LAPORAN

TUGAS SUB-CPMK 2

MATA KULIAH DATA MINING

DISUSUN OLEH:

Muhammad Aryasatya Nugroho (22083010085)

DOSEN PENGAMPU:

Trimono, S.Si., M.Si. (NIP. 211199 50 908269)

PROGRAM STUDI SAINS DATA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR

2024

STUDI KASUS

Diberikan *dataset* perumahan di kota Melbourne, Australia. Lakukan *pre-processing dataset* tersebut menggunakan beberapa metode yang relevan dengan studi kasus agar *dataset* siap untuk dianalisis lebih lanjut.

PENYELESAIAN

Pre-processing

Proses *pre-processing* merupakan sebuah tahap penting dalam analisis data yang bertujuan untuk membersihkan dan memperisapkan data mentah sebelum digunakan untuk pembangunan model atau analisis lebih lanjut. Proses ini meliputi beberapa langkah penting yang membantu memastikan keakuratan, keandalan, serta kegunaan data. Salah satu langkah utama dalam *pre-processing* adalah mendeteksi dan menangani nilai yang hilang (*missing value*). Hal ini dilakukan dengan cara mengidentifikasi apakah ada entri data yang kosong atau tidak lengkap, dan kemudian dapat diisi atau dihapus sesuai kebutuhan.

Selain itu, dalam tahap *pre-processing* juga melibatkan deteksi dan penanganan *outlier*, yaitu data ekstrem atau data yang berbeda secara signifikan dari pola umum dalam *dataset*. Langkah-langkah lain mungkin termasuk mengonversi data kategorikal menjadi bentuk yang dapat diproses oleh model, misalkan menggunakan teknik *one-hot encoding*. Serta normalisasi atau pelaksanaan fitur untuk memastikan bahwa mereka memiliki rentang nilai yang serupa. Keseluruhan dalam proses *pre-processing* ini merupakan pondasi penting dalam memastikan kualitas dan integritas data sebelum dilakukan pemodelan atau analisis lebih lanjut.

Implementasi Studi Kasus Menggunakan Pemrograman Python

Dataset yang digunakan merupakan data perumahan di kota Melbourne. *Dataset* ini memuat kumpulan data yang mengandung informasi tentang properti yang dijual. *Dataset* ini mencakup berbagai atribut seperti lokasi properti, jumlah kamar, harga jual, metode penjualan, dan lainnya. Secara keseluruhan, *dataset* terdiri dari 19 kolom yang mencakup berbagai aspek dari properti, mulai dari detail fisik seperti jumlah kamar dan luas tanah, hingga informasi mengenai lokasi geografis properti di pinggiran kota. Namun perlu diperhatikan *dataset* ini masih mengandung beberapa *missing value* di beberapa kolom, yang perlu ditangani sebelum dilakukan analisis lebih lanjut. Terlebih lagi beberapa kolom yang berisi data kategorikal perlu untuk dilakukan proses *encode* sebelum digunakan dalam pembuatan model. Tahap *pre-processing* akan dijelaskan setiap tahapannya dibawah ini.

1. Persiapan Dataset

```
data = pd.read_csv("Perumahan Kota Melbourne.csv")
data
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	CouncilArea	Lattitude	Longtitude	Regionname	Propertycount
0	Abbotsford	85 Turner St	2	h	1480000	S	Biggin	2.5	2	1	1.0	202	NaN	NaN	Yarra	-37.79960	144.99840	Northern Metropolitan	4019
1	Abbotsford	25 Bloomburg St	2	h	1035000	S	Biggin	2.5	2	1	0.0	156	79.0	1900.0	Yarra	-37.80790	144.99340	Northern Metropolitan	4019
2	Abbotsford	5 Charles St	3	h	1465000	SP	Biggin	2.5	3	2	0.0	134	150.0	1900.0	Yarra	-37.80930	144.99440	Northern Metropolitan	4019
3	Abbotsford	40 Federation La	3	h	850000	PI	Biggin	2.5	3	2	1.0	94	NaN	NaN	Yarra	-37.79690	144.99690	Northern Metropolitan	4019
4	Abbotsford	55a Park St	4	h	1600000	VB	Nelson	2.5	3	1	2.0	120	142.0	2014.0	Yarra	-37.80720	144.99410	Northern Metropolitan	4019
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13575	Wheelers Hill	12 Strada Cr	4	h	1245000	S	Barry	16.7	4	2	2.0	652	NaN	1981.0	NaN	-37.90562	145.16761	South-Eastern Metropolitan	7392
13576	Williamstown	77 Merrett Dr	3	h	1031000	SP	Williams	6.8	3	2	2.0	333	133.0	1995.0	NaN	-37.85927	144.87904	Western Metropolitan	6380
13577	Williamstown	83 Power St	3	h	1170000	S	Raine	6.8	3	2	4.0	436	NaN	1997.0	NaN	-37.85274	144.88738	Western Metropolitan	6380
13578	Williamstown	96 Verdon St	4	h	2500000	PI	Sweeney	6.8	4	1	5.0	866	157.0	1920.0	NaN	-37.85908	144.89299	Western Metropolitan	6380
13579	Yarraville	6 Agnes St	4	h	1285000	SP	Village	6.3	4	1	1.0	362	112.0	1920.0	NaN	-37.81188	144.88449	Western Metropolitan	6543

Load Dataset dan *Outputnya*

2. Pre-Processing

a) Handling Missing Value

```
median_BuildingArea = data['BuildingArea'].median()
data['BuildingArea'].fillna(median_BuildingArea, inplace=True)

data
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	Lattitude	Longtitude	Regionname	Propertycount
0	Abbotsford	85 Turner St	2	h	1480000	S	Biggin	2.5	2	1	1.0	202	126.0	NaN	-37.79960	144.99840	Northern Metropolitan	4019
1	Abbotsford	25 Bloomburg St	2	h	1035000	S	Biggin	2.5	2	1	0.0	156	79.0	1900.0	-37.80790	144.99340	Northern Metropolitan	4019
2	Abbotsford	5 Charles St	3	h	1465000	SP	Biggin	2.5	3	2	0.0	134	150.0	1900.0	-37.80930	144.99440	Northern Metropolitan	4019
3	Abbotsford	40 Federation La	3	h	850000	PI	Biggin	2.5	3	2	1.0	94	126.0	NaN	-37.79690	144.99690	Northern Metropolitan	4019
4	Abbotsford	55a Park St	4	h	1600000	VB	Nelson	2.5	3	1	2.0	120	142.0	2014.0	-37.80720	144.99410	Northern Metropolitan	4019
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13575	Wheelers Hill	12 Strada Cr	4	h	1245000	S	Barry	16.7	4	2	2.0	652	126.0	1981.0	-37.90562	145.16761	South-Eastern Metropolitan	7392
13576	Williamstown	77 Merrett Dr	3	h	1031000	SP	Williams	6.8	3	2	2.0	333	133.0	1995.0	-37.85927	144.87904	Western Metropolitan	6380
13577	Williamstown	83 Power St	3	h	1170000	S	Raine	6.8	3	2	4.0	436	126.0	1997.0	-37.85274	144.88738	Western Metropolitan	6380
13578	Williamstown	96 Verdon St	4	h	2500000	PI	Sweeney	6.8	4	1	5.0	866	157.0	1920.0	-37.85908	144.89299	Western Metropolitan	6380
13579	Yarraville	6 Agnes St	4	h	1285000	SP	Village	6.3	4	1	1.0	362	112.0	1920.0	-37.81188	144.88449	Western Metropolitan	6543

Handling Missing Value dengan Imputasi Median

Kode di atas bertujuan untuk menangani nilai yang hilang (*missing values*) pada kolom 'BuildingArea' dengan mengisi menggunakan metode imputasi median. Median dipilih sebagai metode pengisian karena median lebih tahan terhadap *outlier* daripada *mean*, sehingga lebih cocok digunakan dalam kasus-kasus di mana data memiliki *outlier* atau distribusi yang tidak simetris.

```
median_YearBuilt = data['YearBuilt'].median()
data['YearBuilt'].fillna(median_YearBuilt, inplace=True)

data
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	CouncilArea	Lattitude	Longtitude	Regionname	Propertycount
0	Abbotsford	85 Turner St	2	h	1480000	S	Biggin	2.5	2	1	1.0	202	126.0	1970.0	Yarra	-37.79960	144.99840	Northern Metropolitan	4019
1	Abbotsford	25 Bloomburg St	2	h	1035000	S	Biggin	2.5	2	1	0.0	156	79.0	1900.0	Yarra	-37.80790	144.99340	Northern Metropolitan	4019
2	Abbotsford	5 Charles St	3	h	1465000	SP	Biggin	2.5	3	2	0.0	134	150.0	1900.0	Yarra	-37.80930	144.99440	Northern Metropolitan	4019
3	Abbotsford	40 Federation La	3	h	850000	PI	Biggin	2.5	3	2	1.0	94	126.0	1970.0	Yarra	-37.79690	144.99690	Northern Metropolitan	4019
4	Abbotsford	55a Park St	4	h	1600000	VB	Nelson	2.5	3	1	2.0	120	142.0	2014.0	Yarra	-37.80720	144.99410	Northern Metropolitan	4019
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13575	Wheelers Hill	12 Strada Cr	4	h	1245000	S	Barry	16.7	4	2	2.0	652	126.0	1981.0	NaN	-37.90562	145.16761	South-Eastern Metropolitan	7392
13576	Williamstown	77 Merrett Dr	3	h	1031000	SP	Williams	6.8	3	2	2.0	333	133.0	1995.0	NaN	-37.85927	144.87904	Western Metropolitan	6380
13577	Williamstown	83 Power St	3	h	1170000	S	Raine	6.8	3	2	4.0	436	126.0	1997.0	NaN	-37.85274	144.88738	Western Metropolitan	6380
13578	Williamstown	96 Verdon St	4	h	2500000	PI	Sweeney	6.8	4	1	5.0	866	157.0	1920.0	NaN	-37.85908	144.89299	Western Metropolitan	6380
13579	Yarraville	6 Agnes St	4	h	1285000	SP	Village	6.3	4	1	1.0	362	112.0	1920.0	NaN	-37.81188	144.88449	Western Metropolitan	6543

Handling Missing Value dengan Imputasi Median

Kode di atas bertujuan untuk menangani nilai yang hilang (*missing values*) pada kolom 'YearBuilt' dengan mengisi menggunakan metode imputasi median. Median dipilih sebagai metode pengisian karena median lebih tahan terhadap *outlier* daripada *mean*, sehingga lebih cocok digunakan dalam kasus-kasus di mana data memiliki *outlier* atau distribusi yang tidak simetris.

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# Pemilihan kolom atau fitur yang akan digunakan untuk perhitungan
selected_features = ["Suburb", "Distance", "Regionname"]

# Pemisahan data menjadi dua bagian, satu "CouncilArea" yang tidak memiliki "Nan" dan memiliki "Nan"
data_train = data.dropna(subset=["CouncilArea"]).copy()  # Data training
data_predict = data[data["CouncilArea"].isnull()].copy()  # Data testing

# Pemisahan kolom atau fitur dan target untuk data yang lengkap
X_train = data_train[selected_features]
y_train = data_train["CouncilArea"]

# Penggunaan ColumnTransformer untuk mengubah data kategorikal menjadi numerik
categorical_features = ["Suburb", "Regionname"]
categorical_transformer = Pipeline(steps=[("onehot", OneHotEncoder(handle_unknown="ignore"))])
preprocessor = ColumnTransformer(transformers=[("cat", categorical_transformer, categorical_features)])

# Pembuatan pipeline untuk preprocessing dan model
clf = Pipeline(steps=[("preprocessor", preprocessor), ("classifier", RandomForestClassifier())])

# Pelatihan pembentukan model
clf.fit(X_train, y_train)

# Prediksi untuk data yang memiliki nilai "NaN" pada kolom "CouncilArea"
X_predict = data_predict[selected_features]
predicted_council_area = clf.predict(X_predict)

# Penyimpanan hasil prediksi
data_predict.loc[:, "CouncilArea"] = predicted_council_area
data = pd.concat([data_train, data_predict], axis=0)
data
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	CouncilArea	Lattitude	Longtitude	Regionname	Propertycount
0	Abbotsford	85 Turner St	2	h	1480000	S	Biggin	2.5	2	1	1.0	202	126.0	1970.0	Yarra	-37.79960	144.99840	Northern Metropolitan	4019
1	Abbotsford	25 Bloomburg St	2	h	1035000	S	Biggin	2.5	2	1	0.0	156	79.0	1900.0	Yarra	-37.80790	144.99340	Northern Metropolitan	4019
2	Abbotsford	5 Charles St	3	h	1465000	SP	Biggin	2.5	3	2	0.0	134	150.0	1900.0	Yarra	-37.80930	144.99440	Northern Metropolitan	4019
3	Abbotsford	40 Federation La	3	h	850000	PI	Biggin	2.5	3	2	1.0	94	126.0	1970.0	Yarra	-37.79690	144.99690	Northern Metropolitan	4019
4	Abbotsford	55a Park St	4	h	1600000	VB	Nelson	2.5	3	1	2.0	120	142.0	2014.0	Yarra	-37.80720	144.99410	Northern Metropolitan	4019
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13575	Wheelers Hill	12 Strada Cr	4	h	1245000	S	Barry	16.7	4	2	2.0	652	126.0	1981.0	Monash	-37.90562	145.16761	South-Eastern Metropolitan	7392
13576	Williamstown	77 Merrett Dr	3	h	1031000	SP	Williams	6.8	3	2	2.0	333	133.0	1995.0	Hobsons Bay	-37.85927	144.87904	Western Metropolitan	6380
13577	Williamstown	83 Power St	3	h	1170000	S	Raine	6.8	3	2	4.0	436	126.0	1997.0	Hobsons Bay	-37.85274	144.88738	Western Metropolitan	6380
13578	Williamstown	96 Verdon St	4	h	2500000	PI	Sweeney	6.8	4	1	5.0	866	157.0	1920.0	Hobsons Bay	-37.85908	144.89299	Western Metropolitan	6380
13579	Yarraville	6 Agnes St	4	h	1285000	SP	Village	6.3	4	1	1.0	362	112.0	1920.0	Maribyrnong	-37.81188	144.88449	Western Metropolitan	6543

Handling Missing Value dengan Imputasi Model Random Forest

Kode di atas bertujuan untuk menangani nilai yang hilang (*missing values*) pada kolom 'CouncilArea' dengan menggunakan model *Random Forest* untuk melakukan imputasi. *Random Forest* dipilih karena merupakan salah satu model yang kuat untuk menangani data kategorikal, seperti pada kasus ini di mana kolom 'CouncilArea' merupakan data kategorikal. *Random Forest* memiliki kemampuan untuk menangani kategori yang tidak *linear* dan interaksi antara fitur dengan baik, sehingga cocok digunakan untuk menangani variabel kategorikal seperti 'CouncilArea'. Selain itu, *Random Forest* juga mampu mengatasi *overfitting* dan memiliki kemampuan untuk menangani sejumlah besar fitur dengan baik, sehingga menjadi pilihan yang tepat untuk melakukan imputasi pada data kategorik seperti kolom diatas.

b) Deteksi dan Tangani Outlier

```
numeric_cols = data.select_dtypes(include=['float64', 'int64'])

#define function untuk menangani outlier
def outliers_handling(column):

    Q1 = column.quantile(0.25)
    Q3 = column.quantile(0.75)
    IQR = Q3 - Q1
    low_bound = Q1 - 1.5 * IQR
    up_bound = Q3 + 1.5 * IQR
    column = column.mask(column < low_bound, low_bound)
    column = column.mask(column > up_bound, up_bound)
    return column

#pemrosesan setiap kolom numerik
for col in numeric_cols.columns:
    numeric_cols[col] = outliers_handling(numeric_cols[col])
data[numeric_cols.columns] = numeric_cols

data
```

Suburb	Address	Rooms	Type	Price	Method	SellerG	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	CouncilArea	Lattitude	Longtitude
Abbotsford	85 Turner St	2.0	h	1480000	S	Biggin	2.5	2.0	1.0	1.0	202	126.00	1970.0	Yarra	-37.79960	144.99840
Abbotsford	25 Bloomburg St	2.0	h	1035000	S	Biggin	2.5	2.0	1.0	0.0	156	110.09	1937.5	Yarra	-37.80790	144.99340
Abbotsford	5 Charles St	3.0	h	1465000	SP	Biggin	2.5	3.0	2.0	0.0	134	141.85	1937.5	Yarra	-37.80930	144.99440
Abbotsford	40 Federation La	3.0	h	850000	PI	Biggin	2.5	3.0	2.0	1.0	94	126.00	1970.0	Yarra	-37.79690	144.99690
Abbotsford	55a Park St	4.0	h	1600000	VB	Nelson	2.5	3.0	1.0	2.0	120	141.85	1997.5	Yarra	-37.80720	144.99410

Deteksi dan Penanganan Outliers

Kode diatas digunakan untuk membersihkan *outlier* dalam kolom numerik *DataFrame* menggunakan metode *lower-upper bound*. Pertama kita memikih kolom kemudian mendefinisikan fugnsi *interquartile* untuk setiap kolom, dan menentukan batas bawah dan atas untuk *outlier*. Selanjutnya, fungsi ini mengganti nilai *outlier* yang berada di luar rentang tersebut dengan nilai batas bawah atau atas. Melalui iterasi, kode kemudian menerapkan fungsi ini pada setiap kolom numerik *DataFrame* untuk membersihkan *outlier*.

c) Visualisasi Boxplot

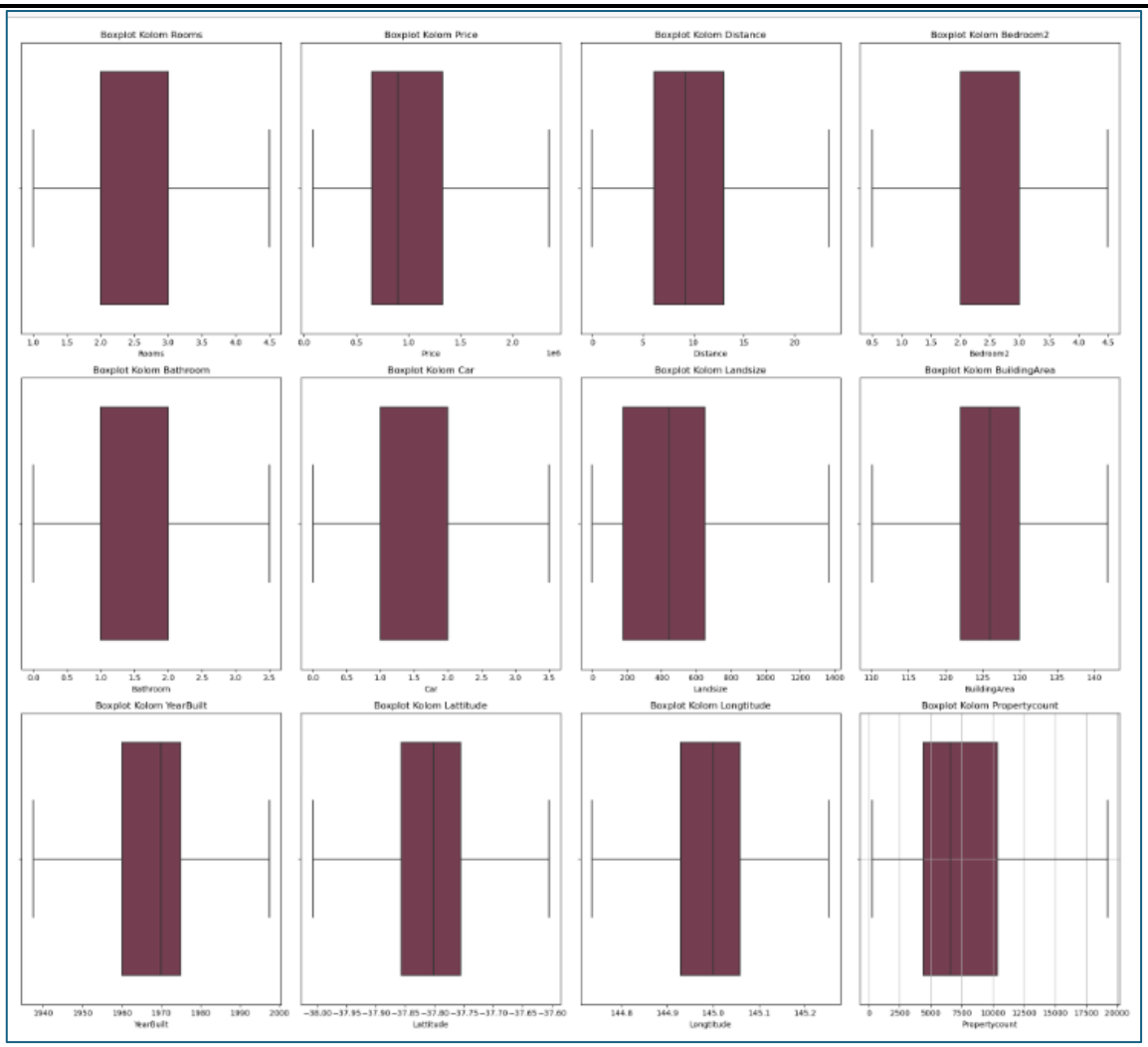
```
import matplotlib.pyplot as plt
import seaborn as sns

color = '#7E354D'
num_cols = numeric_cols.columns
num_plots = len(num_cols)
num_subplots = 4
num_rows = -(-num_plots // num_subplots)

fig, axes = plt.subplots(num_rows, num_subplots, figsize=(20, 6*num_rows))

for i, col in enumerate(num_cols):
    ax = axes[i // num_subplots, i % num_subplots] if num_rows > 1 else axes[i % num_subplots]
    sns.boxplot(x=numeric_cols[col], ax=ax, color=color)
    ax.set_title(f'Boxplot Kolom {col}')

plt.tight_layout()
plt.grid(True)
plt.show()
```



Visualisasi Boxplot dan Outputnya

Kode tersebut digunakan untuk membuat boxplot dari setiap kolom numerik dan menentukan jumlah subplot berdasarkan jumlah kolom numerik. Dalam loop, setiap kolom numerik digunakan untuk membuat boxplot pada subplot yang sesuai, dengan judul yang mencantumkan nama kolom. Terakhir, layout disesuaikan untuk mempermudah interpretasi.

d) Normalisasi Data

```
from sklearn.preprocessing import MinMaxScaler

#hanya untuk kolom numerik
numeric_columns = data.select_dtypes(include=['int', 'float']).columns

#normalisasi menggunakan min-max scaling hanya pada kolom-kolom numerik
scaler = MinMaxScaler()
data[numeric_columns] = scaler.fit_transform(data[numeric_columns])

data
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	CouncilArea	Lattitude	Longtitude	Regionname	Propertycount
0	Abbotsford	85 Turner St	0.285714	h	0.615894	S	Biggin	0.107066	0.375	0.285714	0.285714	0.148311	0.500945	0.541667	Yarra	0.517454	0.508639	Northern Metropolitan	0.198332
1	Abbotsford	25 Bloomburg St	0.285714	h	0.419426	S	Biggin	0.107066	0.375	0.285714	0.000000	0.114537	0.000000	0.000000	Yarra	0.496792	0.498927	Northern Metropolitan	0.198332
2	Abbotsford	5 Charles St	0.571429	h	0.609272	SP	Biggin	0.107066	0.625	0.571429	0.000000	0.098385	1.000000	0.000000	Yarra	0.493306	0.500869	Northern Metropolitan	0.198332
3	Abbotsford	40 Federation La	0.571429	h	0.337748	PI	Biggin	0.107066	0.625	0.571429	0.285714	0.069016	0.500945	0.541667	Yarra	0.524176	0.505725	Northern Metropolitan	0.198332
4	Abbotsford	55a Park St	0.857143	h	0.668874	VB	Nelson	0.107066	0.625	0.285714	0.571429	0.088106	1.000000	1.000000	Yarra	0.498534	0.500287	Northern Metropolitan	0.198332
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13575	Wheelers Hill	12 Strada Cr	0.857143	h	0.512141	S	Barry	0.715203	0.875	0.571429	0.571429	0.478708	0.500945	0.725000	Monash	0.253520	0.837317	South-Eastern Metropolitan	0.375779
13576	Williamstown	77 Merrett Dr	0.571429	h	0.417660	SP	Williams	0.291221	0.625	0.571429	0.571429	0.244493	0.721348	0.958333	Hobsons Bay	0.368907	0.276791	Western Metropolitan	0.322540
13577	Williamstown	83 Power St	0.571429	h	0.479029	S	Raine	0.291221	0.625	0.571429	1.000000	0.320117	0.500945	0.991667	Hobsons Bay	0.385163	0.292991	Western Metropolitan	0.322540
13578	Williamstown	96 Verdon St	0.857143	h	1.000000	PI	Sweeney	0.291221	0.875	0.285714	1.000000	0.635830	1.000000	0.000000	Hobsons Bay	0.369380	0.303888	Western Metropolitan	0.322540
13579	Yarraville	6 Agnes St	0.857143	h	0.529801	SP	Village	0.269807	0.875	0.285714	0.285714	0.265786	0.060139	0.000000	Maribyrnong	0.486884	0.287377	Western Metropolitan	0.331115

Normalisasi Data dan Outputnya

Kode diatas digunakan untuk melakukan normalisasi data menggunakan metode *Min-Max Scaling* pada kolom-kolom numerik dari *DataFrame*. Normalisasi ini bertujuan untuk memastikan rentang nilai setiap kolom numerik seragam, sehingga mencegah keberatannya yang berlebihan pada model yang sensitif terhadap skala seperti klasifikasi dan regresi. Metode *Min-Max Scaling* mengubah setiap nilai dalam kolom menjadi nilai antara 0 dan 1, dengan memperhitungkan nilai minimum dan maksimum dari setiap kolom. Dengan demikian, hasil normalisasi ini dapat meningkatkan kinerja model yang akan dibangun dengan menggunakan data tersebut.

e) Label Encoding

```
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()

#inisialisasi kolom
columns_to_encode = ["Suburb", "Type", "Method", "SellerG", "CouncilArea", "Regionname"]

#label encoding
for column in columns_to_encode:
    data[column] = label_encoder.fit_transform(data[column])
data
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	CouncilArea	Lattitude	Longtitude	Regionname	Propertycount
0	0	85 Turner St	0.285714	0	0.615894	1	23	0.107066	0.375	0.285714	0.285714	0.148311	0.500945	0.541667	31	0.517454	0.508639	2	0.198332
1	0	25 Bloomburg St	0.285714	0	0.419426	1	23	0.107066	0.375	0.285714	0.000000	0.114537	0.000000	0.000000	31	0.496792	0.498927	2	0.198332
2	0	5 Charles St	0.571429	0	0.609272	3	23	0.107066	0.625	0.571429	0.000000	0.098385	1.000000	0.000000	31	0.493306	0.500869	2	0.198332
3	0	40 Federation La	0.571429	0	0.337748	0	23	0.107066	0.625	0.571429	0.285714	0.069016	0.500945	0.541667	31	0.524176	0.505725	2	0.198332
4	0	55a Park St	0.857143	0	0.668874	4	155	0.107066	0.625	0.285714	0.571429	0.088106	1.000000	1.000000	31	0.498534	0.500287	2	0.198332
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13575	302	12 Strada Cr	0.857143	0	0.512141	1	16	0.715203	0.875	0.571429	0.571429	0.478708	0.500945	0.725000	20	0.253520	0.837317	4	0.375779
13576	305	77 Merrett Dr	0.571429	0	0.417660	3	251	0.291221	0.625	0.571429	0.571429	0.244493	0.721348	0.958333	10	0.368907	0.276791	6	0.322540
13577	305	83 Power St	0.571429	0	0.479029	1	194	0.291221	0.625	0.571429	1.000000	0.320117	0.500945	0.991667	10	0.385163	0.292991	6	0.322540
13578	305	96 Verdon St	0.857143	0	1.000000	0	222	0.291221	0.875	0.285714	1.000000	0.635830	1.000000	0.000000	10	0.369380	0.303888	6	0.322540
13579	313	6 Agnes St	0.857143	0	0.529801	3	239	0.269807	0.875	0.285714	0.285714	0.265786	0.060139	0.000000	16	0.486884	0.287377	6	0.331115

Label Encoding dan Outputnya

Kode di atas menggunakan teknik *Label Encoding* untuk mengubah nilai-nilai kategorikal dalam beberapa kolom tertentu menjadi nilai numerik. Proses ini dilakukan untuk kolom-kolom seperti "Suburb", "Type", "Method", "SellerG", "CouncilArea", dan "Regionname". *Label Encoding* bertujuan untuk mengubah setiap nilai dalam kolom kategorikal menjadi bilangan bulat unik, mulai dari 0 hingga jumlah kategori minus 1. Hal ini memungkinkan algoritma *machine learning* yang hanya dapat memproses data numerik untuk dapat bekerja dengan data kategorikal. Dalam proses *preprocessing* data, *Label Encoding* membantu mengonversi data kategorikal menjadi bentuk yang dapat dimengerti oleh model *machine learning*, sehingga meningkatkan akurasi dan kinerja model tersebut.

f) Seleksi Fitur

```
# Drop kolom 'Address' yang tidak relevan
data.drop(columns=['Address'], inplace=True)
data
```

	Suburb	Rooms	Type	Price	Method	SellerG	Distance	Bedroom2	Bathroom	Car	Landsize	BuildingArea	YearBuilt	CouncilArea	Lattitude	Longtitude	Regionname	Propertycount
0	0	0.285714	0	0.615894	1	23	0.107066	0.375	0.285714	0.285714	0.148311	0.500945	0.541667	31	0.517454	0.508639	2	0.198332
1	0	0.285714	0	0.419426	1	23	0.107066	0.375	0.285714	0.000000	0.114537	0.000000	0.000000	31	0.496792	0.498927	2	0.198332
2	0	0.571429	0	0.609272	3	23	0.107066	0.625	0.571429	0.000000	0.098385	1.000000	0.000000	31	0.493306	0.500869	2	0.198332
3	0	0.571429	0	0.337748	0	23	0.107066	0.625	0.571429	0.285714	0.069016	0.500945	0.541667	31	0.524176	0.505725	2	0.198332
4	0	0.857143	0	0.668874	4	155	0.107066	0.625	0.285714	0.571429	0.088106	1.000000	1.000000	31	0.498534	0.500287	2	0.198332
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13575	302	0.857143	0	0.512141	1	16	0.715203	0.875	0.571429	0.571429	0.478708	0.500945	0.725000	20	0.253520	0.837317	4	0.375779
13576	305	0.571429	0	0.417660	3	251	0.291221	0.625	0.571429	0.571429	0.244493	0.721348	0.958333	10	0.368907	0.276791	6	0.322540
13577	305	0.571429	0	0.479029	1	194	0.291221	0.625	0.571429	1.000000	0.320117	0.500945	0.991667	10	0.385163	0.292991	6	0.322540
13578	305	0.857143	0	1.000000	0	222	0.291221	0.875	0.285714	1.000000	0.635830	1.000000	0.000000	10	0.369380	0.303888	6	0.322540
13579	313	0.857143	0	0.529801	3	239	0.269807	0.875	0.285714	0.285714	0.265786	0.060139	0.000000	16	0.486884	0.287377	6	0.331115

Seleksi Fitur dan Outputnya

Kode di atas bertujuan untuk melakukan seleksi fitur dengan menghapus kolom "Address" menggunakan fungsi *drop*. Hal ini dilakukan karena kolom "Address" dianggap tidak cocok untuk *diencode* dan dapat mengganggu analisis perhitungan lebih lanjut pada tahap berikutnya. Dengan menghapus kolom "Address", kita dapat menyederhanakan struktur data dan fokus pada fitur-fitur yang lebih relevan untuk pemodelan atau analisis yang akan dilakukan selanjutnya.

g) Simpan Hasil Pre-Processing ke Dalam File CSV

```
# Simpan DataFrame ke dalam file CSV
data.to_csv('after_preprocessing.csv', index=False)
```

Simpan Hasil Pre-Processing ke Dalam File CSV

Data siap dianalisis lebih lanjut menggunakan model sesuai kebutuhan kita.

KESIMPULAN

Secara keseluruhan, proses *preprocessing* data telah dilakukan dengan beberapa langkah. Pertama, dilakukan penanganan *missing values* dengan mengisi nilai yang hilang menggunakan metode imputasi median pada kolom "YearBuilt". Selanjutnya, *outlier* pada kolom numerik diatasi dengan menggunakan metode *upper-lower bound*. Setelah itu, dilakukan normalisasi data menggunakan *Min-Max Scaling* untuk menjaga konsistensi skala antar fitur. Kemudian, dilakukan label encoding pada beberapa kolom kategorikal untuk mengubah nilai kategorikal menjadi nilai numerik yang dapat diproses oleh model. Terakhir, dilakukan seleksi fitur dengan menghapus kolom "Address" karena dianggap tidak relevan dan tidak cocok untuk diencode. *Preprocessing* data ini bertujuan untuk mempersiapkan data dengan format yang sesuai dan berkualitas baik untuk analisis atau pemodelan selanjutnya.