**Siapkan data untuk pembelajaran mesin dengan Azure Databricks**

Sebelum menggunakan data untuk melatih model pembelajaran mesin, penting untuk menyiapkan data dengan tepat.

**Tujuan pembelajaran**

Setelah menyelesaikan modul ini, Anda akan dapat:

* Memahami konsep pembelajaran mesin
* Lakukan pembersihan data
* Lakukan rekayasa fitur
* Lakukan penskalaan data
* Lakukan pengodean data

1. **Pengantar**

Pelajari cara menyiapkan data untuk pembelajaran mesin di Azure Databricks.

1. **Tujuan pembelajaran**

Setelah menyelesaikan modul ini, Anda akan mampu:

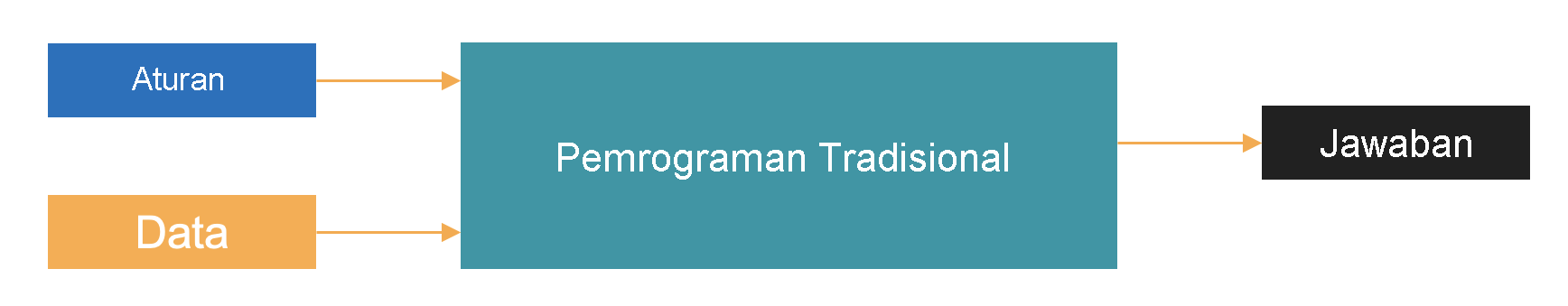
* Menjelaskan konsep pembelajaran mesin.
* Lakukan pembersihan data.
* Melakukan rekayasa fitur.
* Lakukan penskalaan data.
* Lakukan pengodean data.

1. **Memahami konsep pembelajaran mesin**

Pembelajaran mesin adalah teknik ilmu data yang digunakan untuk mengekstrak pola dari data yang memungkinkan komputer mengidentifikasi data terkait, memperkirakan hasil, perilaku, dan tren masa depan.

1. **Pembelajaran mesin sebagai paradigma pemrograman baru**
2. **Pemrograman tradisional**

Dalam pemrograman tradisional, input dari aturan dan data hard-coded digunakan untuk sampai pada output jawaban.



Anda menyediakan program tradisional dengan aturan dan data, dan sebagai imbalannya, itu memberikan hasil atau jawaban Anda.

1. **Pembelajaran Mesin**

Hasil dari pelatihan algoritma pembelajaran mesin adalah algoritma tersebut telah mempelajari aturan untuk memetakan data input menjadi jawaban.



Dalam pembelajaran mesin, Anda melatih algoritme dengan data dan jawaban, juga dikenal sebagai label, dan algoritme mempelajari aturan untuk memetakan data ke label masing-masing.

**3. Lakukan pembersihan data**

**Big Data** telah menjadi bagian dari leksikon organisasi di seluruh dunia, karena semakin banyak organisasi yang memanfaatkan data untuk mendorong keputusan bisnis yang lebih tepat. Dengan evolusi dalam pengambilan keputusan bisnis ini, jumlah data mentah yang dikumpulkan, bersama dengan jumlah dan keragaman sumber data, tumbuh pada tingkat yang mencengangkan.

Data mentah, bagaimanapun, seringkali berisik dan tidak dapat diandalkan dan mungkin mengandung nilai dan outlier yang hilang. Menggunakan data tersebut untuk **Pembelajaran Mesin** dapat menghasilkan hasil yang menyesatkan. Dengan demikian, pembersihan data dari data mentah adalah salah satu langkah terpenting dalam menyiapkan data untuk Pembelajaran Mesin. Karena algoritma Pembelajaran Mesin mempelajari aturan dari data, memiliki data yang bersih dan konsisten merupakan faktor penting dalam mempengaruhi kemampuan prediktif dari algoritma yang mendasarinya.

Jenis data paling umum yang tersedia untuk pembelajaran mesin adalah dalam format tabel. Data tabular biasanya tersedia dalam bentuk baris dan kolom. Dalam data tabular, baris menggambarkan satu pengamatan, dan setiap kolom menggambarkan sifat pengamatan yang berbeda. Nilai kolom dapat terus menerus (numerik), diskrit (kategoris), datetime (deret waktu), atau teks. Kolom yang dipilih sebagai input ke model Pembelajaran Mesin juga dikenal sebagai fitur model.

Pembersihan data berkaitan dengan masalah dalam kualitas data seperti kesalahan, nilai yang hilang, dan outlier. Ada beberapa teknik dalam menangani masalah kualitas data dan kami akan membahas beberapa pendekatan umum di bagian ini.

1. **Imputasi nilai nol**

Nilai nol mengacu pada data yang tidak diketahui atau hilang serta tanggapan yang tidak relevan. Strategi untuk menghadapi skenario ini meliputi:

* Menjatuhkan catatan ini: Berfungsi saat Anda tidak perlu menggunakan informasi untuk beban kerja hilir.
* Menambahkan placeholder (misalnya, -1): Memungkinkan Anda melihat data yang hilang nanti tanpa melanggar skema.
* Imputasi dasar: Memungkinkan Anda memiliki "tebakan terbaik" tentang data yang sebenarnya, seringkali dengan menggunakan rata-rata atau median data yang tidak hilang untuk tipe data numerik, atau Nilai paling\_sering dari data yang tidak hilang untuk tipe data kategorikal.
* Imputasi tingkat lanjut: Menentukan "tebakan terbaik" dari data apa yang harus menggunakan strategi yang lebih canggih seperti pengelompokan algoritma pembelajaran mesin atau teknik oversampling seperti SMOTE (Synthetic Minority Over-sampling Technique).

1. **Mengonversi tipe data**

Nilai nol mengacu pada data yang tidak diketahui atau hilang serta tanggapan yang tidak relevan. Strategi untuk menghadapi skenario ini meliputi:

* Menjatuhkan catatan ini: Berfungsi saat Anda tidak perlu menggunakan informasi untuk beban kerja hilir.
* Menambahkan placeholder (misalnya, -1): Memungkinkan Anda melihat data yang hilang nanti tanpa melanggar skema.
* Imputasi dasar: Memungkinkan Anda memiliki "tebakan terbaik" tentang data yang sebenarnya, seringkali dengan menggunakan rata-rata atau median data yang tidak hilang untuk tipe data numerik, atau Nilai paling\_sering dari data yang tidak hilang untuk tipe data kategorikal.
* Imputasi tingkat lanjut: Menentukan "tebakan terbaik" dari data apa yang harus menggunakan strategi yang lebih canggih seperti pengelompokan algoritma pembelajaran mesin atau teknik oversampling seperti SMOTE (Synthetic Minority Over-sampling Technique).

1. **Mengonversi tipe data**

Dalam beberapa situasi, kolom memiliki tipe data yang tidak konsisten. Misalnya, kolom dapat memiliki kombinasi angka yang disajikan sebagai string, seperti **"44.5" dan "25.1".** Sebagai bagian dari pembersihan data, Anda sering kali harus mengonversi data di kolom ke tipe data yang benar.

1. **Catatan duplikat**

Dalam beberapa situasi, Anda menemukan rekaman duplikat dalam tabel. Solusi termudah adalah membuang catatan duplikat.

1. **Pencilan**

Pencilan didefinisikan sebagai pengamatan yang berbeda secara signifikan dengan semua pengamatan lain dalam kolom tertentu. Ada beberapa cara untuk mengidentifikasi outlier, dan satu pendekatan umum adalah menghitung Z-score untuk pengamatan x.

Anda dapat menggunakan strategi serupa seperti memasukkan nilai nol untuk menangani outlier. Namun, penting untuk dicatat bahwa outlier belum tentu data yang tidak valid dan, dalam beberapa situasi, sangat valid untuk mempertahankan outlier dalam data pelatihan Anda.

**4. Lakukan rekayasa fitur**

Model pembelajaran mesin sekuat data yang mereka latih. Seringkali penting untuk mendapatkan fitur dari data mentah yang ada yang lebih mewakili sifat data dan dengan demikian membantu meningkatkan daya prediksi dari algoritma pembelajaran mesin. Proses menghasilkan fitur prediktif baru dari data mentah yang ada biasanya disebut sebagai **rekayasa fitur.**

1. **Rekayasa fitur**

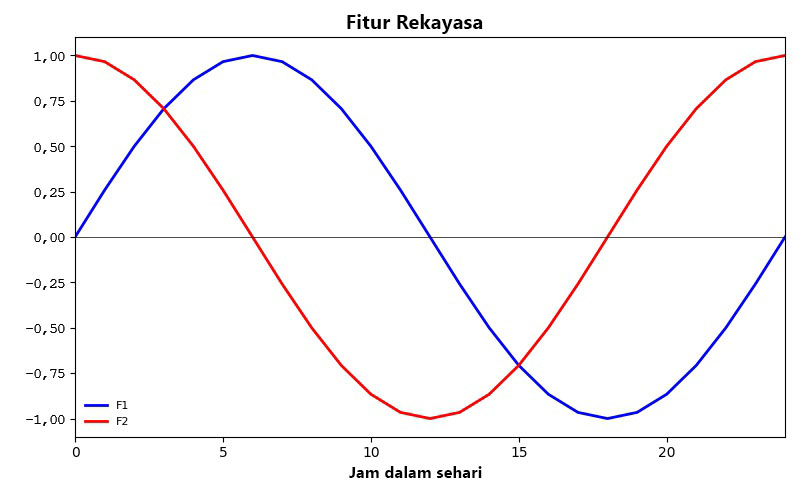
Ada banyak pendekatan yang valid untuk rekayasa fitur dan beberapa yang paling populer, dikategorikan berdasarkan tipe data, adalah sebagai berikut:

* Agregasi (hitungan, jumlah, rata-rata, rata-rata, median, dan sejenisnya)
* Bagian-dari (tahun tanggal, bulan tanggal, minggu tanggal, dan sejenisnya)
* Binning (mengelompokkan entitas ke dalam bin dan kemudian menerapkan agregasi)
* Menandai (kondisi boolean yang menghasilkan True of False)
* Berbasis frekuensi (menghitung frekuensi tingkat satu atau lebih variabel kategoris)
* Embedding (mengubah satu atau lebih fitur kategoris atau teks menjadi satu set fitur baru, mungkin dengan kardinalitas yang berbeda)
* Berasal dari contoh

Rekayasa fitur tidak terbatas pada daftar di atas dan dapat melibatkan pendekatan berbasis pengetahuan domain untuk menurunkan fitur.

Mari bekerja dengan contoh untuk memahami proses rekayasa fitur. Dalam contoh kami, kami bekerja dengan sistem yang memberi kami data cuaca setiap jam, dan kami memiliki kolom dalam data yaitu **hour of day.** Kolom **hour of day** bertipe integer dan dapat mengasumsikan nilai integer dalam rentang **[0, 23].**

Pertanyaannya adalah, bagaimana cara terbaik untuk merepresentasikan data ini ke algoritme pembelajaran mesin yang dapat mempelajari sifat siklusnya? Salah satu pendekatannya adalah merekayasa sekumpulan fitur baru yang mengubah kolom **hour of day** menggunakan fungsi sinus dan kosinus. Fitur turunan ini diplot pada gambar di bagian untuk rentang **[0, 24]:**



Fungsi kosinus memberikan bobot yang sama secara simetris untuk jam AM dan PM yang sesuai, dan fungsi sinus memberikan bobot yang berlawanan secara simetris dengan jam AM dan PM yang sesuai. Kedua fungsi menangkap sifat siklus **hour of day.**

1. **Lakukan penskalaan data**

Penskalaan fitur numerik adalah bagian penting dari data prapemrosesan untuk pembelajaran mesin. Biasanya rentang nilai yang diambil setiap fitur masukan sangat bervariasi antar fitur. Ada banyak algoritme pembelajaran mesin yang peka terhadap besaran fitur input dan dengan demikian tanpa penskalaan fitur, bobot yang lebih tinggi dapat ditetapkan ke fitur dengan magnitudo yang lebih tinggi terlepas dari pentingnya fitur pada keluaran yang diprediksi.

Ada dua pendekatan umum untuk menskalakan fitur numerik:

* **Normalisasi kasus**
* **Standardisasi**

Kami akan membahas masing-masing pendekatan di bagian ini.

1. **Normalisasi kasus**

Normalisasi secara matematis mengubah skala data ke dalam rentang [0, 1].

Misalnya, untuk setiap nilai individual, Anda dapat mengurangi nilai minimum untuk input tersebut dalam set data pelatihan, lalu membaginya dengan rentang nilai dalam set data pelatihan. Rentang nilai adalah selisih antara nilai maksimum dan nilai minimum.

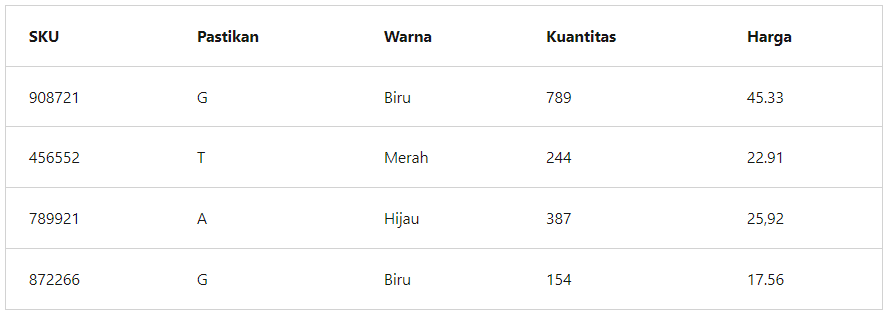
1. **Standardisasi**

Standardisasi mengubah skala data menjadi rata-rata = 0 dan standar deviasi = 1.

Untuk input numerik, pertama-tama Anda menghitung mean dan standar deviasi menggunakan semua data yang tersedia dalam set data pelatihan. Kemudian, untuk setiap nilai input individu, Anda menskalakan nilai tersebut dengan mengurangkan mean dan kemudian membaginya dengan standar deviasi.

1. **Lakukan pengodean data**

Jenis data umum yang lazim dalam pembelajaran mesin disebut data kategoris. Data kategoris menyiratkan diskrit atau seperangkat nilai yang terbatas. Misalnya, jenis kelamin atau etnis seseorang dianggap kategoris. Mari kita perhatikan tabel data berikut:

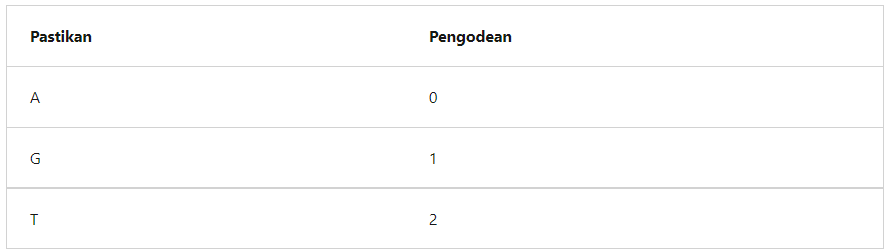


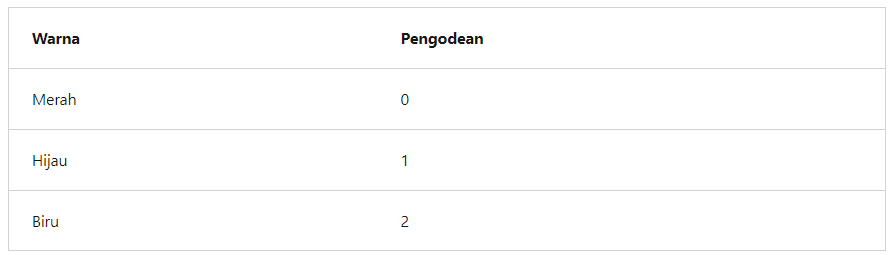
Pada tabel di atas, baris menggambarkan satu pengamatan, dan setiap kolom menggambarkan sifat pengamatan yang berbeda. Dalam tabel, kami memiliki dua jenis data, data numerik seperti **Quantity** dan **Price**, dan data kategorikal seperti **Make** dan **Color**. Dalam pelajaran sebelumnya, kita melihat contoh bagaimana menskalakan tipe data numerik. Selain itu, penting untuk dicatat bahwa dalam pembelajaran mesin, kami pada akhirnya selalu bekerja dengan angka atau secara khusus, **vectors**. Dalam konteks ini, vektor adalah array angka, atau array angka bersarang. Jadi bagaimana seseorang mengkodekan data kategoris untuk tujuan pembelajaran mesin? Kita akan melihat dua pendekatan umum untuk **pengodean** data kategoris:

* Pengkodean ordinal
* Pengodean satu panas

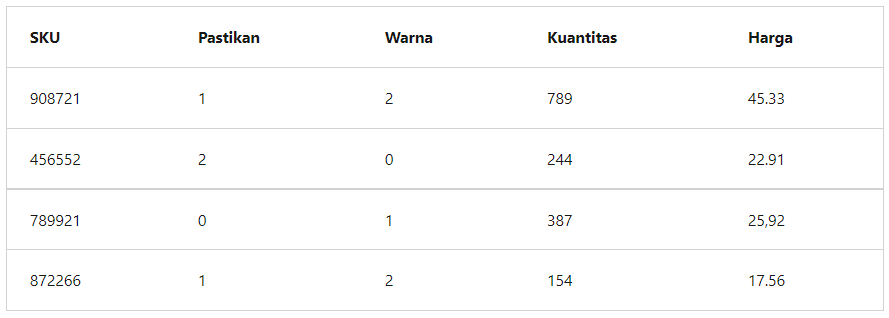
1. **Pengkodean ordinal**

Encoding ordinal, mengubah data kategorikal menjadi kode integer mulai dari 0 hingga (jumlah kategori – 1). Misalnya, kategori **Make** dan **Color** dari tabel di atas dikodekan sebagai:



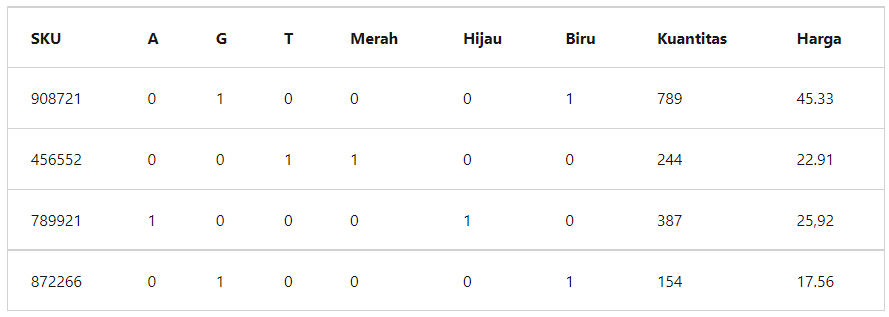


Dengan menggunakan pengodean di atas, tabel yang diubah akan ditampilkan di bawah ini:



1. **Pengodean satu panas**

Pengodean one-hot sering kali merupakan pendekatan yang direkomendasikan, dan melibatkan transformasi setiap nilai kategorikal menjadi n (= jumlah kategori) nilai biner, dengan salah satunya 1, dan yang lainnya 0. Sebagai contoh, tabel di atas dapat diubah menjadi:



Encoding one-hot sering lebih disukai daripada encoding ordinal karena mengkodekan setiap item kategori dengan bobot yang sama. Dalam contoh kami di atas, encoder ordinal menetapkan warna **Green = 1** dan warna **Blue = 2**, dan itu dapat menyiratkan bahwa warna Biru dua kali lebih penting daripada warna Hijau. Padahal, dengan pengodean one-hot, setiap warna berbobot sama.

1. **Latihan - Siapkan data untuk pembelajaran mesin**

Sekarang, ini adalah kesempatan Anda untuk menggunakan Azure Databricks untuk menyiapkan data untuk Pembelajaran Mesin.

Dalam latihan ini, Anda akan:

* Menangani data yang hilang.
* Rekayasa Fitur.
* Penskalaan fitur Numerik.
* Pengodean Fitur Kategoris.

1. **Instruksi**

Ikuti petunjuk berikut untuk menyelesaikan latihan:

* + Buka petunjuk latihan di <https://aka.ms/mslearn-dp090.> \ <https://microsoftlearning.github.io/dp-090-databricks-ml/>
  + Selesaikan latihan Menyiapkan Data untuk Pembelajaran Mesin.

1. **Ringkasan**

Dalam modul ini, Anda mempelajari cara menyiapkan data untuk pembelajaran mesin di Azure Databricks.

Sekarang setelah Anda menyelesaikan modul ini, Anda dapat:

* Menjelaskan konsep pembelajaran mesin.
* Lakukan pembersihan data.
* Melakukan rekayasa fitur.
* Lakukan penskalaan data.
* Lakukan pengodean data.