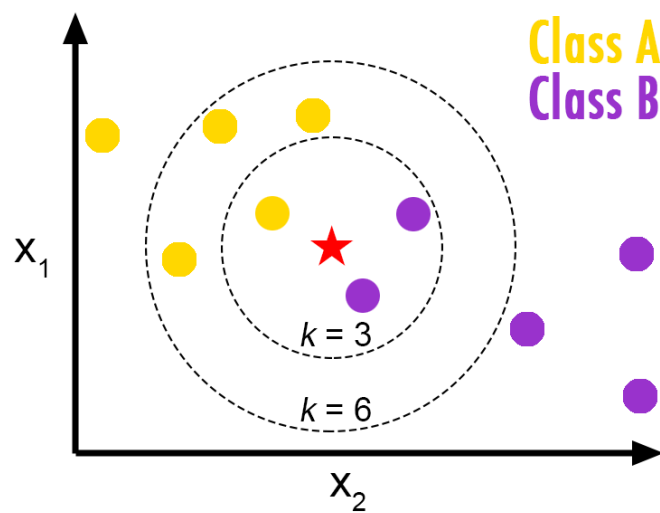


Nama Dosen : Teguh Iman Hermanto, M.Kom
 Mata Kuliah : Machine Learning 1
 Pembahasan : K Nearest Neighbors
 Pokok Pemb : - Konsep Algoritma KNN
 - Membangun Model KNN
 - Simulasi Algoritma KNN
 - Evaluasi Algoritma KNN
 - Aplikasi menggunakan algoritma KNN

1. Konsep Algoritma KNN



- k-NN mengidentifikasi nilai “k” dalam data pelatihan yang merupakan “tetangga terdekat” (Lantz, 2013)
- Metode untuk melakukan klasifikasi terhadap objek berdasarkan pembelajaran yang jaraknya paling dekat dengan objek tersebut

Model algoritma KNN

- Euclidean Distance

$$Ed(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- Manhattan Distance

$$Md(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Langkah-langkah Algoritma KNN :

1. Tentukan data testing dan nilai parameter k

2. Hitung jarak antara data testing dan data training

3. Urutkan data berdasarkan jarak terkecil

4. Kelompokkan data

2. Membangun Model KNN

```
1 import numpy as np
2 import pandas as pd
3
4 from sklearn.preprocessing import LabelEncoder
5 from sklearn.model_selection import train_test_split
6 from sklearn.neighbors import KNeighborsClassifier
7 from sklearn.metrics import accuracy_score, confusion_matrix
8
9 import matplotlib.pyplot as plt
10 import seaborn as sns
```

```
1 df = pd.read_csv('Pinjaman.csv')
```

```
1 df.head()
```



```
1 # Distribution of Age
2 plt.figure(figsize=(8, 6))
3 sns.histplot(df['Usia'], bins=10, kde=True, color='blue')
4 plt.title('Distribution of Age')
5 plt.xlabel('Usia')
6 plt.ylabel('Frequency')
7 plt.show()
```



```
1 # Boxplot of Income (Pendapatan) vs Lulus Kredit
2 plt.figure(figsize=(8, 6))
3 sns.boxplot(x='Lulus_Kredit', y='Pendapatan', data=df, palette='Set1')
4 plt.title('Pendapatan vs Kelayakan Kredit')
5 plt.xlabel('Lulus Kredit')
6 plt.ylabel('Pendapatan (Juta)')
7 plt.xticks(ticks=[0, 1], labels=['Tidak Layak', 'Layak'])
8 plt.show()
```



```
1 # Countplot of Marital Status
2 plt.figure(figsize=(8, 6))
3 sns.countplot(x='Status_Perkawinan', data=df, palette='Set2')
4 plt.title('Count of Marital Status')
5 plt.xticks(ticks=[0, 1], labels=['Belum Menikah', 'Menikah'])
6 plt.ylabel('Frequency')
7 plt.show()
```



```

1 # Pairplot for the numerical features
2 plt.figure(figsize=(8, 6))
3 sns.pairplot(df[['Usia', 'Pendapatan',
4                 'Jumlah_Pinjaman',
5                 'Durasi_Pinjaman',
6                 'Lulus_Kredit']],
7             hue='Lulus_Kredit', palette='coolwarm')
8 plt.show()

```



```

1 # Violin plot of Loan Amount vs Credit Eligibility
2 plt.figure(figsize=(8, 6))
3 sns.violinplot(x='Lulus_Kredit', y='Jumlah_Pinjaman', data=df, palette='Set2')
4 plt.title('Distribusi Jumlah Pinjaman berdasarkan Kelayakan Kredit')
5 plt.xlabel('Lulus Kredit')
6 plt.ylabel('Jumlah Pinjaman (Juta)')
7 plt.xticks(ticks=[0, 1], labels=['Tidak Layak', 'Layak'])
8 plt.show()

```



```

1 label_encoder_status_perkawinan = LabelEncoder()
2 label_encoder_status_pekerjaan = LabelEncoder()
3 label_encoder_lulus_kredit = LabelEncoder()

```



```

1 df['Status_Perkawinan'] = label_encoder_status_perkawinan.fit_transform(df['Status_Perkawinan'])
2 df['Status_Pekerjaan'] = label_encoder_status_pekerjaan.fit_transform(df['Status_Pekerjaan'])
3 df['Lulus_Kredit'] = label_encoder_lulus_kredit.fit_transform(df['Lulus_Kredit'])

```

Status Perkawinan (0:Belum Menikah) | (1:Menikah)

Status Pekerjaan (0:Karyawan Kontrak) | (1:Karyawan Tetap) | (2:Pensiunan) | (3:Wirausaha)

Lulus Kredit (0:layak) | (1:tidak layak)



```
1 df.to_csv('Pinjaman_modif.csv', index=False)
```



```
1 # Splitting the dataset into features and labels
2 X = df.drop(columns=['Lulus_Kredit'])
3 y = df['Lulus_Kredit']
```



```
1 # Splitting data into training and testing sets (80% train, 20% test)
2 X_train, X_test, y_train, y_test = train_test_split(
3     X, y, test_size=0.2, random_state=42)
```



```
1 # Creating and training the KNN model
2 knn_model = KNeighborsClassifier(n_neighbors=5)
3 knn_model.fit(X_train, y_train)
```



```
1 # Predicting the test data
2 y_pred = knn_model.predict(X_test)
```

3. Simulasi Algoritma KNN



```
1  # Example:
2  # [Usia : 30,
3  # Pendapatan : 80,
4  # Status Perkawinan : 0,
5  # Jumlah Pinjaman : 70,
6  # Durasi Pinjaman : 10,
7  # Status Pekerjaan : 1]
8  new_data = np.array([[30, 80, 0, 70, 10, 1]])
9  new_pred = knn_model.predict(new_data)
```



```
1  print('Hasil prediksi: ', new_pred)
```



```
1  label_encoder_lulus_kredit.inverse_transform(new_pred)
```

4. Evaluasi Algoritma KNN



```
1 # Calculating the accuracy
2 accuracy = accuracy_score(y_test, y_pred)
3 accuracy
```



```
1 # Menyimpan model menggunakan pickle
2 import pickle
3
4 filename = 'knn_pinjam_mod.pkl'
5 pickle.dump(knn_model, open(filename, 'wb'))
```

5. Aplikasi Menggunakan Algoritma KNN

Buat file `knn_pinjaman.py` pada direktori project

```

1 import streamlit as st
2 import numpy as np
3 import pickle
4 import plotly.express as px
5 import pandas as pd
6
7 # Load the trained KNN model
8 with open('knn_pinjam_mod.pkl', 'rb') as model_file:
9     knn = pickle.load(model_file)
10
11 # Create a title for the web app
12 st.title('K-Nearest Neighbor (KNN) Loan Eligibility Prediction')
13
14 # Input fields for new data
15 usia = st.sidebar.number_input('Usia', min_value=18, max_value=70, value=30)
16 pendapatan = st.sidebar.number_input('Pendapatan', min_value=10, max_value=200, value=50)
17 status_perkawinan = st.sidebar.selectbox('Status_Perkawinan', ['Belum Menikah', 'Menikah'])
18 if status_perkawinan == 'Belum Menikah':
19     status_perkawinan = 0
20 else:
21     status_perkawinan = 1
22 jumlah_pinjaman = st.sidebar.number_input('Jumlah_Pinjaman', min_value=10, max_value=500, value=100)
23 durasi_pinjaman = st.sidebar.number_input('Durasi_Pinjaman', min_value=1, max_value=30, value=10)
24 #status_pekerjaan = st.sidebar.number_input('Status_Pekerjaan', min_value=0, max_value=3, value=0)
25 status_pekerjaan = st.sidebar.selectbox('Status_Pekerjaan', ['Karyawan Kontrak', 'Karyawan Tetap', 'Pensiunan', 'Wirausaha'])
26 if status_pekerjaan == 'Karyawan Kontrak':
27     status_pekerjaan = 0
28 elif status_pekerjaan == 'Karyawan Tetap':
29     status_pekerjaan = 1
30 elif status_pekerjaan == 'Pensiunan':
31     status_pekerjaan = 2
32 elif status_pekerjaan == 'Wirausaha':
33     status_pekerjaan = 3
34
35 # New data for prediction
36 new_data = np.array([[usia, pendapatan, status_perkawinan, jumlah_pinjaman, durasi_pinjaman, status_pekerjaan]])
37
38 # Predict button
39 if st.sidebar.button('Predict'):
40     # Predict the result
41     prediction = knn.predict(new_data)
42     result = 'Layak' if prediction == 0 else 'Tidak Layak'
43
44     # Show the prediction result
45     st.write(f'Hasil prediksi: **{result}**')
46
47 # 3D Plot for KNN
48 df = pd.DataFrame({
49     'Usia': np.random.randint(18, 70, size=60),
50     'Pendapatan': np.random.randint(10, 200, size=60),
51     'Jumlah_Pinjaman': np.random.uniform(10, 500, size=60),
52     'Lulus_Kredit': np.random.choice(['0', '1'], size=60)
53 })
54
55 fig = px.scatter_3d(df, x='Usia', y='Pendapatan', z='Jumlah_Pinjaman', color='Lulus_Kredit',
56                    title=f'KNN Prediction')
57 st.plotly_chart(fig)

```

Untuk menjalankan aplikasinya buka command prompt, lalu arahkan ke direktori project dan masukan perintah `streamlit run nama_aplikasi.py`