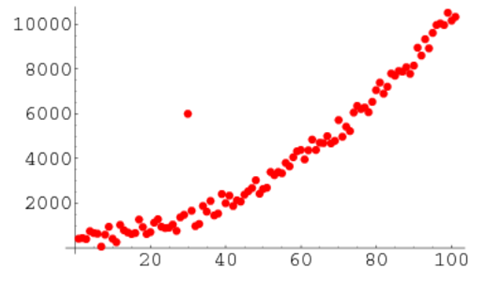
Selain konversi data kategorik menjadi numerik, ada beberapa teknik lain dalam *data preparation*. Teknik yang akan dibahas antara lain membuang *outlier*, *normalization*, dan *standardization*.

**Outlier Removal**

Dalam statistik, outlier adalah sebuah nilai yang jauh berbeda dari kumpulan nilai lainnya dan dapat mengacaukan hasil dari sebuah analisis statistik. Outlier dapat disebabkan oleh kesalahan dalam pengumpulan data atau nilai tersebut benar ada dan memang unik dari kumpulan nilai lainnya.

Apa pun alasan kemunculannya, Anda perlu tahu cara mengidentifikasi dan memproses outlier. Ini adalah bagian penting dalam persiapan data di dalam machine learning. Salah satu cara termudah untuk mengecek apakah terdapat outlier dalam data kita adalah dengan melakukan visualisasi.

Berikut adalah contoh visualisasi terhadap data yang memiliki outlier.



Dapat dilihat dengan jelas bahwa terdapat satu sampel yang jauh berbeda dengan sampel-sampel lainnya. Setelah mengetahui bahwa di data kita terdapat outlier, kita dapat mencari lalu menghapus sampel tersebut dari dataset.

**Normalization**

Normalization adalah salah satu teknik yang dipakai dalam data preparation. Tujuan dari normalisasi adalah mengubah nilai-nilai dari sebuah fitur ke dalam skala yang sama. Normalization memungkinkan kenaikan performa dan stabilitas dari sebuah model machine learning.

| **Nama** | **Gaji** | **Umur** |
| --- | --- | --- |
| A | 12.000.000 | 33 |
| B | 35.000.000 | 45 |
| C | 4.000.000 | 23 |
| D | 6.500.000 | 26 |
| E | 9.000.000 | 29 |

Contoh dari normalization adalah ketika kita memiliki dataset seperti di atas yang memiliki fitur umur dengan skala 23 sampai 45 tahun dan fitur gaji dengan skala 4.000.000 sampai 35.000.000. Di sini kita melihat bahwa fitur gaji sekitar satu juta kali lebih besar dari fitur umur dan menunjukkan kedua fitur ini berada pada skala yang sangat jauh berbeda.

Ketika membangun model seperti regresi linear, fitur gaji akan sangat mempengaruhi prediksi dari model karena nilainya yang jauh lebih besar daripada umur, walaupun tidak berarti fitur tersebut jauh lebih penting dari fitur umur.

Salah satu contoh dari normalization adalah *min-max scaling* di mana nilai-nilai dipetakan ke dalam skala 0 sampai 1. SKLearn menyediakan library untuk normalization

Pada Colab kita **Import** library **MinMaxScaler** dan masukkan data dari tabel sebelumnya.

1. from sklearn.preprocessing import MinMaxScaler
2. data = [[12000000, 33], [35000000, 45], [4000000, 23], [6500000, 26], [9000000, 29]]

Pada cell selanjutnya kita buat sebuah objek MinMaxScaler dan panggil fungsi fit() dan mengisi argumen **data**seperti potongan kode di bawah. Fungsi fit() dari objek MinMaxScaler adalah fungsi untuk menghitung nilai minimum dan maksimum pada tiap kolom.

1. scaler = MinMaxScaler()
2. scaler.fit(data)

Apabila dijalankan, maka hasilnya sebagai berikut.

[dos:258bd96648e41d2363141b20860fe40e20220818143545.png](https://www.dicoding.com/academies/184/tutorials/8352?from=8347)

Sampai pada fungsi fit() ini, komputer baru menghitung nilai minimum dan maksimum pada tiap kolom dan belum melakukan operasi scaler pada data. Terakhir kita panggil fungsi transform() yang akan mengaplikasikan scaler pada data, sebagai berikut.

1. print(scaler.transform(data))

Hasil dari kode di atas seperti ditunjukkan pada gambar berikut.

Setiap nilai dari kolom gaji dan umur telah dipetakan pada skala yang sama seperti di bawah ini.

| **Nama** | **Gaji** | **Umur** |
| --- | --- | --- |
| A | 0.25806452 | 0.45454545 |
| B | 1 | 1 |
| C | 0 | 0 |
| D | 0.08064516 | 0.13636364 |
| E | 0.16129032 | 0.27272727 |

Untuk informasi lebih detail tentang Min Max Scaler, silakan kunjungi [tautan](https://scikit-learn.org/0.16/modules/generated/sklearn.preprocessing.MinMaxScaler.html) berikut.

**Standardization**

Standardization adalah proses konversi nilai-nilai dari suatu fitur sehingga nilai-nilai tersebut memiliki skala yang sama. *Z* score adalah metode paling populer untuk standardisasi di mana setiap nilai pada sebuah atribut numerik akan dikurangi dengan rata-rata dan dibagi dengan standar deviasi dari seluruh nilai pada sebuah kolom atribut.



Fungsi standardisasi itu serupa dengan normalization. Keduanya berfungsi menyamakan skala nilai dari tiap atribut pada data. SKLearn menyediakan library untuk mengaplikasikan standard scaler pada data.

| **Nama** | **Gaji** | **Umur** |
| --- | --- | --- |
| A | 12.000.000 | 33 |
| B | 35.000.000 | 45 |
| C | 4.000.000 | 23 |
| D | 6.500.000 | 26 |
| E | 9.000.000 | 29 |

Pada colab di cell pertama kita akan mengimpor library **preprocessing** dari **scikit learn** lalu membuat data dummy sesuai dengan tabel di atas.

1. from sklearn import preprocessing
2. data = [[12000000, 33], [35000000, 45], [4000000, 23], [6500000, 26], [9000000, 29]]

Selanjutnya kita buat object scaler dan panggil fungsi fit dari scaler pada data. Fungsi fit memiliki fungsi untuk menghitung rata-rata dan deviasi standar dari setiap kolom atribut untuk kemudian dipakai pada fungsi transform.

1. scaler = preprocessing.StandardScaler().fit(data)

Terakhir, kita panggil fungsi transform untuk mengaplikasikan standard scaler pada data. Untuk melihat hasil dari standard scaler kita tinggal memanggil objek scaler yang telah kita buat sebelumnya. Kodenya sebagai berikut.

1. data = scaler.transform(data)
2. data

Hasil akhirnya apabila dijalankan seperti di bawah ini.

Untuk informasi lebih detail tentang standardization, silakan kunjungi [tautan](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html) berikut.