IS411-C

Hearth Disease

Menggunakan Algoritma Logistic Reggresion dan Decision Tree

KELOMPOK 5

Anggota

1. Katherine Allen Lius - 00000044462
2. Reinhard - 00000045346
3. Theresia Cindana - 00000044538
4. Tralya Dharmada - 00000044343
5. Valencia Eurelia A.T - 00000046227

Latar Belakang

Penyakit jantung adalah salah satu penyakit berbahaya yang dapat menjadi alasan kematian seseorang. Penyakit ini memiliki berbagai faktor penyebab seperti usia, jenis kelamin, hipertensi, kadar kolesterol, detak jantung, hingga aktivitas fisik. Banyaknya faktor yang menjadi pemicu seseorang mengalami sakit jantung membuat penyakit ini sulit untuk diprediksi. Namun dengan memanfaatkan machine learning, sebuah rekam medis dapat diolah dan digunakan untuk memprediksi penyakit jantung. Pada penelitian ini, dilakukan perbandingan dua algoritma berbeda, yaitu logisitic regression dan decision tree. Tujuan penggunaan kedua algoritma tersebut adalah untuk membandingkan klasifikasi mana yang lebih akurat untuk memprediksi penyakit jantung.

Research Purpose

Memanfaatkan teknologi machine learning untuk memprediksi dan menganalisa apakah seseorang menderita penyakit koroner sehingga dapat dilakukan penanganan sedini mungkin.

Metode Penelitian

Metodologi yang digunakan dalam penelitian ini yaitu metodologi CRISP-DM:

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modelling
5. Evaluation
6. Deployment

Alasan memilih algoritma

Logistic Regression

- Dapat bekerja dengan variabel numerik maupun kategoris.
- Dapat menjelaskan hubungan antara variabel biner (dependen) dengan variabel independen non biner

Decision Tree

- Dapat digunakan untuk variabel numerik ataupun kategorik.
- Hasil klasifikasi mudah disimpulkan,
- Tingkat akurasi decision tree cukup tinggi.

Alasan memilih algoritma

Kenapa menggunakan 2 algoritma?

Karena penelitian ini ingin mencari tahu, algoritma apa yang menghasilkan nilai akurasi serta model yang paling baik dalam memprediksi penyakit jantung.

Hasil dan Pembahasan

Business Understanding

Tujuan bisnis dari penelitian adalah untuk memprediksi banyaknya masyarakat yang berpotensi memiliki penyakit jantung koroner berdasarkan gejala yang ada.

Data Understanding

Dataset **"Heart Disease Cleveland UCI"** tahun 2020 diperoleh dari website kaggle.

Data penyakit jantung ini terdiri dari **297 baris** dan **14 kolom.**

https://www.kaggle.com/cherngs/heart-disease-cleveland-uci

VARIABLE	DESCRIPTION
age	
sex	Jenis kelamin
cp	Rasa sakit pada dada
trestbps	Tekanan darah saat istirahat
chol	Kolesterol dalam mg/dl
fbs	Gula darah setelah puasa > 120 mg/dl
restecg	Pemeriksaan EKG saat pasien dalam posisi berbaring
thalach	Maksimum detak jantung per menit
exang	Exercise induced angina
oldpeak	Penurunan ST depresi saat pasien beristirahat yang diakibatkan karena berolahraga.
slope	Exercise ST segment
thal	Talasemia (kelainan darah)
condition	Memiliki penyakit jantung. (0 = no disease, 1 = disease)

Data Preparation

Data preparation dibagi ke dalam beberapa cara, yaitu:

1 Missing value

```
In [11]:
1 # mencari missing value
2 heart_disease.isnull().sum()
```

```
Out[11]: age          0
         sex          0
         cp           0
         trestbps     0
         chol         0
         fbs          0
         restecg      0
         thalach      0
         exang        0
         oldpeak      0
         slope        0
         ca           0
         thal         0
         condition    0
         dtype: int64
```

2 Duplikasi data

```
In [12]:
1 # mencari duplikasi data
2 heart_disease.duplicated().sum()
```

```
Out[12]: 0
```

Data Preparation

3 Perhitungan statistik

In [13]:
```
# melihat data dengan perhitungan statistik
heart_disease.describe().transpose()
```

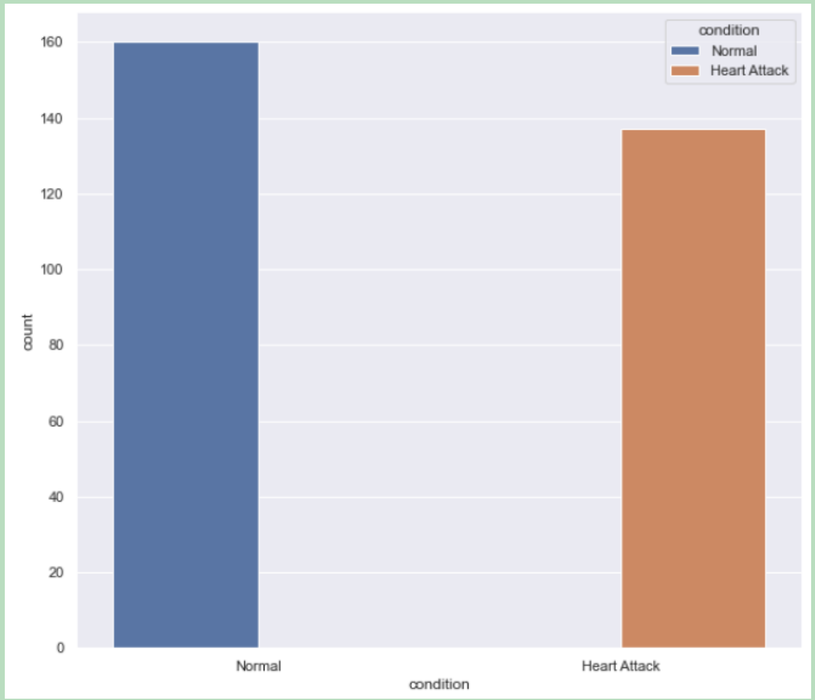
Out[13]:

	count	mean	std	min	25%	50%	75%	max
age	297.0	54.542088	9.049736	29.0	48.0	56.0	61.0	77.0
sex	297.0	0.676768	0.468500	0.0	0.0	1.0	1.0	1.0
cp	297.0	2.158249	0.964859	0.0	2.0	2.0	3.0	3.0
trestbps	297.0	131.693603	17.762806	94.0	120.0	130.0	140.0	200.0
chol	297.0	247.350168	51.997583	126.0	211.0	243.0	276.0	564.0
fbs	297.0	0.144781	0.352474	0.0	0.0	0.0	0.0	1.0
restecg	297.0	0.996633	0.994914	0.0	0.0	1.0	2.0	2.0
thalach	297.0	149.599327	22.941562	71.0	133.0	153.0	166.0	202.0
exang	297.0	0.326599	0.469761	0.0	0.0	0.0	1.0	1.0
oldpeak	297.0	1.055556	1.166123	0.0	0.0	0.8	1.6	6.2
slope	297.0	0.602694	0.618187	0.0	0.0	1.0	1.0	2.0
ca	297.0	0.676768	0.938965	0.0	0.0	0.0	1.0	3.0
thal	297.0	0.835017	0.956690	0.0	0.0	0.0	2.0	2.0
condition	297.0	0.461279	0.499340	0.0	0.0	0.0	1.0	1.0

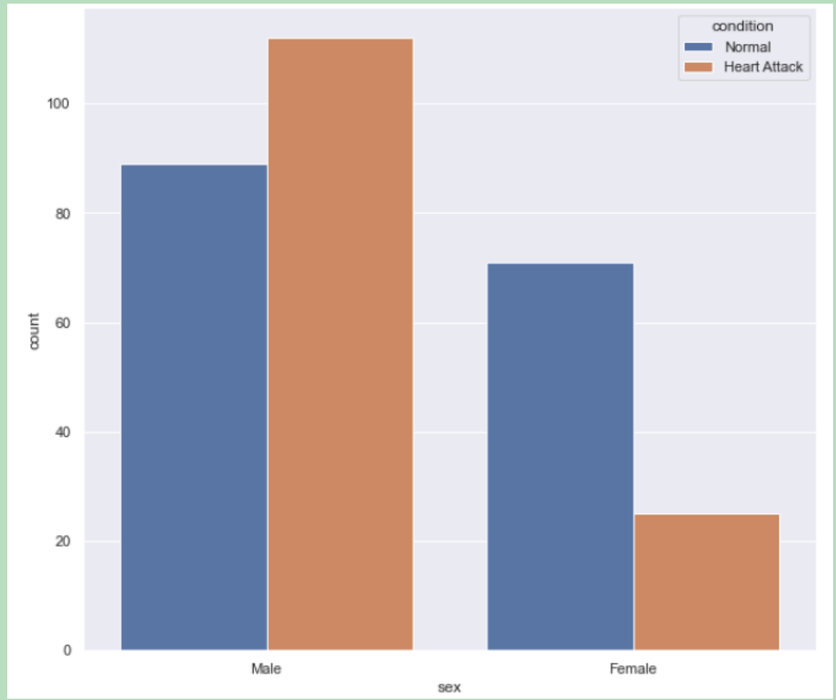
Data Preparation

4 Ekplorasi data (EDA)

VARIABLE CONDITION



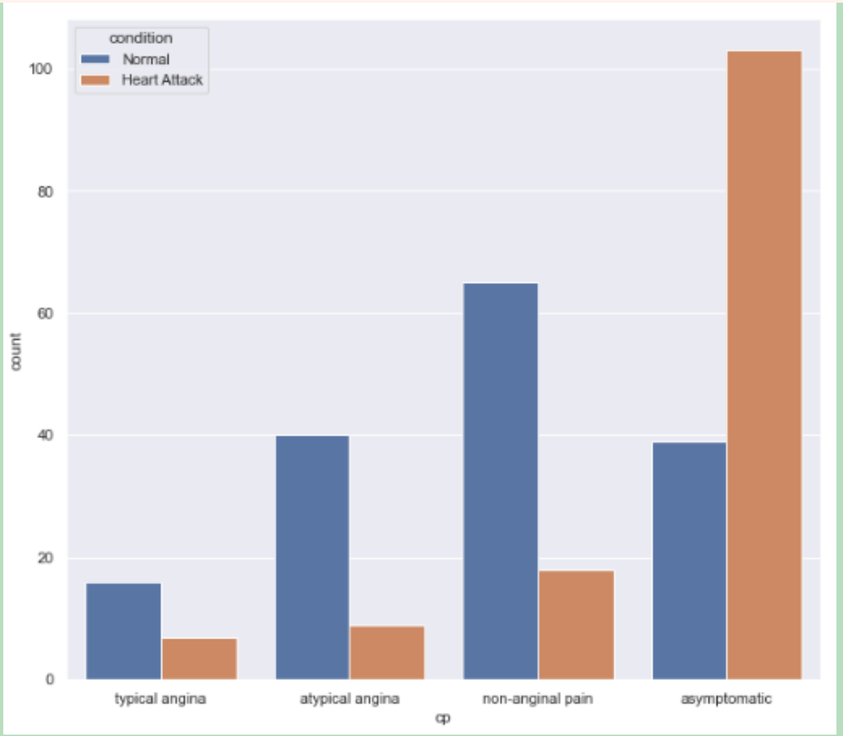
VARIABLE SEX



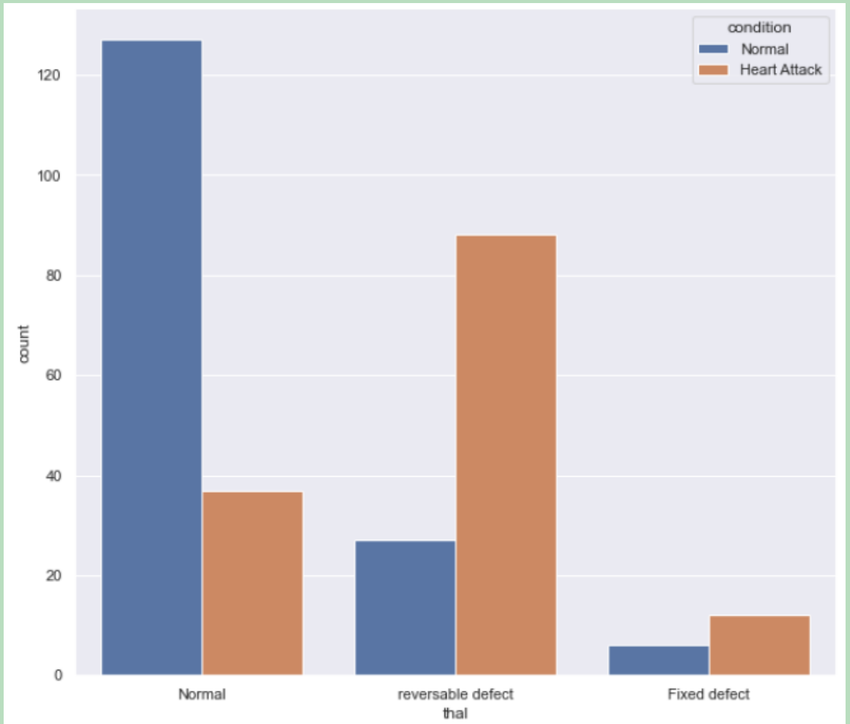
Data Preparation

4 Ekplorasi data (EDA)

VARIABLE CP



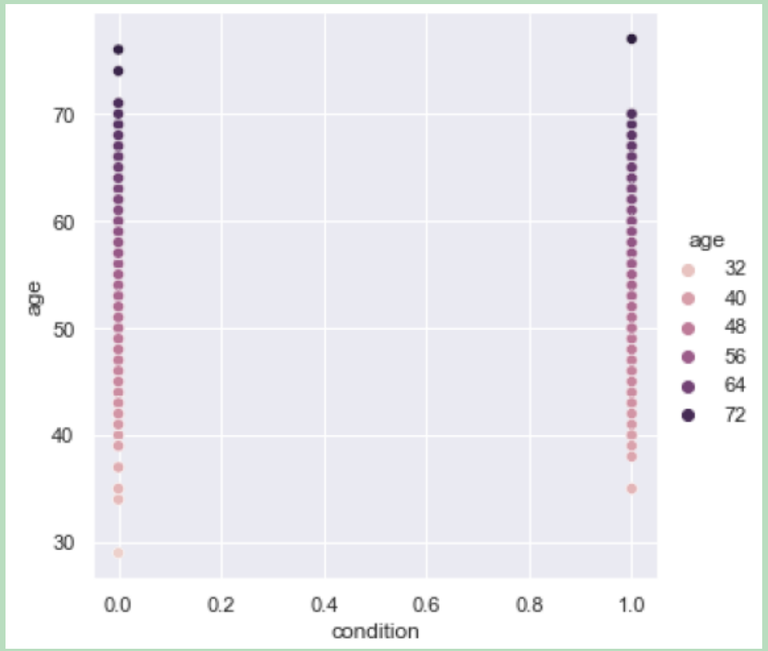
VARIABLE THAL



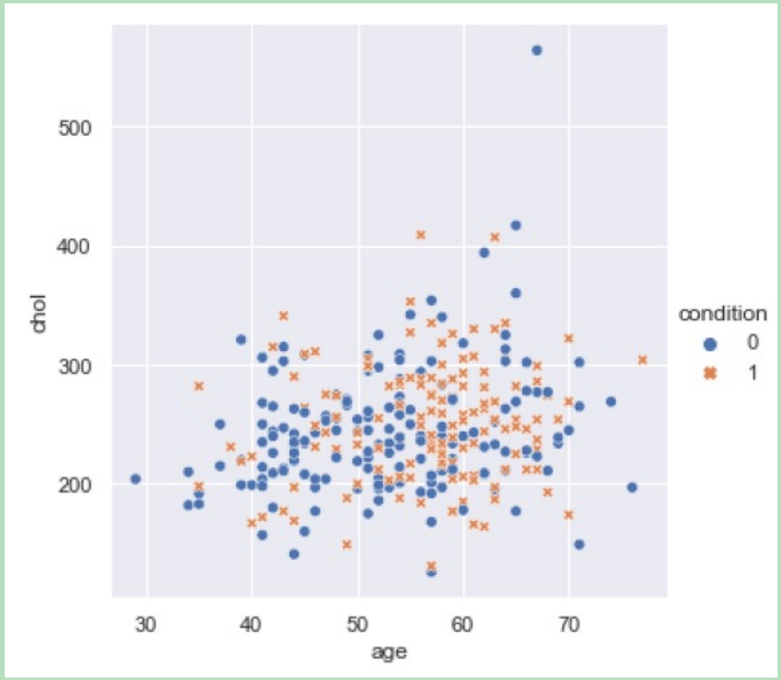
Data Preparation

4 Ekplorasi data (EDA)

VARIABLE CONDITION & AGE



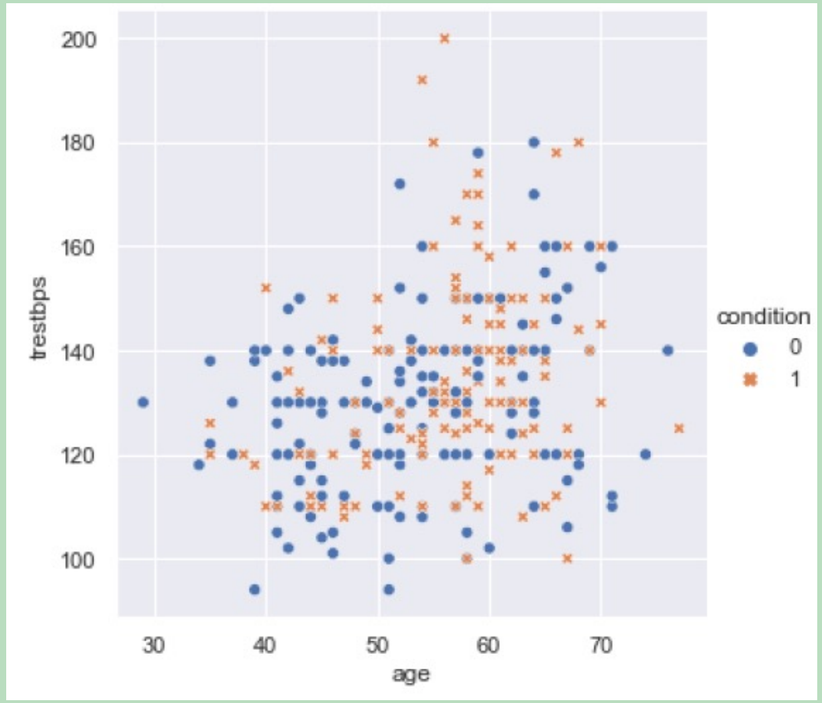
VARIABLE AGE & CHOL



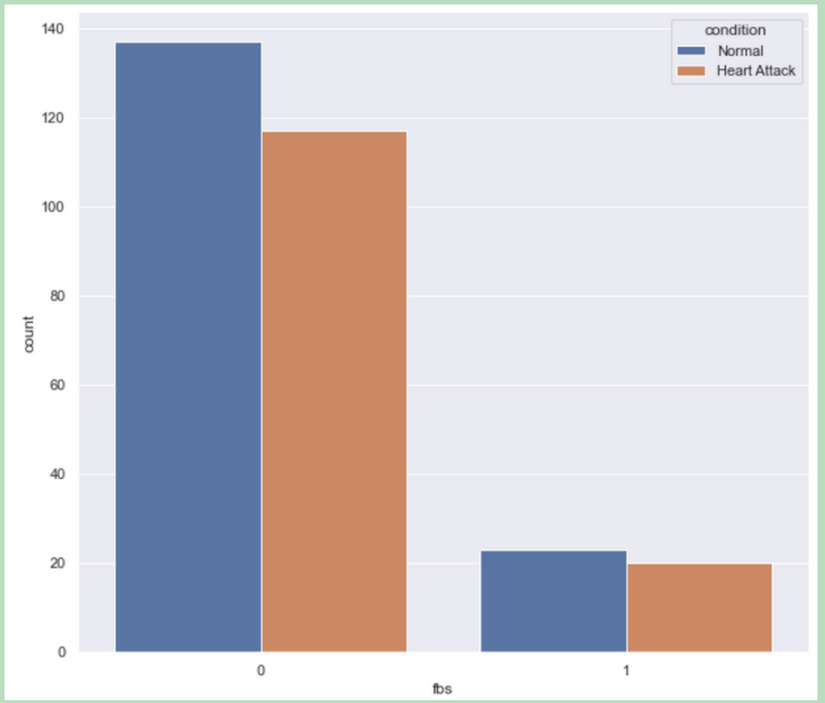
Data Preparation

4 Ekplorasi data (EDA)

VARIABLE AGE & TRESTBPS



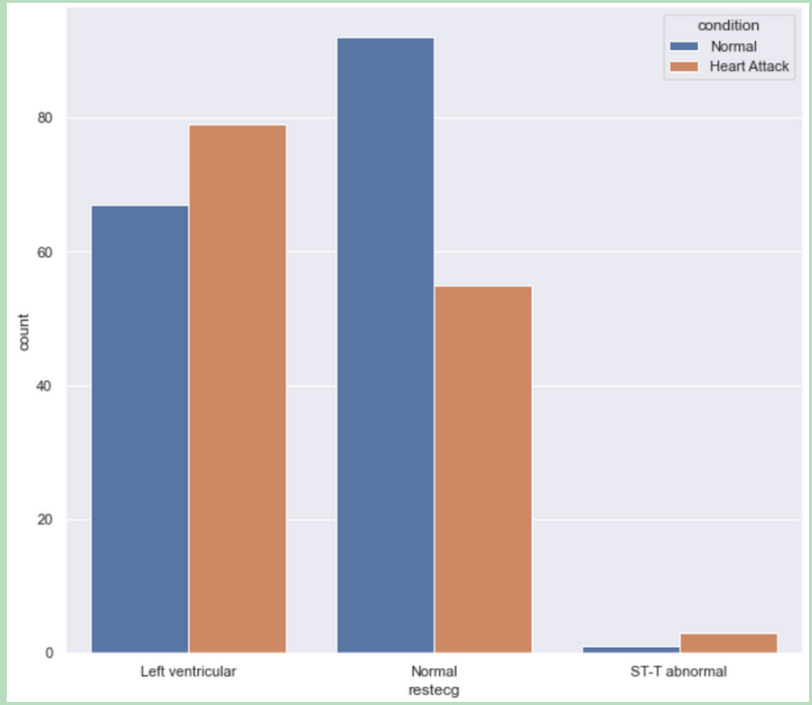
VARIABLE FBS



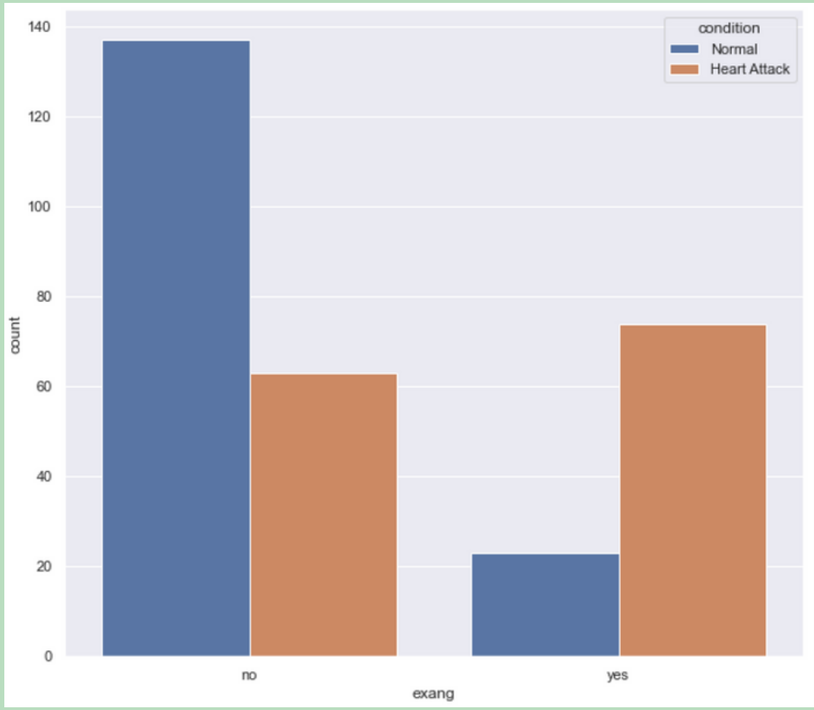
Data Preparation

4 Ekplorasi data (EDA)

VARIABLE RESTECG



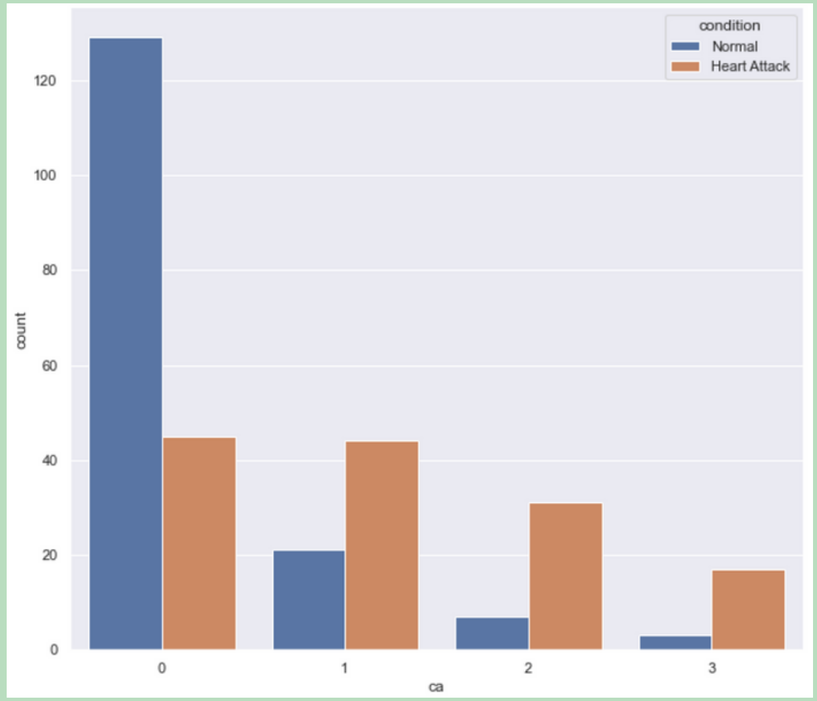
VARIABLE EXANG



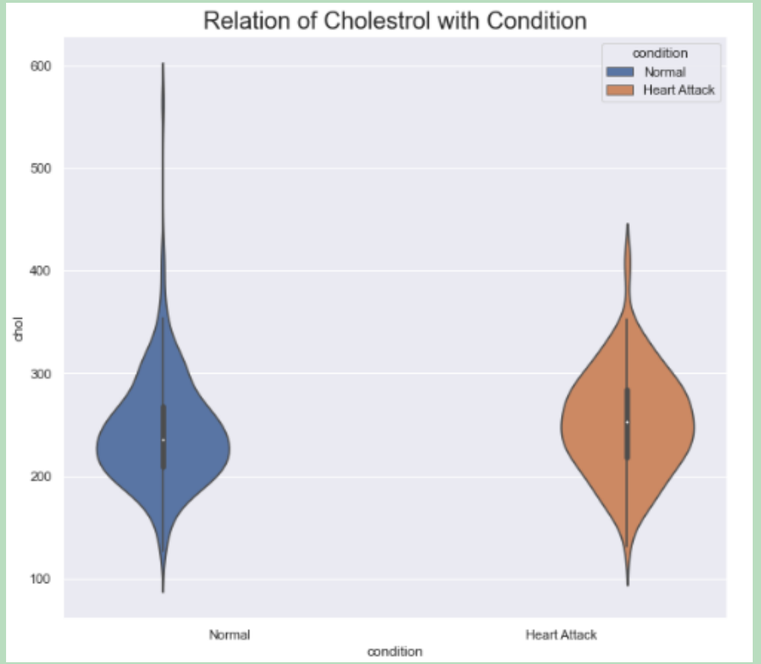
Data Preparation

4 Ekplorasi data (EDA)

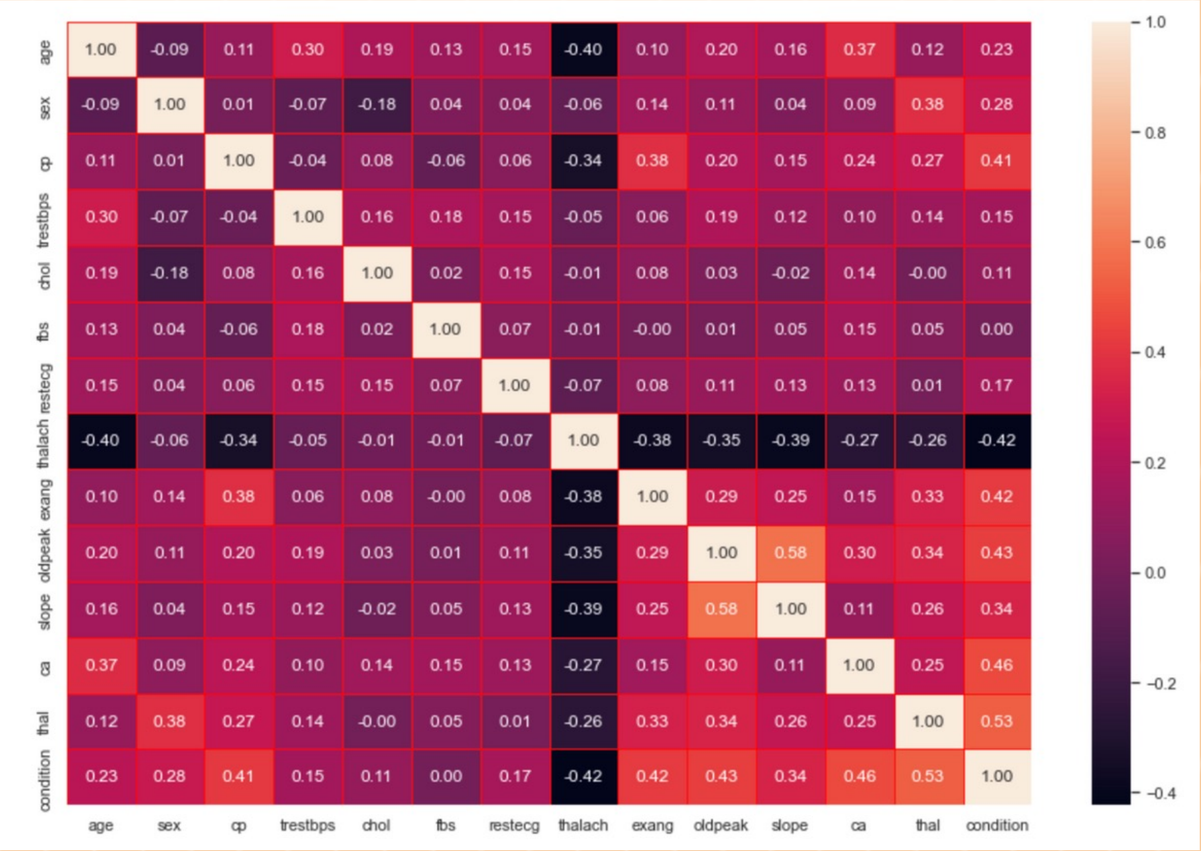
VARIABLE CA



CONNECTION BETWEEN CHOLESTROL AND PATIENT'S CONDITION



5 Visualisasi korelasi antar variabel



Variabel condition menjadi variabel independen dan variabel age, fbs, dan oldpeak menjadi variabel dependen

In [38]:
```
# Memilih variabel independen (target) dan dependen
Y= heart_disease['condition'] #independen
X=heart_disease.drop(['condition','age','fbs','oldpeak'],axis=1) #dependen

X.shape
```

Out[38]: (296, 10)

6 Menentukan variabel independen dan dependen

Modelling & Evaluation

Menggunakan algoritma **Logistic Regression** dan **Decision Tree**

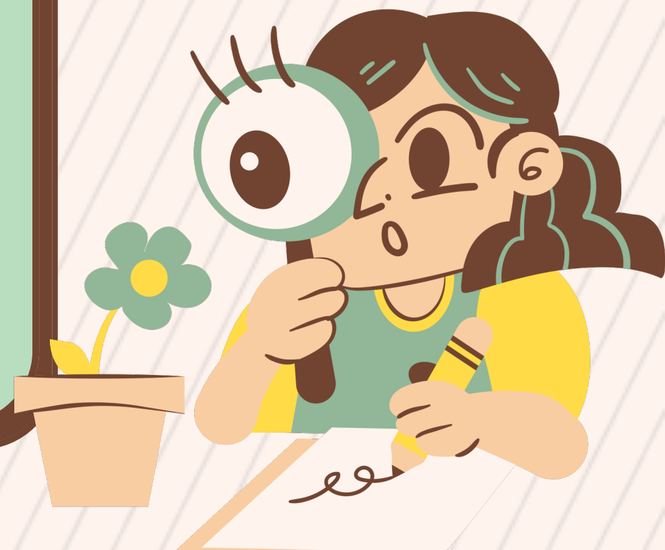
Membagi data ke dalam **training set** dan **testing set**

In [40]:

```
# Split data
# 80% training and 20% testing.
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2,
                                                    random_state = 0)

print("Shape of x_train :", x_train.shape)
print("Shape of x_test :", x_test.shape)
print("Shape of y_train :", y_train.shape)
print("Shape of y_test :", y_test.shape)
```

```
Shape of x_train : (236, 10)
Shape of x_test : (60, 10)
Shape of y_train : (236,)
Shape of y_test : (60,)
```



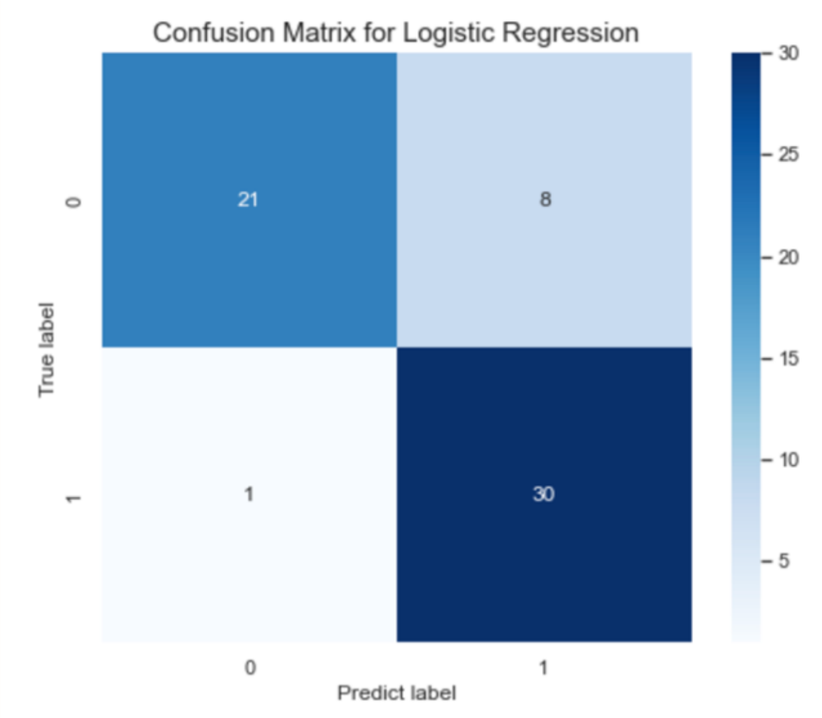
Logistic Reggresion

LOGISTIC REGGRESION

```
Accuracy of Logistic Regression on training set: 0.848
Accuracy of Logistic Regression on testing set: 0.850


How often the classifier model correct?
Accuracy:  0.85
```

Akurasi yang didapatkan untuk model Logistic Reggresion adalah 0.85




```
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.72	0.95	0.82	22
1	0.97	0.79	0.87	38
accuracy			0.85	60
macro avg	0.85	0.87	0.85	60
weighted avg	0.88	0.85	0.85	60

Confusion Matrix Logistic Reggresion

True Positive: 21
False Positive: 8
True Negative: 30
False Negative: 1

Precision (how many are correctly classified among that class)
Recall (how many of this class you find over the whole number of element of this class)

Nilai akurasi pada algoritma **Logistic Reggresion** lebih **rendah** ketika diuji dengan menggunakan metode cross validation, yaitu 0.83423

Cross Validation

Cross Validation

```
CROSS_final = cross_val_score(lr, x_train, y_train, cv=10).mean()
CROSS_final
```

```
0.8342391304347826
```

Perbandingan Algoritma Logistic Regression

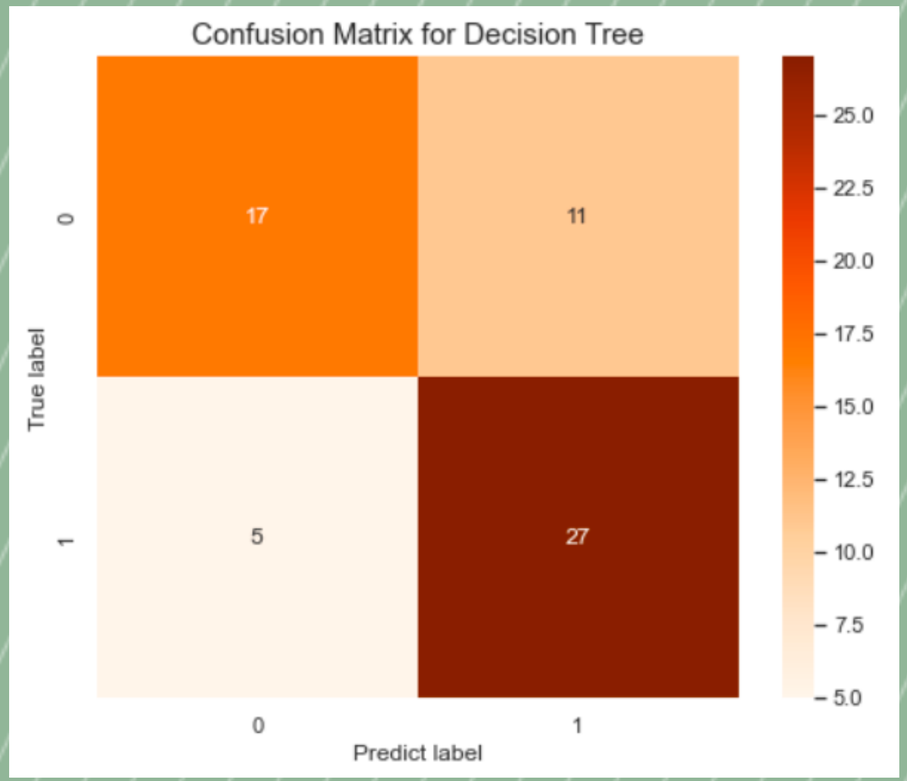
- Sebelum Cross-Validation -- Akurasi = 0.85
- Sesudah Cross-Validation -- Akurasi = 0.8342391304347826

Decision Tree

```
Accuracy of Decision Tree on training set: 1.000
Accuracy of Decision Tree on testing set: 0.733


How often the classifier model correct?
Accuracy:  0.7333333333333333
```

Akurasi untuk model Decision Tree adalah 0.7333



Confusion Matrix Decision Tree

```
In [49]: 1 print(classification_report(y_test, y_pred))
```

Classification Report

	precision	recall	f1-score	support
0	0.61	0.80	0.69	25
1	0.81	0.63	0.71	35
accuracy			0.70	60
macro avg	0.71	0.71	0.70	60
weighted avg	0.73	0.70	0.70	60

Nilai akurasi pada algoritma **Decision Tree** lebih **rendah** ketika diuji dengan menggunakan metode cross validation, yaitu 0.72047

Cross Validation

Cross Validation

```
CROSS_final = cross_val_score(dt, x_train, y_train, cv=10).mean()
CROSS_final
```

```
0.7204710144927535
```

Perbandingan Algoritma Decision Tree

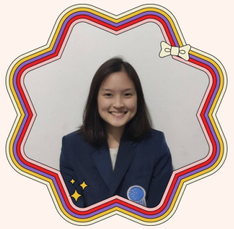
- Sebelum Cross-Validation -- Akurasi = 0.7333333333333333
- Sesudah Cross-Validation -- Akurasi = 0.7204710144927535

Perbandingan 2 algoritma

	Model	Training Accuracy %	Testing Accuracy %
0	Logistic Regression	84.810127	85.000000
1	Decision Tree	100.000000	73.333333

Logistic Regression memiliki performa yang lebih baik untuk dataset heart dilihat dari nilai akurasi testing yang lebih tinggi.
Kemudian untuk nilai true positive pada Algoritma Logistic Regression juga lebih banyak yaitu sebesar 27 dibandingkan dengan DT

Thank You



FOR YOUR ATTENTION