Analisis Perbandingan Support Vector Machine dan KNearest Neighbors Dalam Membangun Model Klasifikasi Indeks Pembangunan Manusia.

Latifatuzikra Suhairi
ASIMO

Latar Belakang & Rumusan Masalah

Latar Belakang

- IPM (Indeks Pembangunan Manusia) sebagai indicator untuk mengukur keberhasilan dalam upaya membangun kualitas hidup manusia.
- IPM menurut BPS dibagi menjadi 4 kategori yaitu rendah/low, sedang/normal, tinggi/high, dan sangat tinggi/very-high.
- Komponen IPM terdiri dari bidang pendidikan, kependudukan, dan kesehatan.
- Komponen tersebut akan menjadi nilai penentu kategori Indeks Pembangunan Manusia di suatu wilayah.
- Untuk mempercepat penentuan Indeks Pembangunan Manusia, dibutuhkan pemodelan yang mampu mengklasifikan IPM dengan mudah dan akurasi yang baik.
- Dalam tugas ini, model akan dibangun menggunakan algoritma SVM dan KNN.

Latar Belakang & Rumusan Masalah

Rumusan Masalah

- Bagaimana perbandingan model klasifikasi Indeks Pembangunan Manusia menggunakan algoritma SVM dan KNN?

Deskripsi singkat data

Dataset

IPM.xlsx

- Berisikan 2196 baris data dengan 5 kolom tentang variable-variable yang mempengaruhi indeks pembangunan manusia beserta indeksnya.
- Indeks pembangunan manusia terdiri atas data kategori

	Harapan_Lama_Sekolah	Pengeluaran_Perkapita	Rerata_Lama_Sekolah	Usia_Harapan_Hidup	IPM
0	14.36	9572	9.37	69.96	High
1	13.9	7148	9.48	65.28	Normal
2	14.32	8776	8.68	67.43	Normal
3	14.6	8180	8.88	64.4	Normal
4	14.01	8030	9.67	68.22	Normal
...	...	...	...	...	...
2191	10.13	5522	4.91	65.32	Low
2192	7.11	5440	2.51	65.26	Low
2193	9.79	4761	2.99	64.83	Low
2194	14.99	14922	11.3	70.15	High
2195	12.91	11059	8.17	71.2	High

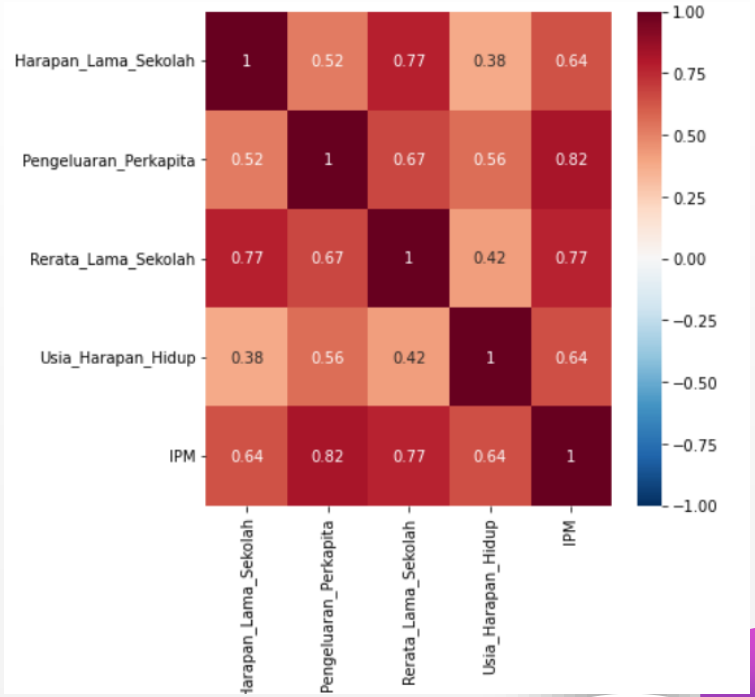
2196 rows × 5 columns

Deskripsi singkat data

Variabel Yang Digunakan

Feature:
- Harapan_Lama_Sekolah
- Pengeluaran_Perkapita
- Rerata_Lama_Sekolah
- Usia_Harapan_Hidup

Target:
- IPM



Preprocessing Data

1. Pembersihan data

Memeriksa nilai missing dan duplikat

```
# missing value
df.isnull().sum()
```

```
Harapan_Lama_Sekolah     0
Pengeluaran_Perkapita    0
Rerata_Lama_Sekolah      0
Usia_Harapan_Hidup       0
IPM                      0
dtype: int64
```

Tidak ada missing value pada dataset

```
# duplikat
df.duplicated().sum()
```

```
0
```

Tidak ada data yang duplikat pada dataset

2. Pengubahan Tipe Data

```
#mendapatkan informasi dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2196 entries, 0 to 2195
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Harapan_Lama_Sekolah  2196 non-null   object
 1   Pengeluaran_Perkapita 2196 non-null   object
 2   Rerata_Lama_Sekolah   2196 non-null   object
 3   Usia_Harapan_Hidup    2196 non-null   object
 4   IPM                   2196 non-null   object
dtypes: object(5)
memory usage: 85.9+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2196 entries, 0 to 2195
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Harapan_Lama_Sekolah  2196 non-null   float64
 1   Pengeluaran_Perkapita 2196 non-null   int32
 2   Rerata_Lama_Sekolah   2196 non-null   float64
 3   Usia_Harapan_Hidup    2196 non-null   float64
 4   IPM                   2196 non-null   object
dtypes: float64(3), int32(1), object(1)
memory usage: 77.3+ KB
```

Preprocessing Data

3. Encode pada variabel IPM

```
df['IPM'].unique()
```

```
array(['High', 'Normal', 'Very-High', 'Low'], dtype=object)
```

```
ipm_data['IPM'] = ipm_data['IPM'].replace({'Low':'0', 'Normal': '1',
                                           'High': '2', 'Very-High': '3'}).astype(int)
ipm_data.head()
```

	Harapan_Lama_Sekolah	Pengeluaran_Perkapita	Rerata_Lama_Sekolah	Usia_Harapan_Hidup	IPM
0	14.36	9572	9.37	69.96	2
1	13.90	7148	9.48	65.28	1
2	14.32	8776	8.68	67.43	1
3	14.60	8180	8.88	64.40	1
4	14.01	8030	9.67	68.22	1

4. Feature selection

```
x = ipm_data.drop(["IPM"], axis=1).to_numpy()
y = ipm_data['IPM'].to_numpy()
```

5. Standardization

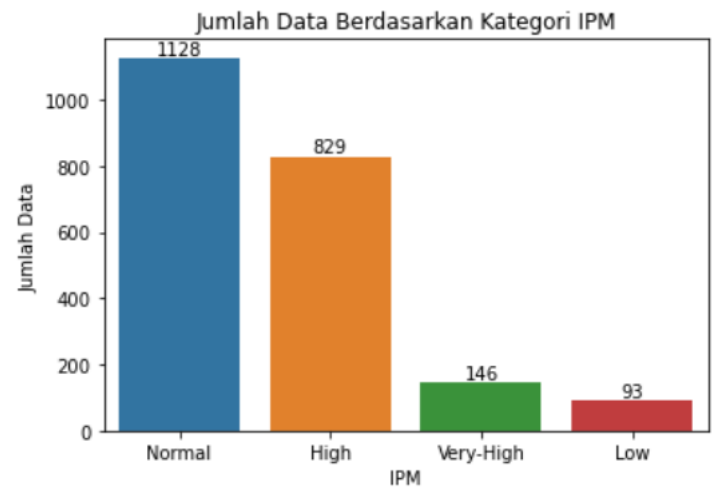
```
sc = StandardScaler()
sc.fit_transform(x)
```

```
array([[ 1.08824282, -0.28194717,  0.66945354,  0.1433277 ],
       [ 0.73781156, -1.19181217,  0.73773882, -1.21842258],
       [ 1.05777054, -0.58073122,  0.24111861, -0.59283217],
       ...,
       [-2.3932156 , -2.08778895, -3.29109263, -1.34936011],
       [ 1.56818129,  1.72621194,  1.8675498 ,  0.19861243],
       [-0.01637746,  0.27620845, -0.07547677,  0.50413333]])
```

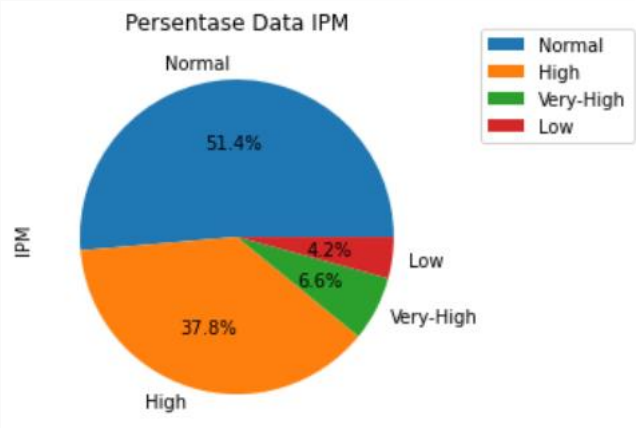
Preprocessing Data

6. Visualisasi data

a. Jumlah data berdasarkan kategori IPM



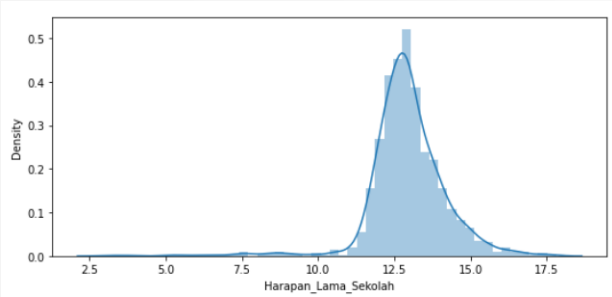
b. Persentase data IPM

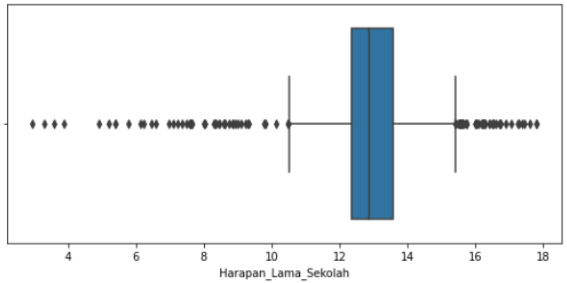


Preprocessing Data

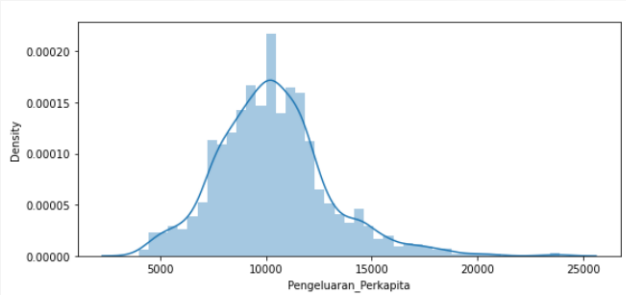
6. Visualisasi data

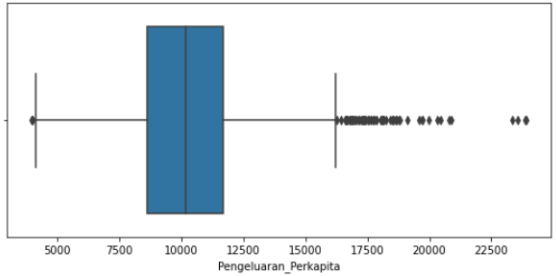
c. Analisis univariate Harapan Lama Sekolah





d. Analisis univariate Pengeluaran per kapita

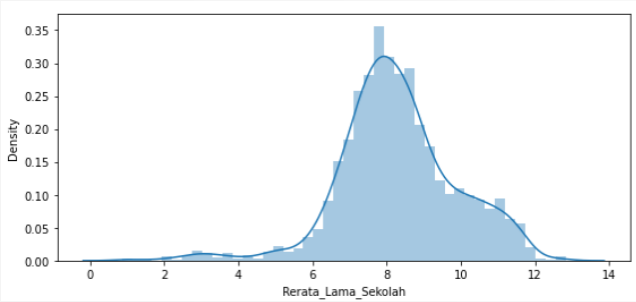


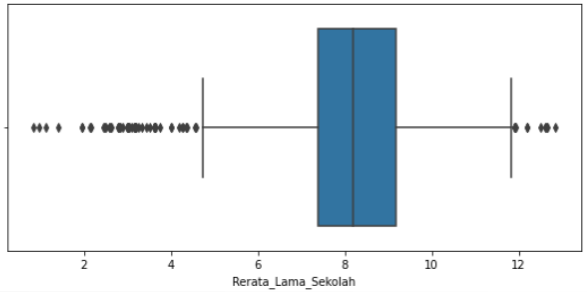


Preprocessing Data

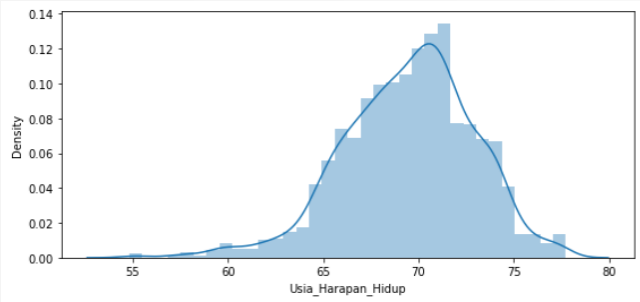
6. Visualisasi data

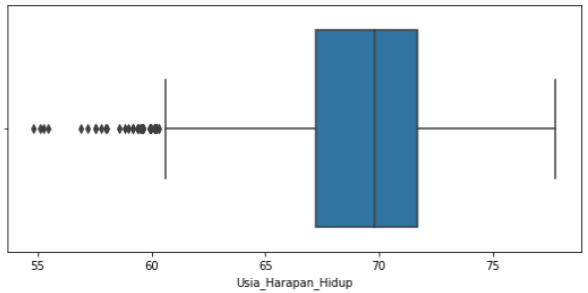
e. Analisis univariate Rerata Lama Sekolah





f. Analisis univariate Usia harapan hidup

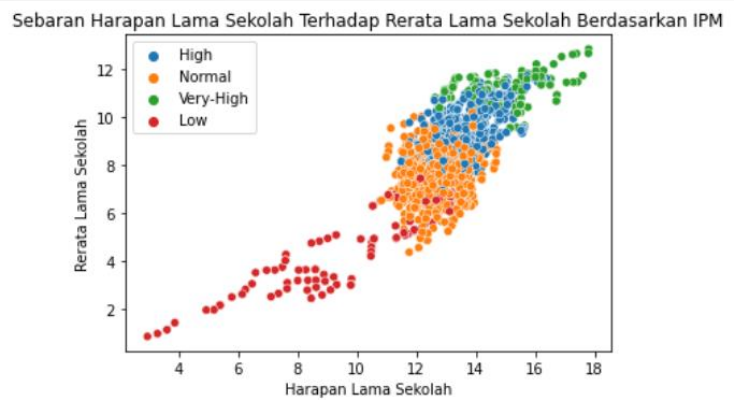




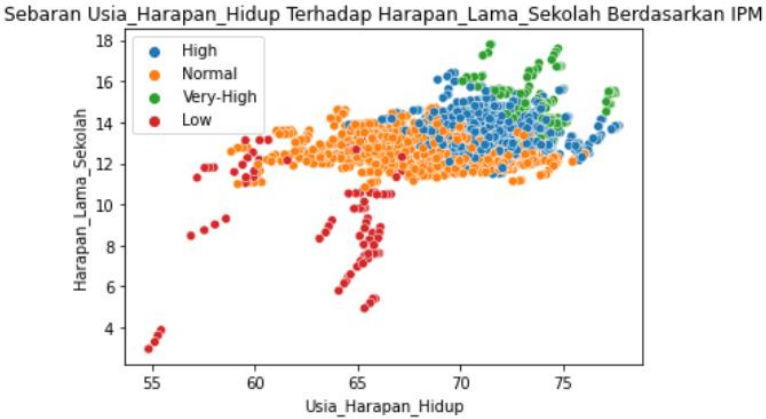
Preprocessing Data

6. Visualisasi data

g. Sebaran Harapan Lama Sekolah thdp Rerata Lama Sekolah berdasarkan IPM



h. Sebaran Usia_Harapan_Hidup Terhadap Harapan_Lama_Sekolah Berdasarkan IPM



Model 1 : SVM Classifier

01

Model SVM

Algoritma supervised learning untuk klasifikasi dan regresi yang bekerja menggunakan konsep Structural Risk Minimization. dirancang untuk mengolah data menjadi Hyperplane yang mengklasifikasikan ruang input menjadi dua kelas.

02

Parameter

```
SVC(C=0.1,
decision_function_shape='ovo',
gamma=1, kernel='linear',
    probability=True)
```

Model 2 : KNN

01 Model KNN

K-nearest neighbors atau knn adalah algoritma yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (*train data sets*), yang diambil dari k tetangga terdekatnya (*nearest neighbors*).

02 Parameter

KNN(leaf_size=1; p=1; n_neighbors=22)

Ukuran Kebaikan Model

Model SVM

1. Classification report

	precision	recall	f1-score	support
0	1.00	0.96	0.98	28
1	0.96	1.00	0.98	274
2	0.97	0.95	0.96	208
3	1.00	0.85	0.92	39
accuracy			0.97	549
macro avg	0.98	0.94	0.96	549
weighted avg	0.97	0.97	0.96	549

2. ROC

```
print(roc_auc_svc_1)
```

```
99.80661499778219
```

Model KNN

1. Classification report

	precision	recall	f1-score	support
0	1.00	0.82	0.90	28
1	0.81	0.84	0.83	274
2	0.74	0.76	0.75	208
3	0.96	0.69	0.81	39
accuracy			0.80	549
macro avg	0.88	0.78	0.82	549
weighted avg	0.81	0.80	0.80	549

2. ROC

```
print(roc_auc_knn)
```

```
94.09117125836138
```

Ukuran Kebaikan Model

Model SVM

1. Classification report

	precision	recall	f1-score	support
0	1.00	0.96	0.98	28
1	0.96	1.00	0.98	274
2	0.97	0.95	0.96	208
3	1.00	0.85	0.92	39
accuracy			0.97	549
macro avg	0.98	0.94	0.96	549
weighted avg	0.97	0.97	0.96	549

2. ROC

```
print(roc_auc_svc_1)
```

```
99.80661499778219
```

Model KNN

1. Classification report

	precision	recall	f1-score	support
0	1.00	0.82	0.90	28
1	0.81	0.84	0.83	274
2	0.74	0.76	0.75	208
3	0.96	0.69	0.81	39
accuracy			0.80	549
macro avg	0.88	0.78	0.82	549
weighted avg	0.81	0.80	0.80	549

2. ROC

```
print(roc_auc_knn)
```

```
94.09117125836138
```

Model SVM lebih baik: 97%

Kesimpulan

01

Dari dataset **IPM.xlsx,** dapat dibuatkan model klasifikasi untuk mengklasifikasikan indeks pembangunan manusia yang memiliki 4 kategori: low, normal, high, very high berdasarkan variable harapanlama sekolah, pengeluaran perkapita, rerata lama sekolah, dan usia harapan hidup

02

Model klasifikasi menggunakan metode Support Vector Machine menghasilkan akurasi model lebih baik, yaitu 97% dibandingkan menggunakan model KNN, yaitu 80%.