DQLab

liveClass

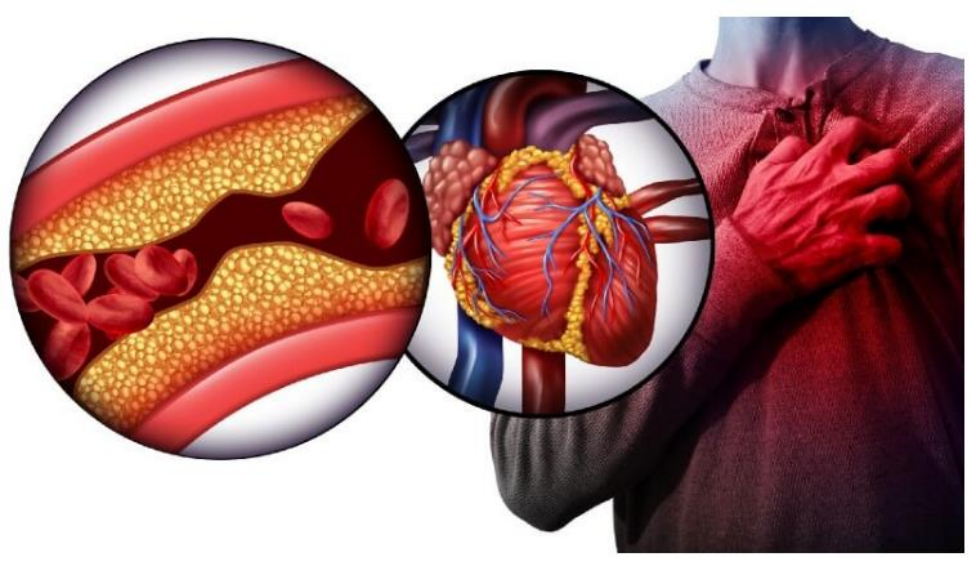
Project Implementation

Outline

- Introduction to Capstone Project
- Data Preprocessing and Cleaning
- Exploratory Data Analysis
- Dimensionality Reduction

Heart Disease:

- Penyebab kematian nomor satu secara global
- Membangun kesadaran dalam menjaga pola hidup
- Prediksi penyakit jantung akan berguna untuk perawatan dini



PROBLEM STATEMENT

Masalah yang ingin kita selesaikan adalah melakukan diagnosa pasien penderita penyakit jantung secara tepat dan akurat. Perlu dilakukan analisis faktor-faktor penyebab dan gejala penyakit jantung pada pasien.

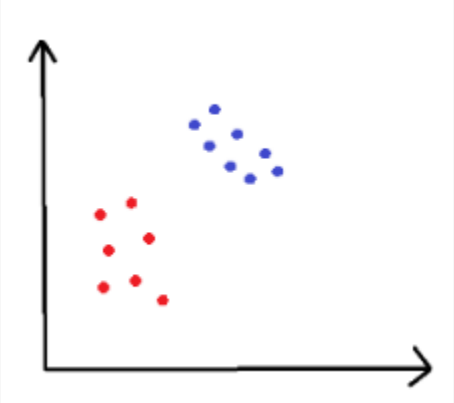
SUMBER DATASET

Dari 76 attribute, terpilih 14 attribute dalam dataset

https://archive.ics.uci.edu/dataset/45/heart+disease

Supervised Learning

- Classification – menetapkan klasifikasi data baru berdasarkan observasi
- Classifiers – mempelajari data training untuk melakukan prediksi data testing
- Target: a binary variable 0/1 untuk tidak penyakit jantung dan penyakit jantung
- Feature: variabel yang digunakan untuk memprediksi target



Liat Sampel Data

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

- Setiap baris memiliki target penyakit jantung dan tidak berdasarkan hal-hal yang mendorong itu terjadi
- Kita dapat melakukan analisa lebih lanjut dari data feature, misal filtering atau combine. Contoh `.isin()` : `df.columns.isin(["1"])` – Memilah pria

Analisis Feature

```
print(df.sex.value_counts())
```

```
sex
1    713
0    312
```

```
print(df.groupby("sex")["target"].sum())
```

```
sex
0    226
1    300
```

Review

Manakah yang tepat dari eksplorasi sebelumnya:

a) Pria memiliki potensi penyakit jantung lebih besar dari wanita

b) Wanita memiliki potensi penyakit jantung lebih besar dari pria

c) Jumlah wanita lebih banyak dari pria

d) Target (0/1) dapat digunakan sebagai features

Exploratory Data Analysis

Penting dilakukan untuk mengali informasi dari fitur yang kita miliki.

- Lihat fitur lebih dekat dengan print(df.columns) dan print(df.dtypes)
- Untuk melihat tipe data tertentu print(df.select_dtypes(include=["int"])

```
Index(['cp', 'thalach', 'slope', 'oldpeak', 'exang', 'ca', 'thal', 'sex',
       'age', 'target'],
      dtype='object')
```

```
cp            int64
thalach       int64
slope         int64
oldpeak     float64
exang         int64
ca            int64
thal          int64
sex           int64
age           int64
target        int64
dtype: object
```

```
df.select_dtypes(include=["int64"])
```

	cp	thalach	slope	exang	ca	thal	sex	age	target
0	0	168	2	0	2	3	1	52	0
1	0	155	0	1	0	3	1	53	0
2	0	125	0	1	0	3	1	70	0
3	0	161	2	0	1	3	1	61	0
4	0	106	1	0	3	2	0	62	0
...	...	...	...	...	...	...	...	...	...
723	2	115	1	0	0	2	0	68	1
733	2	175	1	0	0	2	0	44	1
739	0	161	2	1	1	3	1	52	0
843	3	125	2	0	0	2	1	59	0
878	0	113	1	0	1	3	1	54	0

Missing Value

Kita menemukan row yang missing dari dataset yang kita miliki dengan

- df.info() untuk mendapatkan informasi dari dataset
- df['id'].isnull() untuk mengetahui jumlah null dalam satu kolom
- Untuk sumbu = 1 adalah baris dan sumbu = 0 adalah kolom, maka dari untuk mendapatkan jumlah missing value dalam column dengan df.isnull().sum(axis=0) atau df.isnull().sum(axis=0).sum()
- Missing value bisa digantikan mean/median untuk numerikal dan modus untuk kategorikal

Distribusi Data

Langkah yang bisa kita lakukan untuk mengetahui distribusi dengan

- df.groupby(['cp', 'target']).size().unstack() untuk mendapatkan distribusi 1 dan 0 dalam grup cp (dengan melihat pengaruh target/fitur, kita bisa melakukan analisa lebih lanjut)

target	0	1
cp		
0	93	37
1	8	41
2	17	65
3	7	15

Feature Engineering

Hal yang dapat kita lakukan mengubah variabel kategorikal menjadi numerik:

Salah satu cara dengan **hashing** (mengubah input arbitrer menjadi bilangan bulat, dan memberikan output yang sama untuk setiap input)

Lambda func: lambda x: f(x) dengan memasukan f(x) = hash(x), sehingga

df[“gender”] = df[‘gender’].apply(lambda x: hash(x))

```
df.sex.replace({0:"female",
                1:"male"}).apply(lambda x: hash(x)).value_counts()
```

```
sex
-5289500399743831148    198
-4475904408058165709     85
Name: count, dtype: int64
```

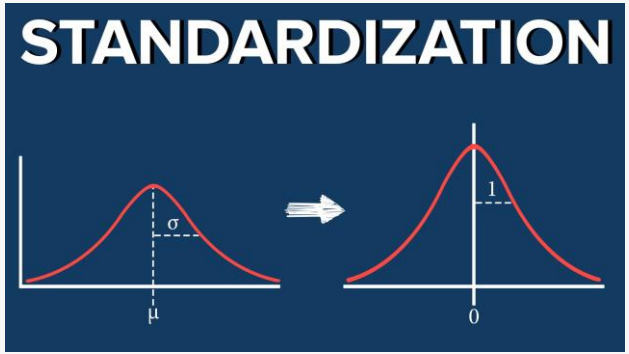
Ingat, bahwa banyak data adalah categorical, dan kita perlu melaukan feature engineering

Standardizing Feature

Memastikan bahwa data sesuai dengan asumsi yang dimiliki model tentang fitur-fiturnya (maksudnya adalah, mungkin ada beberapa fitur yang memiliki high variance yang mendominasi models)

Hal ini mendorong untuk mendapatkan model prediksi yang sebaik mungkin.

Note. Standardisasi tidak cocok diterapkan apabil data yang kita miliki adalah object.



Scaling mengubah semua fitur memiliki rata-rata 0 dengan standar deviasi 1


Thanks!