Machine Learning

M Octaviano Pratama, S.Kom., M.Kom

Chief Scientist BISA AI, Co-Founder BISA AI

Contact: info@bisa.ai , https://bisa.ai

BISA AI

We Empower **Artificial Intelligence and its Related Fields** to Everyone through Academy, Webinar, and Workshop

https://bisa.ai

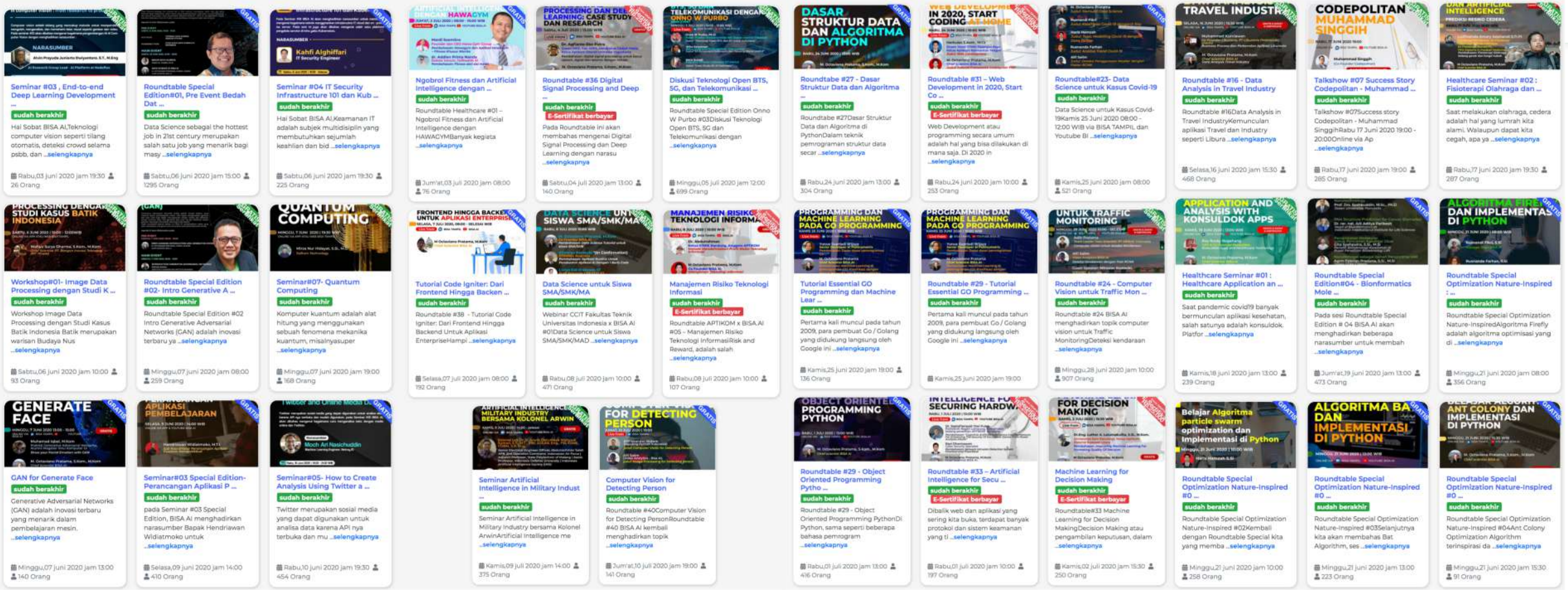
https://Instagram.com/bisa.ai

https://youtube.com/bisaai

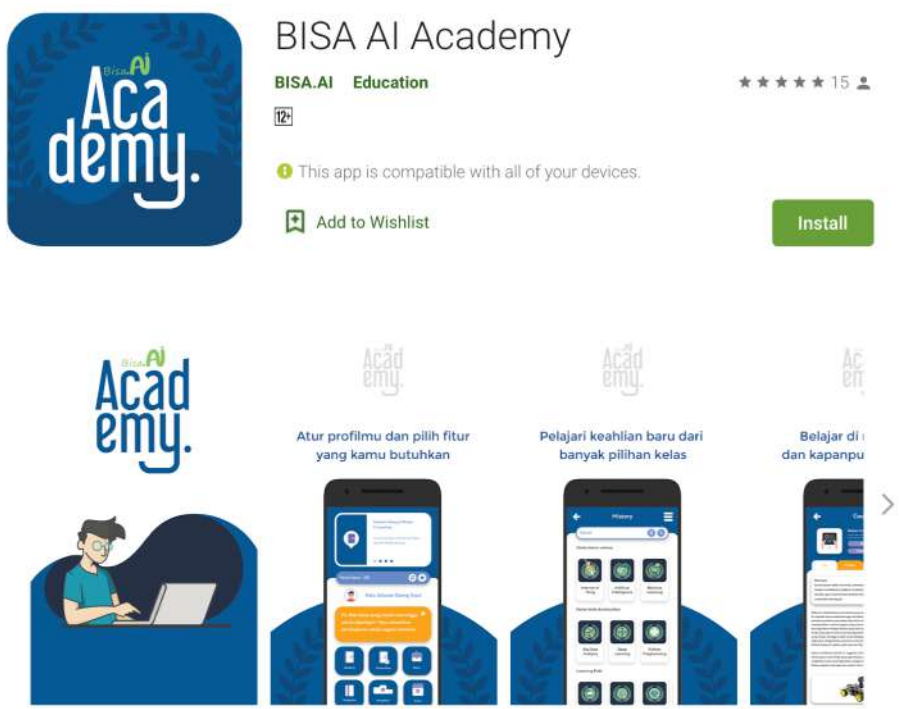
BISA TAMPIL by BISA AI

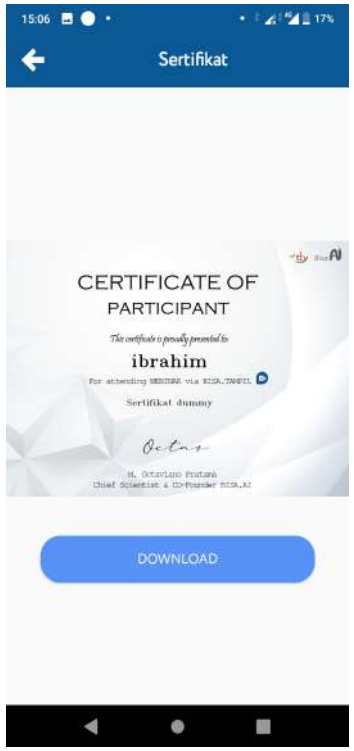
Video Conference and Webinar Community for Everyone. Starting from May 2020, **we organize +400 Artificial Intelligence and its related fields webinar**

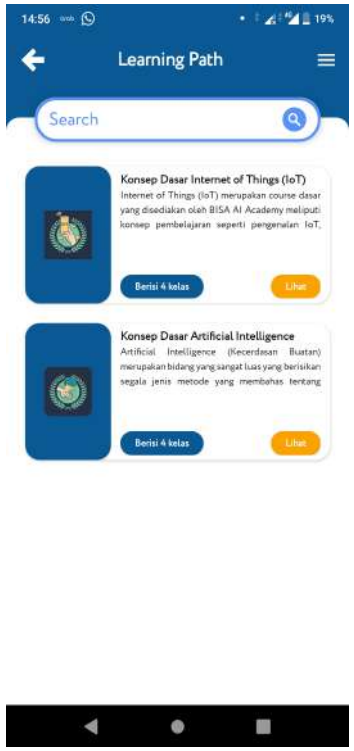
+400 AI Webinar using BISA Tampil (May 2020 – Now)



BISA AI Academy





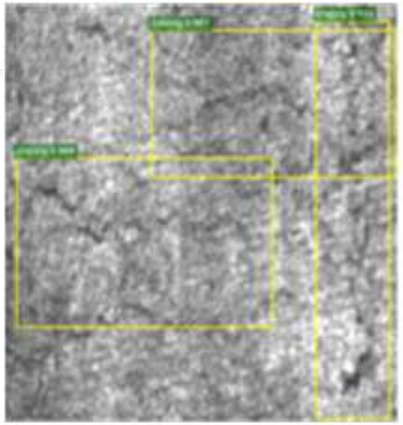


Outline

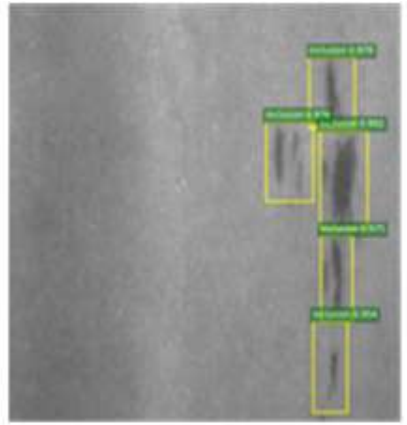
- **Trend Machine Learning Industri**
- Pendekatan Machine Learning
- Signal Processing dengan Machine Learning

Sensor and IoT Industry: Surface Defect Detection

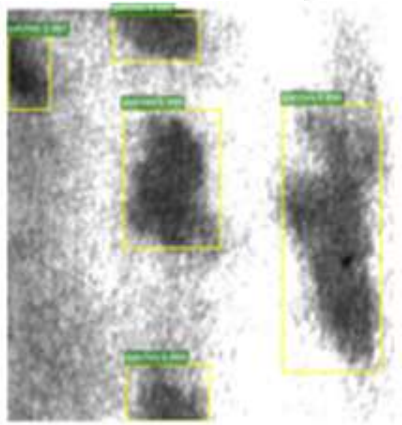
six kinds of typical surface defects of the hot-rolled steel strip are collected, i.e., rolled-in scale (RS), patches (Pa), crazing (Cr), pitted surface (PS), inclusion (In) and scratches (Sc). The database includes 1,800 grayscale images: 300 samples each of six different kinds of typical surface defects.



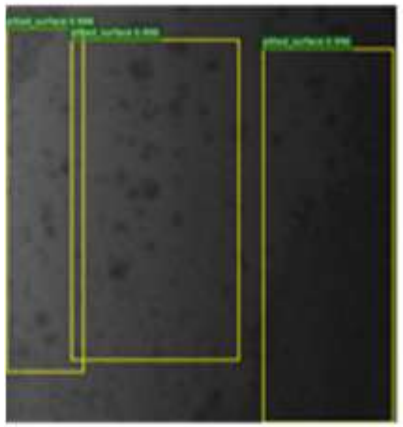
(a) crazing



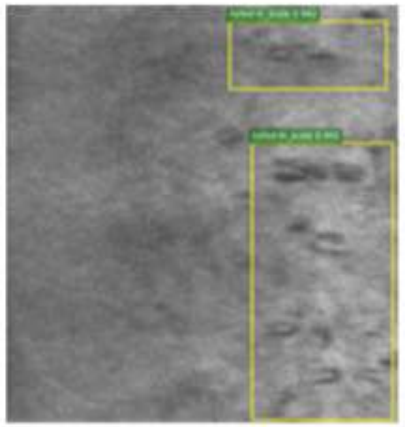
(b) inclusion



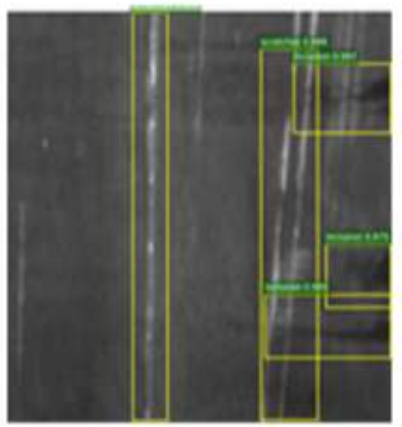
(c) patches



(d) pitted surface



(e) rolled-in scale



(f) scratches

Sensor and IoT Industry: Precision Agriculture

INTRODUCTION

- Population of Paddy should be catered in order to keep sustainability of primary food for human
- By knowing harvest time, stakeholder like farmers even governments can prepare when plant should be catered, watered, and harvested that forwarded into yields distribution right on target and time
- **Before knowing harvest or area estimation of paddy, the first tasks to do is to know paddy growth level**
- there are three phases of paddy development: (1) vegetative, (2) reproductive, and (3) maturation

Sensor and IoT Industry: Precision Agriculture

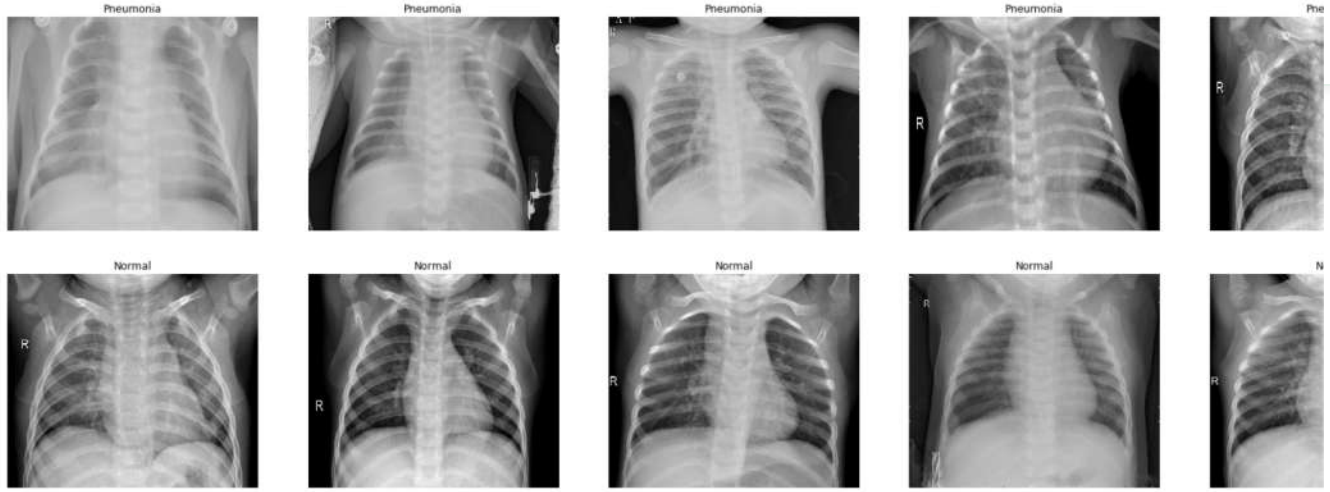
INTRODUCTION

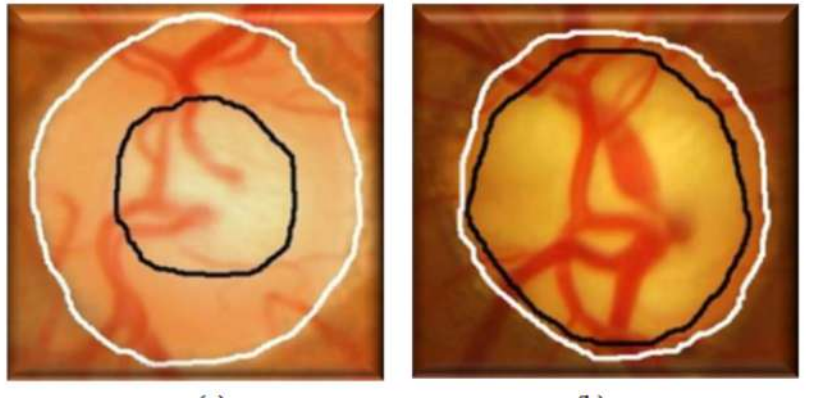
- Monitoring and Prediction of paddy growth can be taken by Technology like **Precision Agriculture** in order to enhance farming techniques
- There are several Precision Agriculture classification techniques that can be used like **Machine Learning or Deep Learning**
- Previous research by Sadiyah et al [3] produce 99.58% overall accuracy of plant image dataset using Convolutional Neural Networks, meanwhile other Sadiyah et al paper [4] produce 99.8% overall accuracy using the same model, meanwhile Mulyono et al [5] produce 50% overall accuracy using Support Vector Machine to paddy growth classification task
- Paddy should be monitored well like time to harvest, time to watering and Growth Level of Paddy

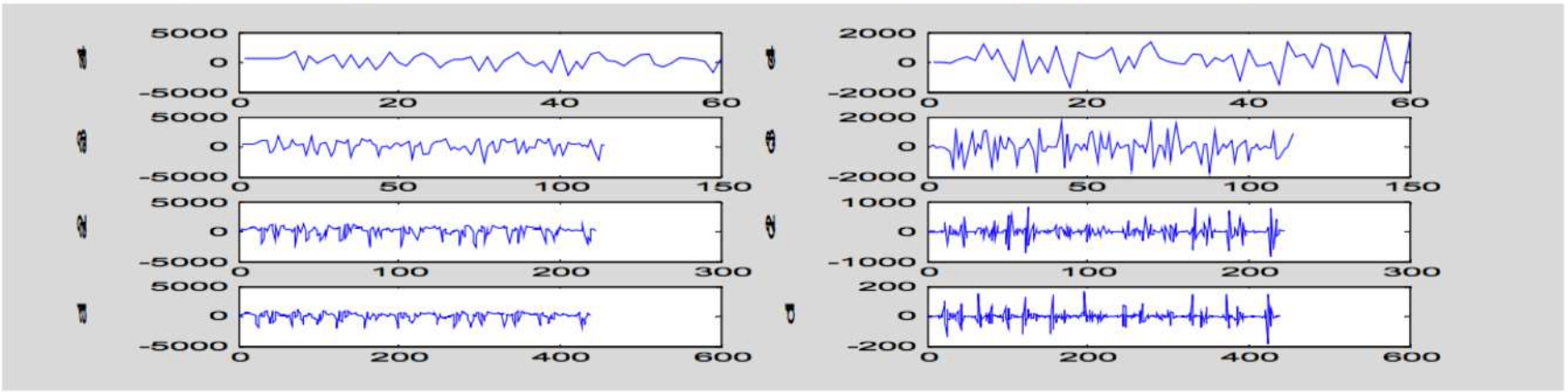
Sensor and IoT Industry: Precision Agriculture

- 300 paddy images classified into 3 labels: vegetative, reproductive, and maturation
- We used 2 annotators
- Kappa score is used to measure the agreement between two annotators.

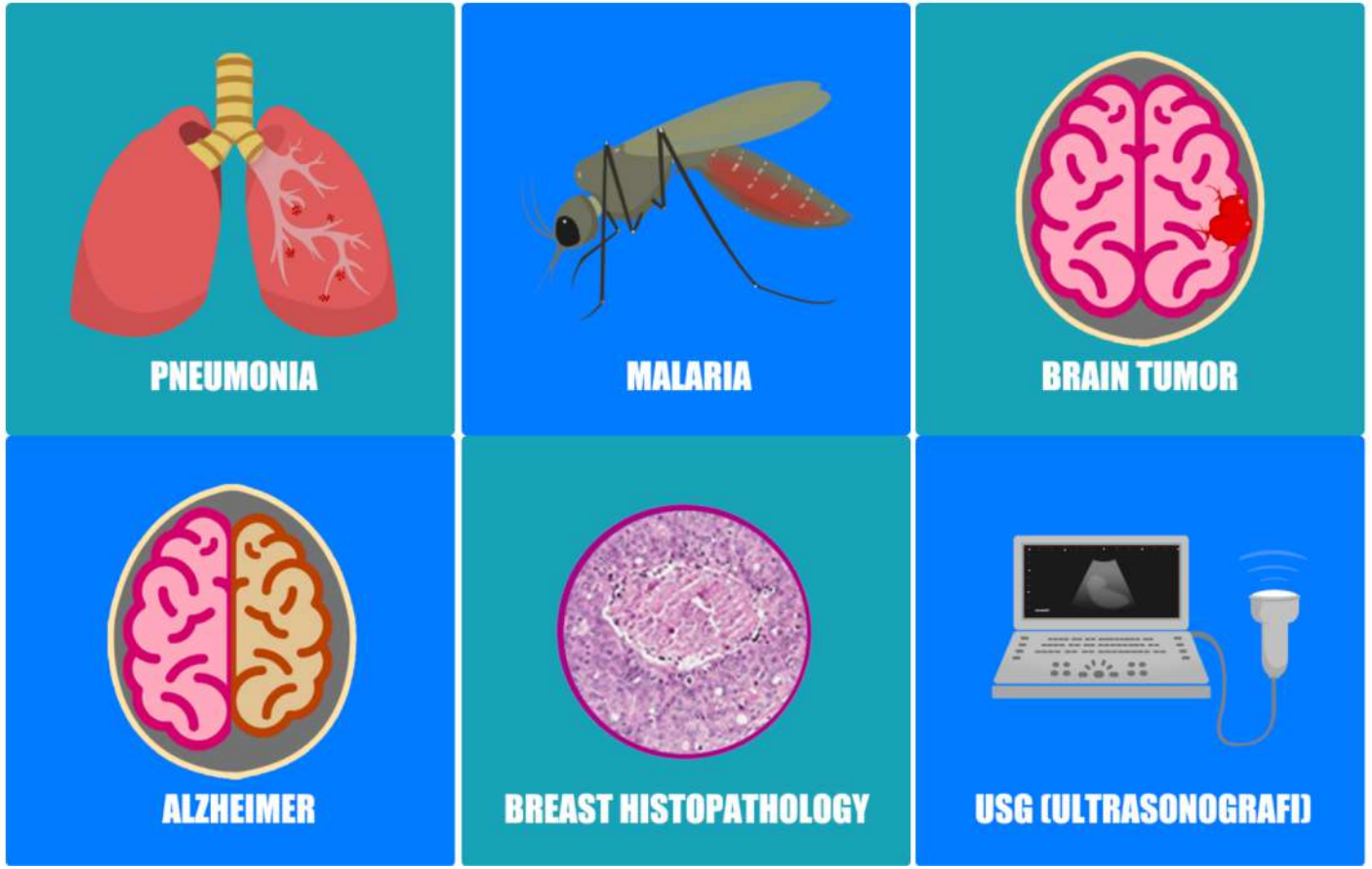
Healthcare Industry







Healthcare Industry: Tellhealth.ai



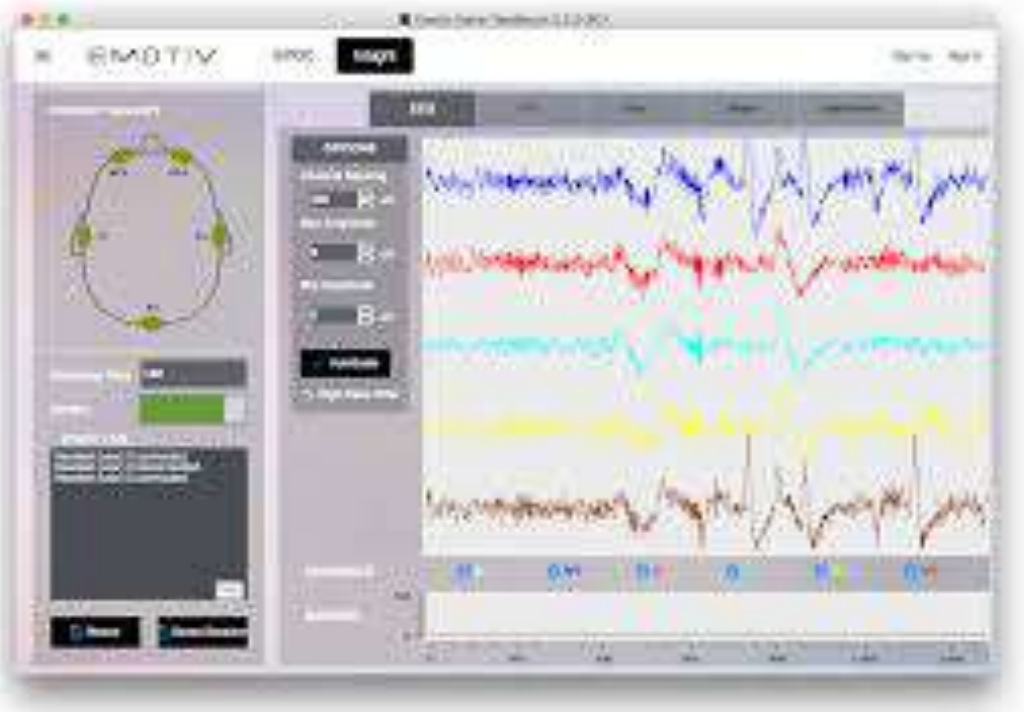
Healthcare Industry: Prediksi Usia kehamilan

USG

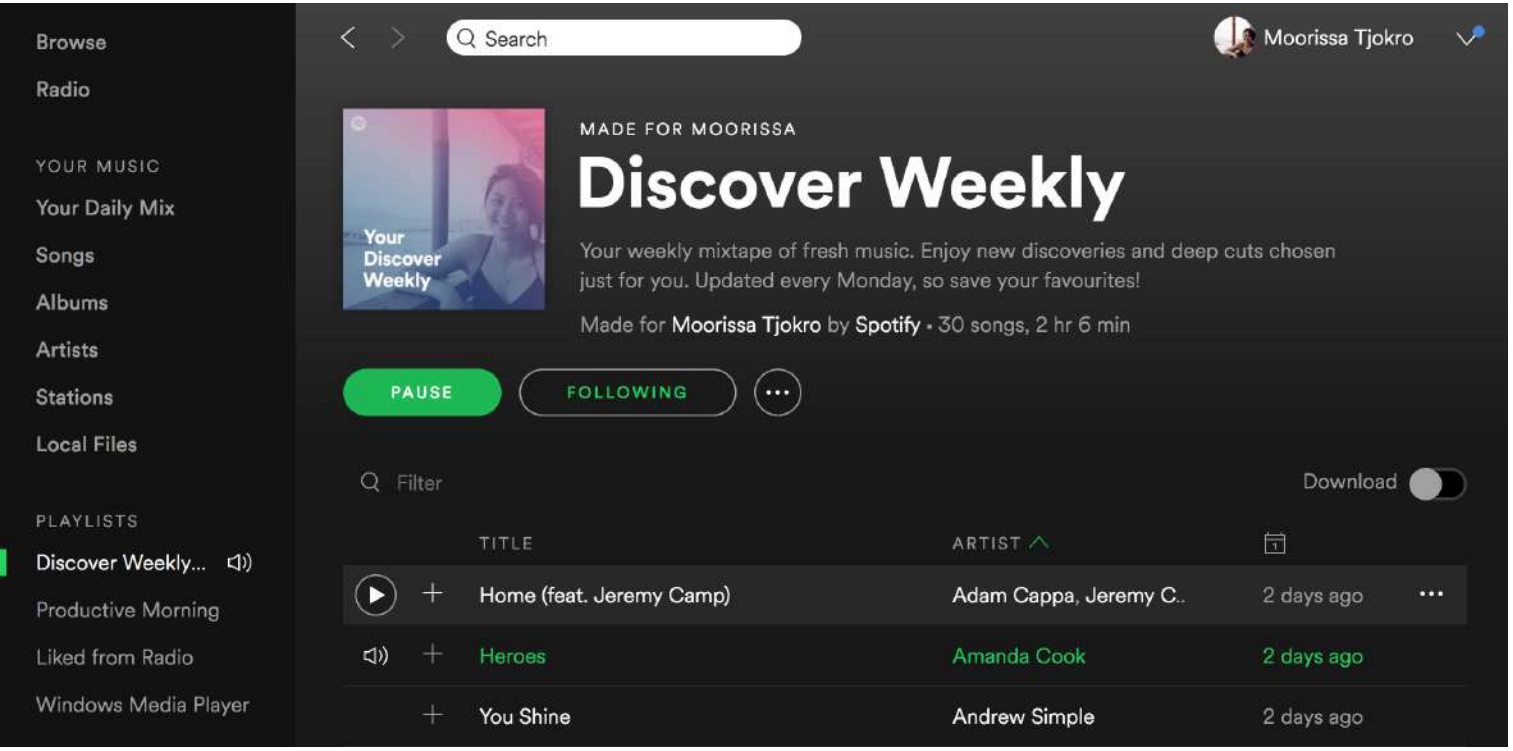
GA (weeks)	*BPD*	*HC*	*AC*	*FL*
12	12	12	0	13
13	11	12	0	12
14	18	18	17	18
15	26	26	26	26
16	34	34	34	34
17	27	27	27	27
18	29	29	29	29
19	28	28	28	28
20	28	28	28	28
21	28	28	28	28
22	27	27	26	27
23	26	26	26	26
24	29	29	29	29
25	32	32	30	32
26	27	27	26	27
27	26	26	26	25
28	23	23	23	23
29	26	26	26	26
30	23	23	23	23
31	23	23	23	23
32	25	25	25	25
33	23	23	23	23
34	24	24	25	25
35	25	25	26	26
36	20	20	20	20
37	22	22	22	22
38	25	25	25	25
39	24	24	24	24
40	14	14	14	14
Total	705	706	679	708

Signal Processing Industry

Emotive EEG 16 channel



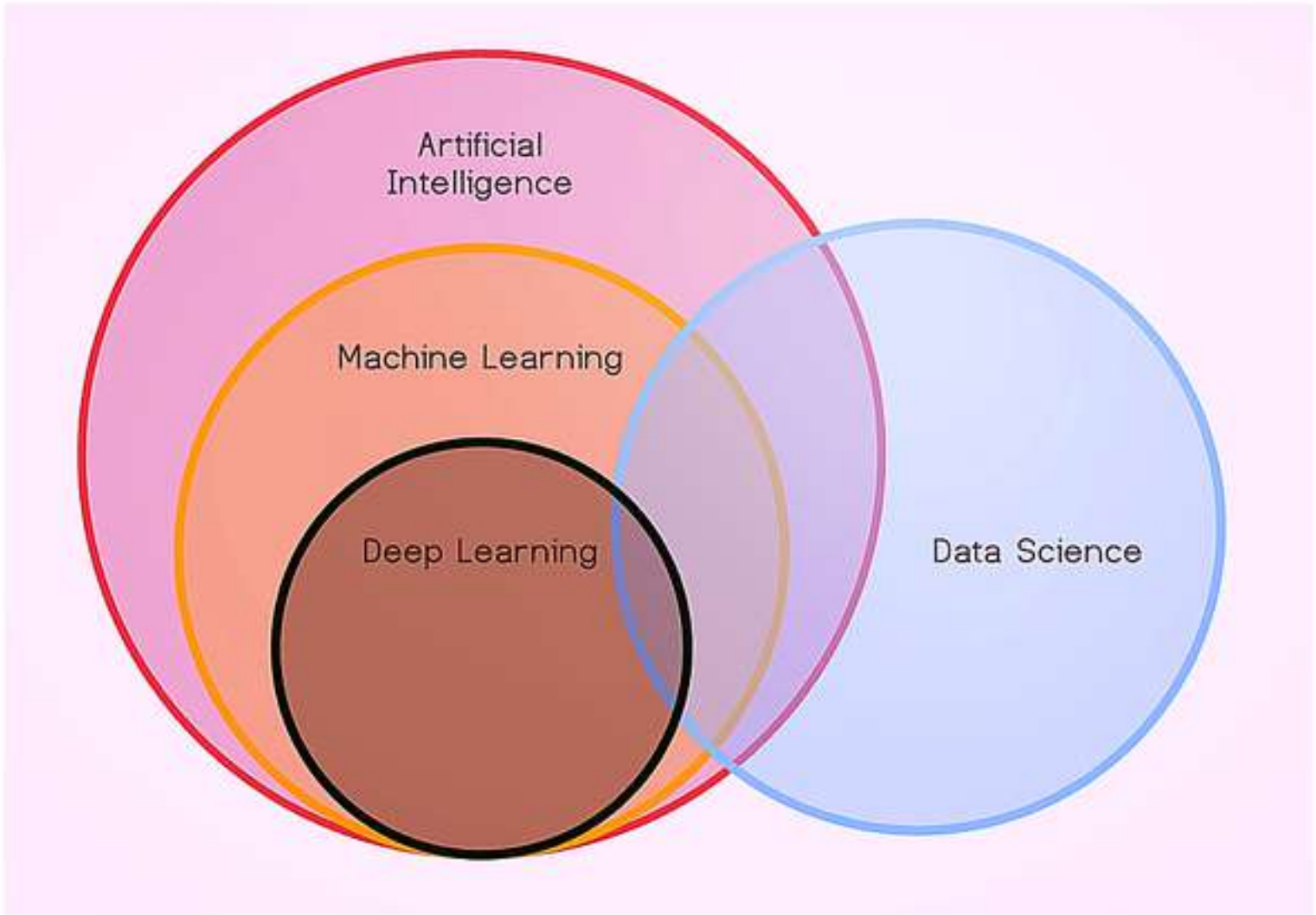
Signal Processing Industry



Outline

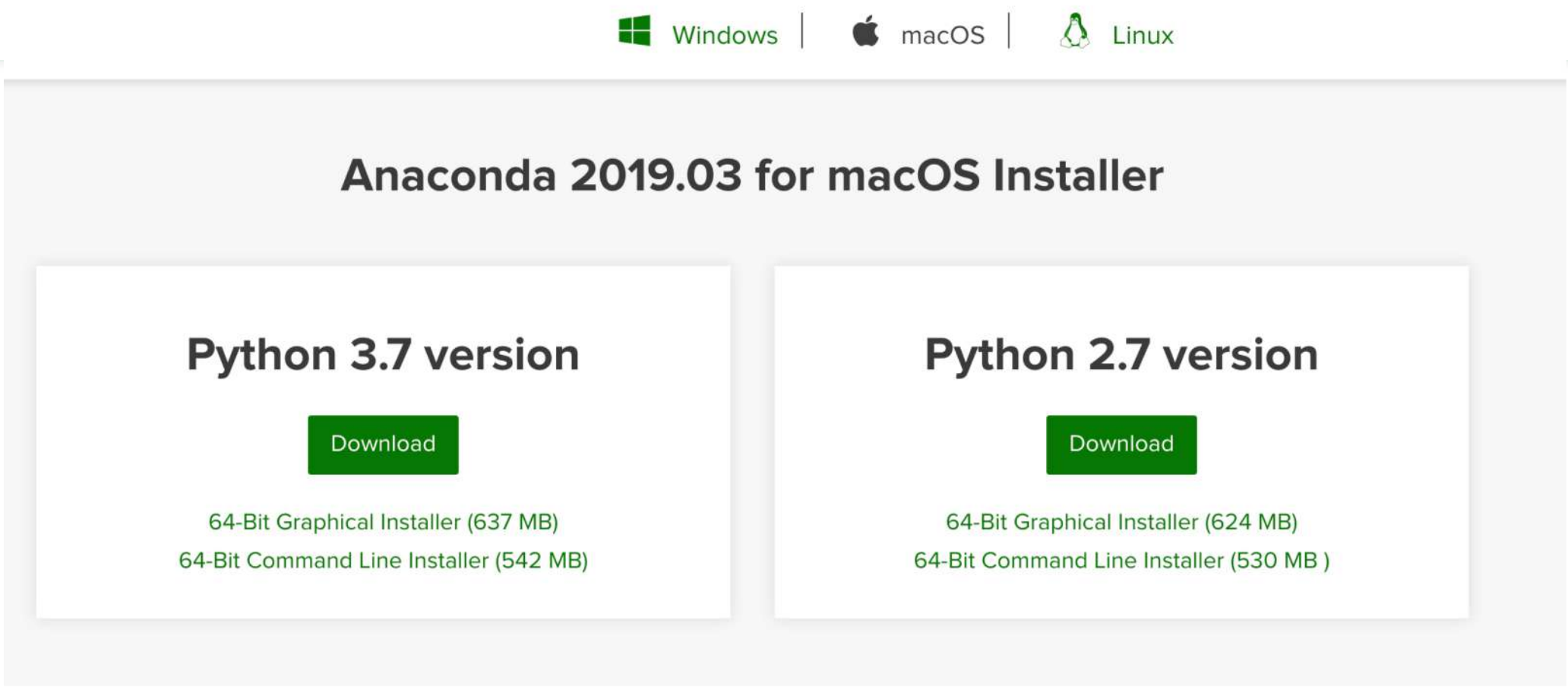
- Trend Machine Learning Industri
- **Pendekatan Machine Learning**
- Signal Processing dengan Machine Learning

Machine Learning



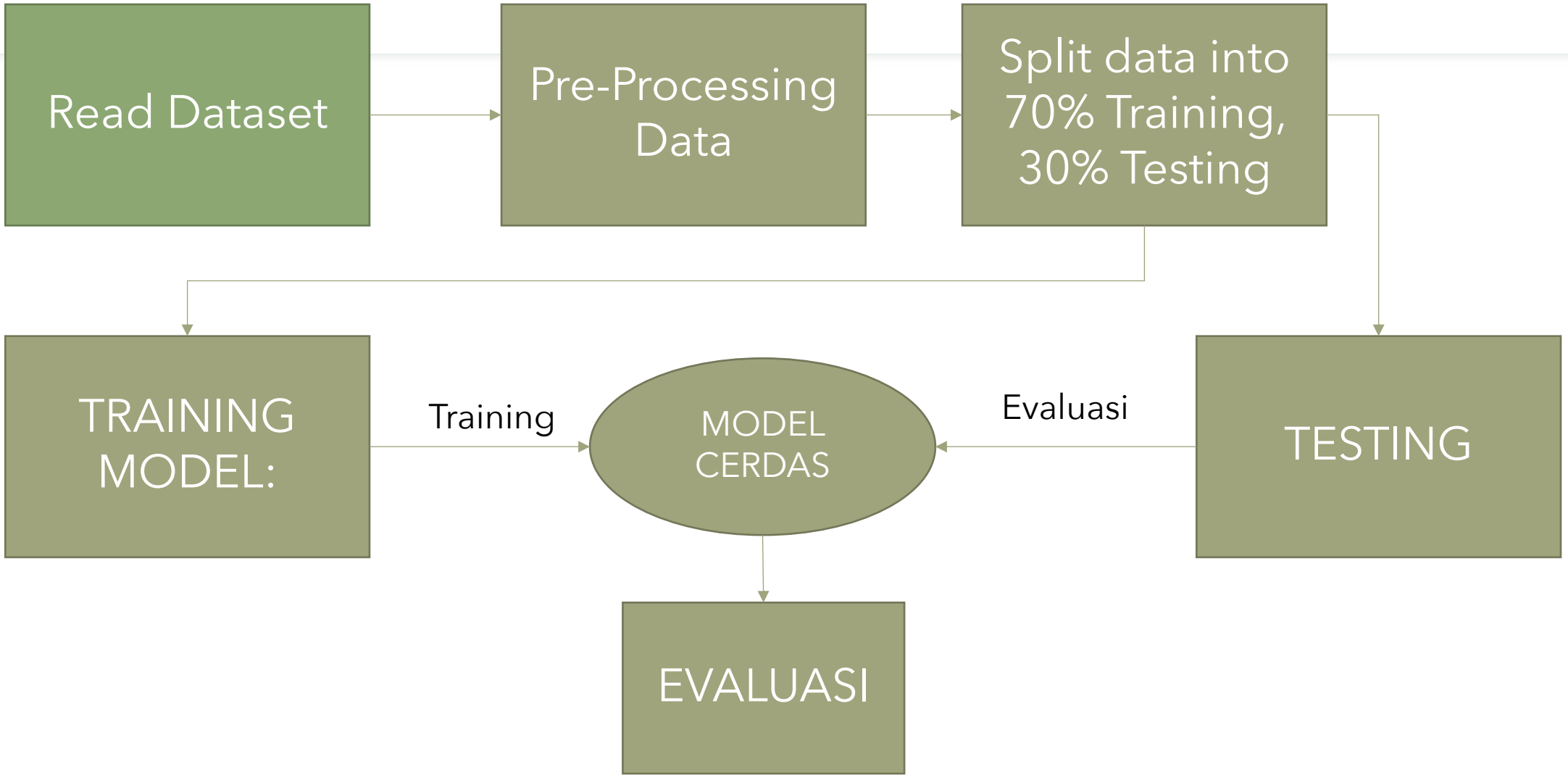
Programming

Python Install



https://www.anaconda.com/distribution/

Machine Learning: Klasifikasi



We Need Data!

- https://archive.ics.uci.edu/ml/index.php
- https://www.kaggle.com/datasets
- https://data.go.id/

- https://www.kaggle.com/ronitf/heart-disease-uci
- http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html

Exploratory Data Analysis

Exploratory Data Analysis (EDA) are very important steps in any analysis task.

get to know your data!

- distributions (symmetric, normal, skewed)
- data quality problems
- outliers
- correlations and inter-relationships
- subsets of interest
- suggest functional relationships

Summary Statistics

- mean: $\mu = \sum_i X_i / n$
- mode: most common value in X
- median: $\mathbf{X}$=sort(X), median = $\mathbf{X}_{n/2}$ (half below, half above)
- variance: $\sigma^2 = \sum_i (X_i - \mu)^2 / n$

	Unnamed: 0	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke
count	201.000000	201.000000	164.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000
mean	100.000000	0.840796	122.000000	98.797015	174.200995	65.889055	53.766667	2555.666667	126.875622	3.319154	3.256766
std	58.167861	1.254802	35.442168	6.066366	12.322175	2.101471	2.447822	517.296727	41.546834	0.280130	0.316049
min	0.000000	-2.000000	65.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000
25%	50.000000	0.000000	NaN	94.500000	166.800000	64.100000	52.000000	2169.000000	98.000000	3.150000	3.110000
50%	100.000000	1.000000	NaN	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	3.310000	3.290000
75%	150.000000	2.000000	NaN	102.400000	183.500000	66.600000	55.500000	2926.000000	141.000000	3.580000	3.410000
max	200.000000	3.000000	256.000000	120.900000	208.100000	72.000000	59.800000	4066.000000	326.000000	3.940000	4.170000

Data Normalization

- Simple Featur
- Min-Max
- Z-score

age	income
20	100000
30	20000
40	500000

Not-normalized

- "age" and "income" are in different range.
- hard to compare
- "income" will influence the result more

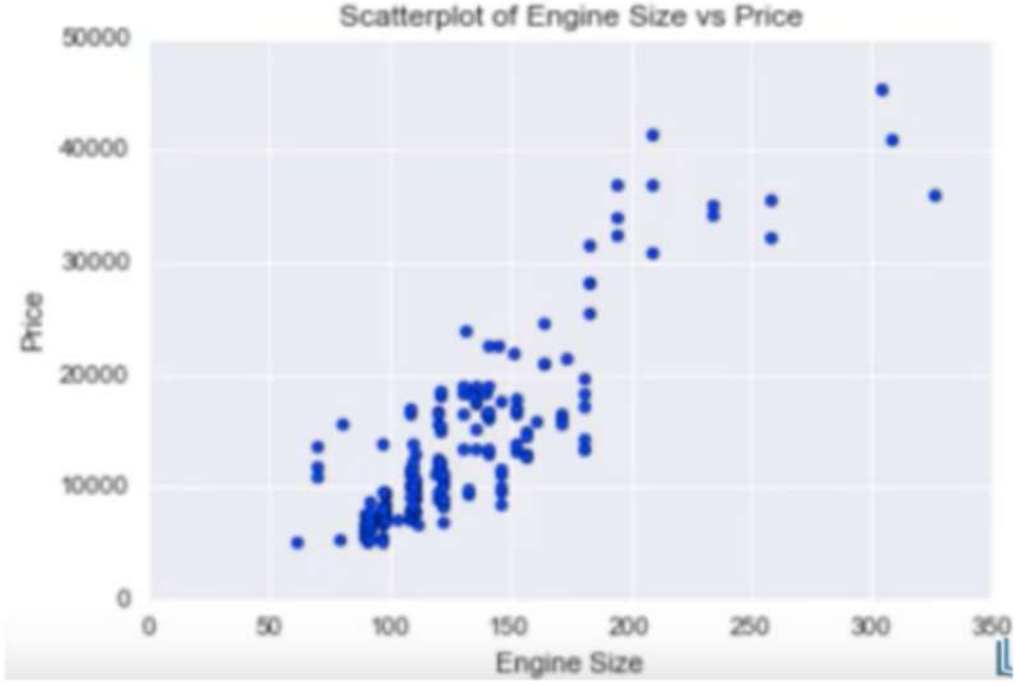
age	income
0.2	0.2
0.3	0.04
0.4	1

Normalized

- similar value range.
- similar intrinsic influence on analytical model.

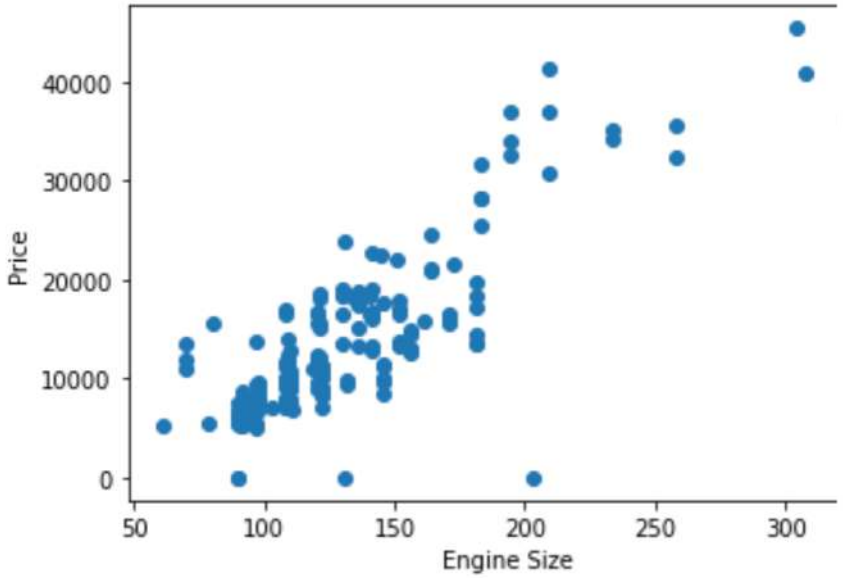
Visualisasi Hubungan Antar Variable Data

- This is giving us an initial indication that there is a positive linear relationship between these two variables.

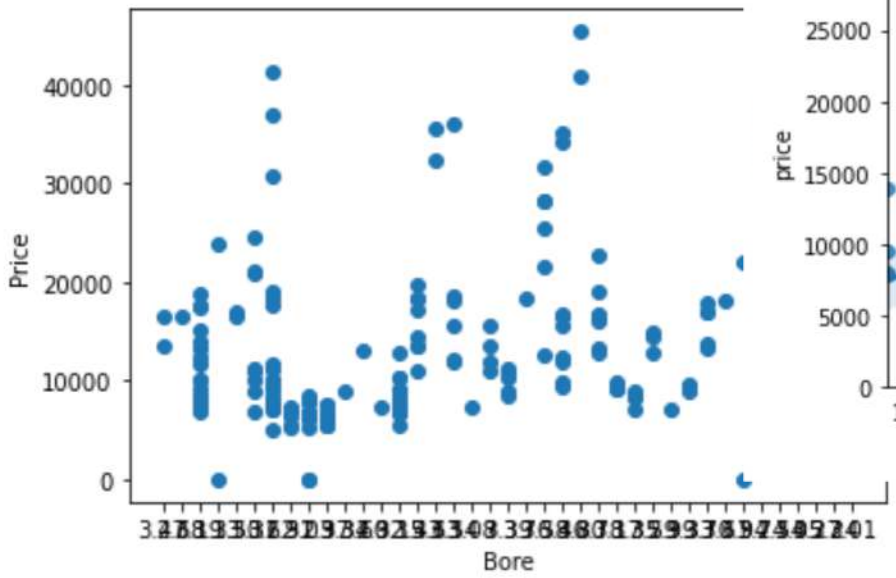


Scatter Plot

```
import matplotlib.pyplot as plt

plt.scatter(df['engine-size'], df['price'])
plt.xlabel("Engine Size")
plt.ylabel("Price")
plt.show()
```

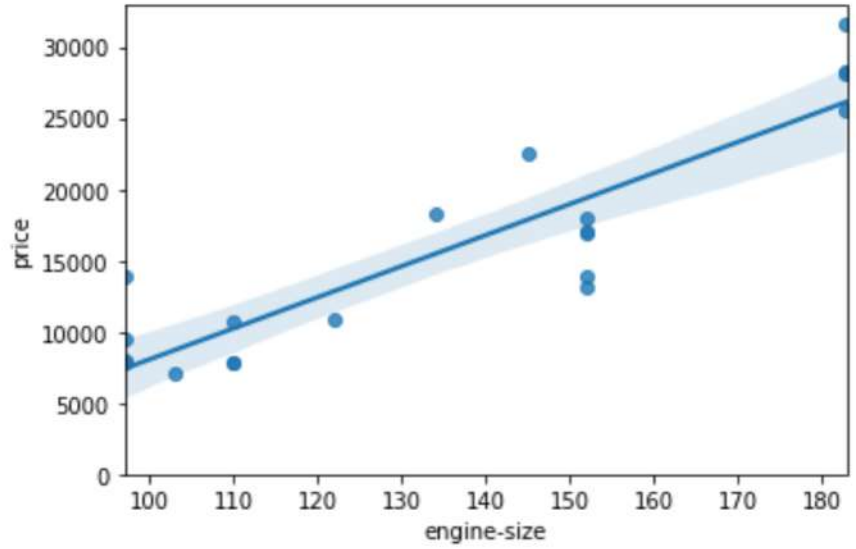



```
import matplotlib.pyplot as plt

plt.scatter(df['bore'], df['price'])
plt.xlabel("Bore")
plt.ylabel("Price")
plt.show()
```




```
sns.regplot(x='engine-size',y='price',data=df_new)
plt.ylim(0,)
```

```
(0.0, 32915.73541112133)
```



Pearson Correlation

```
import scipy

x, y = scipy.stats.pearsonr(df['engine-size'],df['price'])
print("pearson Coef: " + str(x))
print("p value: " + str(y))
```

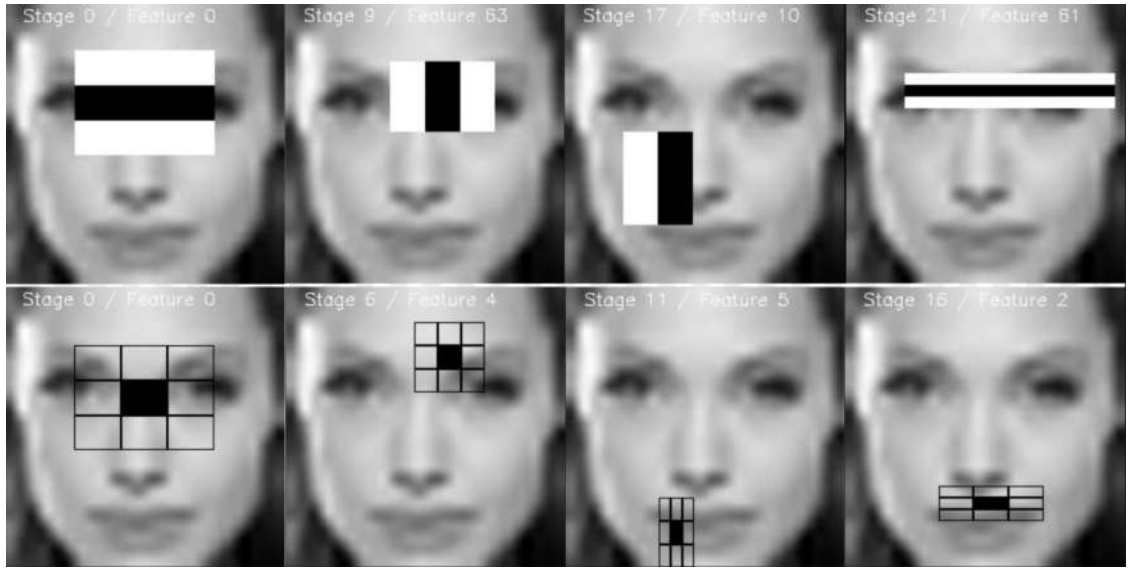
```
pearson Coef: 0.838097285838633
p value: 2.4898087727398237e-55
```

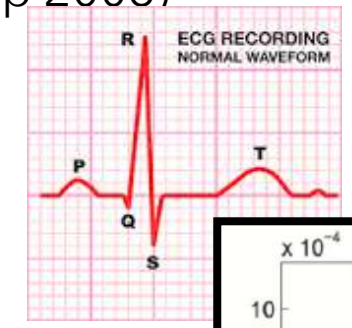
```
df['bore'] = df['bore'].replace(to_replace ="?",value ="0")
df['bore'] = df['bore'].astype("float")

x, y = scipy.stats.pearsonr(df['bore'],df['price'])
print("pearson Coef: " + str(x))
print("p value: " + str(y))
```

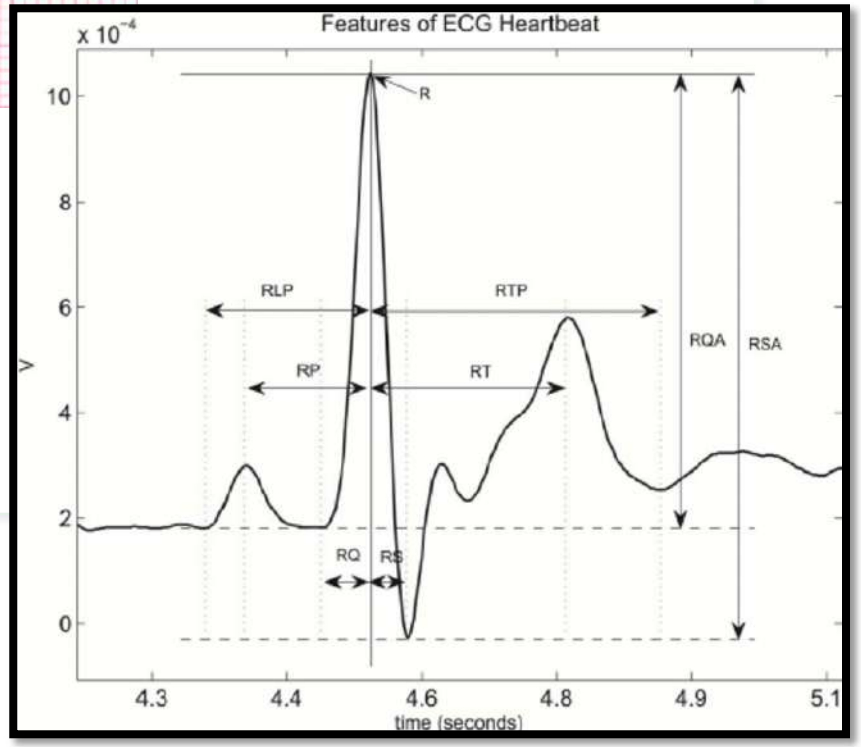
```
pearson Coef: 0.2640960301221321
p value: 0.00013007599065564113
```

What is a feature?

Features are the variables that describe the data. A **feature** is an individual measurable property or characteristic of a phenomenon being observed (Bishop 2006)

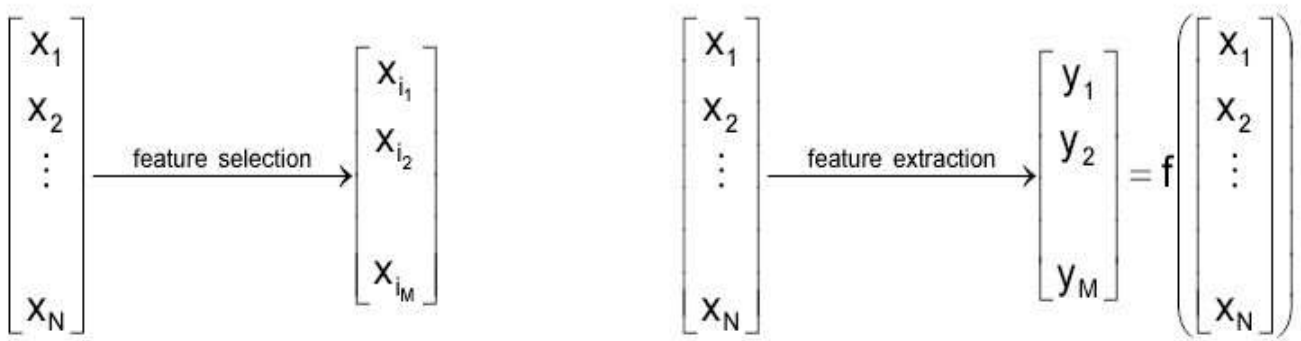






Feature Extraction

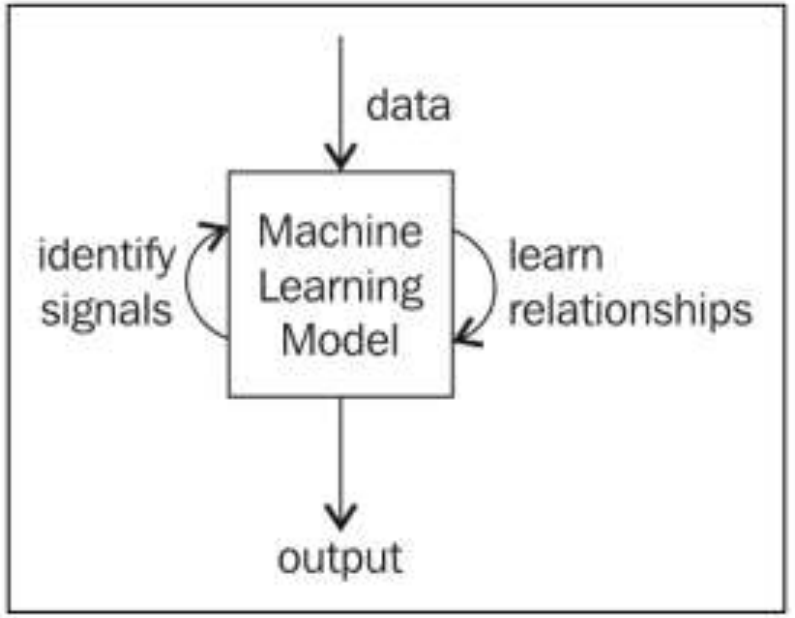
- **Two approaches are available to perform dimensionality reduction**
  - **Feature extraction**: creating a subset of new features by combinations of the existing features
  - **Feature selection**: choosing a subset of all the features (the ones more informative)



- **The problem of feature extraction can be stated as**
  - Given a feature space $\mathbf{x}_i \in \mathbf{R}^N$ find a mapping $\mathbf{y=f(x):R^N \rightarrow R^M}$ with $\mathbf{M<N}$ such that the transformed feature vector $y_i \in R^M$ preserves (most of) the information or structure in R^N.
  - An **optimal** mapping **y=f(x)** will be one that results in **no increase in the minimum probability of error**
    - This is, a Bayes decision rule applied to the initial space R^N and to the reduced space R^M yield the same classification rate

How does Machine Learning work?

- Machine learning works by taking in data, finding relationships within the data, and giving as output what the model learned, as illustrated in the following diagram:
- Three types of Machine Learning
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning

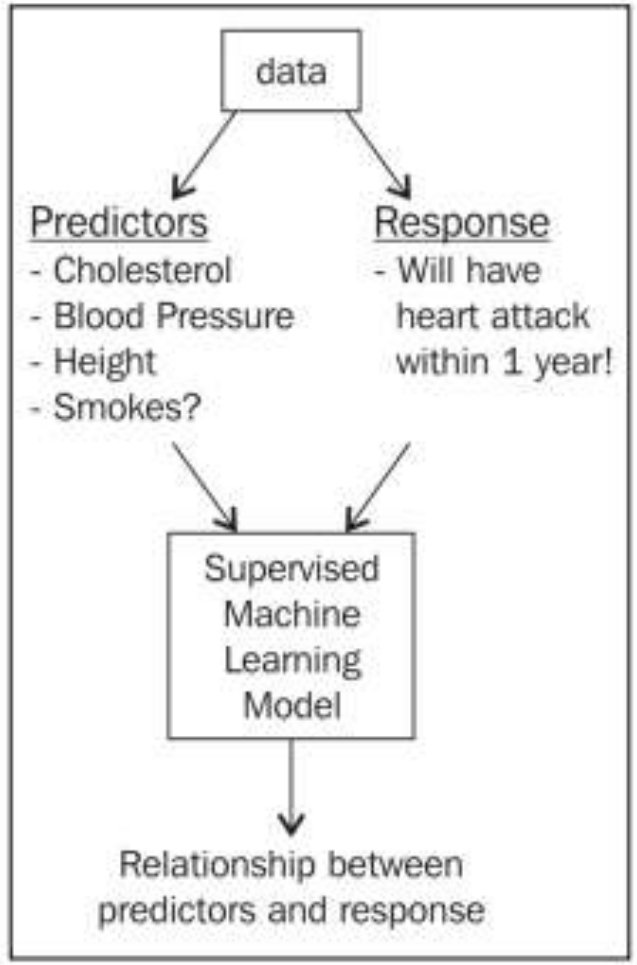


Supervised Learning

- Simply put, supervised learning finds associations between features of a dataset and a target variable.
- Supervised machine learning requires a certain type of data called labeled data.
- This means that we must teach our model by giving it historical examples that are labeled with the correct answer.
- we must separate data into two parts, as follows:
  - The predictors, which are the columns that will be used to make our prediction.
    - These are sometimes called features, inputs, variables, and independent variables.
  - The response, which is the column that we wish to predict.
    - This is sometimes called outcome, label, target, and dependent variable.

Example – Heart Attack Predication

- Suppose we wish to predict if someone will have a heart attack within a year. To predict this we need predictors, we are given that person's
  - cholesterol,
  - blood pressure,
  - height,
  - their smoking habits,
- The response is previous heart attack that was ever happened

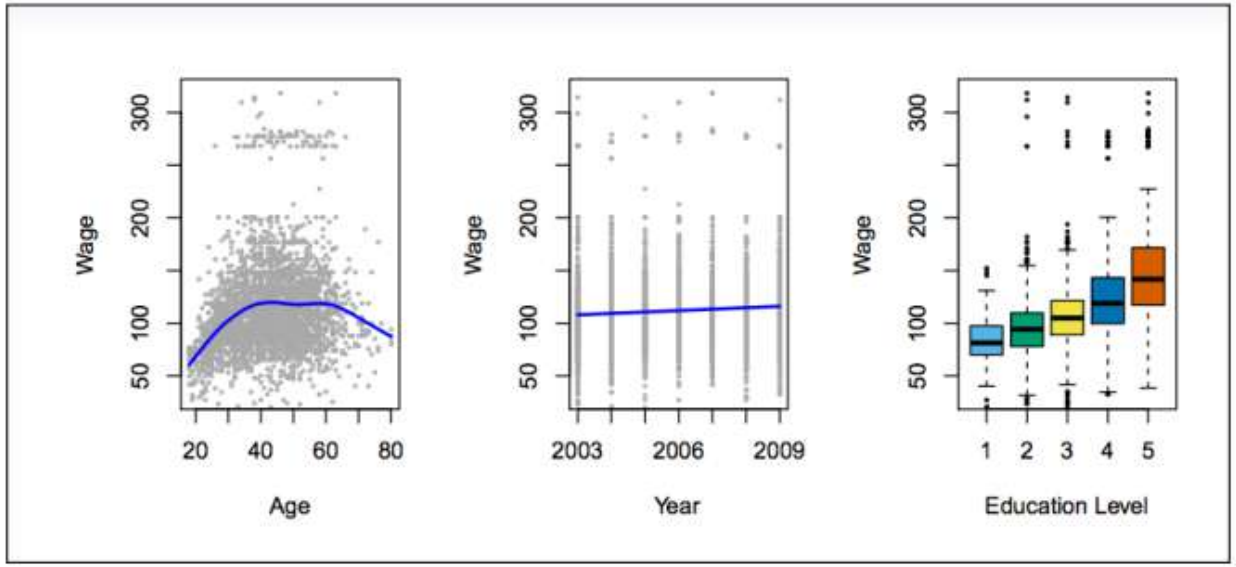


Types of supervised learning

- Two types of supervised learning models:
- Regression
  - Regression models attempt to predict a continuous response. This means that th eresponse can take on a range of infinite values.
    - Example : Salary, Budget, Temperature, Amount of cars
- Classification
  - Classification attempts to predict a categorical response, which means that the response only has a finite amount of choices.
    - Example : Cancer grade (1, 2, 3, 4, 5), facial recognition

Example – Supervised Learning- Regression

- The following graphs show a relationship between three categorical variables (age, year they were born, and education level) and a person's wage:



Source: https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/introduction.pdf

- Note that even though each predictor is categorical, this example is regressive because the y axis, our dependent variable, our response, is continuous.

Unsupervised learning

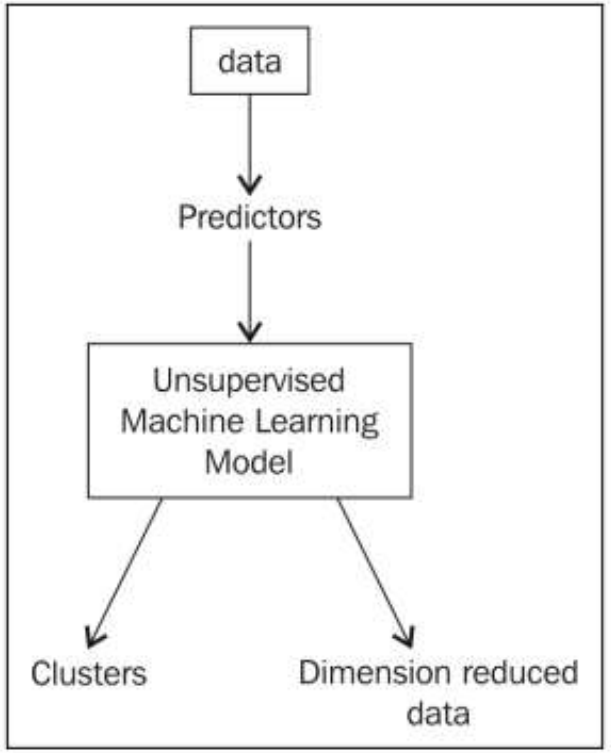
- The second type of machine learning does not deal with predictions but has a much more open objective.
- Unsupervised learning takes in a set of predictors and utilizes relationships between the predictors in order to accomplish tasks, such as
  - Dimension Reduction
    - Reducing the dimension of the data by condensing variables together. An example of this would be file compression. Compression works by utilizing patterns in the data and representing the data in a smaller format.
  - Clustering
    - Finding groups of observations that behave similarly and grouping them together.

Unsupervised Learning

- Both of these are examples of unsupervised learning because they **do not attempt** to **find a relationship** between **predictors** and a **specific response** and therefore are not used to make predictions of any kind.
- Unsupervised models, utilized to find organizations and representations of the data that were previously unknown.
- A big **advantage** for unsupervised learning is that it does not require labeled data, which means that it is much easier to get data that complies with unsupervised learning models.
- A **drawback** to this is that we lose all predictive power because the response variable holds the information to make predictions and without it our model will be hopeless in making any sort of predictions

Unsupervised Learning

- A big drawback is that it is difficult to see how well we are doing.
- In a regression or classification problem, we can easily tell how well our models are predicting by comparing our models' answers to the actual answers.
  - For example, if our supervised model predicts rain and it is sunny outside, the model was incorrect.
  - If our supervised model predicts the price will go up by 1 dollar and it goes up by 99 cents, our model was very close!



Classification

Classification attempts to predict a categorical response, which means that the response only has a finite amount of choices.

Classification Algorithm :

1. Naïve Bayes
2. Decision Tree
3. Support Vector Machine (SVM)
4. Multi Layer Perceptron
5. K-Nearest Neighbor (k-NN)
6. Etc

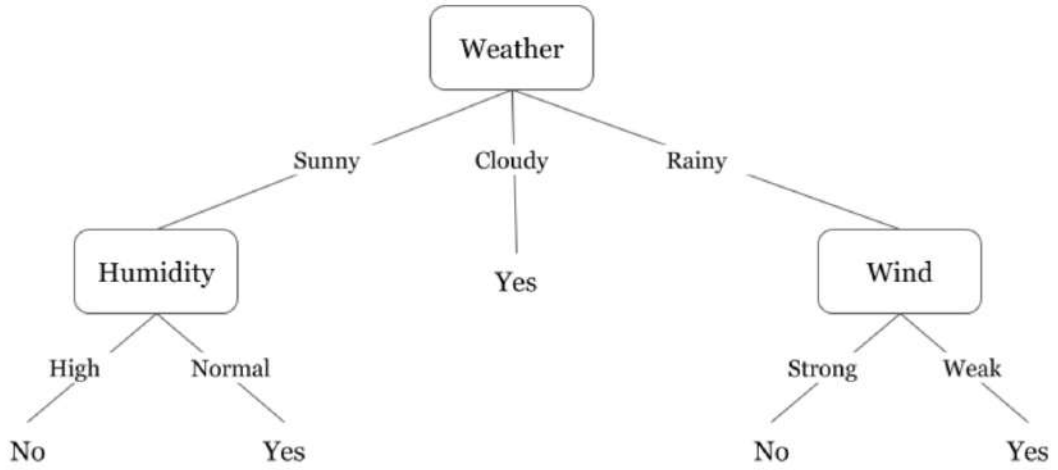
Decision Tree

Let's assume we want to play badminton on a particular day – say Saturday – how will you decide whether to play or not. Let's say you go out and check if it's hot or cold, check the speed of the wind and humidity, how the weather is, i.e. is it sunny, cloudy, or rainy. You take all these factors into account to decide if you want to play or not.

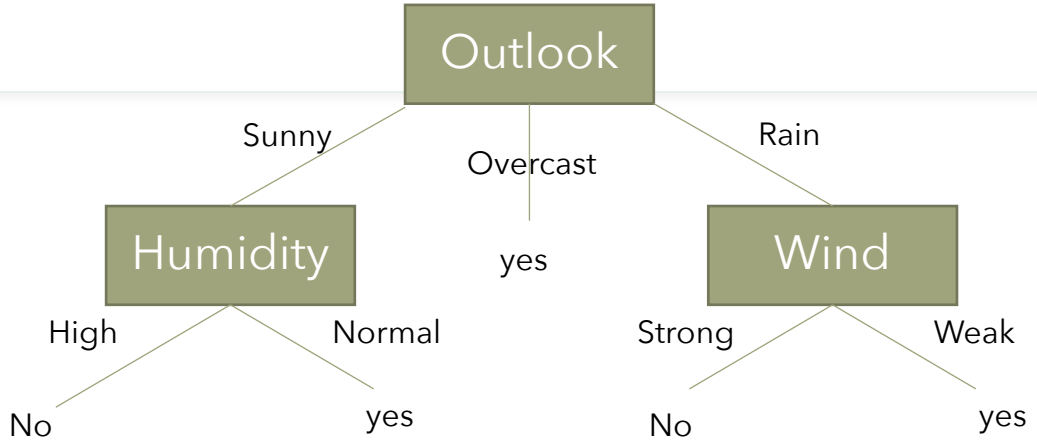
Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

Decision Tree

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No



Decision Tree Learning



- **Problem Settings**
  - Set of possible Instances X
    - Each instances x in X is a feature vector
    - e.g <Humidity=low, Wind=weak, Outlook=rain, Temp=hot>
  - Unknown target function f:X→Y
    - Y is a discrete value (No = 0, Yes = 1)
  - Set of function hypothesis H={ h | h:X→Y}
    - Each hypothesis h is a decision tree
    - Trees sorts x to leaf, which assign y

ID3 Algorithm

- Create root node for the tree
- If all examples are positive, return leaf node ‘positive’
- Else if all examples are negative, return leaf node ‘negative’
- Calculate the entropy of current state H(S)
- For each attribute, calculate the entropy with respect to the attribute ‘x’ denoted by H(S, x)
- Select the attribute which has maximum value of IG(S, x)
- Remove the attribute that offers highest IG from the set of attributes
- Repeat until we run out of all attributes, or the decision tree has all leaf nodes.

Entropy

- **Entropy H(X) of random variable X:**

$$H(X) = -\sum_{i=1}^{n} P(X = i)\ log_2\ P(X = i)$$

- **Specific conditional entropy H(X|Y=v) of given Y=v:**

$$H(X|Y = v) = -\sum_{i=1}^{n} P(X = i|Y = v)\ log_2\ P(X = i|Y = v)$$

- **Conditional Entropy H(X|Y) of given Y:**

$$H(X|Y) = -\sum_{v \epsilon values(Y)} P(Y = v)\ H(X = Y|v)$$

- **Mutual Information (Information Gain) of X and Y:**

$$I(X|Y) = H(X) - H(Y|X) = H(Y) - H(Y|X)$$

Information Gain: Mutual Information between input attribute A and target Y.
IG is expected reduction in entropy of target Y for data sample S

Training Example 1

Record	Attribute1	Attribute2	Kelas
1	M	N	1
2	T	N	1
3	F	N	1
4	F	N	1
5	T	O	1
6	M	N	0
7	F	O	0
8	T	O	0
9	T	N	0
10	F	O	0

a. Carilah nilai entropi utuk menentukan suatu record masuk kelas 0 atau 1. (10)
b. Bentuklah Decision Tree yang terbaik berdasarkan konsep Information Gain setiap atribut. (20)
c. Prediksi nilai kelas untuk record baru di bawah ini (jelaskan). (10)

Record	Attribute1	Attribute2	Kelas
11	F	N	?
12	T	N	?

d. Berapakah tingkat kepercayaan (*confidence*) prediksi pada tabel bagian (c)? (10)

Solution Training Example 1

a. Terdapat 5 *record* yang masuk ke kelas 1 dan 5 *record* yang masuk ke kelas 0. Perhitungan entropi untuk kelas adalah:

$$H(kelas) = I\left(\frac{5}{10}, \frac{5}{10}\right) = -\log_2 \frac{5}{10} - \log_2 \frac{5}{10} = 1$$

b. Untuk menghitung *information gain* attribute1, perhatikan bahwa terdapat 2 *record* dengan attribute1 = M, 4 *record* dengan attribute1 = T, dan 4 *record* dengan attribute1 = F.

Berdasarkan attribute1 dan kelasnya, maka *information gain* untuk attribute1 adalah:

$$IG(attribute1) = 1 - \left(\frac{2}{10} I\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{4}{10} I\left(\frac{2}{4}, \frac{2}{4}\right) + \frac{4}{10} I\left(\frac{2}{4}, \frac{2}{4}\right)\right)$$

$$= 1 - \left(\frac{2}{10} + \frac{4}{10} + \frac{2}{10}\right) = 0$$

Untuk menghitung *information gain* attribute2, perhatikan bahwa terdapat 6 *record* dengan attribute2 = N dan 4 *record* dengan attribute2 = O.

Berdasarkan attribute2 dan kelasnya, maka *information gain* untuk attribute2 adalah:

$$IG(attribute2) = 1 - \left(\frac{6}{10} I\left(\frac{4}{6}, \frac{2}{6}\right) + \frac{4}{10} I\left(\frac{1}{4}, \frac{3}{4}\right)\right)$$

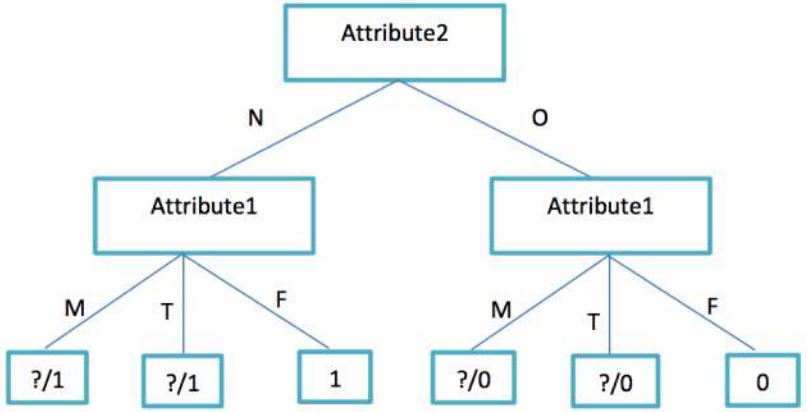
$$= 1 - (0.6(0.92) + 0.4(0.80)) = 0.13$$

$$= 1 - (0.55 + 0.32) = 0.13$$

Karena IG(attribute2) > IG(attribute1), maka attribute2 akan dipilih menjadi *root* untuk *decision tree.*

Solution Training Example 1

Berdasarkan perhitungan sebelumnya, *decision tree* yang terbentuk adalah:



Ukuran dataset sangat kecil dan atributnya sedikit sehingga *decision tree* yang terbentuk tidak dapat menentukan kelas untuk beberapa *record*.

Record dengan nilai Attribute2 = N, Attribute1 = M diprediksi ke kelas 1 karena record dengan Attribute2 = N lebih banyak di kelas 1. Sama dengan record dengan nilai Attribute2 = N, Attribute1 = T. Untuk record dengan Attribute2 = O diprediksi ke kelas 0 karena record dengan Attribute2 = 0 lebih banyak ke kelas 0.

c. Hasil prediksi untuk Record 11 (Attribute1 = F, Attribute2 = N) → Kelas 1
Hasil prediksi untuk Record 12 (Attribute1 = T, Attribute2 = N) → belum dapat diprediksi

Terdapat 2 record dengan attribute1 = T, attribute2 = N. 1 record masuk ke kelas 0 dan 1 record masuk ke kelas 1 sehingga tidak dapat ditentukan. Akan tetapi, lihat bahwa record dengan attribute2 = N lebih banyak masuk ke kelas 1, sehingga prediksinya adalah kelas 1.

Solution Training Example 1

d. Untuk Record 11 lihat di tabel aslinya atribut yang memiliki pola yang sama sebanyak 2 dan semuanya tepat sama jadi confidence-nya 2/2= 100%.
Untuk Record 12 di tabel aslinya attribute yang memiliki pola yang sama sebanyak 6 tapi hanya 4 yang menghasilkan prediksi yang sama jadi confidence-nya 4/6= 67%.

Training Example 2

Dataset apakah mahasiswa malas (M) atau rajin (D) berdasarkan berat
Badan (normal/ underweight), warna mata (amber/violet), dan jumlah Buku (2/3/4)

No.	Berat badan (B)	Warna mata (W)	Jumlah buku (J)	Kelas
1.	N	A	2	L
2.	N	V	2	L
3.	N	V	2	L
4.	U	V	3	L
5.	U	V	3	L
6.	U	A	4	D
7.	N	A	4	D
8.	N	V	4	D
9.	U	A	3	D
10.	U	A	3	D

a. Hitunglah Entropi untuk kelas,
b. Hitunglah entropi H(W|B=N)

Solution

a) Entropi untuk Kelas:

$$H(X) = -\sum_{i=1}^{n} P(X = i) log_2 P(X = i)$$

$$H(K) = -\big(P(K = L) log_2 P(K = L) + P(K = D) log_2 P(K = D)\big)$$

$$= -\left(\frac{5}{10} log_2 \left(\frac{5}{10}\right) + \frac{5}{10} log_2 \left(\frac{5}{10}\right)\right) = -log_2 \left(\frac{5}{10}\right) = -(-1) = 1$$

b) Entropi untuk H(W|B=N)

$$H(X|Y = v) = -\sum_{i=1}^{n} P(X = i|Y = v) log_2 P(X = i|Y = v)$$

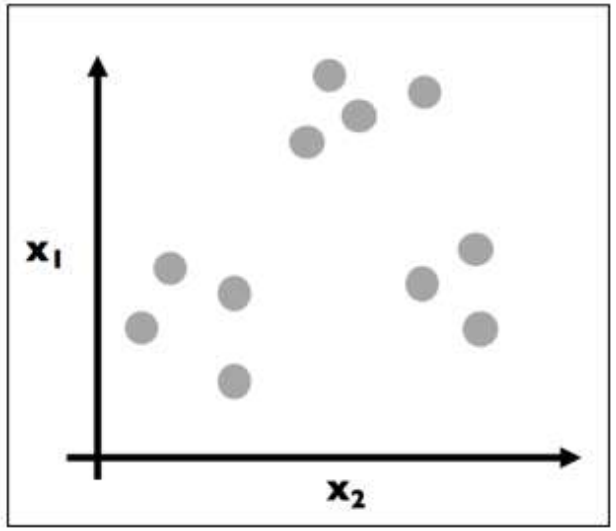
$$H(W|B = N) = -\big(P(W = A|B = N) log_2 P(W = A|B = N) + P(W = V|B = N) log_2 P(W = V|B = N)\big)$$

$$= -\left(\frac{2}{5} log_2 \left(\frac{2}{5}\right) + \frac{3}{5} log_2 \left(\frac{3}{5}\right)\right)$$

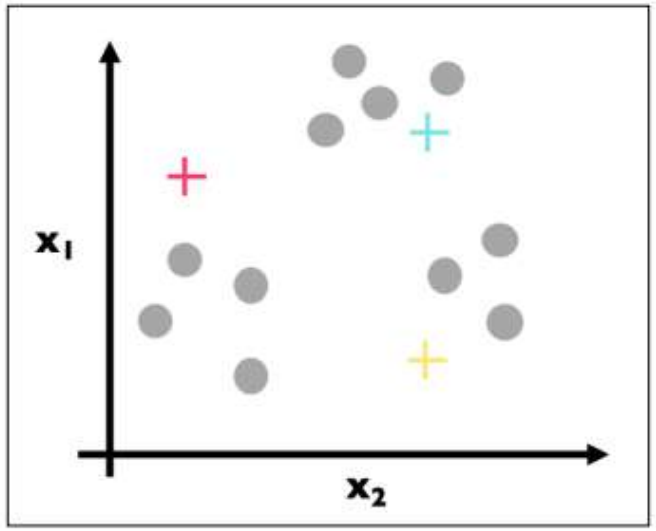
K-Means

- K-means is an iterative method that partitions a data set into k clusters. It works in four steps:
  - 1. Choose k initial centroids (note that k is an input).
  - 2. For each point assign the point to the nearest centroid.
  - 3. Recalculate the centroid positions.
  - 4. Repeat steps 2-3 until stopping criteria is met.

Step by step

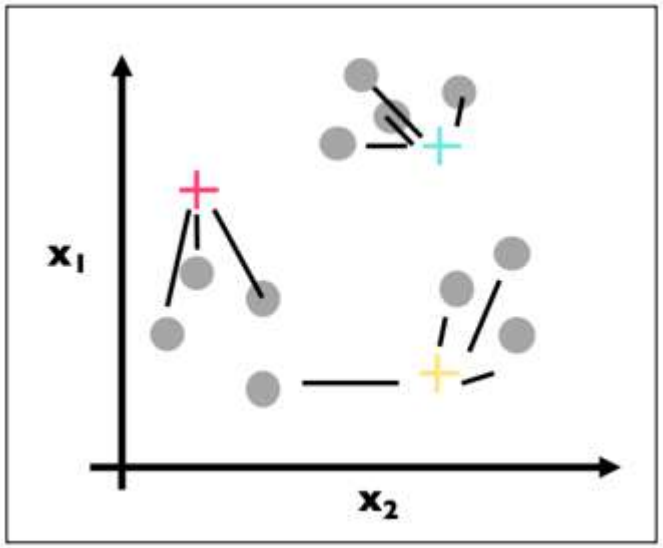


1. Each dot is colored grey so as to assume no prior grouping before applying the K-means algorithm.

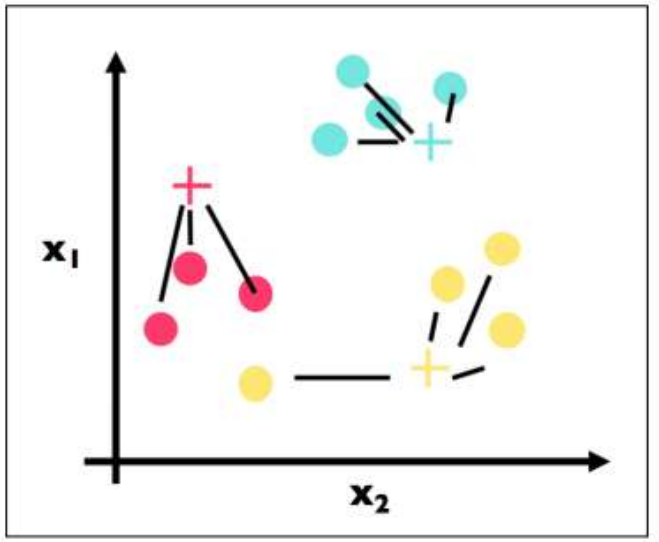


2. step 1 has been applied. We have (randomly) chosen three centroids (red, blue, and yellow).

Step by step

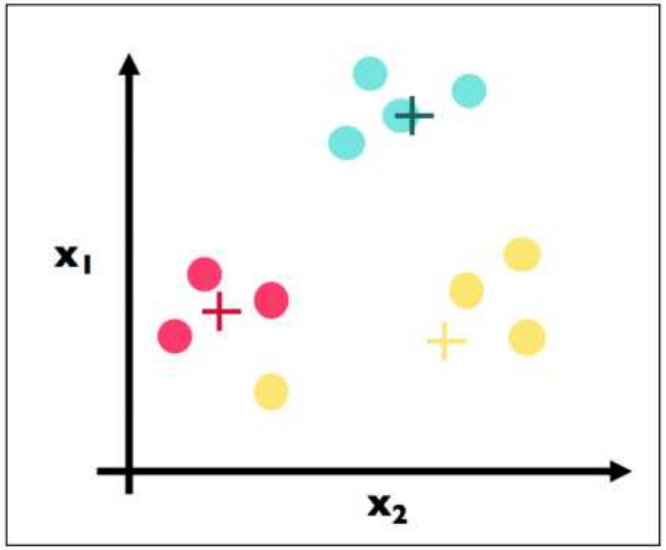


3. The first part of step 2 has been applied. For each data point, we found the most similar centroid (closest).

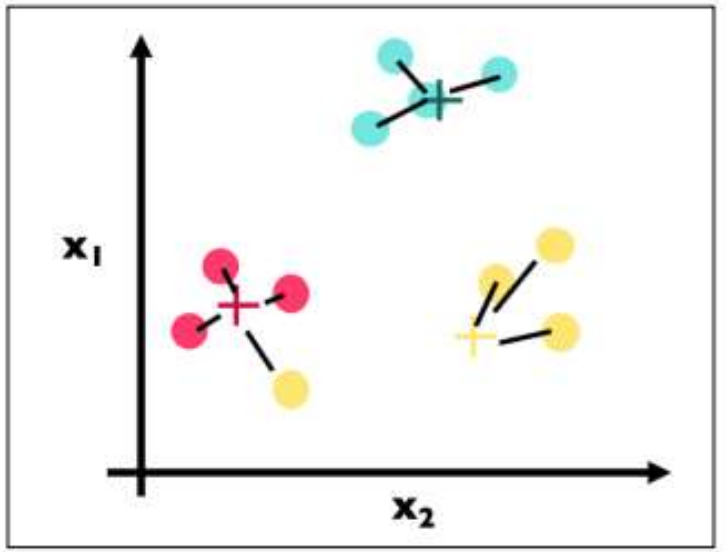


4. The second part of step 2 has been applied here. We have colored in each data point in accordance with its most similar centroid.

Step by step

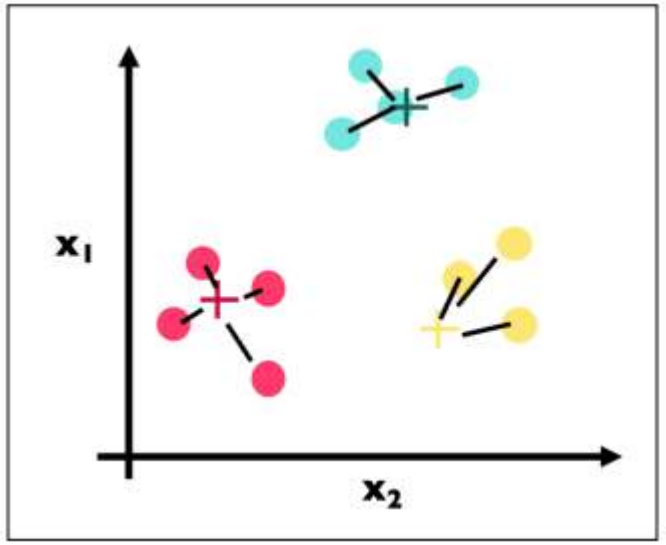


5. This is step 3 and the crux of K-Means. Note that we have physically moved the centroids to be the actual center of each cluster).

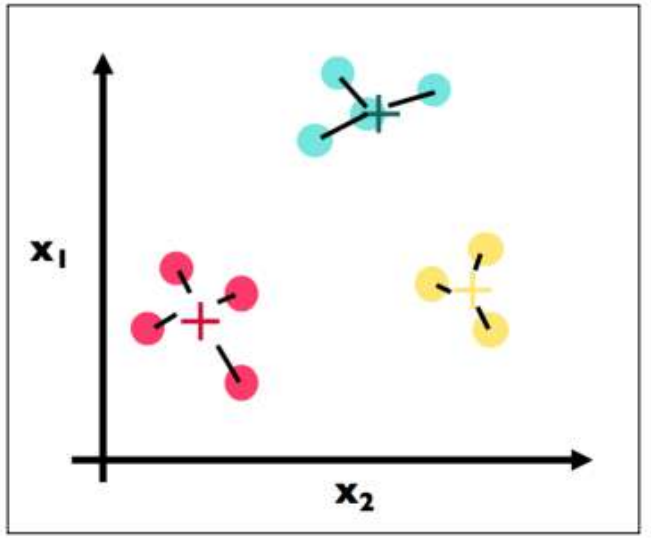


6. We continue with our algorithm by repeating step 2. Here is the first part where we find the closest center for each point. Note a big change: The point that is circled in the following figure used to be a yellow point, but has changed to be a red cluster point because the yellow cluster moved closer to its yellow constituents.

Step by step

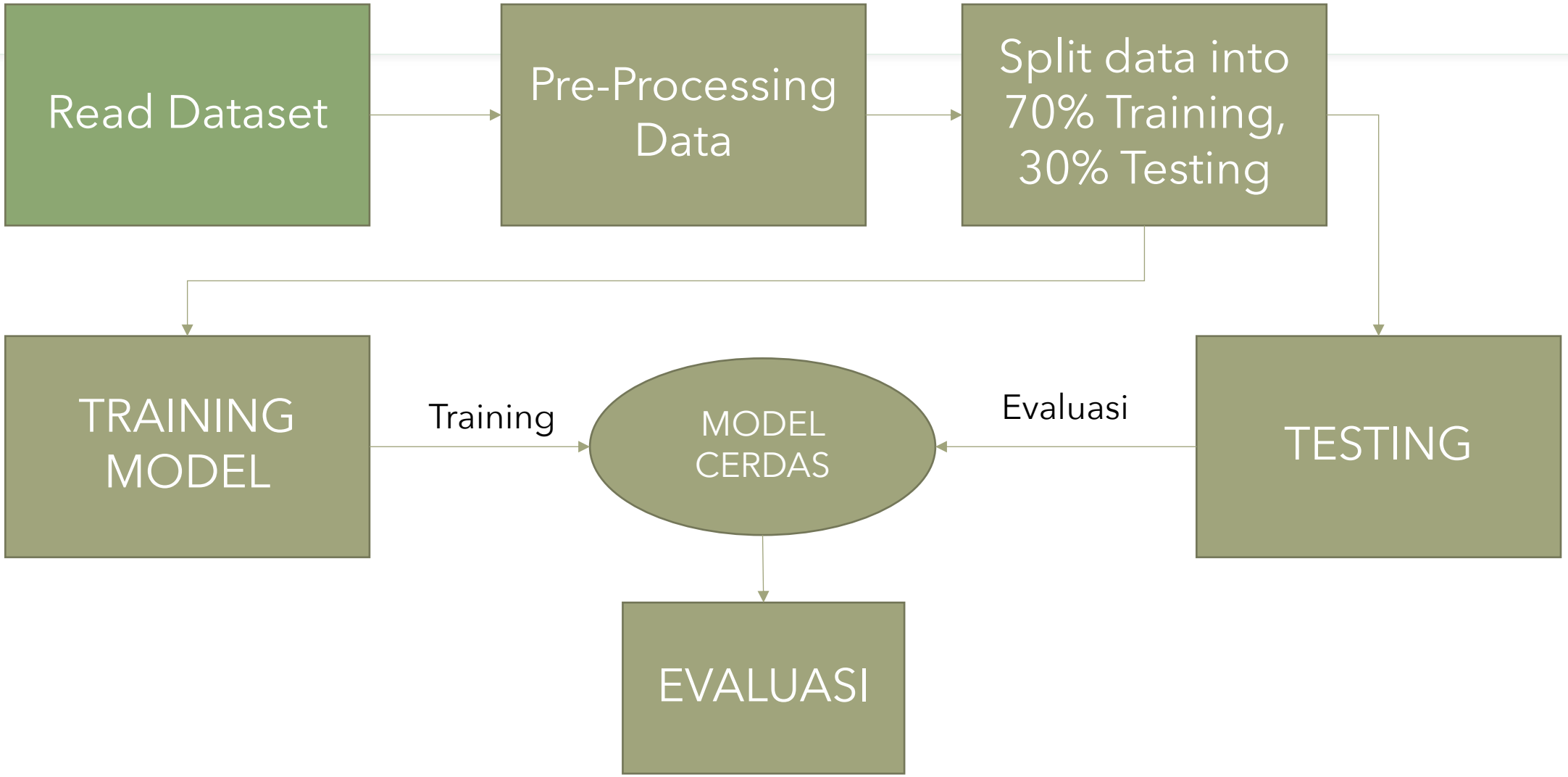


7. Here is the second part of step 2 again. We have assigned each point to the color of the closest cluster.



8. Here, we recalculate once more the centroids for each cluster (step 3). Note that the blue center did not move at all, while the yellow and red centers both moved.

Flow Classification Heart Disease



Read Dataset

- *Data = pd.read_csv("../dataset/diabetes.csv")*

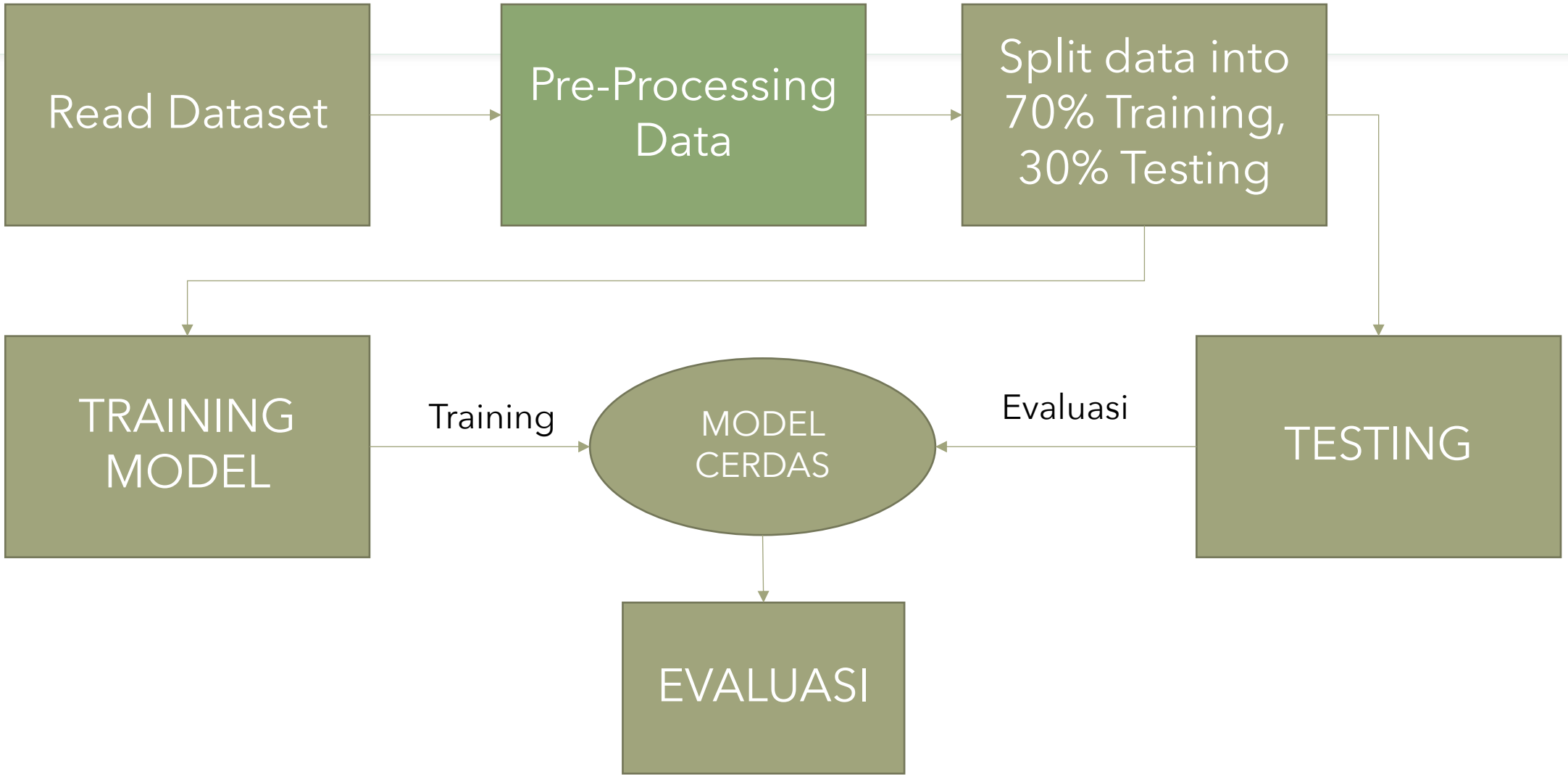
  There is a number of pandas commands to read other data formats:

```
pd.read_excel('myfile.xlsx',sheet_name='Sheet1')
pd.read_stata('myfile.dta')
pd.read_sas('myfile.sas7bdat')
pd.read_hdf('myfile.h5','df')
```

Data Frame Pandas

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Flow Classification Heart Disease



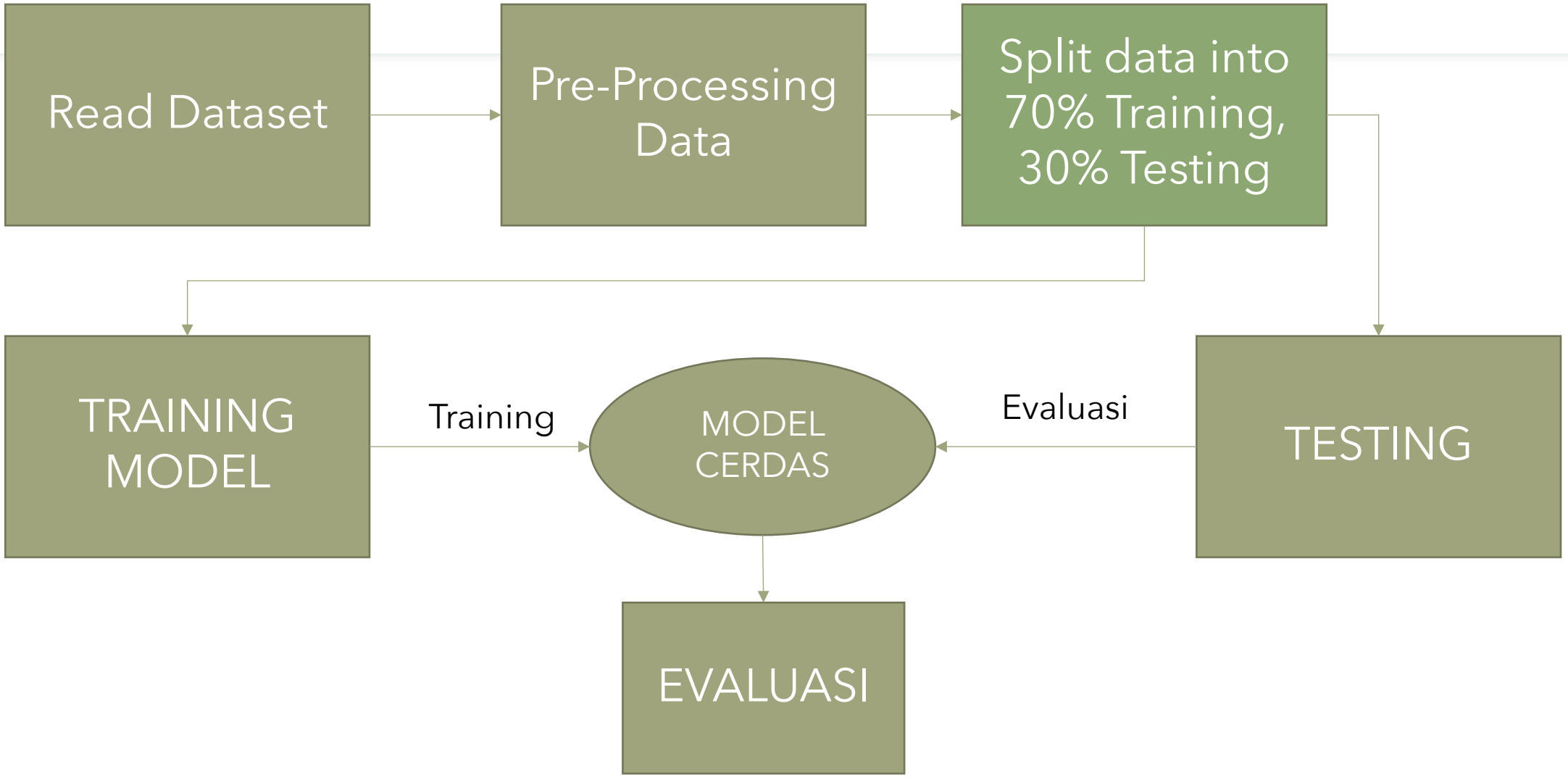
Data Frame iLoc

- If we need to select a range of rows and/or columns, using their positions we can use method iloc:

```
X = data.iloc[:,0:13]
#Save first 13 column and all row to X
X.head()
```

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
63	1	3	145	233	1	0	150	0	2.3	0	0	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2
41	0	1	130	204	0	0	172	0	1.4	2	0	2
56	1	1	120	236	0	1	178	0	0.8	2	0	2
57	0	0	120	354	0	1	163	1	0.6	2	0	2

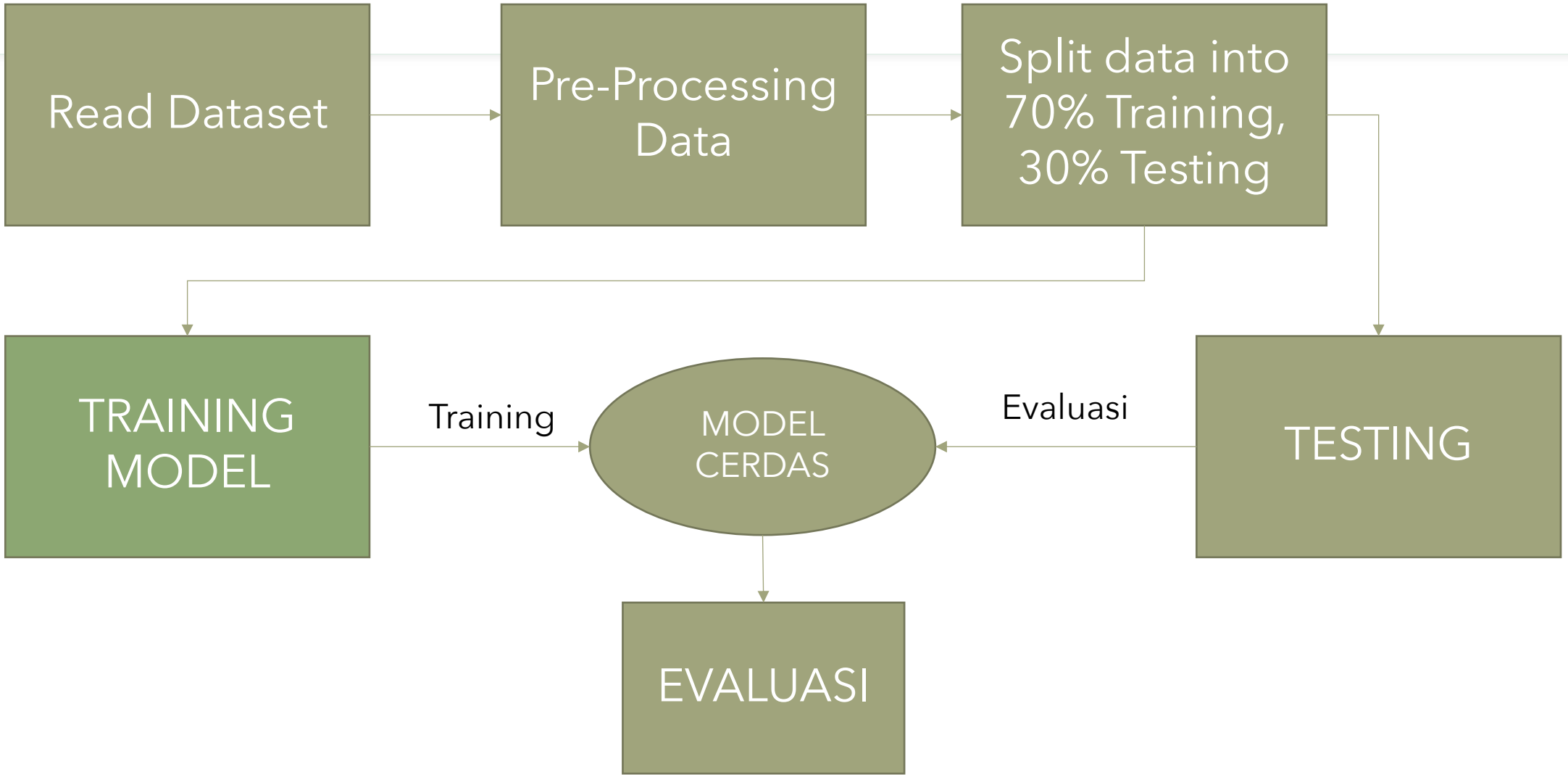
Flow Classification Heart Disease



Train Test Split

- `X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3, random_state=42)`

Flow Classification Heart Disease

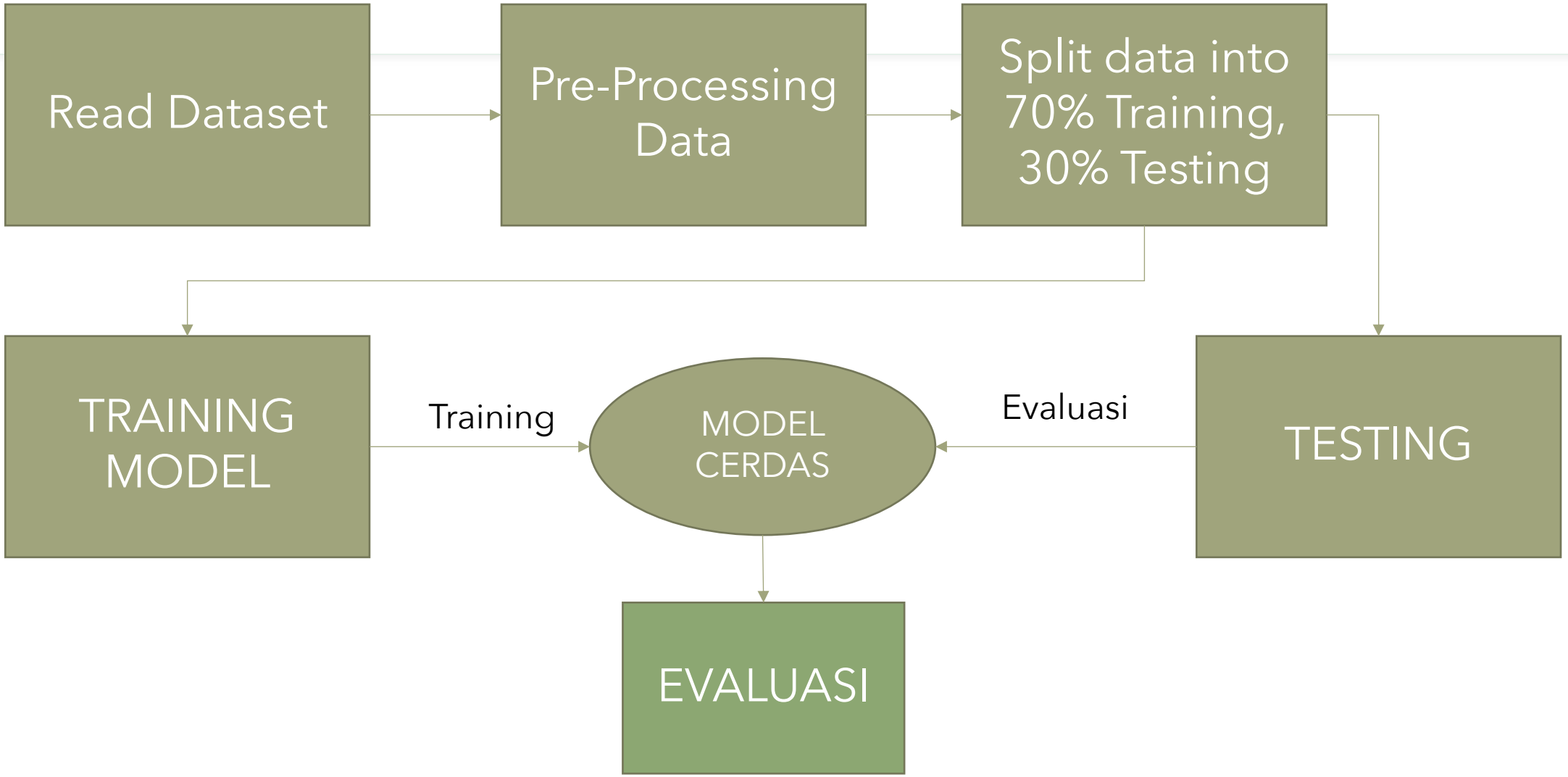


Training Model

- Pelajari Module Decision Tree dari https://scikit-learn.org/stable/modules/tree.html

- Pelajari dan Implementasikan Module Naïve Bayes dari https://scikit-learn.org/stable/modules/naive_bayes.html

Flow Classification Heart Disease



Precision Recall + Confusion Matrix

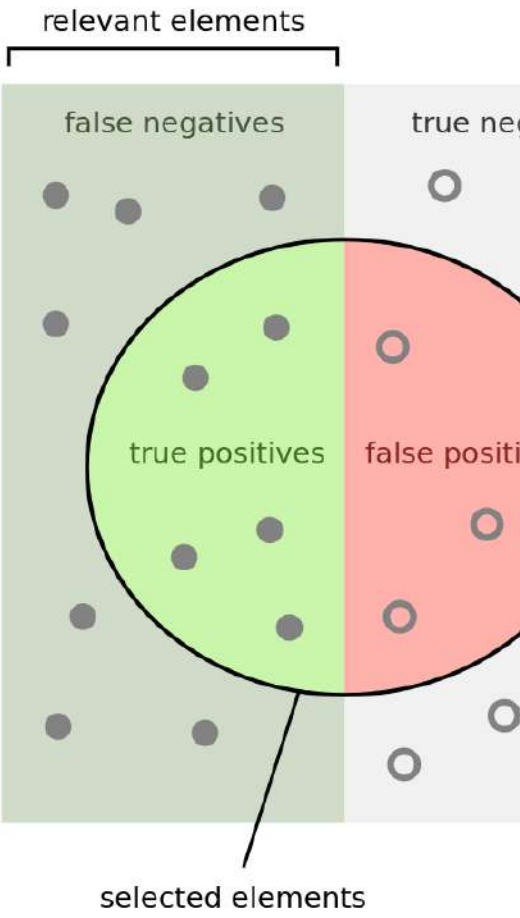
$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

```
[[249,   0,   8,   0,  10,   0,   7,   4,   0],
 [  0, 261,   4,   0,   0,   0,   0,   1,   4],
 [ 15,   3, 232,   0,   1,   0,   0,   2,   0],
 [  0,   0,   0, 363,   0,   7,   1,   0,   0],
 [ 63,   1,   7,  16,  14,   5,  13,  12,   0],
 [  1,   0,   0,  35,   1,  15,  11,   0,   0],
 [  0,   0,   0,   0,   0,   0, 393,   1,   0],
 [  2,   0,   0,   0,   0,   0,   2, 514,   0],
 [  0,  55,   2,   0,   0,   0,   0,   0,  50]])
```



How many selected items are relevant?

Precision =

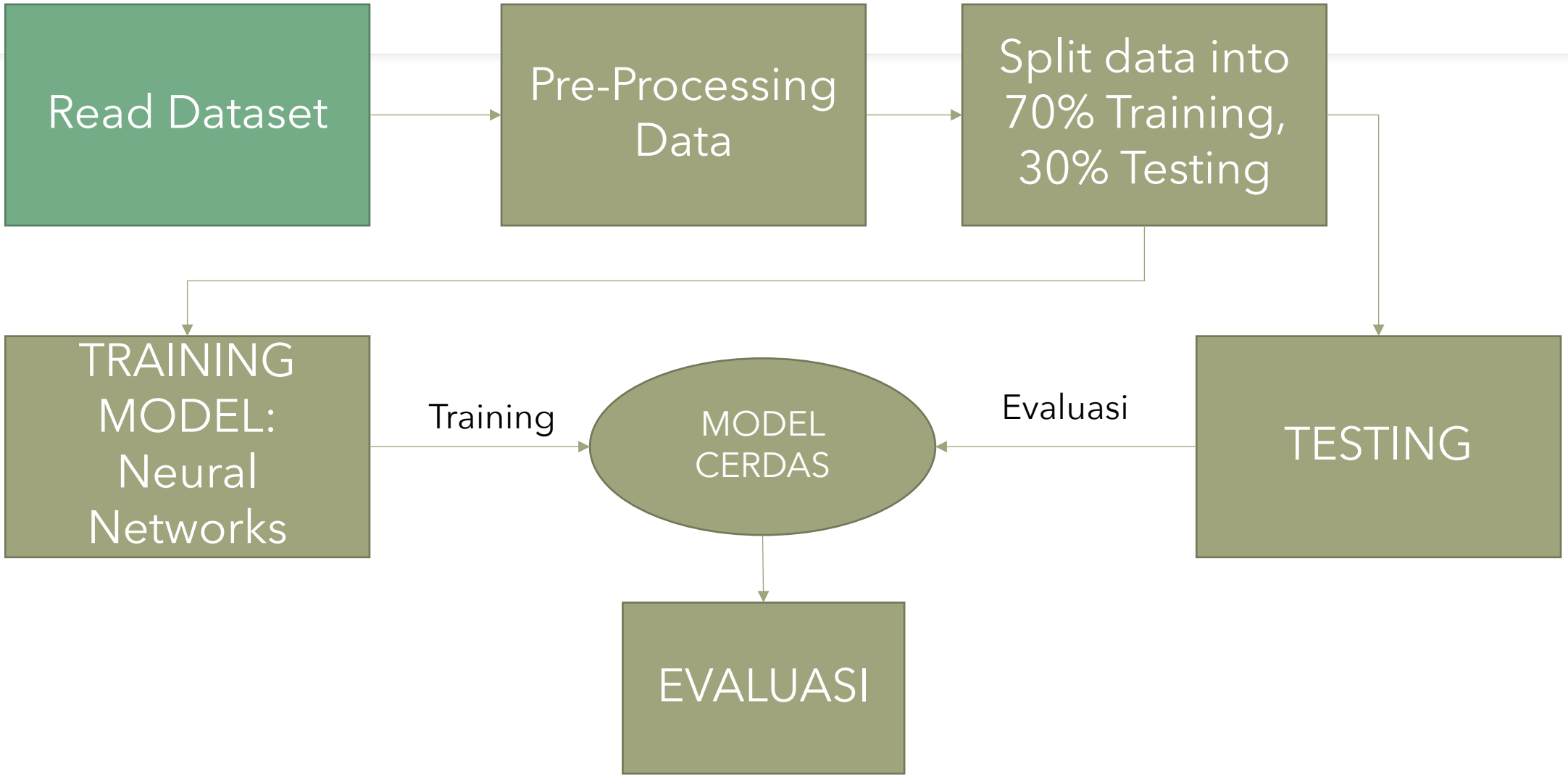
How many items are s

Recall =

Outline

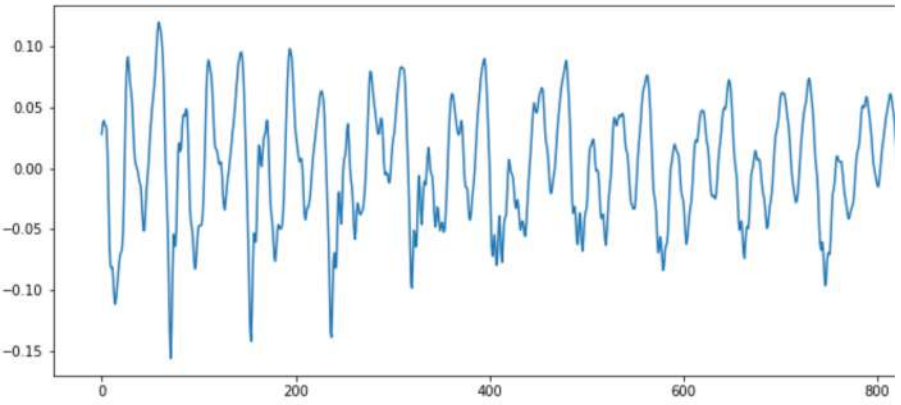
- Trend Machine Learning Industri
- Pendekatan Machine Learning
- **Signal Processing dengan Machine Learning**

Flow Classification: Voice Gender



Flow Classification: Voice Gender Recognition

emosi	sentiment	ZCR	SC	RMSE	SB	SROLL	SFLAT	SCON
4	3	0.430664	5395.540679	0.000003	2970.705638	8914.746094	0.305180	24.323693
4	3	0.040527	1180.375774	0.026604	1557.050021	2713.183594	0.000447	29.543887
4	3	0.068848	1617.700879	0.000417	1895.989101	3186.914062	0.007749	10.379656
4	3	0.074707	2067.990375	0.000701	1784.612375	3552.978516	0.011723	22.355055
4	3	0.065918	2118.206491	0.000601	2251.859553	4618.872070	0.010714	10.943335

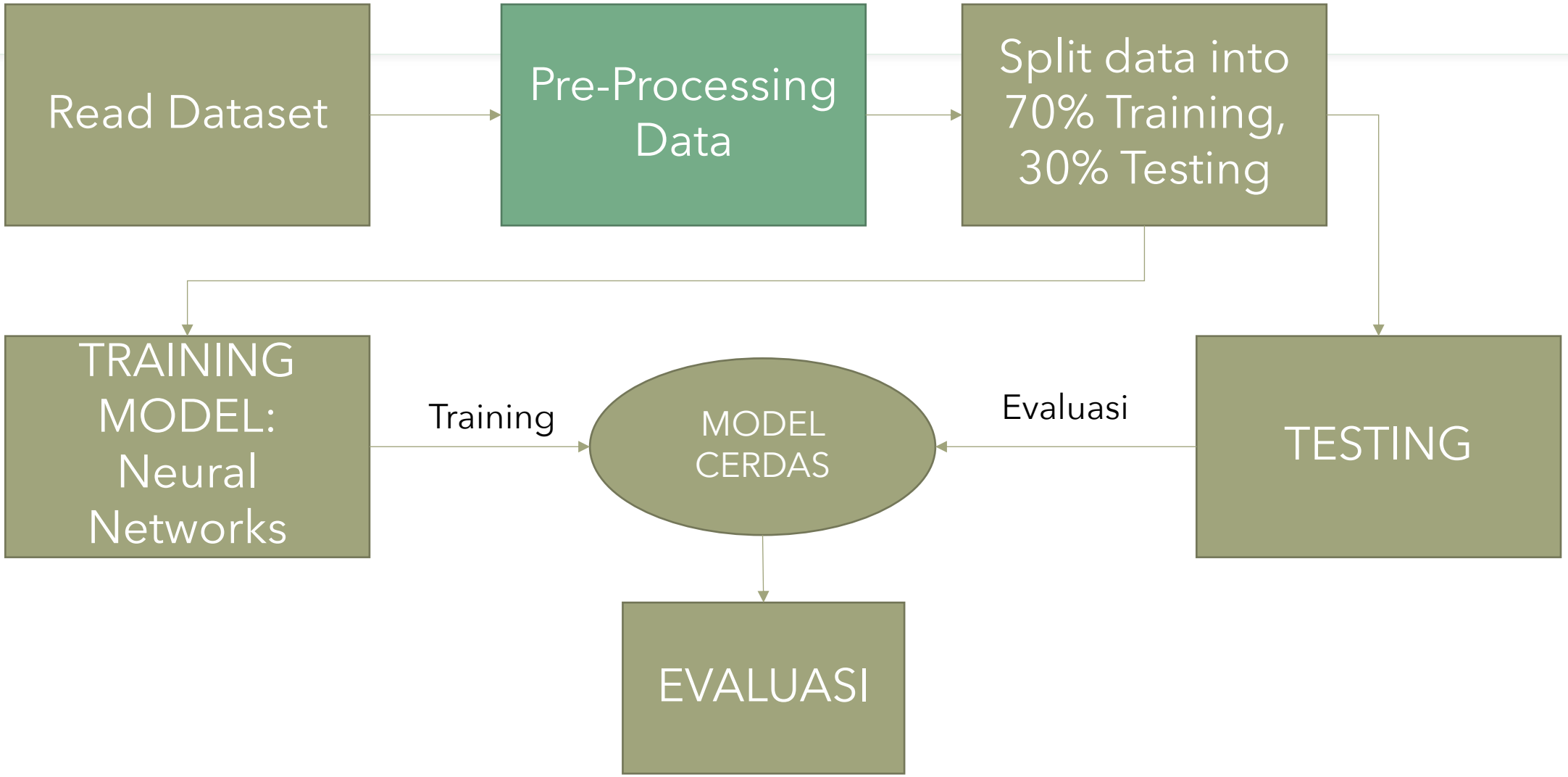


meanfreq	sd	median	Q25	Q75	IQR	skew
0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462
0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285
0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155

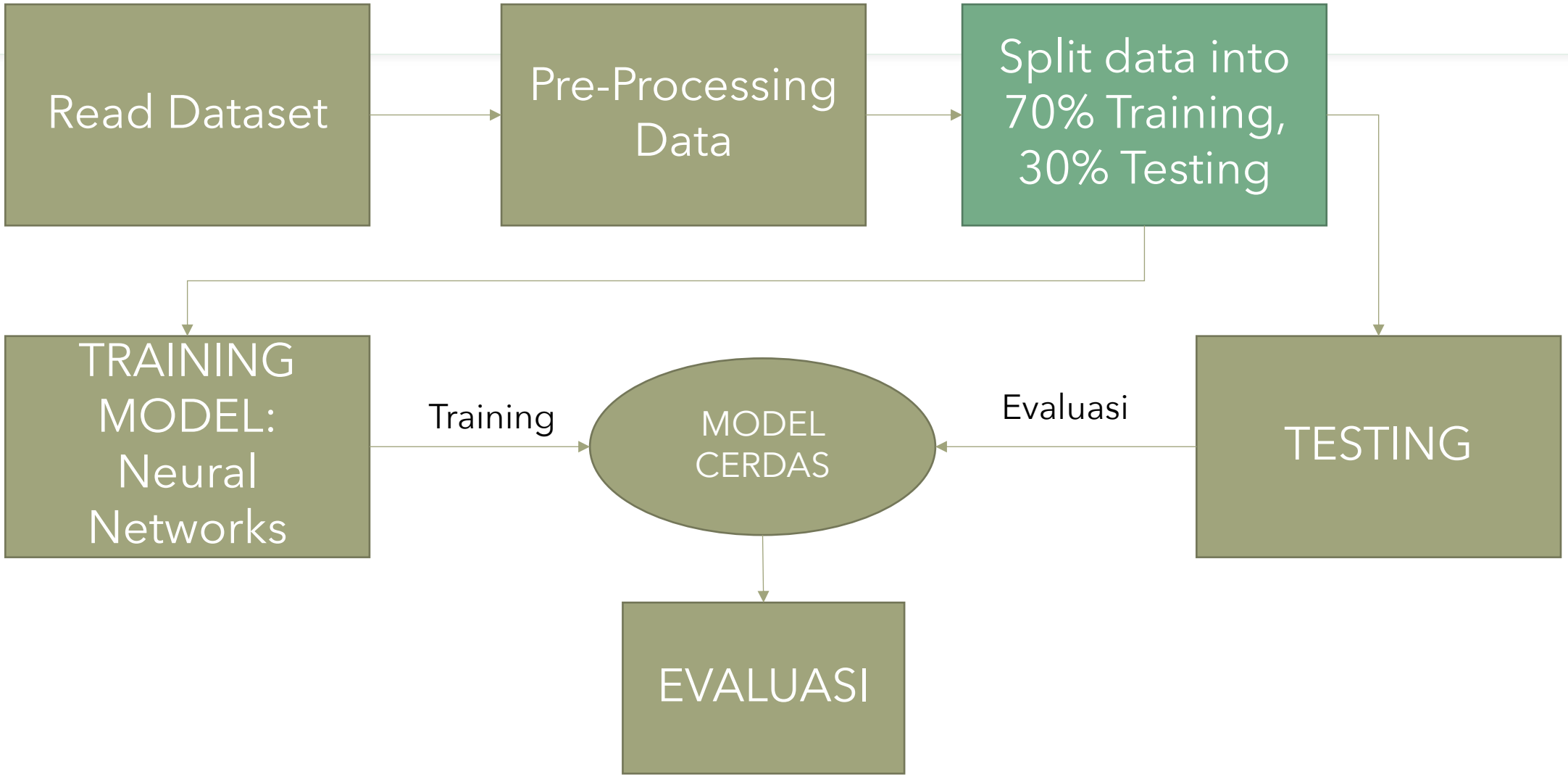
Flow Classification: Voice Gender Recognition

```
data_low_level = []
def extract_low_features(signal):
  zcr = librosa.feature.zero_crossing_rate(signal[0][0])[0, 0]
  sc  = librosa.feature.spectral_centroid(signal[0][0])[0, 0] #average freq
  sb  = librosa.feature.spectral_bandwidth(signal[0][0])[0, 0] #varian
  sroll  =  librosa.feature.spectral_rolloff(signal[0][0])[0, 0] #max freq
  sflat  =  librosa.feature.spectral_flatness(signal[0][0])[0, 0] #flat
  scon  = librosa.feature.spectral_contrast(signal[0][0])[0, 0] #contrast
  rmse = librosa.feature.rmse(signal[0][0])[0, 0]
  mfcc = librosa.feature.mfcc(y=signal[0][0], sr=signal[0][1], n_mfcc=40)

  return zcr, sc, rmse, mfcc, sb, sroll, sflat, scon

for x in audio_spec:
  try:
    data_low_level.append(extract_low_features(x))
  except:
    print("Error Baca File")
```

Flow Classification: Voice Gender



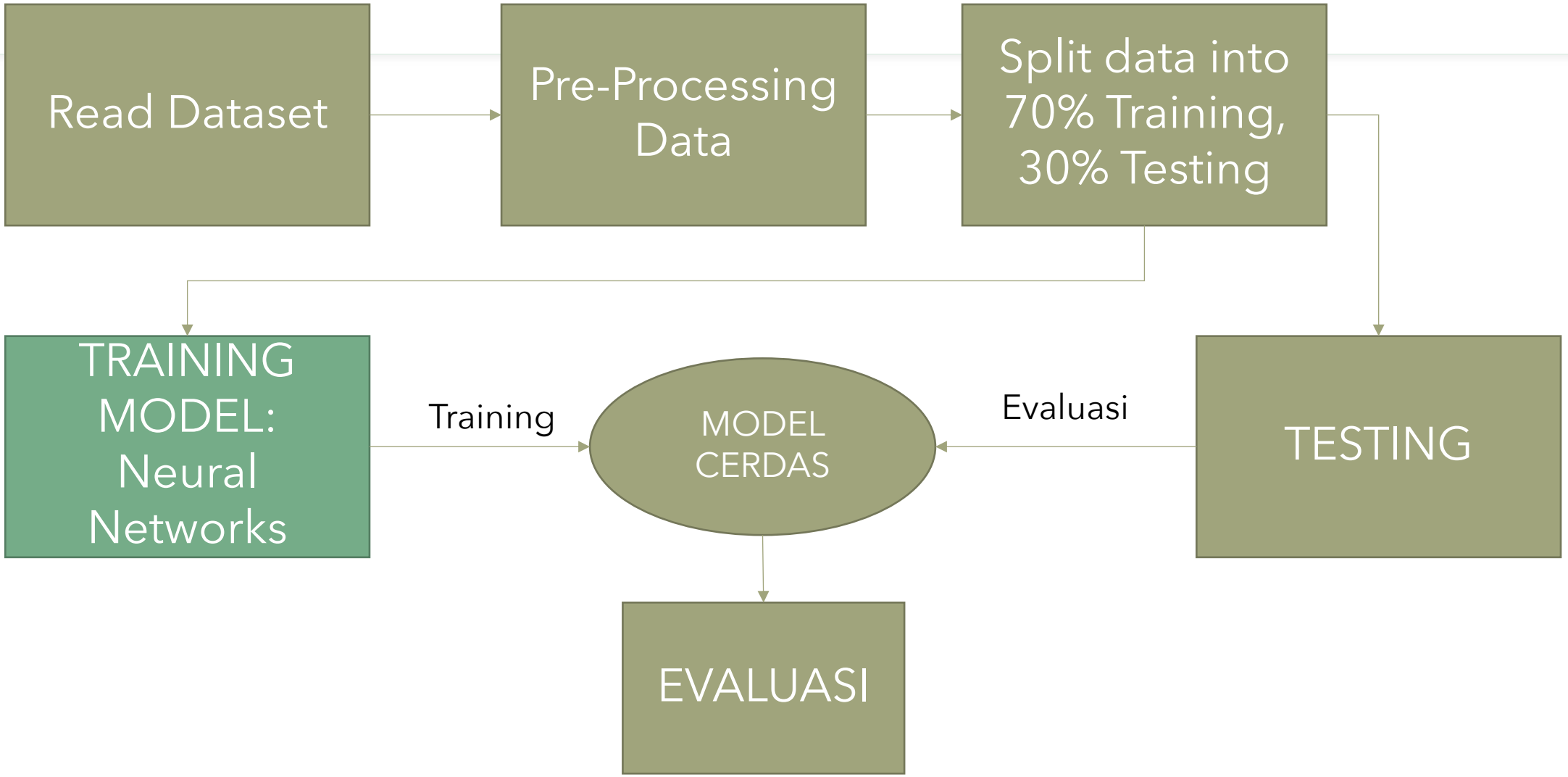
Flow Classification: Voice Gender



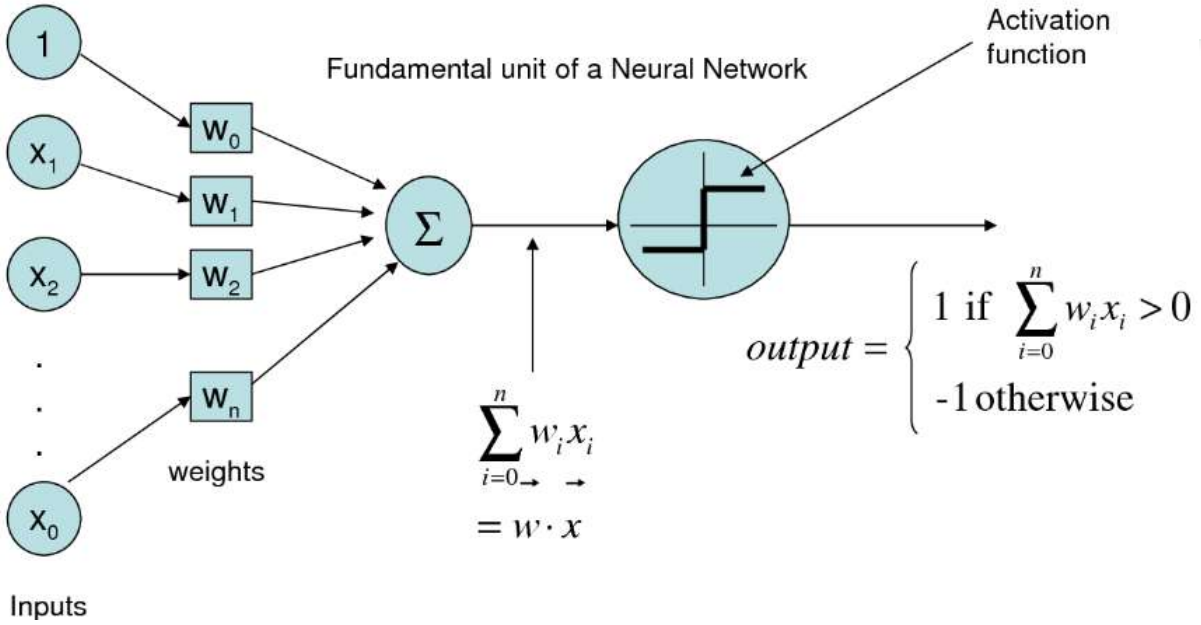
Flow Classification: Voice Gender

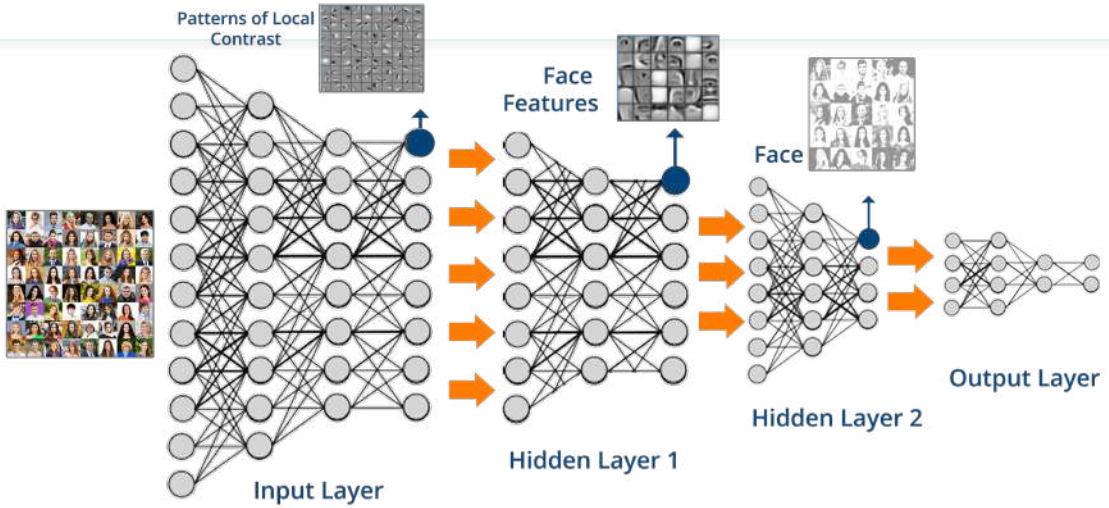
```
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
from sklearn import preprocessing #label encoder: categorical --> numeric
from keras.utils import np_utils

X = df.iloc[:, 0:df.shape[1]-1] #dataset_fix yang isinya low level feature kit
y = df.iloc[:, df.shape[1]-1] #dataset_fix untuk class label kita jadikan y

le = preprocessing.LabelEncoder() #panggil LE
le.fit(y)
y = le.transform(y) #ubah class yang masih text ke numeric

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)

y_train_ = to_categorical(y_train, 2) #change label to binary / categorical: [
y_test_ = to_categorical(y_test, 2) #change label to binary / categorical
```

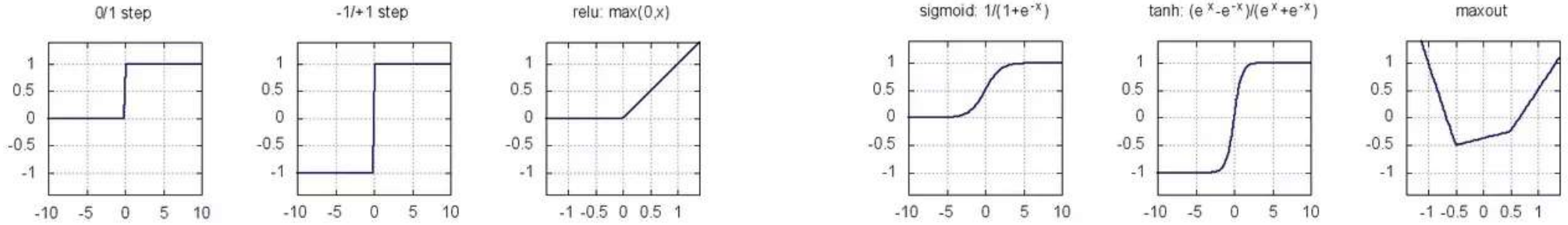
Flow Classification: Contoh Klasifikasi



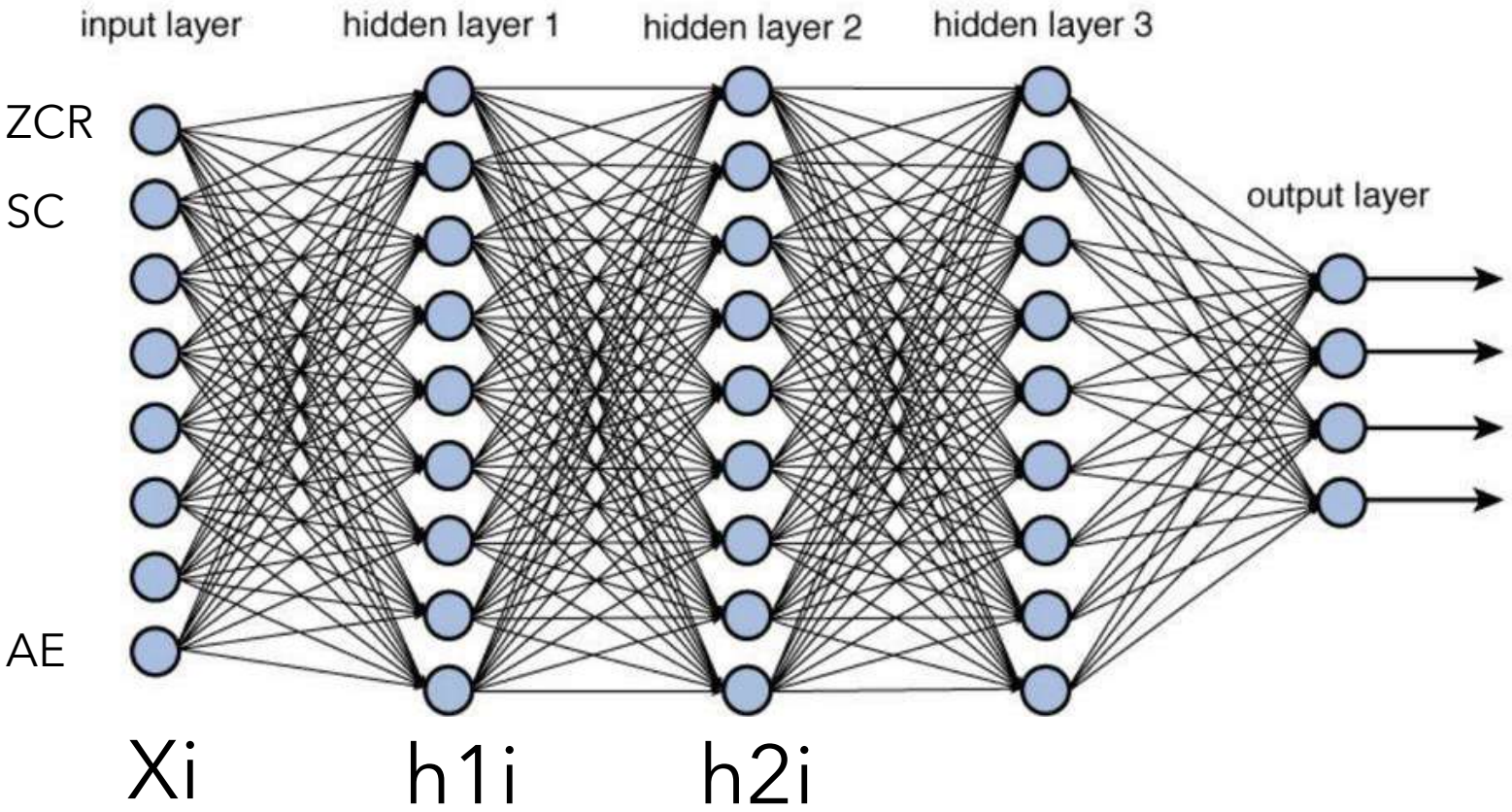
Neural Networks







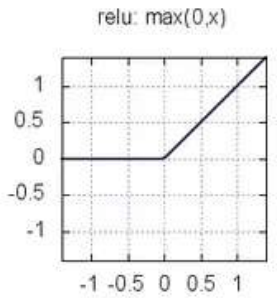
Flow Classification: Machine Learning Model



1 = male
0 = female

$$S(x,W) = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} W_{(i-m,j-n)}$$

$$Z(S,U) = \sum_{i=1}^{n}\sum_{j=1}^{m} S_{ij} U_{(i-m,j-n)}$$



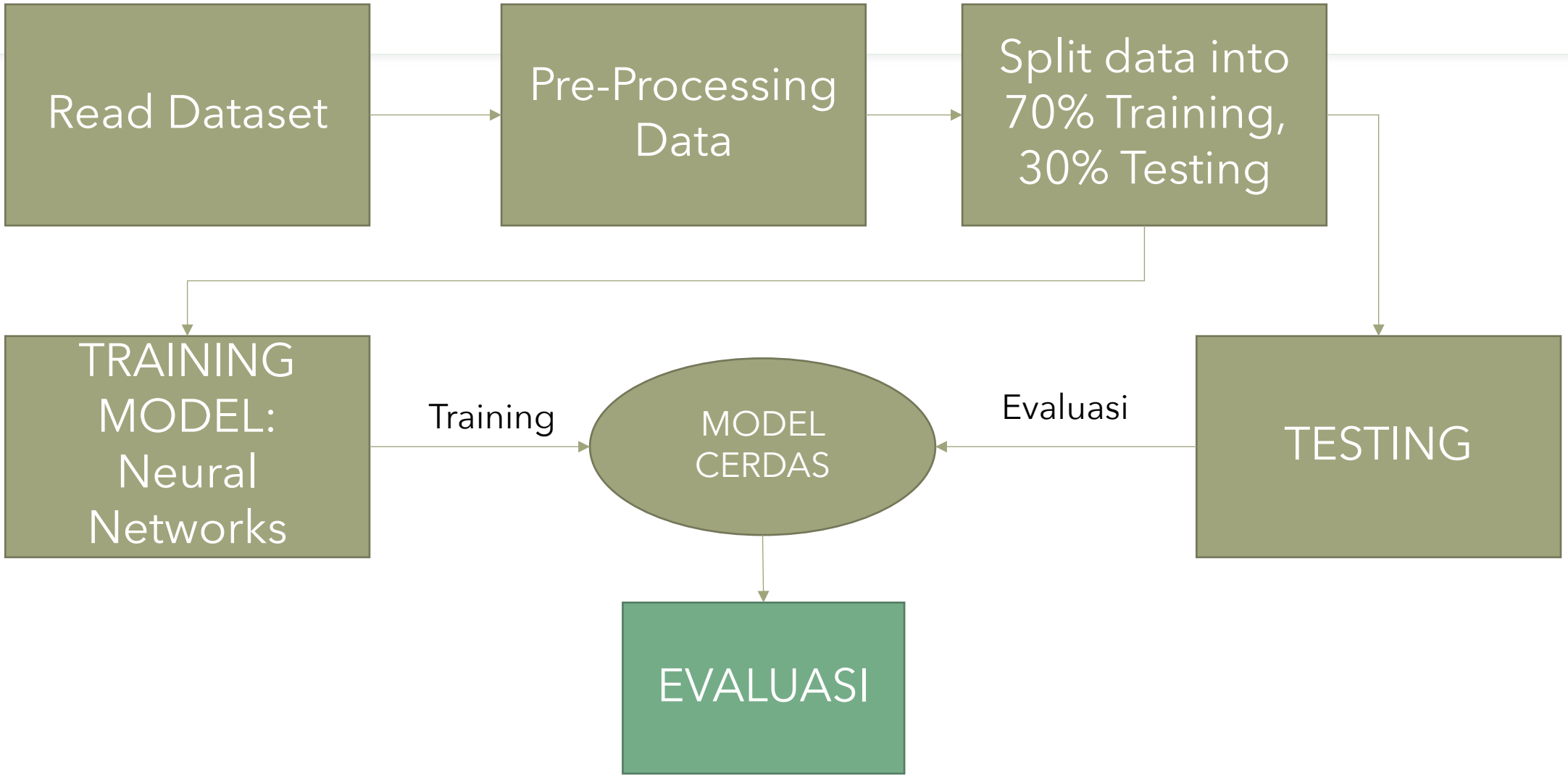
$$ReLU = \begin{cases} x, if\ x > 0 \\ 0, otherwise \end{cases}$$

$$Softmax(z_i) = \frac{\exp(Dl(B_{ij} + h_i, W))}{\sum_{i=1}^{n} \exp(Dl(B_{ij} + h_i, W))}$$

Training Process

```
loss: 0.4712 - acc: 0.8066 - val_loss: 0.4341 - val_acc: 0.8494

loss: 0.4568 - acc: 0.8184 - val_loss: 0.4301 - val_acc: 0.8564

loss: 0.4561 - acc: 0.8189 - val_loss: 0.4374 - val_acc: 0.8546

loss: 0.4509 - acc: 0.8202 - val_loss: 0.4273 - val_acc: 0.8476
```

Flow Classification: Contoh Klasifikasi



Precision Recall + Confusion Matrix

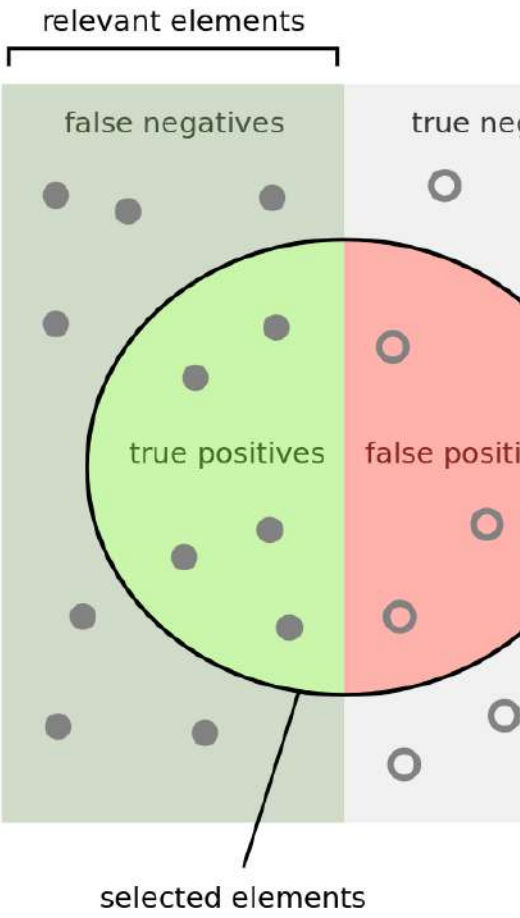
$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

```
[[249,   0,   8,   0,  10,   0,   7,   4,   0],
 [  0, 261,   4,   0,   0,   0,   0,   1,   4],
 [ 15,   3, 232,   0,   1,   0,   0,   2,   0],
 [  0,   0,   0, 363,   0,   7,   1,   0,   0],
 [ 63,   1,   7,  16,  14,   5,  13,  12,   0],
 [  1,   0,   0,  35,   1,  15,  11,   0,   0],
 [  0,   0,   0,   0,   0,   0, 393,   1,   0],
 [  2,   0,   0,   0,   0,   0,   2, 514,   0],
 [  0,  55,   2,   0,   0,   0,   0,   0,  50]])
```



How many selected items are relevant?

Precision =

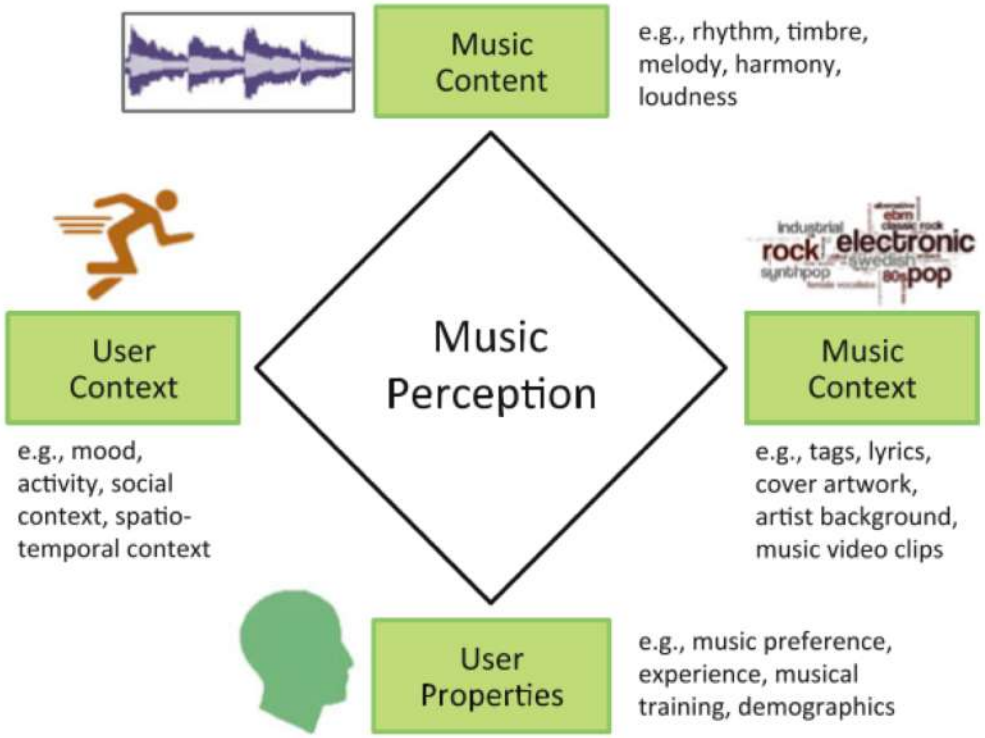
How many items are s

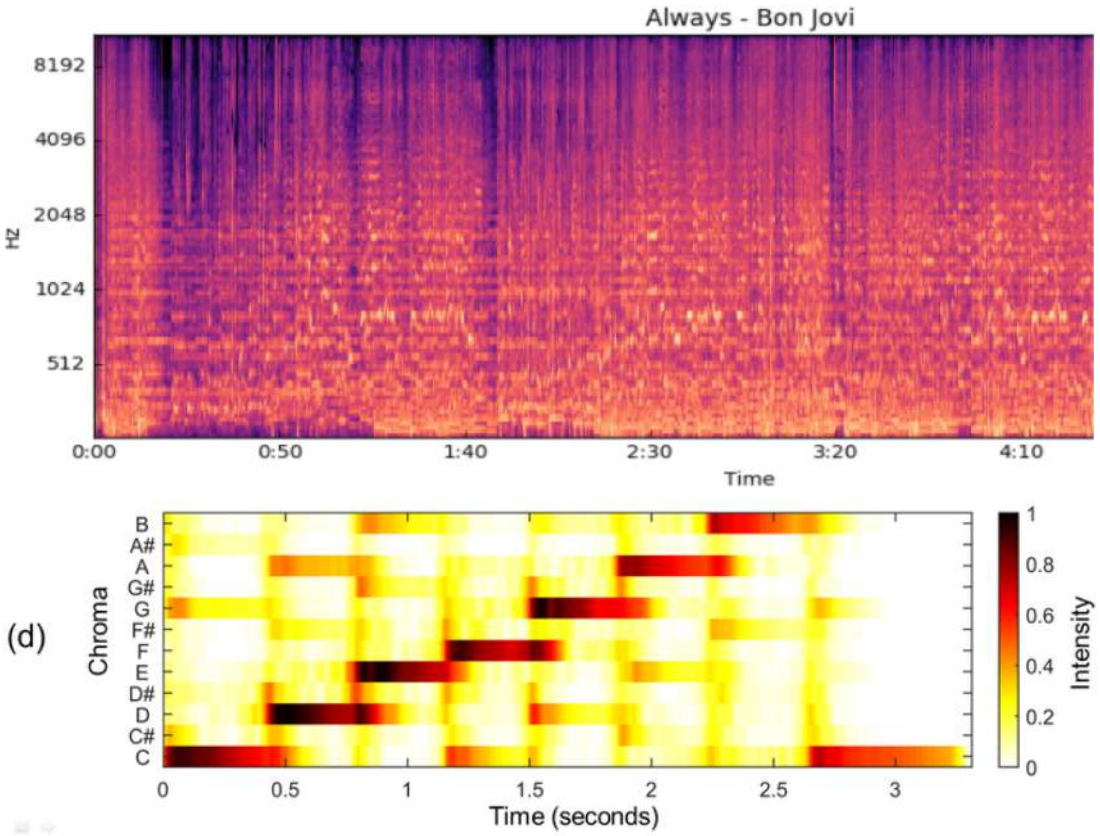
Recall =

Flow Classification: Evaluasi

	precision	recall	f1-score	support
	0.77	0.80	0.79	41
	0.83	0.80	0.82	50
	0.80	0.80	0.80	91

Music Retrieval



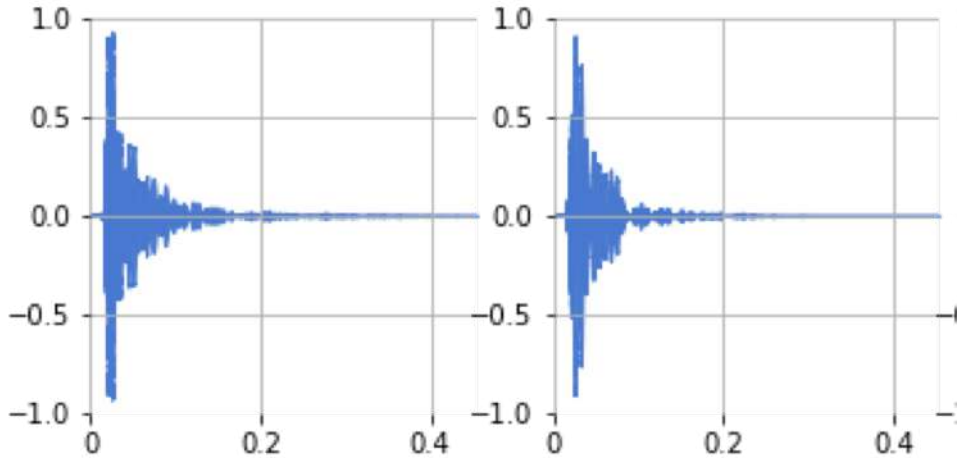


Acoustic Feature

- **Low Level Feature:** Zero Crossing Rate, Bandwidth, Spectral Coeficient, Energy, RMSE
- **High Level Feature:** MFCC, Spectogram, Chroma Feature

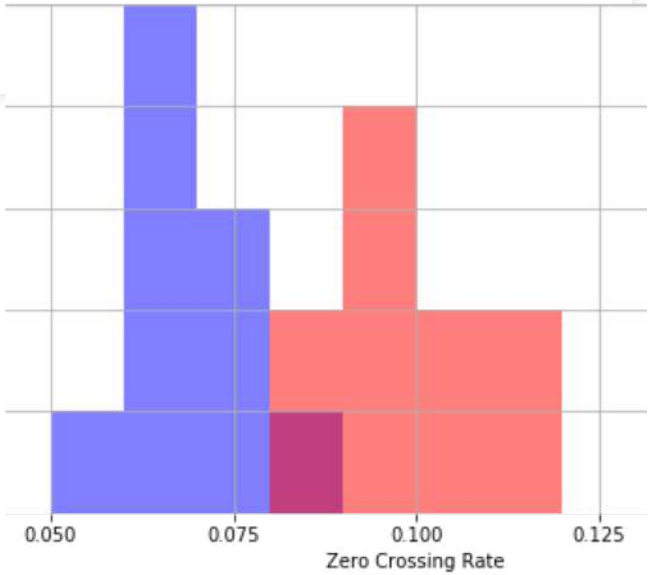
Low Level: Basic Feature Extraction

```
kick_signals = [
        librosa.load(p)[0] for p in Path().glob('kick*.mp3')
]
snare_signals = [
        librosa.load(p)[0] for p in Path().glob('snare_*.mp3')
]
len(kick_signals)
plt.figure(figsize=(15, 6))
for i, x in enumerate(kick_signals):
        plt.subplot(2, 5, i+1)
        librosa.display.waveplot(x[:10000])
        plt.ylim(-1, 1)
```

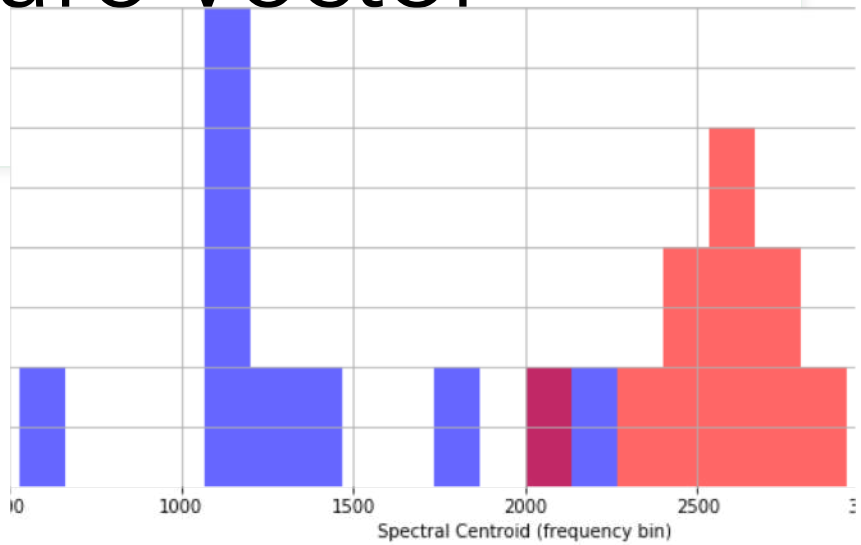


Low Level: Constructing Feature Vector

```
def extract_features(signal):
        return [
                librosa.feature.zero_crossing_rate(signal)[0, 0],
                librosa.feature.spectral_centroid(signal)[0, 0]
        ]
kick_f = numpy.array([extract_features(x) for x in kick_signals])
snare_f = numpy.array([extract_features(x) for x in snare_signals])
plt.figure(figsize=(14, 5))
plt.hist(kick_features[:,0], color='b', range=(0, 0.2), alpha=0.5, bins=20)
plt.hist(snare_features[:,0], color='r', range=(0, 0.2), alpha=0.5, bins=20)
plt.legend(('kicks', 'snares'))
plt.xlabel('Zero Crossing Rate') plt.ylabel('Count')
```



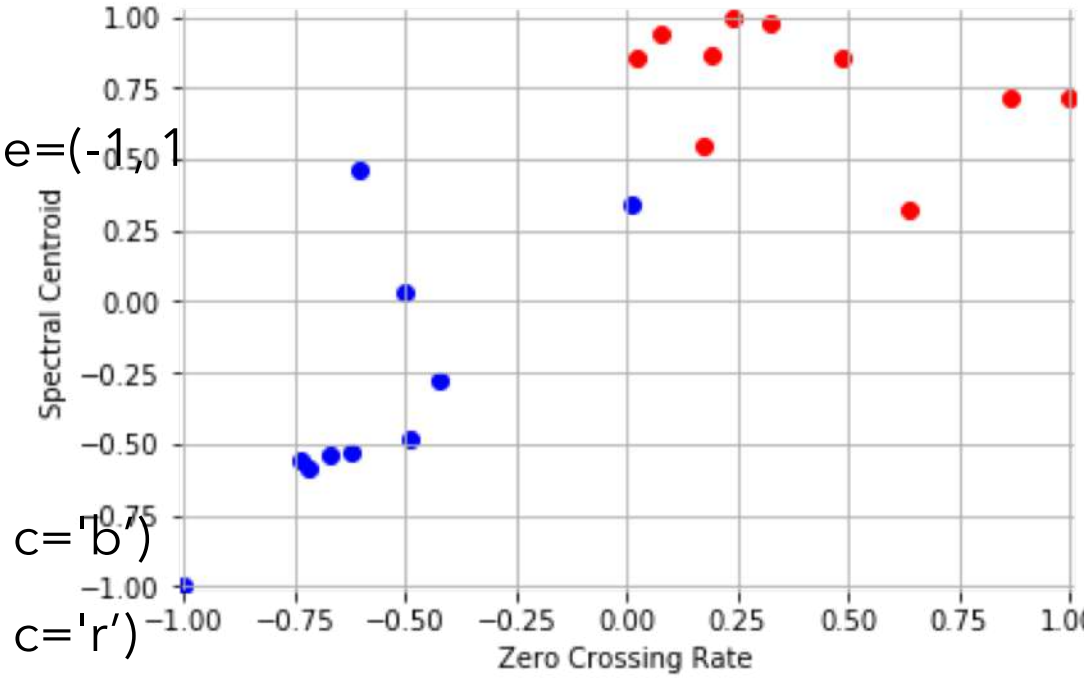
Low Level: Constructing Feature Vector



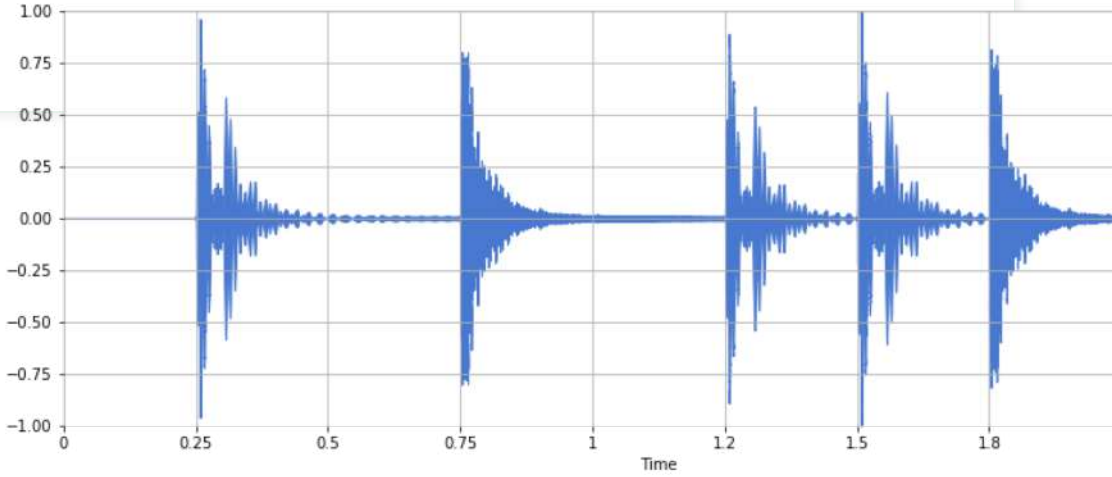

```
def extract_features(signal):
        return [
                librosa.feature.zero_crossing_rate(signal)[0, 0],
                librosa.feature.spectral_centroid(signal)[0, 0]
        ]
kick_f = numpy.array([extract_features(x) for x in kick_signals])
snare_f = numpy.array([extract_features(x) for x in snare_signals])
plt.figure(figsize=(14, 5))
plt.hist(kick_features[:,1], color='b', range=(0, 4000), bins=30, alpha=0.6)
plt.hist(snare_features[:,1], color='r', range=(0, 4000), bins=30, alpha=0.6)
plt.legend(('kicks', 'snares'))
plt.xlabel('Spectral Centroid (frequency bin)') plt.ylabel('Count')
```

Low Level: Feature Scaling

```
feature_table = numpy.vstack((kick_features, snare_features))
print(feature_table.shape)
scaler = sklearn.preprocessing.MinMaxScaler(feature_range=(-1,1
training_features = scaler.fit_transform(feature_table)
print(training_features.min(axis=0))
print(training_features.max(axis=0))
plt.scatter(training_features[:10,0], training_features[:10,1], c='b')
plt.scatter(training_features[10:,0], training_features[10:,1], c='r')
plt.xlabel('Zero Crossing Rate') plt.ylabel('Spectral Centroid')
```



Low Level: Energy and RMSE



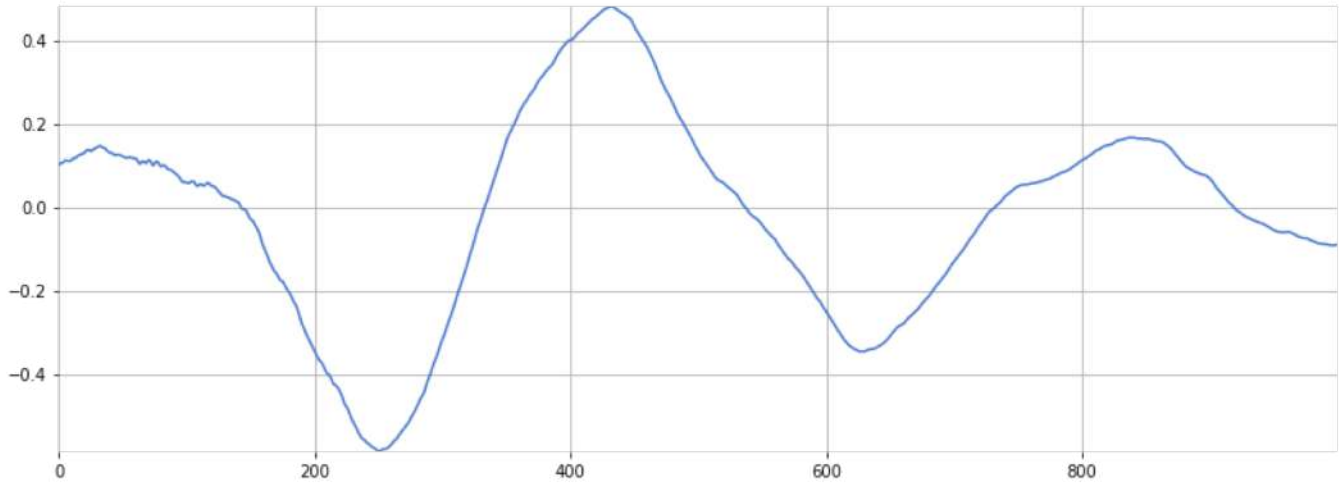

```
x, sr = librosa.load(simple_loop.wav')
librosa.get_duration(x, sr)
hop_length = 256
frame_length = 512
energy = numpy.array(
        [ sum(abs(x[i:i+frame_length]**2))
        for i in range(0, len(x), hop_length) ]
)
rmse = librosa.feature.rmse(x, frame_length=frame_length, hop_length=hop_length, center=True)
frames = range(len(energy)) t = librosa.frames_to_time(frames, sr=sr, hop_length=hop_length)
```

$$Energy = \sum |x(n)|^2 \qquad RMSE = \frac{1}{n}\sqrt{\sum |x(n)|^2}$$

Low Level: Zero Crossing Rate

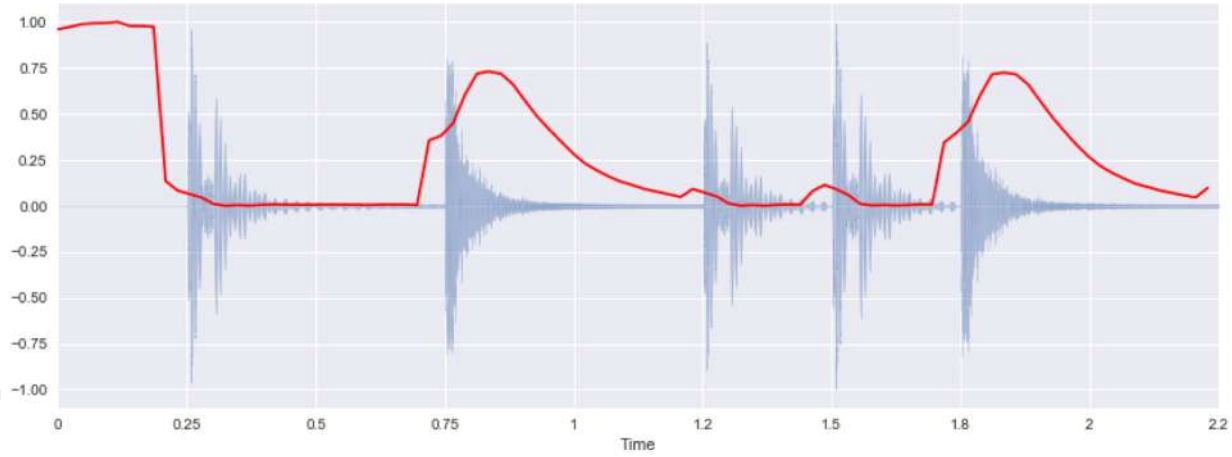
```
x, sr = librosa.load('audio/simple_loop.wav')
plt.figure(figsize=(14, 5))
librosa.display.waveplot(x, sr=sr)
```

```
n0 = 6500
n1 = 7500
plt.figure(figsize=(14, 5))
plt.plot(x[n0:n1])
```



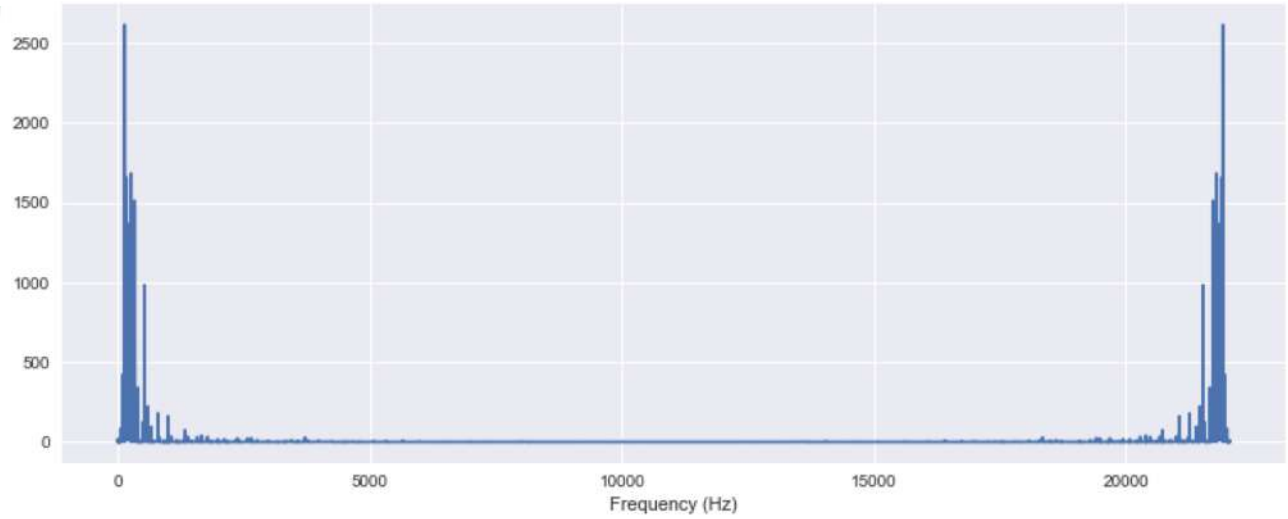
Low Level: Spectral Centroid

```
def normalize(x, axis=0):
        return sklearn.preprocessing.minmax_scale(x, axis=axis)


x, sr = librosa.load('audio/simple_loop.wav')
ipd.Audio(x, rate=sr)
spectral_centroids = librosa.feature.spectral_centroid(x, sr=sr)[0]
spectral_centroids.shape
frames = range(len(spectral_centroids))
 t = librosa.frames_to_time(frames)
librosa.display.waveplot(x, sr=sr, alpha=0.4)
plt.plot(t, normalize(spectral_centroids), color='r
```

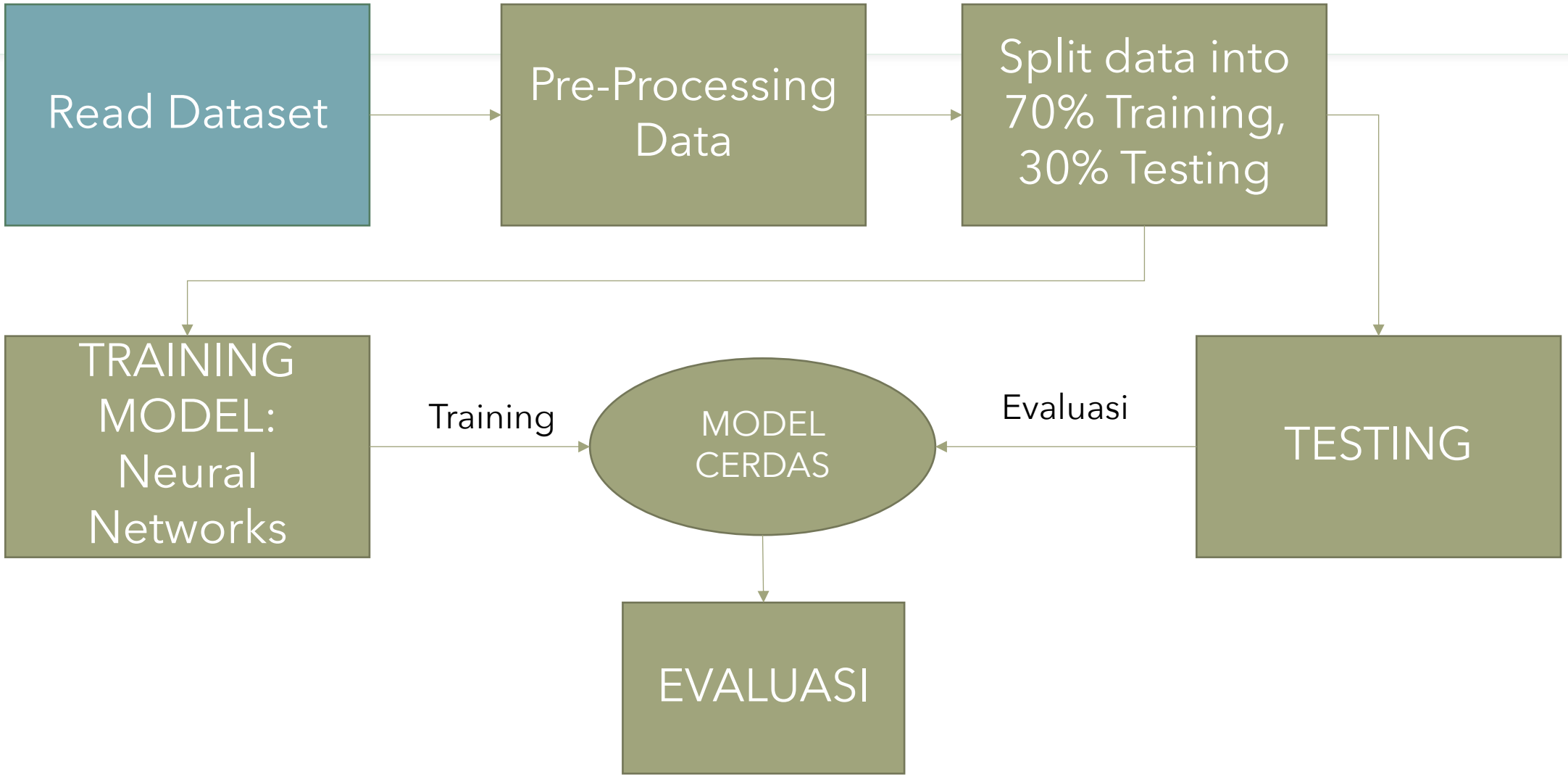


Low Level: Fourier Transform

```
X = scipy.fft(x)
X_mag = numpy.absolute(X)
f = numpy.linspace(0, sr, len(X_mag))


plt.figure(figsize=(13, 5))
plt.plot(f, X_mag)
plt.xlabel('Frequency (Hz)')


#ZOOM IN
plt.figure(figsize=(13, 5))
plt.plot(f[:5000], X_mag[:5000])
plt.xlabel('Frequency (Hz)')
```



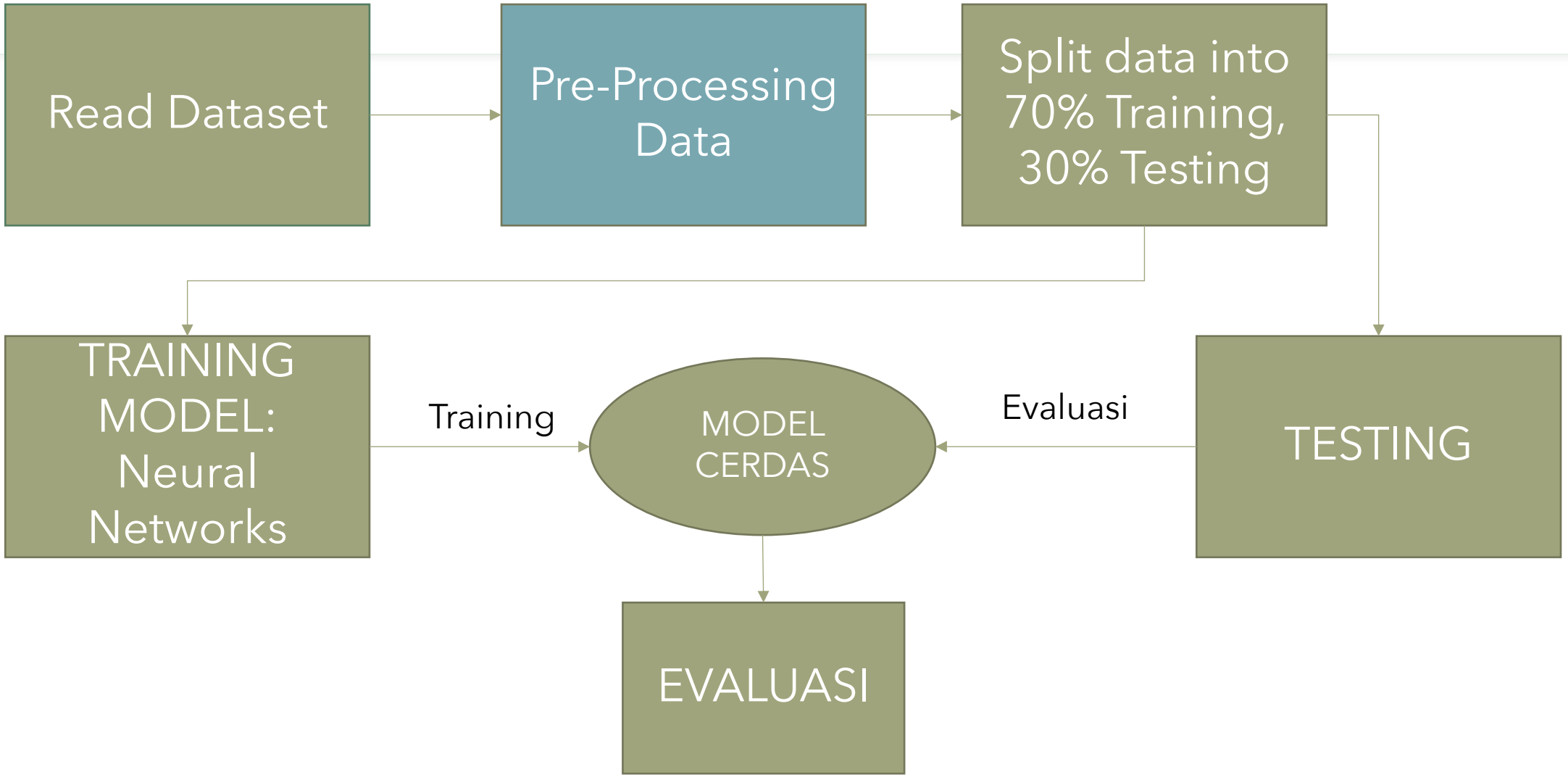
Flow Classification: Voice Gender



Method: Dataset Collection

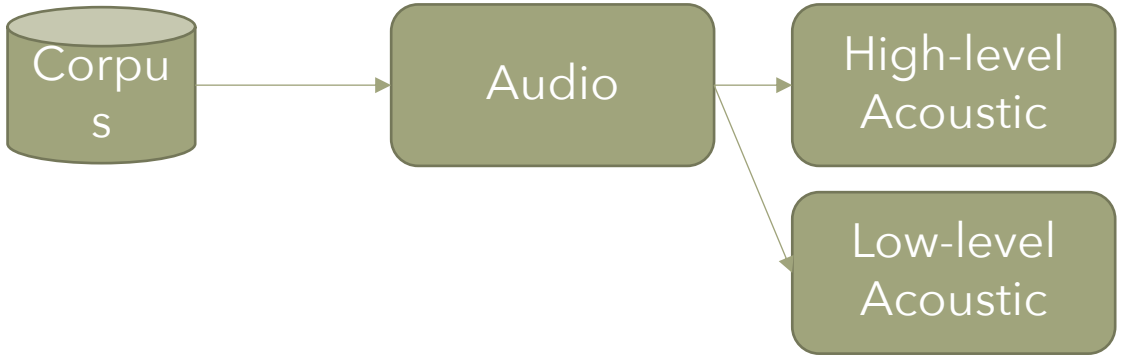
- Dataset diambil dari Talkshow Indonesia Lawyer Club.
- Segmentasi oleh 2 orang annotators. Segmentasi dilakukan per 1 kalimat di speech. Setelah dipisahkan, dicoba diberikan label emosi sesuai dengan konsep Emosi Valence-Arousal. Persetujuan diukur oleh Kappa Score untuk level agreement.
- Setiap segment akan dipisahkan audio dan text.

Flow Classification: Voice Gender



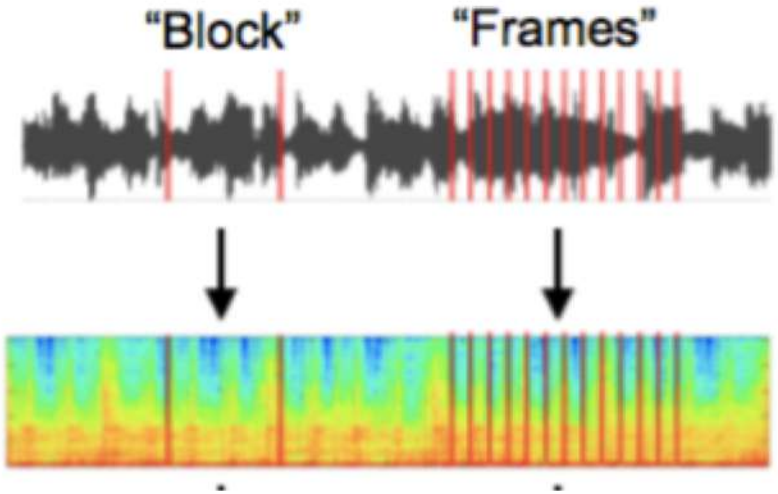
Method: Features

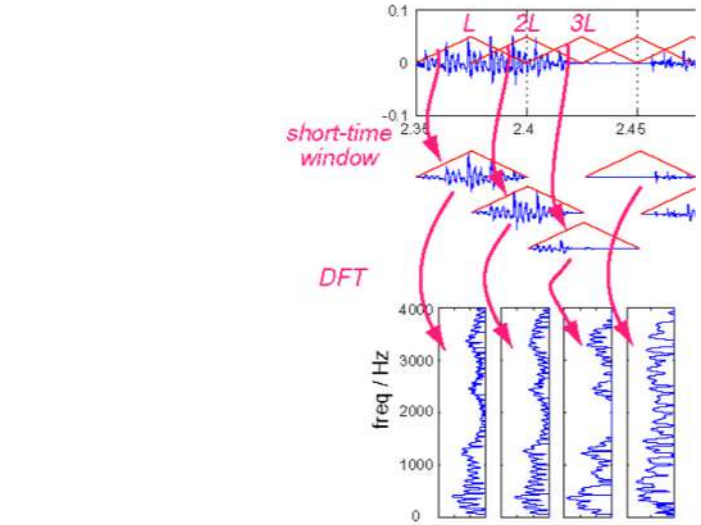
- Extract from High-level Acoustic and Low-level Acoustic Features



Ekstraksi Mel-Spectrogram

1. Bagi setiap 3 detik *track* lagu kedalam *overlapping frame,* setiap durasi 25ms. Umumnya, dari satu frame ke frame lainnya digunakan pergeseran 5ms.
2. Berikan *Fourier transform* pada setiap *frame* dan tumpuk dalam sumbu frekuensi dan waktu
3. Berikan *Triangular Filter Bank* untuk mendapatkan respon frekuensi setiap *frame* pada *mel-scale.*
4. Untuk mendapatkan *mel-spectrogram*, berikan logaritma pada intensitas spectral.
5. Setiap 3 detik lagu direpresentasikan sebagai 600 x 128 tensor
6. Fitur disediakan oleh library Librosa



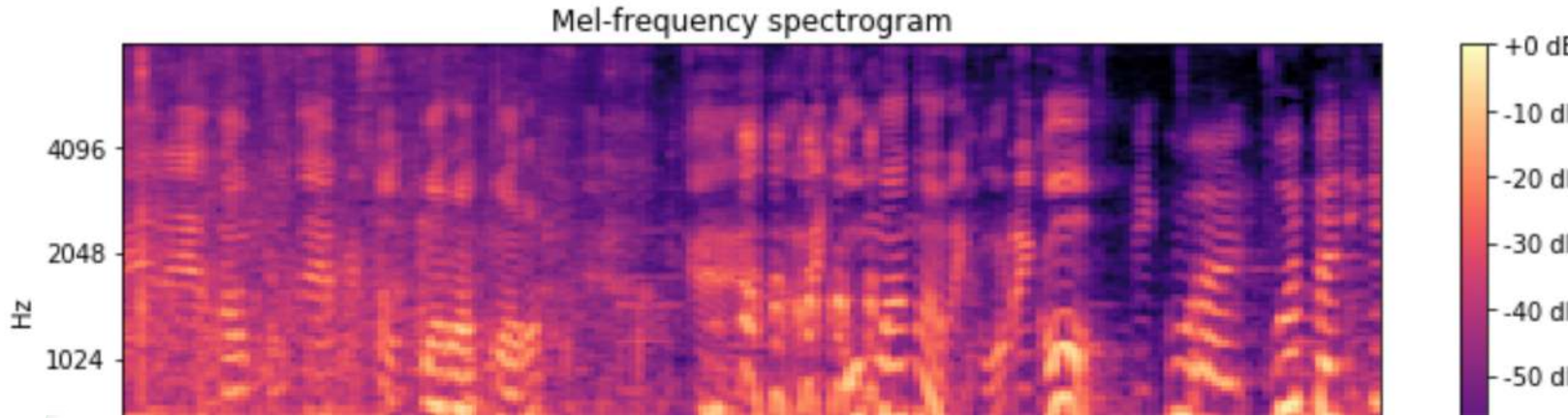


Read, preprocessing

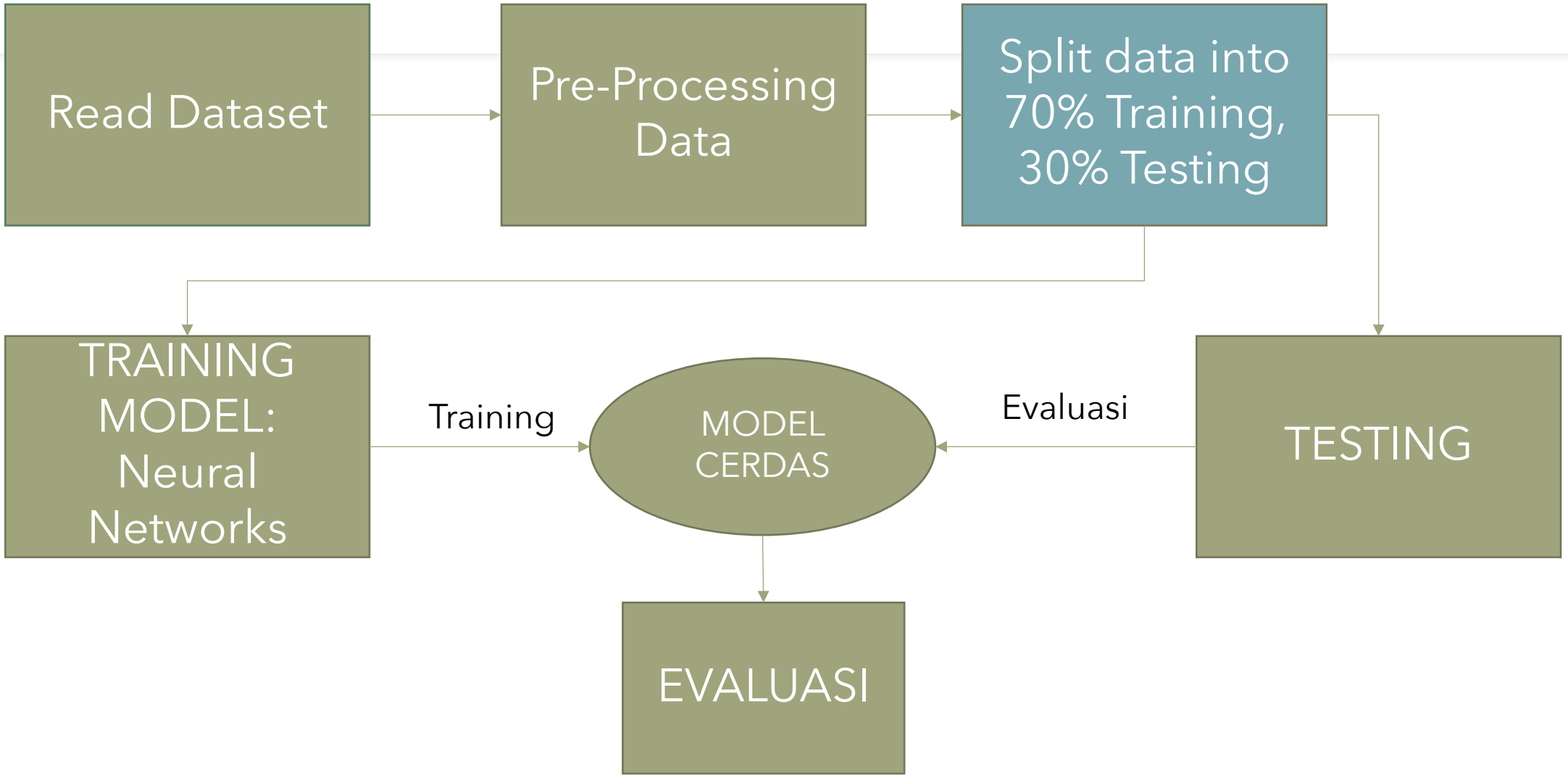
```
dirs = os.listdir('/content/drive/My Drive/DATASET/spectro
label = 0
im_arr = []
lb_arr = []
X = []
y = []
for i in dirs: #loop all directory
    count = 0
    for pic in glob.glob('/content/drive/My Drive/DATASET/
        im = cv2.imread(pic) #open image
        im = cv2.resize(im,(70,70))
        im = np.array(im) #change into array
        count = count + 1
        X.append(im)
        y.append(label)
        if(count == 3): #SAmple
            im_arr.append({str(i):im})
    print("Jumlah "+str(i)+" : "+str(count))
    label = label + 1
    lb_arr.append(i)
X = np.array(X)
y = np.array(y);
```

Read, preprocessing

```
import matplotlib.pyplot as plt
import librosa

y, sr = librosa.load("/content/drive/My Drive/DATASET/chunk2.wav")
S = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=128, fmax=8000)
plt.figure(figsize=(10, 4))
S_dB = librosa.power_to_db(S, ref=np.max)
librosa.display.specshow(S_dB, x_axis='time', y_axis='mel', sr=sr,fmax=8000)
plt.colorbar(format='%+2.0f dB')
plt.title('Mel-frequency spectrogram')
plt.tight_layout()
plt.show()
```



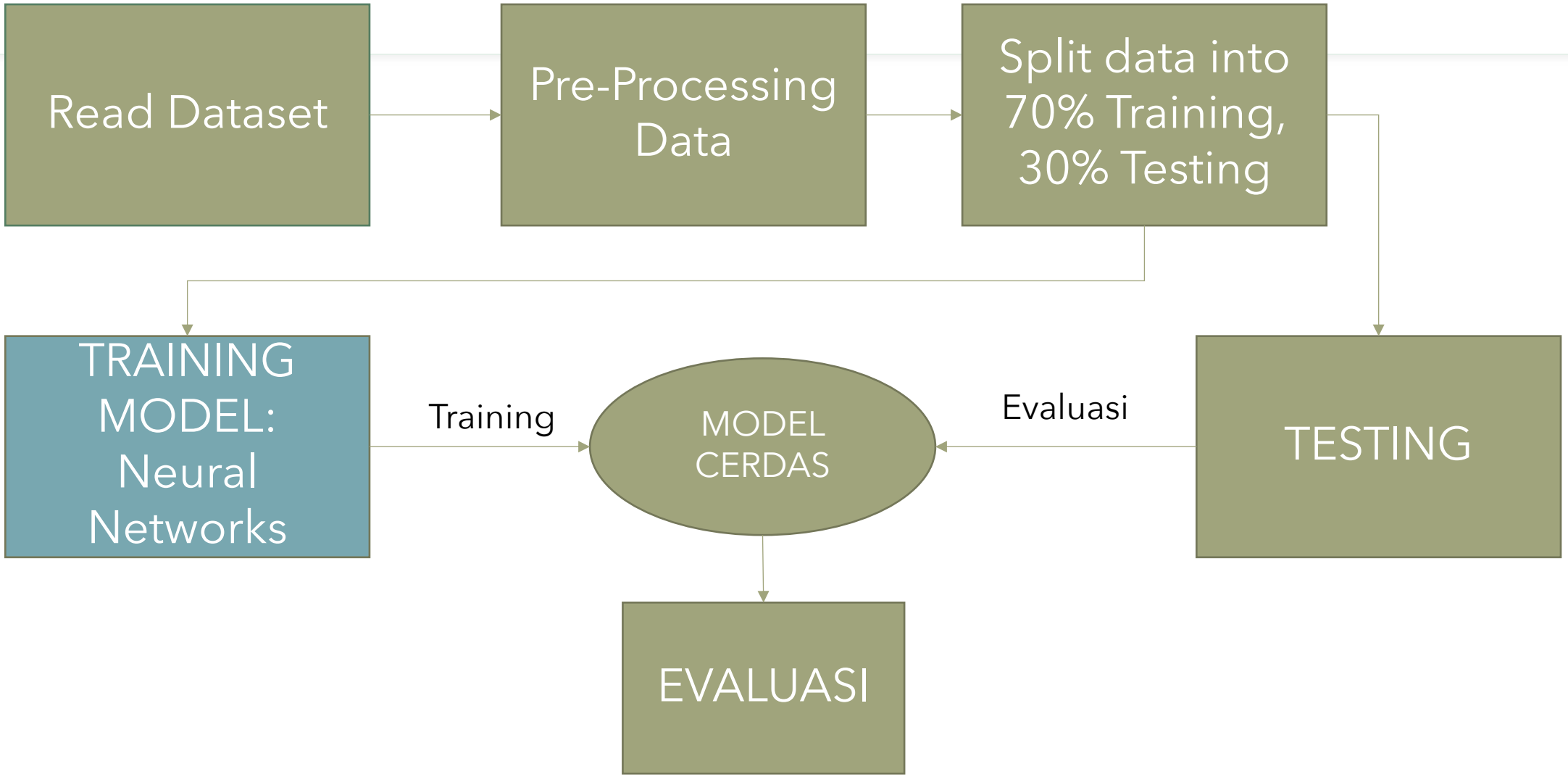
Flow Classification: Voice Gender



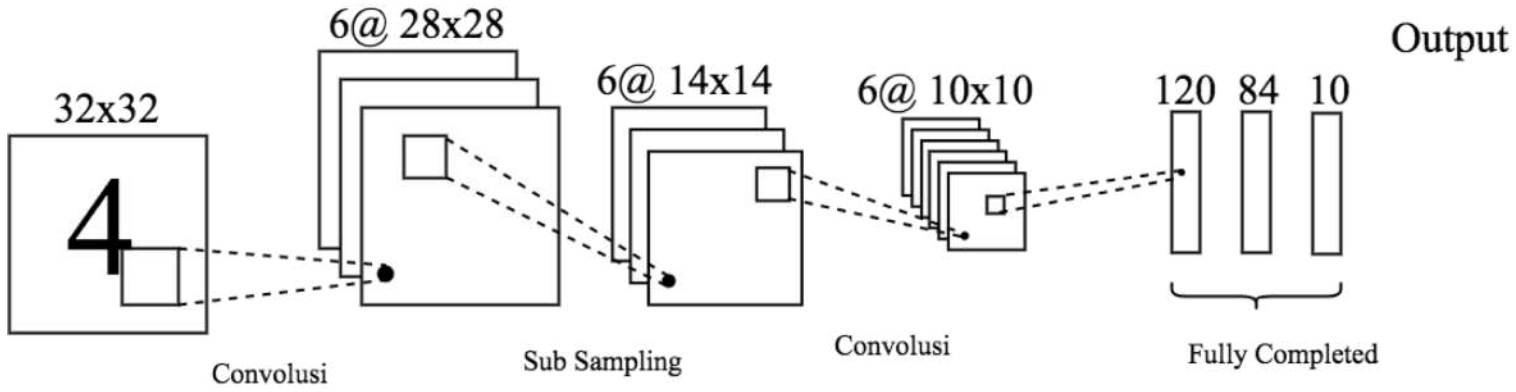
Split Data

```
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

X_train, X_test, y_train, y_test = train_test_split(X, y,

X_train = X_train.astype('float32') #set x_train data typ
X_test = X_test.astype('float32') #set x_test data type a
X_train /= 255 #change x_train value between 0 - 1
X_test /= 255 #change x_test value between 0 - 1
y_train = to_categorical(y_train, 5) #change label to bin
y_test = to_categorical(y_test, 5) #change label to binar
```

Flow Classification: Voice Gender



Convolutional Neural Networks



$$S(x, W) = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} W_{(i-m, j-n)}$$

$$Softmax(z_i) = \frac{\exp(z_i)}{\sum_{i=1}^{n} \exp(z_j)}$$

Training Neural Networks

```
# ARSITEKTUR
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropo
model = Sequential() #model = sequential
model.add(Conv2D(32, kernel_size=(3, 3),activation='
model.add(MaxPooling2D(pool_size=(2,2))) #max poolin
model.add(Conv2D(32, (3, 3), activation='relu')) #la
model.add(MaxPooling2D(pool_size=(2,2))) #max poolin
model.add(Dropout(0.25)) #delete neuron randomly whi
model.add(Flatten()) #make layer flatten
model.add(Dense(128, activation='relu')) #fully conn
model.add(Dropout(0.5)) #delete neuron randomly and
model.add(Dense(5, activation='softmax')) #softmax w
```