Nama : Daffa Asyqar Ahmad Khalisheka
NIM : 1103200034

UTS Machine Learning Semester Ganjil 2023/2024

Klasifikasi Data Kualitas Udara di Provinsi DKI Jakarta dengan menggunakan model Support Vector Machine

Support Vector Machine (SVM) dapat digunakan untuk mengklasifikasikan data udara Jakarta berdasarkan parameter seperti tingkat polusi udara, suhu, kelembaban, dan lainnya. Model ini memisahkan data ke dalam kategori yang berbeda, membantu identifikasi kondisi udara yang potensial menjadi masalah kesehatan atau lingkungan.

```
# Import library Google Colab untuk mengakses Google Drive
from google.colab import drive

# Mount (pasang) Google Drive pada direktori /content/drive
drive.mount('/content/drive')
```

Kode di atas menggunakan library dari Google Colab untuk mengakses Google Drive. Dengan melakukan mounting (pemasangan) Google Drive pada direktori tertentu di Colab, pengguna dapat dengan mudah berinteraksi dengan file yang ada di Google Drive mereka.

Langkah-langkahnya secara singkat:

1. Import library `drive` dari `google.colab`.
2. Gunakan fungsi `drive.mount()` untuk memasang Google Drive pada direktori `/content/drive` di Google Colab.

Setelah langkah-langkah tersebut dieksekusi, pengguna akan diminta untuk memberikan otorisasi terhadap Google Drive mereka. Setelah itu, Google Drive akan terhubung dan dapat diakses langsung dari Google Colab pada direktori yang telah ditentukan.

Langkah-langkah di atas adalah contoh kode Python yang digunakan untuk mengimpor beberapa library populer, yaitu pandas, seaborn, matplotlib.pyplot, dan numpy, dengan memberikan alias pada setiap library. Berikut penjelasan langkah-langkahnya:

1. **Mengimpor Library Pandas dengan Alias pd:**
   ```python
   import pandas as pd
   ```
   Langkah ini mengimpor library pandas dan memberikan alias "pd" pada library tersebut. Alias ini berguna untuk mempersingkat penulisan kode selanjutnya.

2. **Mengimpor Library Seaborn dengan Alias sns:**
   ```python
   import seaborn as sns
   ```

Pada langkah ini, library seaborn diimpor dengan memberikan alias "sns". Alias ini akan digunakan saat memanggil fungsi-fungsi dari library seaborn.

3. **Mengimpor Library Matplotlib.pyplot dengan Alias plt:**
```python
import matplotlib.pyplot as plt
```
Di langkah ini, library matplotlib.pyplot diimpor dengan memberikan alias "plt". Penggunaan alias ini mempermudah saat membuat plot atau visualisasi menggunakan Matplotlib.

4. **Mengimpor Library Numpy dengan Alias np:**
```python
import numpy as np
```
Pada langkah terakhir, library numpy diimpor dengan memberikan alias "np". Alias ini memudahkan penggunaan fungsi-fungsi numpy dalam kode selanjutnya.

Dengan memberikan alias pada library, kita dapat menggunakan singkatan yang lebih pendek saat memanggil fungsi atau metode dari library tersebut. Ini membantu dalam menulis kode yang lebih ringkas dan mudah dibaca.

EDA adalah singkatan dari "Exploratory Data Analysis" atau Analisis Data Eksploratif dalam bahasa Indonesia. Ini adalah proses analisis data statistik yang digunakan untuk mengeksplorasi, meringkas, dan memahami karakteristik utama dari suatu kumpulan data. Tujuannya adalah untuk mengidentifikasi pola, hubungan, anomali, dan tren dalam data secara visual dan deskriptif. EDA biasanya dilakukan sebagai langkah awal dalam analisis data sebelum penerapan metode statistik atau model prediktif lebih lanjut.

membaca beberapa lembar (sheet) dari file Excel ke dalam DataFrame menggunakan pandas. Berikut ini penjelasan langkah-langkahnya:

1. **Tentukan Nama-nama Sheet yang Ingin Dibaca:**
- Anda menentukan daftar nama sheet yang ingin dibaca dari file Excel. Dalam hal ini, sheet_names berisi nama-nama sheet, seperti 'DKI1', 'DKI2', 'DKI3', 'DKI4', dan 'DKI5'.

```python
sheet_names = ['DKI1', 'DKI2', 'DKI3', 'DKI4', 'DKI5']
```

2. **Berikan Path File:**
- Anda memberikan path atau lokasi file Excel yang akan dibaca.

```python
file_path = '/content/drive/MyDrive/Dataset/DATA ISPU - Impute (1).xlsx'
```

3. **Baca Semua Sheet ke dalam Dictionary DataFrame:**

- Anda menggunakan fungsi `pd.read_excel` untuk membaca semua sheet yang telah ditentukan ke dalam bentuk dictionary DataFrame. Setiap elemen dictionary akan memiliki kunci berupa nama sheet dan nilai berupa DataFrame.

```python
all_data = pd.read_excel(file_path, sheet_name=sheet_names)
```

4. **Loop Melalui Setiap Sheet dan DataFrame pada Dictionary:**
- Anda menggunakan loop `for` untuk melakukan iterasi melalui setiap item dalam dictionary. Pada setiap iterasi, Anda mencetak beberapa baris pertama dari DataFrame.

```python
for sheet_name, df in all_data.items():
    print(f"Beberapa baris pertama dari {sheet_name}:")
    print(df.head())
    print("\n")
```

5. **Gabungkan Semua DataFrames Menjadi Satu DataFrame:**
- Terakhir, Anda menggunakan `pd.concat` untuk menggabungkan semua DataFrames dalam dictionary menjadi satu DataFrame tunggal, menggunakan `ignore_index=True` untuk mengatur ulang indeks DataFrame hasil gabungan.

```python
dataudara = pd.concat(all_data.values(), ignore_index=True)
```

Dengan langkah-langkah tersebut, Anda telah berhasil membaca dan menggabungkan data dari beberapa sheet Excel menjadi satu DataFrame yang lebih besar.

Outlier dalam visualisasi data adalah data yang secara signifikan berbeda dari pola umum atau mayoritas data. Outlier dapat mempengaruhi interpretasi keseluruhan data dan mungkin menunjukkan anomali, kesalahan pengukuran, atau kejadian luar biasa. Identifikasi outlier penting untuk memahami distribusi data secara lebih akurat dan membuat keputusan yang lebih tepat. Visualisasi outlier biasanya menggunakan metode seperti box plot atau scatter plot untuk menyoroti nilai-nilai yang berbeda secara visual.

Transformasi data adalah proses mengubah data dari satu bentuk atau format ke bentuk atau format lainnya. Ini melibatkan manipulasi dan restrukturisasi data agar sesuai dengan kebutuhan analisis atau aplikasi tertentu. Transformasi data dapat mencakup penggabungan, pemfilteran, pengurutan, dan perubahan format data. Tujuannya adalah untuk membuat data lebih mudah dimengerti, relevan, atau sesuai dengan persyaratan spesifik yang diperlukan oleh sistem atau alat analisis data. Transformasi data seringkali merupakan langkah penting dalam pra-pemrosesan data sebelum dilakukan analisis lebih lanjut.

Membangun model machine learning melibatkan penggunaan algoritma dan data untuk melatih sebuah program komputer agar dapat membuat prediksi atau pengambilan keputusan tanpa di-

program secara eksplisit. Proses ini melibatkan tahap pembagian data menjadi set pelatihan dan pengujian, pelatihan model menggunakan algoritma tertentu, dan evaluasi kinerja model untuk memastikan keakuratannya dalam membuat prediksi pada data yang belum pernah dilihat sebelumnya.

SMOTE (Synthetic Minority Over-sampling Technique), SMOTE-N, dan SMOTE-ENN adalah teknik sampling dalam konteks penanganan ketidakseimbangan kelas pada dataset:

1. **SMOTE (Synthetic Minority Over-sampling Technique):**
   - **Deskripsi Singkat:** SMOTE mengatasi ketidakseimbangan kelas dengan membuat sampel sintetis untuk kelas minoritas. Ini dilakukan dengan membuat contoh baru di antara data minoritas yang sudah ada.

2. **SMOTE-N (SMOTE for Nominal features):**
   - **Deskripsi Singkat:** Varian SMOTE yang dirancang khusus untuk menangani fitur nominal dalam dataset. Biasanya, SMOTE bekerja dengan baik pada data numerik, tetapi SMOTE-N memperluas konsep tersebut untuk fitur-fitur kategorikal.

3. **SMOTE-ENN (SMOTE-Edited Nearest Neighbors):**
   - **Deskripsi Singkat:** Kombinasi SMOTE dengan algoritma Edited Nearest Neighbors (ENN) untuk membersihkan sampel sintetis yang mungkin ambigu. Ini melibatkan penghapusan sampel sintetis yang dihasilkan yang berada di dekat batas keputusan dan dapat menyebabkan kebingungan model.

SMOTE dan varian-varian ini membantu meningkatkan kinerja model pada dataset yang tidak seimbang dengan menciptakan keseimbangan antara kelas mayoritas dan minoritas.

Inputting new data ke dalam model machine learning adalah proses memasukkan data baru ke dalam model yang sudah dilatih sebelumnya. Ini memungkinkan model untuk membuat prediksi atau menghasilkan output berdasarkan informasi terbaru yang diberikan. Inputing data baru ini penting untuk memperbarui model dan meningkatkan kinerjanya seiring waktu, terutama ketika model dihadapkan pada data yang belum pernah dilihat sebelumnya. Proses ini melibatkan penggunaan model yang sudah ada untuk mengevaluasi dan memproses data baru sehingga model dapat terus belajar dan beradaptasi dengan lingkungan yang berubah.