Tugas perbaikan
Bab 4

Nama : muhammad makhlufi makbullah

Kelas : TK 45 01

NIM : 1103210171

"Representing Data and Engineering Features"

1. Pentingnya Representasi Data

- **Garbage In, Garbage Out**: Model machine learning hanya sebaik data yang digunakan. Jika data tidak diwakili dengan baik, performa model akan buruk.
- Data harus diubah menjadi representasi numerik yang dapat diproses oleh algoritma machine learning.

2. Apa Itu Fitur Engineering?

- **Feature Engineering** adalah proses mengubah data mentah menjadi fitur yang dapat digunakan oleh algoritma.
- Proses ini mencakup:
    - Seleksi fitur yang relevan.
    - Transformasi atau encoding data menjadi bentuk yang lebih bermakna.
    - Pembuatan fitur baru yang mengandung lebih banyak informasi.

3. Representasi Numerik untuk Kategori

- Data kategori sering muncul dalam dataset, seperti jenis kelamin, kota, atau preferensi.
- Dua metode umum untuk menangani data kategori:
    - **One-Hot Encoding**: Mengubah kategori menjadi vektor biner. Setiap kategori diwakili sebagai satu kolom.
    - **Ordinal Encoding**: Memberikan nilai numerik pada kategori, tetapi hanya cocok jika kategori memiliki urutan logis.

4. Fitur Derivatif

- Fitur baru dapat dibuat dengan menggabungkan atau memodifikasi fitur yang ada, misalnya:
    - Menggabungkan kolom tanggal menjadi kolom seperti bulan, hari, atau tahun.
    - Menggunakan operasi matematis seperti penjumlahan, pengurangan, atau rasio dari dua fitur.

5. Skala Data

- Banyak algoritma machine learning peka terhadap skala data. Fitur dengan rentang yang sangat berbeda dapat mendominasi hasil model.
- Metode normalisasi umum:
    - **Min-Max Scaling**: Menyesuaikan data agar berada di antara 0 dan 1.
    - **Standard Scaling**: Menstandarkan data menjadi distribusi normal dengan mean 0 dan standar deviasi 1.

6. Missing Data

- Data yang hilang seringkali perlu ditangani sebelum digunakan dalam model.
- Teknik-teknik untuk menangani data yang hilang:
    - Menghapus baris atau kolom dengan data hilang.
    - Mengisi nilai hilang dengan mean, median, mode, atau teknik lainnya.
    - Menggunakan model machine learning untuk memperkirakan nilai yang hilang.

7. Interaksi Fitur

- Interaksi antara fitur dapat memberikan informasi tambahan.
- Contoh: Kombinasi antara lokasi dan waktu dapat memberikan wawasan lebih daripada hanya menggunakan masing-masing fitur secara terpisah.

8. Representasi untuk Data Teks

- Data teks perlu diubah menjadi vektor numerik untuk digunakan dalam model:
    - **Bag-of-Words**: Menghitung frekuensi kata dalam dokumen.
    - **TF-IDF (Term Frequency-Inverse Document Frequency)**: Mengukur relevansi kata berdasarkan frekuensi relatif.

9. Representasi untuk Data Gambar

- Data gambar sering diubah menjadi array numerik berdasarkan nilai piksel.
- Teknik preprocessing tambahan, seperti pengubahan ukuran atau normalisasi intensitas piksel, sering digunakan.

10. Pipelines untuk Preprocessing

- **Pipelines** memungkinkan preprocessing dan pelatihan model dilakukan dalam satu langkah.
- Menggunakan **Pipeline** dari scikit-learn, Anda dapat memastikan bahwa langkah preprocessing diterapkan secara konsisten.

Penjelasan untuk masing-masing sub-bab dari Bab 4:

1. Categorical Variables

- **Variabel kategori** adalah fitur yang nilainya berbentuk kategori atau label, seperti nama kota, jenis kelamin, atau warna.
- Machine learning algoritma memerlukan data numerik, sehingga kategori harus diubah menjadi bentuk numerik sebelum digunakan dalam model.

---

2. One-Hot-Encoding (Dummy Variables)

- Salah satu metode paling umum untuk menangani variabel kategori adalah **one-hot encoding**.
- Setiap kategori diubah menjadi kolom biner (1 untuk kehadiran kategori, 0 untuk ketidakhadirannya).
- Contoh:
    - Kategori: [“Merah”, “Hijau”, “Biru”]
    - One-hot encoding:

Merah	Hijau	Biru
1	0	0
0	1	0
0	0	1

- Memungkinkan model untuk menangani data kategori tanpa memberikan bobot numerik yang tidak relevan.

---

3. Numbers Can Encode Categoricals

- Menggunakan angka untuk merepresentasikan kategori dapat menyebabkan model menganggap ada urutan atau hubungan numerik antara kategori.
- Contoh:
    - ["Kecil", "Sedang", "Besar"] direpresentasikan sebagai [1, 2, 3].
    - Representasi ini hanya cocok jika kategori memiliki urutan logis.
- Metode ini dapat berbahaya jika diterapkan pada data kategori tanpa urutan.

---

4. Binning, Discretization, Linear Models, and Trees

- **Binning**: Membagi fitur numerik menjadi beberapa kategori atau interval.
    - Contoh: Usia dibagi menjadi kelompok (0-18, 19-35, 36-50, dst.).
- Discretization membantu algoritma menangkap pola non-linear.
- Linear models: Membutuhkan transformasi data agar pola non-linear terlihat.
- Tree-based models: Tidak memerlukan binning karena pohon keputusan dapat menangani fitur dengan skala kontinu.

---

5. Interactions and Polynomials

- **Interaksi**: Menggabungkan dua atau lebih fitur untuk membuat fitur baru.
    - Contoh: “Lokasi” + “Waktu” untuk fitur baru seperti “Waktu Lokasi Spesifik”.
- **Polynomials**: Menambahkan pangkat dari fitur, misalnya, x2x^2x2, x3x^3x3, dst.
    - Contoh: Model linier sederhana dapat diperluas untuk menangkap hubungan non-linear dengan menambahkan fitur polinomial.

---

6. Univariate Nonlinear Transformations

- Transformasi non-linear diterapkan pada fitur individu untuk menangani distribusi yang tidak normal atau pola non-linear.
- Contoh transformasi:
    - **Log** untuk data dengan distribusi eksponensial.
    - **Square root** untuk mengurangi efek outlier.
    - **Exponential** untuk meningkatkan fitur dengan nilai rendah.

---

7. Automatic Feature Selection

- Proses otomatis untuk memilih subset fitur yang paling relevan untuk model.
- Membantu mengurangi kompleksitas model, meningkatkan akurasi, dan mengurangi overfitting.
- Ada tiga metode utama:
    - **Univariate Statistics**
    - **Model-Based Feature Selection**
    - **Iterative Feature Selection**

---

8. Univariate Statistics

- Memilih fitur berdasarkan hubungan statistik antara fitur dan target.
- Contoh:
    - Menggunakan pengujian statistik seperti $\chi 2$\chi^2$\chi 2$ atau F-test untuk memilih fitur yang paling relevan.
- Hanya mempertimbangkan satu fitur pada satu waktu, sehingga tidak menangkap interaksi antar fitur.

---

9. Model-Based Feature Selection

- Menggunakan model machine learning untuk menilai pentingnya fitur.
- Contoh:
    - Pohon keputusan atau model berbasis ensemble seperti Random Forest memberikan nilai penting fitur.
- Fitur dengan kontribusi kecil dapat dihapus untuk menyederhanakan model.

---

10. Iterative Feature Selection

- Memilih fitur secara iteratif berdasarkan performa model:
    - **Forward Selection**: Menambahkan fitur satu per satu ke model.
    - **Backward Elimination**: Menghapus fitur satu per satu dari model.
    - **Recursive Feature Elimination (RFE)**: Menghapus fitur dengan kontribusi terkecil secara iteratif.

---

11. Utilizing Expert Knowledge

- Menggunakan pengetahuan domain atau wawasan ahli untuk menciptakan atau memilih fitur.
- Contoh: Dalam bidang medis, memilih fitur berdasarkan relevansinya dengan diagnosis penyakit.

---

12. Summary and Outlook

- Pentingnya representasi data dan feature engineering untuk meningkatkan performa model.
- Berbagai teknik, seperti one-hot encoding, interaksi, dan transformasi non-linear, memberikan fleksibilitas dalam menangani berbagai jenis data.
- Pemilihan fitur secara otomatis atau manual memainkan peran penting dalam mengurangi kompleksitas dan meningkatkan efisiensi model.