



Kampus
Merdeka
INDONESIA JAYA

MODUL × ×

Fundamental Data Analyst

Minggu ke-8



Kata Pengantar

Puji syukur Alhamdulillah, penulis panjatkan kehadiran Allah ta'ala, yang telah melimpahkan Rahmat dan Karunia-Nya sehingga pada akhirnya penulis dapat menyelesaikan modul ini dengan baik. Dimana modul ini penulis sajikan dalam bentuk modul yang sederhana. Adapun modul ini penulis buat untuk menambah wawasan para pembaca pada umumnya dan untuk menambah bahan materi untuk mata kuliah Fundamental Data Analyst bagi mahasiswa prodi Sistem Informasi Universitas Bina Sarana Informatika.

Sebagai bahan penulisan diambil berdasarkan pencarian di beberapa sumber, seperti buku, internet dan masih banyak lagi yang lainnya. Dalam modul ini menjelaskan materi Fundamental Data Analyst pertemuan 8 yang membahas tentang Unsupervised Learning (Dimensional Reduction). Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka penulisan dan pembuatan modul ini tidak akan berjalan dengan lancar.

Penulis mengucapkan terima kasih kepada tim sehingga bisa menyelesaikan penyusunan modul ini. Semoga modul ini berguna bagi para pembaca baik mahasiswa ataupun siapapun yang bisa dijadikan bahan referensi untuk pembelajaran.

Agustus 2024

Tim Penyusun

Unit Pengembangan Akademik

Program Studi Sistem Informasi

Daftar Isi

Kata Pengantar	1
Daftar Isi	3
PEMBAHASAN	4
1. Unsupervised Learning (Dimensional Reduction)	4
2. CRISP-DM: Metode Principal Component Analysis (PCA)	4
3. Evaluasi	Error! Bookmark not defined.
Referensi	16

PEMBAHASAN

1. Unsupervised Learning (Dimensional Reduction)

Dimensional Reduction (Reduksi Dimensi) adalah Teknik dalam data mining dan machine learning yang digunakan untuk mengurangi jumlah variable acak dibawah pertimbangan,dengan cara memperoleh sekumpulan variable utama.

Tujuan utama dari reduksi dimensi adalah menyederhanakan data tanpa kehilangan informasi penting, yang pada akhirnya meningkatkan efisiensi dan performa model machine learning.

2. Tujuan Reduksi Dimensi (Dimensional Reduction)

- a. **Mengurangi Komputasi:** Dengan mengurangi jumlah fitur, kita dapat mengurangi waktu komputasi dan kebutuhan memori.
- b. **Mengatasi Curse of Dimensionality:** Ketika jumlah dimensi (fitur) meningkat, volume ruang meningkat secara eksponensial, menyebabkan data menjadi lebih jarang dan sulit untuk dianalisis.
- c. **Memperbaiki Visualisasi Data:** Data dengan dimensi tinggi sulit untuk divisualisasikan. Reduksi dimensi membantu dalam memvisualisasikan data dalam 2D atau 3D.
- d. **Menghilangkan Noise:** Fitur yang tidak relevan atau redundant dapat dihapus, meningkatkan performa model dan interpretabilitas.
- e. **Mengatasi Overfitting:** Dengan mengurangi jumlah fitur, kita dapat mengurangi kompleksitas model, yang dapat membantu mencegah overfitting.

3. Implementasi Dimensional Reduction dengan Python

Studi kasus : House Price

Dataset : <https://s/id/DatasetFDA>

Total Variable : 81 Variabel

4. CRISP-DM (Cross-Industry Standard Process for Data Mining)

a. Business Understanding

Langkah pertama ini berfokus pada pemahaman tujuan bisnis dan persyaratan proyek.

Ini melibatkan:

- 1) Menentukan tujuan bisnis dan kebutuhan yang spesifik.
- 2) Mengidentifikasi masalah bisnis yang ingin diselesaikan.

- 3) Menyusun rencana proyek, termasuk sasaran, anggaran, dan waktu.

Tujuan Bisnis (Study Kasus: House Price):

- 1) Menyederhanakan data tanpa kehilangan informasi penting, yang pada akhirnya meningkatkan efisiensi dan performa model machine learning.
- 2) Mengurangi jumlah Fitur dengan harapan dapat mengurangi waktu komputasi dan kebutuhan memori serta mengurangi kompleksitas model, yang dapat membantu mencegah overfitting.

b. Data Understanding

Langkah kedua melibatkan pengumpulan data dan familiarisasi dengan data yang tersedia. Hal Ini mencakup:

- 1) Mengumpulkan data awal yang diperlukan untuk analisis.
- 2) Menjelajahi data untuk memahami strukturnya, kualitas, dan pola yang ada.
- 3) Mengidentifikasi masalah kualitas data seperti nilai yang hilang atau data yang tidak konsisten.

Studi Kasus : House Price

Dataset : <https://s.id/DatasetFDA>

Deskripsi : dataset house prices memiliki 81 fitur, diantaranya seperti OverallQual, GrLivArea, GarageCars, GarageArea, TotalBsmtSF, FullBath, YearBuilt dan seterusnya (check informasi dasar dari dataset pada pengolahan data)

Sebelum melakukan import library harap install library yellowbrick dengan scrip:
pip install yellowbrick

```

1 import pandas as pd
2 import numpy as np
3 from sklearn.preprocessing import StandardScaler
4 from sklearn.decomposition import PCA
5 from sklearn.discriminant_analysis import
  LinearDiscriminantAnalysis as LDA
6 from sklearn.model_selection import train_test_split
7 from sklearn.metrics import accuracy_score
8 from mlxtend.feature_selection import
  SequentialFeatureSelector
9 from sklearn.preprocessing import OneHotEncoder
10 import matplotlib.pyplot as plt

```

Keterangan Library:

- 1) Manipulasi dan analisis data tabular dimuat dalam bentuk data frame
- 2) Manipulasi data dalam bentuk array
- 3) Bagian dari fungsi library sklearn untuk melakukan standardisasi fitur-fitur dalam dataset
- 4) Mengimpor kelas PCA (Principal Component Analysis) dari modul decomposition yang terdapat dalam Pustaka scikit-learn
- 5) Library untuk untuk menemukan kombinasi fitur yang memisahkan dua atau lebih kelas dari objek
- 6) Bagian dari fungsi library sklearnnya itu untuk Membagi dataset kedalam data training dan testing
- 7) Library untuk menghitung akurasi dari suatu model klasifikasi.
- 8) Library untuk memilih subset fitur yang optimal dari sebuah dataset.
- 9) Library untuk mengubah fitur kategorikal (nominal) menjadi representasi numerik yang dapat digunakan dalam algoritma machine learning
- 10) Library untuk visualisasi data/grafik plot dalam python

```

# Memuat dataset
df = pd.read_csv('HousePriceTrain.csv')
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Id                     1460 non-null   int64
1   MSSubClass             1460 non-null   int64
2   MSZoning               1460 non-null   object
3   LotFrontage           1201 non-null   float64
4   LotArea               1460 non-null   int64
5   Street                1460 non-null   object
6   Alley                 91 non-null     object
7   LotShape              1460 non-null   object
8   LandContour           1460 non-null   object
9   Utilities             1460 non-null   object
10  LotConfig             1460 non-null   object
11  LandSlope             1460 non-null   object
12  Neighborhood          1460 non-null   object
13  Condition1            1460 non-null   object
14  Condition2            1460 non-null   object
15  BldgType              1460 non-null   object
16  HouseStyle            1460 non-null   object
17  OverallQual           1460 non-null   int64
18  OverallCond           1460 non-null   int64
19  YearBuilt              1460 non-null   int64
...
79  SaleCondition         1460 non-null   object
80  SalePrice             1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB

```

c. Data Preparation

Langkah ketiga ini melibatkan pembersihan dan transformasi data agar siap untuk pemodelan. Hal ini termasuk:

- 1) Menghapus fitur yang tidak memiliki pengaruh
- 2) Check Baris yang Kosong
- 3) Hapus Fitur dengan Data Kosong
- 4) Check Fitur dengan type categorical
- 5) Konversi Fitur dengan type categorical Menjadi angka
- 6) Memisahkan fitur dan Target
- 7) Standarisasi fitur.

- 1) Menghapus fitur yang tidak memiliki pengaruh

Menghapus kolom fitur "Id" karena tidak memiliki pengaruh pada pembelajaran model

```

df = df.drop('Id', axis=1)
print(df.shape)

```

- 2) Check Baris yang Kosong

Setelah fitur Id dihapus ukuran dataset saat ini:1460 baris dan 80 kolom, kemudian lakukan pengecekan apakah terdapat kolom data dengan data kosong atau NaN

```

df.loc[:, df.isna().any()]

```

	LotFrontage	Alley	MasVnType	MasVnArea	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinType2	Electrical	FireplaceQu	GarageType	GarageYrBlt	GarageFinish	GarageQual	GarageCond	PoolQC	Fence	MiscFeature
0	65.0	NaN	Brkface	196.0	Gd	TA	No	GLQ	Unf	SBrkr	NaN	Attchd	2003.0	RFin	TA	TA	NaN	NaN	NaN
1	80.0	NaN	NaN	0.0	Gd	TA	Gd	ALQ	Unf	SBrkr	TA	Attchd	1976.0	RFin	TA	TA	NaN	NaN	NaN
2	68.0	NaN	Brkface	162.0	Gd	TA	Mn	GLQ	Unf	SBrkr	TA	Attchd	2001.0	RFin	TA	TA	NaN	NaN	NaN
3	60.0	NaN	NaN	0.0	TA	Gd	No	ALQ	Unf	SBrkr	Gd	Detchd	1998.0	Linf	TA	TA	NaN	NaN	NaN
4	84.0	NaN	Brkface	350.0	Gd	TA	Av	GLQ	Unf	SBrkr	TA	Attchd	2000.0	RFin	TA	TA	NaN	NaN	NaN
...
1455	62.0	NaN	NaN	0.0	Gd	TA	No	Unf	Unf	SBrkr	TA	Attchd	1999.0	RFin	TA	TA	NaN	NaN	NaN
1456	85.0	NaN	Stone	119.0	Gd	TA	No	ALQ	Rec	SBrkr	TA	Attchd	1978.0	Linf	TA	TA	NaN	MFRW	NaN
1457	66.0	NaN	NaN	0.0	TA	Gd	No	GLQ	Unf	SBrkr	Gd	Attchd	1941.0	RFin	TA	TA	NaN	GdPrv	Shed
1458	68.0	NaN	NaN	0.0	TA	TA	Mn	GLQ	Rec	FuseA	NaN	Attchd	1950.0	Linf	TA	TA	NaN	NaN	NaN
1459	75.0	NaN	NaN	0.0	TA	TA	No	BLQ	LwQ	SBrkr	NaN	Attchd	1965.0	Fin	TA	TA	NaN	NaN	NaN

1460 rows x 19 columns

3) Hapus Fitur dengan Data Kosong

Berdasarkan hasil diatas terdapat 19 kolom data yang memiliki data kosong, data kosong dapat diganti dengan data yang sering muncul atau nilai rata atau nilai lainnya. Namun pada studi kasus ini, data kosong akan dihapus

```
new_df = df.dropna(axis=1) # hapus fitur dengan data kosong
print(new_df.shape)
```

Jumlah data sekarang sebanyak 1460 baris dan 61 kolom

4) Check Fitur dengan type categorical

```
s = (new_df.dtypes == 'object')
object_cols = list(s[s].index)
print("Variabel/fitur dengan bertipe Categorical:")
print(object_cols)
print('\n Jumlah Variabel/fitur dengan bertipe Categorical ',
      len(object_cols))
```

Hasil:

Variabel/fitur dengan bertipe Categorical:

```
['MSZoning', 'Street', 'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
 'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'RoofStyle',
 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'ExterQual', 'ExterCond', 'Foundation',
 'Heating', 'HeatingQC', 'CentralAir', 'KitchenQual', 'Functional', 'PavedDrive',
 'SaleType', 'SaleCondition']
```

Jumlah Variabel/fitur dengan bertipe Categorical 27

5) Konversi Fitur dengan type categorical Menjadi angka

Fitur yang memiliki tipe categorical tidak akan dapat dihitung atau dikalkulasi, karena itu kita melakukan konversi/encode menjadi angka berdasarkan jumlah kelompok datanya

```
OH_encoder = OneHotEncoder(sparse_output=False)
OH_cols =
pd.DataFrame(OH_encoder.fit_transform(new_df[object_cols]))
OH_cols.index = new_df.index
OH_cols.columns = OH_encoder.get_feature_names_out()
df_final = new_df.drop(object_cols, axis=1)
df_final = pd.concat([df_final, OH_cols], axis=1)
```

- 6) Memisahkan fitur dan Target

```
X = df_final.drop('SalePrice', axis=1) #data fitur
y = df_final['SalePrice'] #data target
```

- 7) Standarisasi fitur.

Dalam kasus ini akan menggunakan StandardScaler() untuk menstandarisasi fitur-fitur dalam dataset sebelum melakukan tahap modeling.

```
# Standarisasi fitur [lihat materi minggu ke-1]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

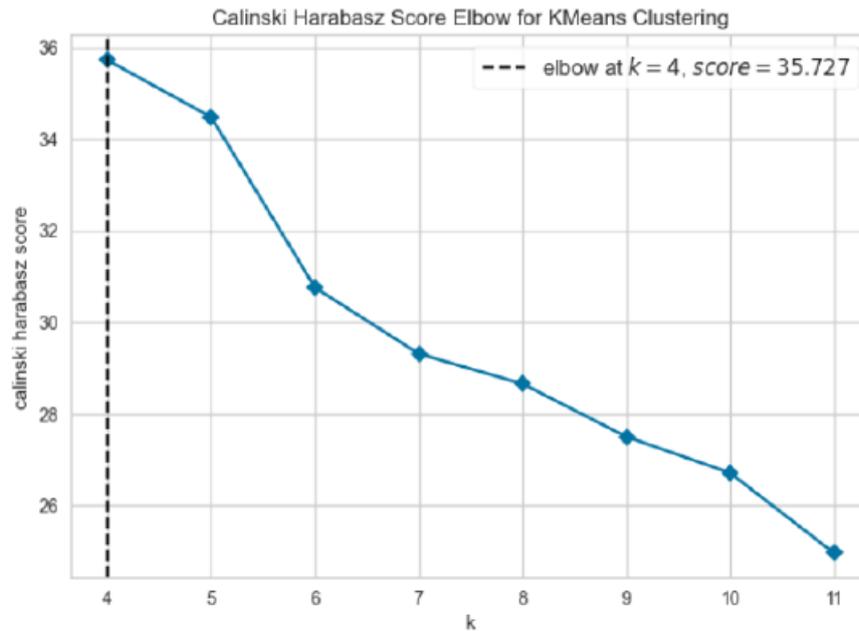
- 8) Mengetahui berapa jumlah komponen utama.

Untuk mengetahui berapa jumlah komponen utama yang paling optimum kita dapat menggunakan fungsi "elbow", disini kita dapat menggunakan fungsi model **KMEANS** untuk mengetahui sebaran data/cluster dan metrik perhitungan dalam kasus ini menggunakan **calinski_harabasz**

```
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer

# Inisiasi model cluster dan visualisasi
model = KMeans()
visualizer = KElbowVisualizer(model, k=(4,12),
metric='calinski_harabasz', timings=False)

visualizer.fit(X_pca) # Fit the data to the visualizer
visualizer.show() # Finalize and render the figure
```



d. Modeling

1) Menerapkan PCA dengan 4 komponen

Langkah keempat adalah membangun model menggunakan Teknik data mining.

Hal Inimencakup:

- a) Memilih Teknik pemodelan yang sesuai dengan masalah bisnis dan jenis data.
- b) Melatih model menggunakan data pelatihan.

Dalam tahap pembelajaran ini algoritma yang akan digunakan adalah Principal Component Analysis (PCA)

```
# Menerapkan PCA dengan 4 komponen
pca = PCA(n_components=4)
X_pca = pca.fit_transform(X_scaled)
```

```
# Menampilkan komponen utama
components = pca.components_
explained_variance = pca.explained_variance_ratio_
cumulative_explained_variance = np.sum(explained_variance)
print('\nVariansi yang Dijelaskan oleh Setiap Komponen Utama:')
print(explained_variance)
print('\nKumulatif Variansi yang Dijelaskan oleh Setiap Komponen Utama:')
print(cumulative_explained_variance)
```

```
Variansi yang Dijelaskan oleh Setiap Komponen Utama: [0.06191951 0.03106124 0.02721189 0.02220717]
Kumulatif Variansi yang Dijelaskan oleh Setiap Komponen Utama: 0.14239980834011928
```

Cumulative explained variance sebesar 0.142 berarti bahwa beberapa komponen utama pertama secara kumulatif menjelaskan 14.2% dari total variansi dalam dataset asli. Ini membantu dalam menentukan seberapa banyak informasi yang berhasil dijelaskan oleh komponen-komponen utama tersebut dan dapat digunakan untuk memilih jumlah fitur yang optimal untuk reduksi dimensi yang efektif.

e. Evaluation Model

1) Membuat DataFrame untuk Visualisasi yang lebih baik

```
components_df = pd.DataFrame(components, columns=X.columns,
index=[f'PC{i+1}' for i in range(4)])
explained_variance_df = pd.DataFrame(explained_variance, index=[f'PC{i+1}'
for i in range(4)], columns=['Explained Variance'])
print('\nKomponen Utama dalam DataFrame:')
print(components_df)
```

Komponen Utama dalam DataFrame:

	MSSubClass	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd
PC1	-0.012161	0.034845	0.220096	-0.063255	0.224988	0.195788
PC2	-0.158167	0.143939	-0.021403	0.027134	0.039110	-0.050673
PC3	-0.018377	0.172206	0.080387	0.021680	-0.142541	-0.041960
PC4	0.108603	-0.024003	0.009148	0.095095	0.007715	0.012116

2) Menampilkan nilai kontribusi dari PC1 hingga PC4

```
for i in range(4):
    print(f'\nFitur dan Nilai yang paling berkontribusi pada PC{i+1}:')
    print(components_df.iloc[i].sort_values(ascending=False))
```

Fitur dan Nilai yang paling berkontribusi pada PC1:

YearBuilt	0.224988
OverallQual	0.220096
Foundation_PConc	0.205327
YearRemodAdd	0.195788
GarageCars	0.192168
...	
CentralAir_N	-0.109302
HeatingQC_TA	-0.122672
Foundation_CBlock	-0.132970
KitchenQual_TA	-0.182024
ExterQual_TA	-0.208705
Name: PC1, Length: 215, dtype: float64	

Fitur dan Nilai yang paling berkontribusi pada PC2:

MSZoning_RL	0.225842
Foundation_CBlock	0.201313
BsmtFinSF1	0.173538
1stFlrSF	0.159055
Exterior2nd_Plywood	0.157176
...	
Foundation_BrkTil	-0.131523
HouseStyle_2Story	-0.138878
2ndFlrSF	-0.152415
MSSubClass	-0.158167
MSZoning_RM	-0.170702
Name: PC2, Length: 215, dtype: float64	

Penjelasan Hasil Nilai Kontribusi

- 1) Explained Variance PC1=0.061920: Komponen utama pertama (PC1) menjelaskan sekitar 6.192% dari variansi total dalam data asli.

- 2) Explained Variance PC2=0.031061: Komponen utama kedua (PC2) menjelaskan sekitar 3.106% dari variansi total dalam data asli.
- 3) Explained Variance PC3=0.027212: Komponen utama ketiga (PC3) menjelaskan sekitar 2.7212% dari variansi total dalam data asli.
- 4) Explained Variance PC4=0.022207: Komponen utama keempat (PC4) menjelaskan sekitar 2.2207% dari variansi total dalam data asli.
- 5) Komponen utama pertama (PC1) memiliki kontribusi terbesar dalam menjelaskan variansi data. Ini berarti PC1 adalah arah dalam ruang fitur yang memaksimalkan pemisahan data.
- 6) Jika kita ingin mempertahankan komponen yang menjelaskan setidaknya 10% dari variansi total, kita mungkin hanya perlu mempertahankan PC1, karena PC1 sendiri menjelaskan sekitar 6.192%.

3) Menentukan dan menentukan jumlah fitur yang optimal

```
print('\nFitur yang paling berkontribusi pada PC1:')
pc1_loadings = components_df.iloc[1].sort_values(ascending=False)
print("Loadings untuk PC1:")
print(pc1_loadings)
```

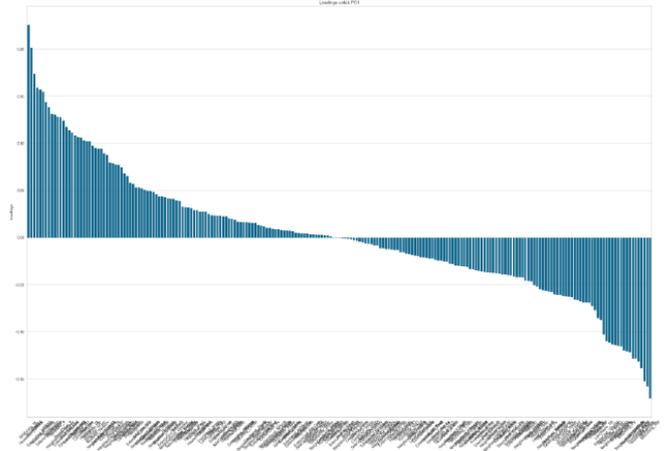
Fitur yang paling berkontribusi pada PC1:
 Loadings untuk PC1:

YearBuilt	0.224988
OverallQual	0.220096
Foundation_PConc	0.205327
YearRemodAdd	0.195788
GarageCars	0.192168
	...
CentralAir_M	-0.109302
HeatingQC_TA	-0.122672
Foundation_CBlock	-0.132970
KitchenQual_TA	-0.182024
ExterQual_TA	-0.208705

Name: PC1, Length: 215, dtype: float64

4) Visualisasi nilai loadings untuk PC1

```
import seaborn as sns
plt.figure(figsize=(30, 20))
sns.barplot(x=pc1_loadings.index, y=pc1_loadings.values)
plt.xlabel('Fitur')
plt.ylabel('Loadings')
plt.title('Loadings untuk PC1')
plt.xticks(rotation=45)
plt.show()
```



Catatan:

Hasil Selengkapnya bisa dilihat pada IDE.

5) Menetapkan threshold untuk loadings

```
# Menetapkan threshold untuk loadings
# Threshold ini menjadi batas pemilihan fitur dengan kontribusi tertinggi.
# Fitur yang memiliki nilai loadings di atas atau di bawah threshold tertentu dapat dianggap penting.
```

```
threshold = 0.1 # Misalnya, threshold ini bisa disesuaikan
# Memilih fitur yang kontribusinya lebih besar dari threshold
important_features_pc1 = pc1_loadings[pc1_loadings.abs() >
threshold].index.tolist()
important_features_pc1_count = len(important_features_pc1)
print("Fitur penting untuk berdasarkan threshold:")
print(important_features_pc1)
print("\nJumlah Fitur penting untuk berdasarkan threshold:")
print(important_features_pc1_count)
```

Fitur penting untuk berdasarkan threshold:

```
['YearBuilt', 'OverallQual', 'Foundation_PConc', 'YearRemodAdd', 'GarageCars', 'ExterQual_Gd', 'C
```

Jumlah Fitur penting untuk berdasarkan threshold:

28

Keterangan:

detail pada IDE, untuk menampilkan fitur penting hasil seleksi PCA, gunakan script berikut:

```
new_X = X[['YearBuilt', 'OverallQual', 'Foundation_PConc', 'YearRemodAdd',  
'GarageCars', 'ExterQual_Gd', 'GarageArea', 'FullBath', 'HeatingQC_Ex',  
'Exterior1st_VinylSd', 'Exterior2nd_VinylSd', 'TotalBsmtSF',  
'KitchenQual_Gd', 'GrLivArea', '1stFlrSF', 'SaleType_New',  
'SaleCondition_Partial', 'CentralAir_Y', 'TotRmsAbvGrd', 'PavedDrive_Y',  
'KitchenQual_Ex', 'Neighborhood_Nridght', 'MSZoning_RM', 'CentralAir_N',  
'HeatingQC_TA', 'Foundation_CBlock', 'KitchenQual_TA', 'ExterQual_TA']]  
  
print(new_X.shape)  
new_X.head()
```

	YearBuilt	OverallQual	Foundation_PConc	YearRemodAdd	GarageCars	ExterQual_Gd	GarageArea	FullBa
0	2003	7	1.0	2003	2	1.0	548	
1	1976	6	0.0	1976	2	0.0	460	
2	2001	7	1.0	2002	2	1.0	608	
3	1915	7	0.0	1970	3	0.0	642	
4	2000	8	1.0	2000	3	1.0	836	

5 rows × 28 columns

Catatan:

Hasil Selengkapnya bisa dilihat pada IDE.

Full Script: <https://drive.google.com/file/d/1ppkCbcH8iQRDYFPFLcos7s-zkrGikvxq/view?usp=sharing>

f. Deployment

Langkah terakhir adalah mengimplementasikan model kedalam lingkungan operasional. Hal Ini mencakup:

- 1) Merencanakan dan menjalankan implementasi model dalam system produksi.
- 2) Memantau dan memelihara model untuk memastikan kinerjanya tetap optimal.
- 3) Mengkomunikasikan hasil dan manfaat model kepada pemangku kepentingan.

Penjelasan Project untuk penilaian CPMK-3

Langkah Terakhir dalam menerapkan model pembelajaran pada dataset, mahasiswa diwajibkan membuat laporan terkait dengan dataset yang telah digunakan pada Tugas1 dan Tugas2 dengan object laporan Project:

1. Business understanding (Tugas1)
2. Data Understanding (Tugas1)
3. Data Preparation (Tugas2)
4. Modelling (Tugas2)
5. Evaluation

Adapun detail format penulisan pedoman Laporan dapat dilihat

<https://drive.google.com/drive/folders/1hSos7W1rdXfYq3QjPoVvKViwL8tCKHy>

Waktu Pengumpulan/Presentasi Project: Minggu Ke-10 s/d 16

Referensi

Steele, B., Chandler, J., Reddy, S. (2016). Algorithms for Data Science. Jerman: Springer International Publishing.

Abdussomad, dkk. (2021). Dasar Pemrograman Python. Yogyakarta: Teknosain.

Saeful Bahri, dkk (2019), Data mining : algoritma klasifikasi & penerapannya dalam aplikasi, Grha Ilmu

Segmentasi Pelanggan Menggunakan Python. (n.d.). (n.p.): Kreatif

Data Mining Menggunakan Android, Weka, dan SPSS. (2020). (n.p.): Airlangga University Press.

Abdussomad, A., Kurniawan, I., & Wibowo, A. (2023). Implementation of the Decission Tree Algorithm to Determine Credit Worthiness. *Compiler*, 12(2), 103-108