

TUGAS PERTEMUAN 3

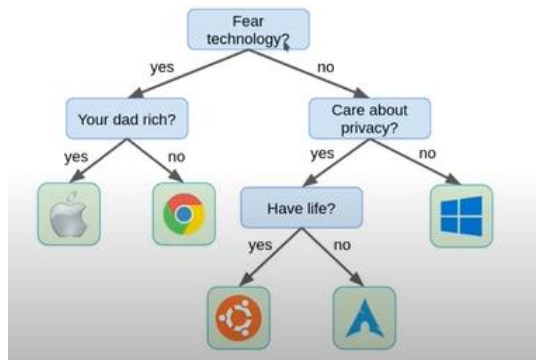
NAMA : MAYA NURROHMAH

NPM : 41155050210019

KELAS : INF_A1

1. Lakukan praktik dari https://youtu.be/5wwXKtLkyqs?si=fn88eveu_qbCC6b3 , buat screenshot dengan nama kalian pada coding, kumpulkan dalam bentuk pdf, dari kegiatan ini:

- 1) Pengenalan komponen Decision Tree: root, node, leaf



Gambar diatas merupakan semacam struktur yang dikenal dengan sebagai tree atau lebih tepatnya binary tree karena tiap notnya akan memiliki dua cabang.

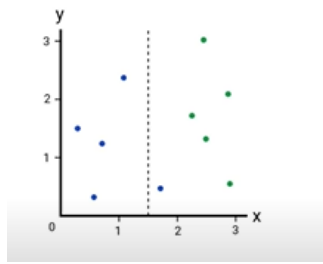
Gambar diatas juga bisa disebut dengan struktur pohon terbalik, disini setidaknya ada 3 komponen yang perlu diketahui :

- Root node atau bisa atau akar
Karena struktur pohonnya terbalik maka root akan menempati posisi paling atas (fear technology merupakan root node nya)
- Leaf node (daun)
Adalah sejumlah not yang posisinya diujung bawah, pada gambar diatas kita memilki 5 buah leaf, setiap leaf dari decision tree akan merepresentasikan prediksi kelas yang akan dihasilkan oleh struktur tree tersebut
- Internal node
Sejumlah node yang bearada diantara root dan leaf, pada gambar diatas node nya ada 3 (you dad rich?, Care about privacy, Have life)

- 2) Pengenalan Gini Impurity

Gini Impurity memiliki jangkauan nilai antara nol dan satu, dimana nilai nol mengindikasikan nilai murni yang sempurna sedangkan nilai satu mengindikasikan nilai impur atau paling tidak murni

Gini Impurity



Dalam kasus ini terdapat 10 data point yang terbagi dalam 2 kelas ada kelas biru dan ada kelas hijau, selanjutnya pisahkan ke dalam dua bagian, sehingga dihasilkan dua buah ruas, sebut saja ruas kiri dan ruas kanan, garis putus putus berperan sebagai splitter atau pemisahannya

Cara melakukan pengukuran pada ruas kiri dan ruas kanan

Ruas Kiri:

$$\begin{aligned} G &= 1 - \sum_i P_i^2 \\ &= 1 - P(\text{biru})^2 \\ &= 1 - \left(\frac{4}{4}\right)^2 = 0 \end{aligned}$$

Ruas Kanan:

$$\begin{aligned} G &= 1 - \sum_i P_i^2 \\ &= 1 - (P(\text{biru})^2 + P(\text{hijau})^2) \\ &= 1 - \left(\left(\frac{1}{6}\right)^2 + \left(\frac{5}{6}\right)^2\right) = 0.278 \end{aligned}$$

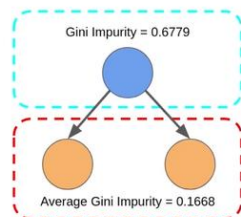
Cara Kalkulasinya

Average Gini Impurity:

$$\begin{aligned} G &= \frac{4}{4+6} \times 0 + \frac{6}{4+6} \times 0.278 \\ &= 0.1668 \end{aligned}$$

3) Pengenalan Information Gain

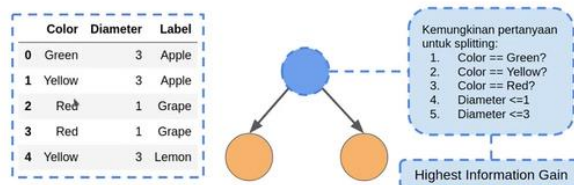
Information Gain



$$\text{Information Gain} = 0.6779 - 0.1668 = 0.51$$

Information gain bisa kita peroleh dengan menselisihkan nilai gini impurity sebelum proses splitting dengan nilai rata rata nilai gini impurity setelah proses splitting

4) Membangun Decision Tree



$$\begin{aligned} G &= 1 - (P(\text{apple})^2 + P(\text{grape})^2 + P(\text{lemon})^2) \\ &= 1 - \left(\left(\frac{2}{5}\right)^2 + \left(\frac{2}{5}\right)^2 + \left(\frac{1}{5}\right)^2\right) \\ &= 0.63 \end{aligned}$$

5) Persiapan dataset: Iris Dataset

```
[3]: from sklearn.datasets import load_iris

X, y = load_iris(return_X_y=True)

print(f'Dimensi Feature: {X.shape}')
print(f'Class: {set(y)}')
```

Dimensi Feature: (150, 4)
Class: {np.int64(0), np.int64(1), np.int64(2)}

```
[4]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.3,
                                                    random_state=0)
```

```
[5]: nama = "Maya Nurrohmah"

print(f>Nama : {nama}")
```

Nama : Maya Nurrohmah

6) Training model Decision Tree Classifier

```
[6]: from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier(max_depth=4)

model.fit(X_train, y_train)
```

```
[6]: DecisionTreeClassifier
DecisionTreeClassifier(max_depth=4)
```

```
[7]: nama = "Maya Nurrohmah"

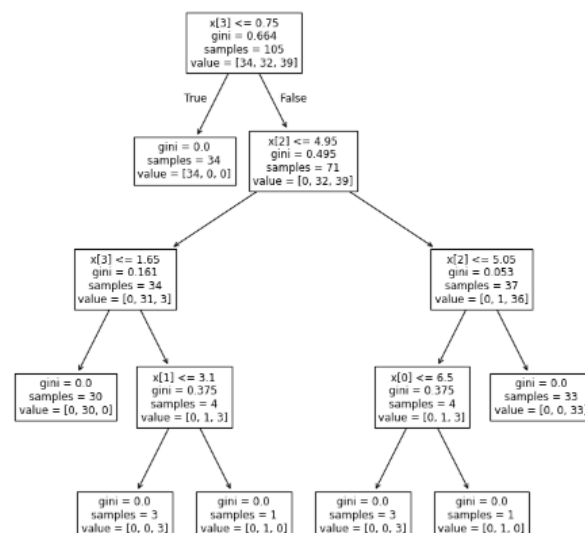
print(f>Nama : {nama}")
```

Nama : Maya Nurrohmah

7) Visualisasi model Decision Tree

```
[9]: import matplotlib.pyplot as plt
from sklearn import tree

plt.rcParams['figure.dpi'] = 85
plt.subplots(figsize=(10, 10))
tree.plot_tree(model, fontsize=10)
plt.show()
```



8) Evaluasi model Decision Tree

```
[12]: from sklearn.metrics import classification_report

y_pred = model.predict(X_test)

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

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	1.00	0.94	0.97	18
2	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45
weighted avg	0.98	0.98	0.98	45

```
[13]: nama = "Maya Nurrohmah"

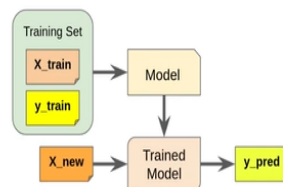
print(f"Nama : {nama}")
```

Nama : Maya Nurrohmah

2. Lakukan praktik dari https://youtu.be/yKovaQ6tyV8?si=HnHG6kcoCsDwvo_0 , buat screenshot dengan nama kalian pada coding, kumpulkan dalam bentuk pdf, dari kegiatan ini:

- 1) Proses training model Machine Learning secara umum

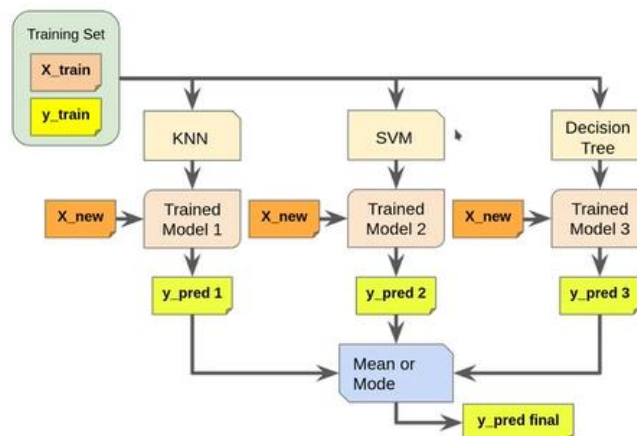
General ML Model Training



Training set digunakan untuk melakukan training model machine learning, dan model yang sudah ditraining ini akan dikenal dengan istilah trained model, trained model ini akan digunakan untuk melakukan prediksi terhadap sekumpulan untuk nilai yang baru

- 2) Pengenalan Ensemble Learning

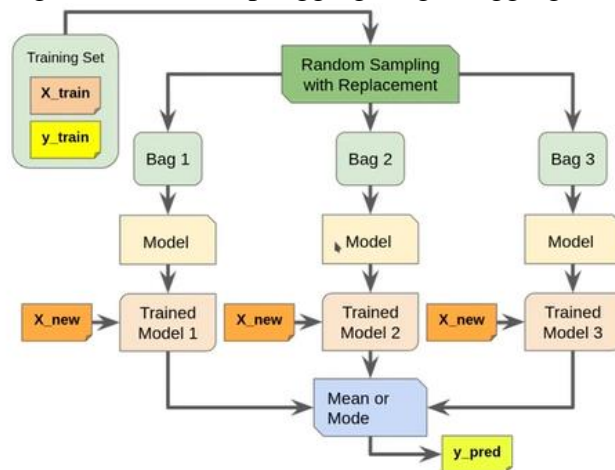
Ensemble learning merupakan suatu Teknik yang dikenal dalam machine learning dimana kita menggabungkan beberapa model untuk melakukan prediksi.



Pada gambar diatas kita memadukan sejumlah model machine learning yang berbeda maka seringkali juga dikenal sebagai heterogeneous ensemble learning atau model ensemble learning yang heterogen.

Sedangkan untuk kasus dimana kita memadukan sejumlah model machine learning yang sejenis akan dikenal dengan istilah homogeneous ensemble learning atau model ensemble learning yang homogen.

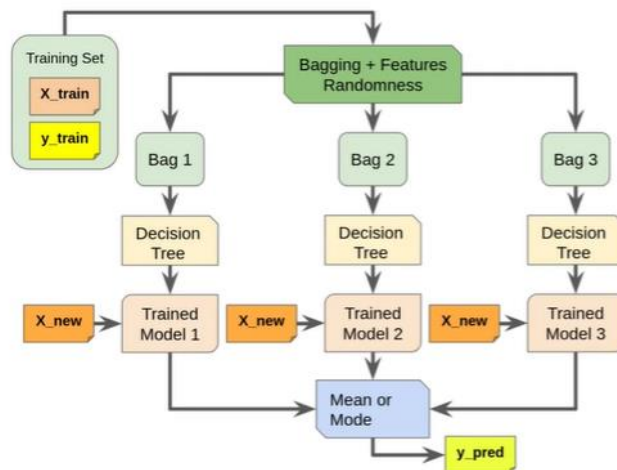
3) Pengenalan Bootstrap Aggregating | Bagging



Pada gambar diatas merupakan contoh homogeneous ensemble learning. Karena pada kasus diatas menerapkan model yg sejenis maka melakukan trained terhadap setiap model menggunakan trained set yang sama bisa dibayangkan, karena akan menghasilkan trained model yang sama persis.

Nah disini kita akan berkenalan dengan suatu Teknik yang dikenal dengan istilah Bootstrap Aggregating atau disingkat dengan (Bagging). Mekanisme Bagging pada dasarnya menerapkan proses random sampling with replacement terhadap trained set yang kita miliki dan akan menghasilkan beberapa trained set baru sejumlah model yang akan kita trained. Trained set baru yang dihasilkan dari random sampling with replacement dikenal dengan istilah Bag.

4) Pengenalan Random Forest | Hutan Acak



Random Forest merupakan implementasi dari homogeneous ensemble learning yang menerapkan decision tree.

Random Forest terbentuk dari sekumpulan decision tree atau pohon keputusan. Selain bagging random forest juga menerapkan features randomness dimana untuk setiap bag yang dihasilkan akan mengadopsi sejumlah features yang dipilih secara acak dari trained set sumbernya.

5) Persiapan dataset | Iris Flower Dataset

```
[2]: from sklearn.datasets import load_iris

X, y = load_iris(return_X_y=True)

print(f'Dimensi Feature: {X.shape}')
print(f'Class: {set(y)}')
```

Dimensi Feature: (150, 4)
Class: {np.int64(0), np.int64(1), np.int64(2)}

```
[3]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.3,
                                                    random_state=0)
```

```
[4]: nama = "Maya Nurrohmah"

print(f>Nama : {nama}")
```

Nama : Maya Nurrohmah

6) Implementasi Random Forest Classifier dengan Scikit Learn

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100,
                              random_state=0)

model.fit(X_train, y_train)
```

RandomForestClassifier

RandomForestClassifier(random_state=0)

```
nama = "Maya Nurrohmah"

print(f>Nama : {nama}")
```

Nama : Maya Nurrohmah

7) Evaluasi model dengan Classification Report

```
[6]: nama = "Maya Nurrohmah"

print(f>Nama : {nama}")
```

Nama : Maya Nurrohmah

```
[7]: from sklearn.metrics import classification_report

y_pred = model.predict(X_test)

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

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	1.00	0.94	0.97	18
2	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45
weighted avg	0.98	0.98	0.98	45