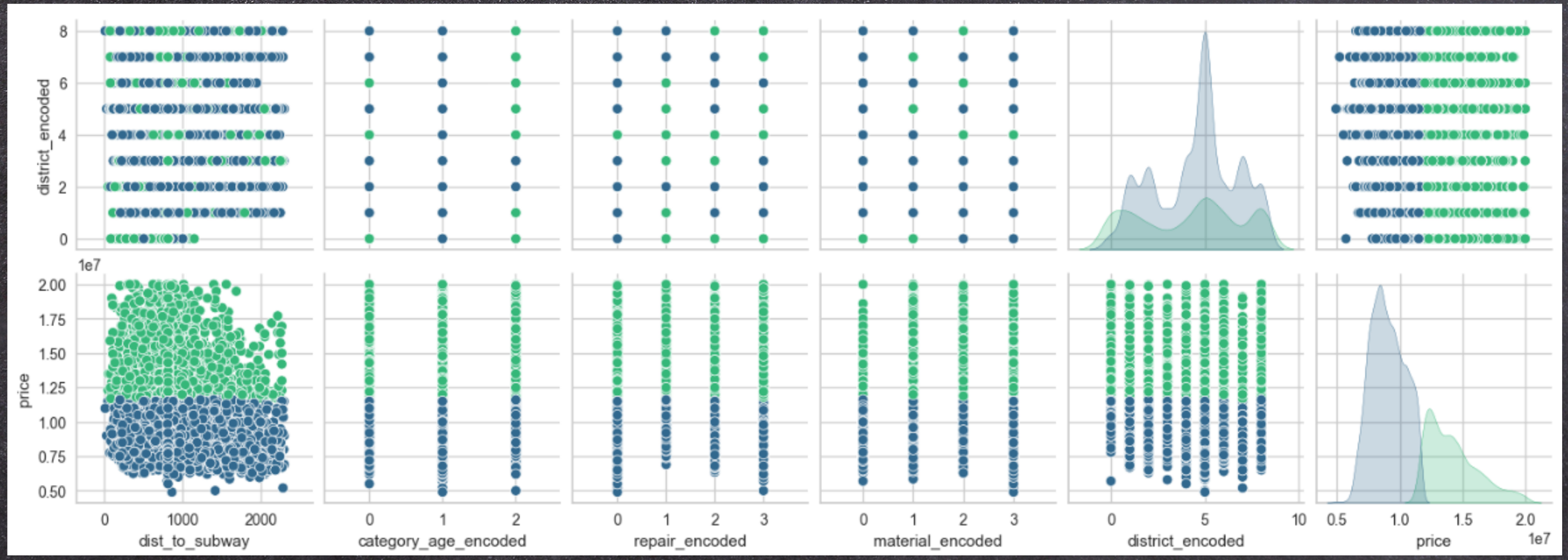
Hasil Clustering

Terhadap kebutuhan repair dan material bangunan



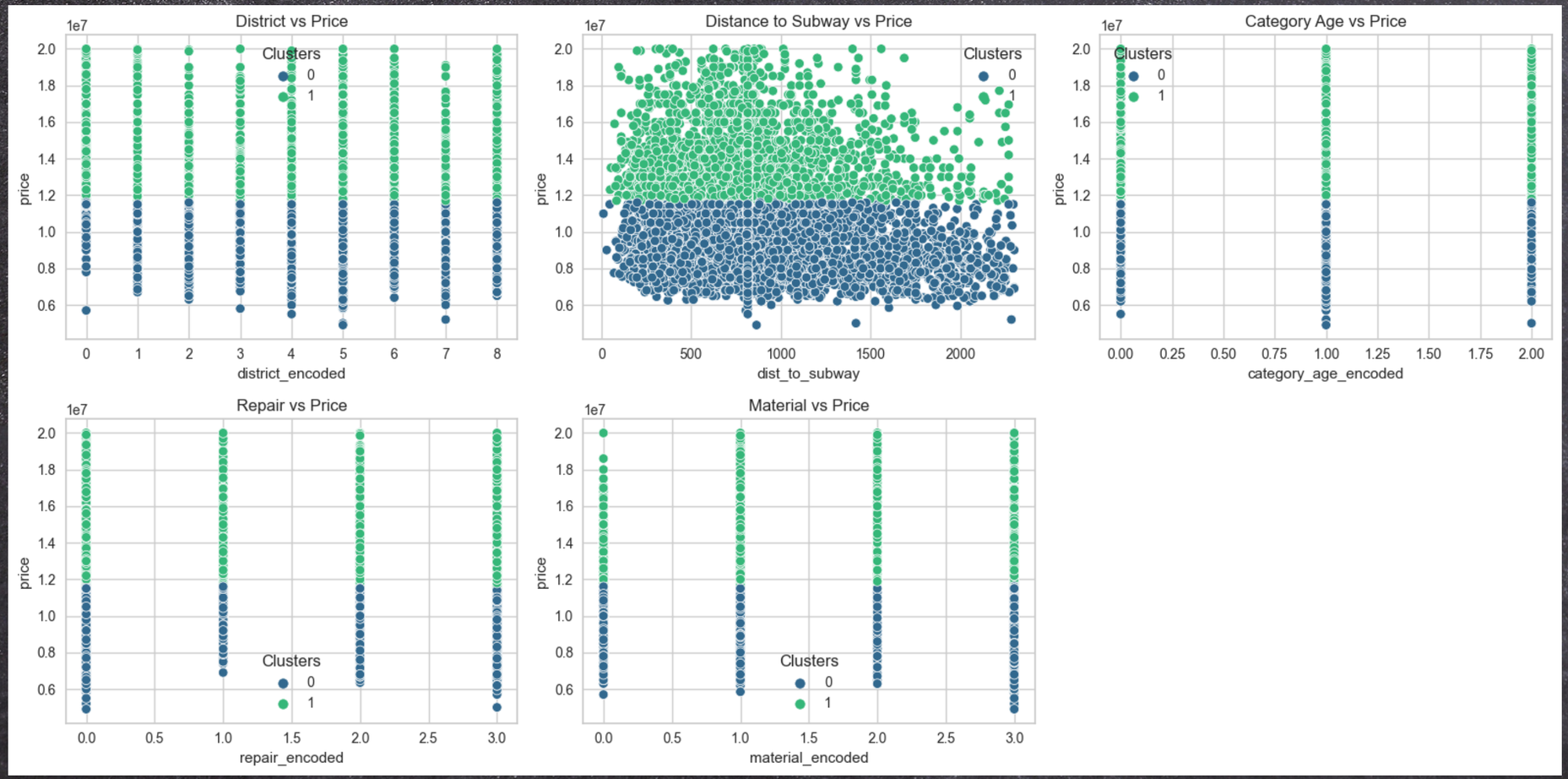
Hasil Clustering

Terhadap distrik dan harga bangunan



Hasil Clustering

Terkhusus terhadap harga



Hasil Clustering

Kesimpulan

Dengan asumsi bahwa hijau = mahal dan biru = murah, maka harga suatu rumah tidak terikat dengan material, kebutuhan repair, umur bangunan, jarak dengan subway, dan distrik rumah tersebut. Kesimpulan tersebut diperoleh dari ratanya distribusi kedua cluster.


THANK
YOU