**Apa Yang Harus Dilakukan Dalam Proses Data Cleaning?**

*Data cleaning* adalah salah satu tahapan dalam *data preparation*. Proses *data cleaning* merupakan proses yang penting dilakukan karena akan mempengaruhi hasil *modelling* *machine learning*.

Hampir semua dataset yang akan digunakan untuk modelling tidak dalam kondisi siap pakai, yang artinya kita harus membersihkan dan memperbaiki struktur data yang kita miliki.

*Data preparation* terdiri dari beberapa proses seperti *data cleaning*, *data transformation*, dan *data reduction*. Dalam artikel ini, kita akan membahas tentang hal apa saja yang harus dilakukan dalam proses *data cleaning*.

**Identifikasi dan hapus data duplikat**

Baris data yang memiliki nilai yang sama persis untuk tiap kolomnya kurang berguna untuk dipertahankan. Data duplikat juga sangat mungkin akan mempengaruhi akurasi model *machine learning*.

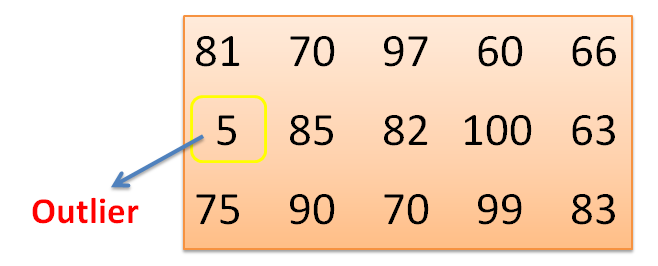
Misalnya saja data duplikat bisa jadi muncul sebagai data *training* dan data *testing* saat kita menggunakan train/test split ataupun K-Fold cross validation untuk evaluasi model. Jika itu terjadi tentunya ini akan menghasilkan performa prediksi yang bias untuk data baru yang belum pernah diujikan.

Di Python, kita dapat menemukan data duplikat dengan *function* [duplicated()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html) dan menghapusnya dengan [drop\_duplicated()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop_duplicates.html).

**Identifikasi dan hapus outliers**

*Outliers* merupakan data yang nilainya terlalu jauh dengan data lainnya atau disebut juga dengan pencilan. Keberadaan *outliers* akan membuat analisis terhadap data tersebut menjadi bias atau dengan kata lain tidak mencerminkan fenomena yang sebenarnya.

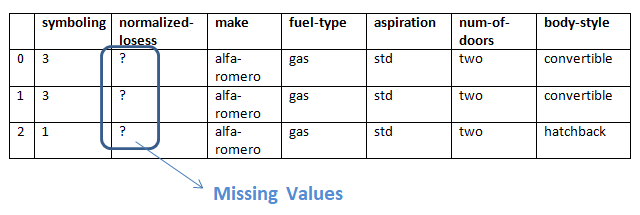
Contoh misalnya nilai mahasiswa dalam satu kelas berkisar antara 60 sampai 100, namun ada satu mahasiswa yang nilainya 5. Nah, nilai 5 ini dianggap sebagai *outliers* karena tentunya akan mempengaruhi statistik data seperti  *mean* alias nilai rata-rata kelas. Oleh sebab itu, biasanya data yang mengandung *outliers* akan dihapus.



Untuk menemukan *outliers*, kita dapat memvisualisasikannya dengan menggunakan Box plot atau Scatter plot. Kita juga dapat menggunakan perhitungan IQR score (*Interquartile Score*) yang merupakan selisih dari kuartil ketiga (persentil 75) dan kuartil pertama (persentil 25) untuk identifikasi *outliers* di dataframe.

**Identifikasi dan tangani missing values**

*Missing values* adalah nilai yang tidak terdefinisi di dataset. Bentuknya beragam, bisa berupa blank cell, ataupun simbol-simbol tertentu seperti NaN (Not a Number), NA (Not Available), ?, -, dan sebagainya. *Missing values* dapat menjadi masalah dalam analisis data serta tentunya dapat mempengaruhi hasil *modelling machine learning*.



Di Python, kita dapat mengidentifikasi *missing values* dengan *function* [isnull()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isnull.html) ataupun [isna()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isna.html) dan menghapusnya dengan *function* [dropna()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html).

Jika tidak ingin menghapusnya, kita bisa mengganti nilai yang hilang tersebut dengan nilai rata-rata atau *mean* jika kolom yang mengandung *missing values* adalah kolom numerik dengan nilai kontinu. Atau menggunakan *mode* atau nilai yang paling banyak muncul jika kolom tersebut bertipe kategori.

Selain yang telah disebutkan, kita juga dapat mengganti *missing values* dengan nilai apapun yang dirasa lebih pas untuk menggantikan data yang hilang tersebut sesuai dengan kondisi dataset.

Untuk mengganti *missing values* di Python dapat menggunakan *function* [replace()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.replace.html) ataupun [fillna()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html).

**Hapus kolom yang hanya memiliki satu nilai**

Selanjutnya yang dapat dilakukan dalam proses *data cleaning* adalah menghapus kolom yang hanya memiliki satu nilai atau nilai tunggal. Kolom yang hanya memiliki satu nilai sejatinya tidak akan memberikan dampak yang signifikan pada model *machine learning* dan mungkin tidak berguna untuk proses *modelling*.

Contoh data yang memiliki satu nilai atau nilai tunggal misalnya data siswa dari sekolah khusus wanita, dimana kolom jenis kelamin tentunya tidak kita butuhkan untuk proses modelling karena sudah pasti nilainya hanya ada satu, yaitu ‘wanita’, tidak mungkin ada jenis kelamin ‘laki-laki’ di data tersebut.

Bagaimana cara mengetahui sebuah kolom di dataframe memiliki nilai tunggal atau tidak?

Di Python, kita dapat mengetahui nilai unik pada setiap kolom dengan menggunakan *function* [unique()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.unique.html). Jika menemukan kolom yang hanya memiliki satu nilai unik, kita dapat menghapusnya dengan *function* [drop()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html).

**Hapus kolom yang tidak representatif**

Dalam proses *data cleaning* ini, kita harus memahami lebih dalam tentang dataset yang sedang kita kerjakan. Kita dapat menentukan kolom apa saja yang sekiranya penting dan tidak penting sebagai fitur untuk proses modelling.

Dalam beberapa kasus, misalnya kolom ID dalam dataset mungkin tidak diperlukan untuk *modelling*, namun mungkin juga diperlukan sehingga tidak masalah jika kita membiarkannya sebagai fitur. Untuk itu, diperlukan pemahaman tentang dataset dan apa yang ingin dihasilkan dari dataset tersebut sebagai pertimbangan kita apakah ingin menghapus atau membiarkannya.

Jika Anda ingin belajar *data science* dengan bahasa pemrograman Python lebih dalam, IlmudataPy telah menyediakan sarana belajar tersebut yaitu [Paket E-Modul Data Science](https://ilmudatapy.com/paket-e-modul-data-science/) untuk pemula yang akan membantu Anda belajar *data science* dengan Python dengan lebih terstruktur. [Paket E-Modul Data Science](https://ilmudatapy.com/paket-e-modul-data-science/) dapat Anda peroleh dengan harga terjangkau dan sangat cocok untuk Anda yang terbiasa belajar secara mandiri dengan mengamati langsung kode program dan menjalankannya pada Jupyter Notebook.

Namun jangan khawatir, bagi Anda yang belum memahami kode program, paket modul ini juga disertai dengan penjelasan untuk tiap blok kodenya karena memang sudah dirancang agar dapat dipelajari secara mandiri.