PRAKTIKUM MACHINE LEARNING

UNIT 3 : DATA VALIDATION

adult_klasifikasi.ipynb

Eka Praja Wiyata Mandala, S.Kom, M,Kom, CADS

Unit 3 : Data Validation

Unit 3: Memvalidasi Data

Tujuan: Memastikan kualitas dan integritas data sebelum analisis lebih lanjut

Langkah-langkah :

- Periksa Missing Values
- Periksa Duplikat
- Validasi Tipe Data
- Validasi Nilai Range
- Periksa Konsistensi Data

Unit 3 : Data Validation

Periksa Missing Values

```
[ ]  # Hitung jumlah missing values
     missing_values = df.isnull().sum()

     # Hitung persentase missing values
     missing_percentage = 100 * df.isnull().sum() / len(df)

     # Gabungkan informasi missing values
     missing_table = pd.concat([missing_values, missing_percentage], axis=1, keys=['Total', 'Percent']

     print(missing_table)
```

Unit 3 : Data Validation

Visualisasi Missing Values

```
[ ]  # Visualisasi missing values
     plt.figure(figsize=(10, 6))
     missing_percentage.plot(kind='bar')
     plt.title('Persentase Missing Values per Kolom')
     plt.xlabel('Kolom')
     plt.ylabel('Persentase Missing')
     plt.xticks(rotation=45, ha='right')
     plt.tight_layout()
     plt.show()
```

Unit 3 : Data Validation

Periksa Duplikat

```
[ ]  # Hitung jumlah duplikat
     duplicates = df.duplicated().sum()
     print(f"Jumlah baris duplikat: {duplicates}")

     # Tampilkan beberapa baris duplikat (jika ada)
     if duplicates > 0:
         print("\nContoh baris duplikat:")
         print(df[df.duplicated(keep=False)].head())
```

Unit 3 : Data Validation

Validasi Tipe Data

```
[ ]  # Tampilkan tipe data setiap kolom
     print(df.dtypes)

     # Periksa apakah ada nilai non-numerik dalam kolom numerik
     numeric_columns = df.select_dtypes(include=[np.number]).columns
     for col in numeric_columns:
         non_numeric = df[pd.to_numeric(df[col], errors='coerce').isna()]
         if len(non_numeric) > 0:
             print(f"\nNilai non-numerik dalam kolom {col}:")
             print(non_numeric[col].unique())
```

Unit 3 : Data Validation

Validasi Nilai Range

```
[ ]  # Periksa range nilai untuk kolom numerik
     for col in numeric_columns:
         print(f"\nRange nilai untuk {col}:")
         print(f"Min: {df[col].min()}, Max: {df[col].max()}")

     # Periksa kategori unik untuk kolom kategorikal
     categorical_columns = df.select_dtypes(include=['object']).columns
     for col in categorical_columns:
         print(f"\nKategori unik dalam {col}:")
         print(df[col].unique())
```

Unit 3 : Data Validation

Periksa Konsistensi Data

```
[ ]  # Contoh: Periksa konsistensi antara 'education' dan 'education_num'
     education_mapping = df.groupby('education')['education_num'].mean().sort_values()
     print("\nPemetaan rata-rata 'education_num' untuk setiap 'education':")
     print(education_mapping)
```

Unit 3 : Data Validation

Visualisasi Konsistensi

```
[ ]  # Visualisasi konsistensi
     plt.figure(figsize=(10, 6))
     education_mapping.plot(kind='bar')
     plt.title('numerisasi education_num untuk Setiap Kategori Education')
     plt.xlabel('Education')
     plt.ylabel('Rata-rata education_num')
     plt.xticks(rotation=45, ha='right')
     plt.tight_layout()
     plt.show()
```

Unit 3 : Data Validation

Ringkasan Validasi

a. Missing Values:

- Kolom mana yang memiliki missing values?
- Berapa persentase missing values di setiap kolom?

b. Duplikat:

- Apakah ada baris duplikat? Jika ya, berapa banyak?

c. Tipe Data:

- Apakah semua kolom memiliki tipe data yang sesuai?
- Adakah nilai non-numerik dalam kolom numerik?

d. Range Nilai:

- Apakah range nilai untuk setiap kolom masuk akal?
- Adakah outlier yang perlu diperhatikan?

Unit 3 : Data Validation

e. Konsistensi Data:

- Apakah ada inkonsistensi antara 'education' dan 'education_num'?
- Adakah kategori yang tidak masuk akal atau salah eja?

f. Rekomendasi:

- Langkah-langkah apa yang perlu diambil untuk membersihkan data?
- Apakah ada fitur yang perlu ditransformasi atau di-encode?

Unit 3 : Data Validation

Jawaban Rekomendasi

Penanganan Missing Values:

Untuk kolom '**workclass**', '**occupation**', dan '**native_country**' yang memiliki **missing values**, rekomendasi:

- Imput dengan **modus** (nilai yang paling sering muncul) untuk setiap kolom karena tipe data adalah **Object atau Kategorikal**
- Atau, buat kategori baru '**Unknown**' untuk missing values.

Justifikasi: **Metode ini mempertahankan informasi tanpa menghilangkan data.**

Unit 3 : Data Validation

Penanganan Outlier:

Untuk '**capital_gain**' dan '**capital_loss**', rekomendasi:

- Gunakan **winsorization** (membatasi nilai ekstrem ke persentil tertentu, misalnya 1% dan 99%).
- Atau, **log-transform** untuk mengurangi skewness (kecondongan).

Justifikasi: **Mengurangi pengaruh outlier ekstrem tanpa menghilangkan data**.

Encoding Variabel Kategorikal:

Untuk '**workclass**', '**education**', '**marital_status**', '**occupation**', '**relationship**', '**race**', '**sex**', '**native_country**':

- Gunakan **one-hot encoding** untuk variabel dengan **kardinalitas rendah**.
- Gunakan **target encoding** untuk variabel dengan **kardinalitas tinggi** seperti 'native_country'.

Justifikasi: **Memungkinkan model untuk memahami variabel kategorikal dengan lebih baik**.

Unit 3 : Data Validation

Feature Engineering:

- Buat fitur baru **'age_group'** berdasarkan **'age'**.
- Kombinasikan **'capital_gain'** dan **'capital_loss'** menjadi **'net_capital'**.

Justifikasi: **Menyederhanakan informasi dan potensial meningkatkan prediktabilitas**.

Normalisasi/Standardisasi:

- **Standardisasi fitur numerik** seperti **'age', 'fnlwgt', 'education_num', 'hours_per_week'**.

Justifikasi: **Memastikan semua fitur memiliki skala yang sebanding, penting untuk beberapa algoritma machine learning**.

Unit 3 : Data Validation

Penanganan Kelas Tidak Seimbang:

Gunakan teknik seperti **SMOTE (Synthetic Minority Over-sampling Technique)** untuk menyeimbangkan kelas target.

Teknik **SMOTE** yaitu **membangkitkan sampel baru yang berasal dari kelas minoritas untuk membuat proporsi data menjadi lebih seimbang dengan cara sampling ulang sampel kelas minoritas**

Justifikasi: **Meningkatkan kemampuan model untuk memprediksi kelas minoritas dengan akurat**.

Validasi Silang:

Implementasikan **stratified k-fold cross-validation** untuk memastikan representasi yang konsisten dari kedua kelas target dalam setiap fold.

Justifikasi: **Meningkatkan robustness evaluasi model, terutama dengan adanya ketidakseimbangan kelas**.