Portofolio Data Science

# Breast Cancer Wisconsin Classification Using Decision Tree Algorithm

DSF 35.0 - Data Science

By : Septa Syaidin Aksa

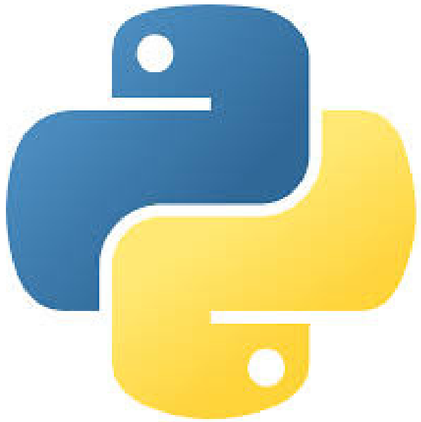
# Breast Cancer Dataset Classification

Breast Cancer Wisconsin (Diagnostic) Dataset adalah kumpulan data yang sering digunakan untuk tugas Machine Learning. Dataset ini juga tersedia di pustaka Skicit-learn, yang dirancang untuk memprediksi apakah suatu tumor bersifat jinak (benign) atau ganas (malignant). Dataset ini terdiri dari 569 sampel dengan 30 fitur numerik yang mencakup karakteristik massa jaringan tumor, seperti ukuran, keliling, tekstur, dan kerapian tepi sel tumor.

# Decision Tree

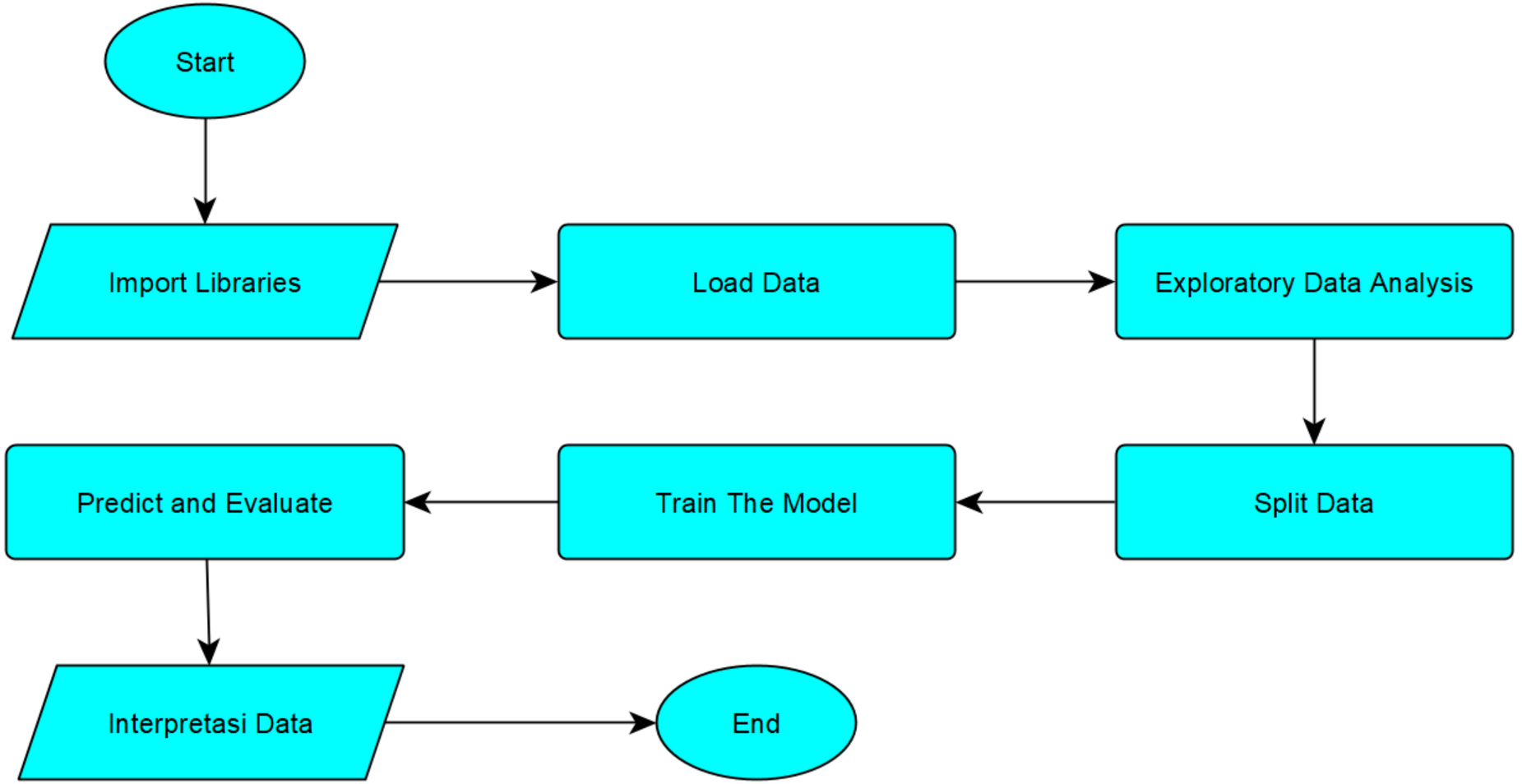
Decision Tree dipilih karena memiliki berbagai keunggulan yang relevan dengan kebutuhan diagnostik medis. Salah satu alasan utamanya adalah kemampuannya memberikan hasil yang mudah diinterpretasikan. Selain itu, Decision Tree dapat menangani data yang kompleks dengan banyak fitur, seperti dataset Breast Cancer ini yang memiliki 30 fitur numerik terkait karakteristik tumor.

# Tools Used






# Flowchart


Start
Import Libraries
Load Data
Exploratory Data Analysis
Split Data
Train The Model
Predict and Evaluate
Interpretasi Data
End

# Import Library

Berikut adalah beberapa daftar library Python yang digunakan untuk membangun model Machine Learning Breast Cancer Classification

## 1. Import Library

0s

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
```

1s

```
[2] import pandas as pd

    data = pd.read_csv('/content/breast-cancer.csv')
    data.head(5)
```

# Load Dataset

Breast Cancer dataset memiliki beberapa fitur, saya hanya memanggil 10 baris saja

## 2. Read Dataset

```python
import pandas as pd
from sklearn import datasets

# Memuat dataset Breast Cancer dari scikit-learn dan mengonversinya menjadi DataFrame
breast_cancer = datasets.load_breast_cancer()

X = breast_cancer.data    # inputan untuk machine learning
y = breast_cancer.target  # output yang dinginkan dari machine learning

# Mengonversi data fitur dan target menjadi DataFrame
df_X = pd.DataFrame(X, columns=breast_cancer.feature_names)
df_y = pd.Series(y, name='target')

# Gabungkan fitur dan target dalam satu DataFrame
df = pd.concat([df_X, df_y], axis=1)

df.head(10)
```

# Overview Data

Berikut adalah hasil dari Load Data yang sudah dibuat dan mencakup banyak fitur

| | id | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mean | compactness_mean | concavity_mean | concave points_mean | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 842302 | M | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | 0.27760 | 0.3001 | 0.14710 | ... |
| 1 | 842517 | M | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | 0.07864 | 0.0869 | 0.07017 | ... |
| 2 | 84300903 | M | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | 0.15990 | 0.1974 | 0.12790 | ... |
| 3 | 84348301 | M | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | 0.28390 | 0.2414 | 0.10520 | ... |
| 4 | 84358402 | M | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | 0.13280 | 0.1980 | 0.10430 | ... |

5 rows × 32 columns

| | mean radius | mean texture | mean perimeter | mean area | mean smoothness | mean compactness | mean concavity | mean concave points | mean symmetry | mean fractal dimension | ... | worst texture | worst perimeter | worst area | worst smoothness |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | 0.27760 | 0.30010 | 0.14710 | 0.2419 | 0.07871 | ... | 17.33 | 184.60 | 2019.0 | 0.1622 |
| 1 | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | 0.07864 | 0.08690 | 0.07017 | 0.1812 | 0.05667 | ... | 23.41 | 158.80 | 1956.0 | 0.1238 |
| 2 | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | 0.15990 | 0.19740 | 0.12790 | 0.2069 | 0.05999 | ... | 25.53 | 152.50 | 1709.0 | 0.1444 |
| 3 | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | 0.28390 | 0.24140 | 0.10520 | 0.2597 | 0.09744 | ... | 26.50 | 98.87 | 567.7 | 0.2098 |
| 4 | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | 0.13280 | 0.19800 | 0.10430 | 0.1809 | 0.05883 | ... | 16.67 | 152.20 | 1575.0 | 0.1374 |
| 5 | 12.45 | 15.70 | 82.57 | 477.1 | 0.12780 | 0.17000 | 0.15780 | 0.08089 | 0.2087 | 0.07613 | ... | 23.75 | 103.40 | 741.6 | 0.1791 |
| 6 | 18.25 | 19.98 | 119.60 | 1040.0 | 0.09463 | 0.10900 | 0.11270 | 0.07400 | 0.1794 | 0.05742 | ... | 27.66 | 153.20 | 1606.0 | 0.1442 |
| 7 | 13.71 | 20.83 | 90.20 | 577.9 | 0.11890 | 0.16450 | 0.09366 | 0.05985 | 0.2196 | 0.07451 | ... | 28.14 | 110.60 | 897.0 | 0.1654 |
| 8 | 13.00 | 21.82 | 87.50 | 519.8 | 0.12730 | 0.19320 | 0.18590 | 0.09353 | 0.2350 | 0.07389 | ... | 30.73 | 106.20 | 739.3 | 0.1703 |
| 9 | 12.46 | 24.04 | 83.97 | 475.9 | 0.11860 | 0.23960 | 0.22730 | 0.08543 | 0.2030 | 0.08243 | ... | 40.68 | 97.65 | 711.4 | 0.1853 |

10 rows × 31 columns

# Exploratory Data Analysis (EDA)

EDA pertama dilakukan untuk mengetahui data sampel yang kita masukan, dapat dilihat bahwa terdapat 569 data sampel dengan 30 fitur

```
0s  df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   mean radius             569 non-null    float64
 1   mean texture            569 non-null    float64
 2   mean perimeter          569 non-null    float64
 3   mean area               569 non-null    float64
 4   mean smoothness         569 non-null    float64
 5   mean compactness        569 non-null    float64
 6   mean concavity          569 non-null    float64
 7   mean concave points     569 non-null    float64
 8   mean symmetry           569 non-null    float64
 9   mean fractal dimension  569 non-null    float64
 10  radius error            569 non-null    float64
 11  texture error           569 non-null    float64
 12  perimeter error         569 non-null    float64
 13  area error              569 non-null    float64
 14  smoothness error        569 non-null    float64
 15  compactness error       569 non-null    float64
 16  concavity error         569 non-null    float64
 17  concave points error    569 non-null    float64
```

```
 16  concavity error          569 non-null    float64
 17  concave points error     569 non-null    float64
 18  symmetry error           569 non-null    float64
 19  fractal dimension error  569 non-null    float64
 20  worst radius             569 non-null    float64
 21  worst texture            569 non-null    float64
 22  worst perimeter          569 non-null    float64
 23  worst area               569 non-null    float64
 24  worst smoothness         569 non-null    float64
 25  worst compactness        569 non-null    float64
 26  worst concavity          569 non-null    float64
 27  worst concave points     569 non-null    float64
 28  worst symmetry           569 non-null    float64
 29  worst fractal dimension  569 non-null    float64
 30  target                   569 non-null    int64
dtypes: float64(30), int64(1)
memory usage: 137.9 KB
```

# Exploratory Data Analysis (EDA)

EDA kedua dilakukan untuk melihat jumlah array pada tabel, dan selanjutnya dapat mengetahui statistik deskriptif yang digunakan untuk memberikan gambaran singkat tentang suatu data

```
[8]  df['target'].unique()
```

```
array([0, 1])
```

```
[9]  df.describe()
```

| | mean radius | mean texture | mean perimeter | mean area | mean smoothness | mean compactness | mean concavity | mean concave points | mean symmetry | mean fractal dimension | ... | worst texture | worst perimeter |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | 569.000000 | ... | 569.000000 | 569.000000 |
| mean | 14.127292 | 19.289649 | 91.969033 | 654.889104 | 0.096360 | 0.104341 | 0.088799 | 0.048919 | 0.181162 | 0.062798 | ... | 25.677223 | 107.261213 |
| std | 3.524049 | 4.301036 | 24.298981 | 351.914129 | 0.014064 | 0.052813 | 0.079720 | 0.038803 | 0.027414 | 0.007060 | ... | 6.146258 | 33.602542 |
| min | 6.981000 | 9.710000 | 43.790000 | 143.500000 | 0.052630 | 0.019380 | 0.000000 | 0.000000 | 0.106000 | 0.049960 | ... | 12.020000 | 50.410000 |
| 25% | 11.700000 | 16.170000 | 75.170000 | 420.300000 | 0.086370 | 0.064920 | 0.029560 | 0.020310 | 0.161900 | 0.057700 | ... | 21.080000 | 84.110000 |
| 50% | 13.370000 | 18.840000 | 86.240000 | 551.100000 | 0.095870 | 0.092630 | 0.061540 | 0.033500 | 0.179200 | 0.061540 | ... | 25.410000 | 97.660000 |
| 75% | 15.780000 | 21.800000 | 104.100000 | 782.700000 | 0.105300 | 0.130400 | 0.130700 | 0.074000 | 0.195700 | 0.066120 | ... | 29.720000 | 125.400000 |
| max | 28.110000 | 39.280000 | 188.500000 | 2501.000000 | 0.163400 | 0.345400 | 0.426800 | 0.201200 | 0.304000 | 0.097440 | ... | 49.540000 | 251.200000 |

# Split Data and Train The Model

Split data dilakukan untuk membagi data training dan data testing, yang digunakan untuk melatih dan menguji model klasifikasi. Train model dilakukan untuk modeling pada klasifikasi dengan menggunakan algoritma decision tree

## 3. Split Data

```
[11] from sklearn.model_selection import train_test_split

     # Membagi data menjadi train dan test
     X_train, X_test, y_train, y_test = train_test_split(df_X, df_y, test_size=0.2, random_state=70)
```

## 4. Train the Model

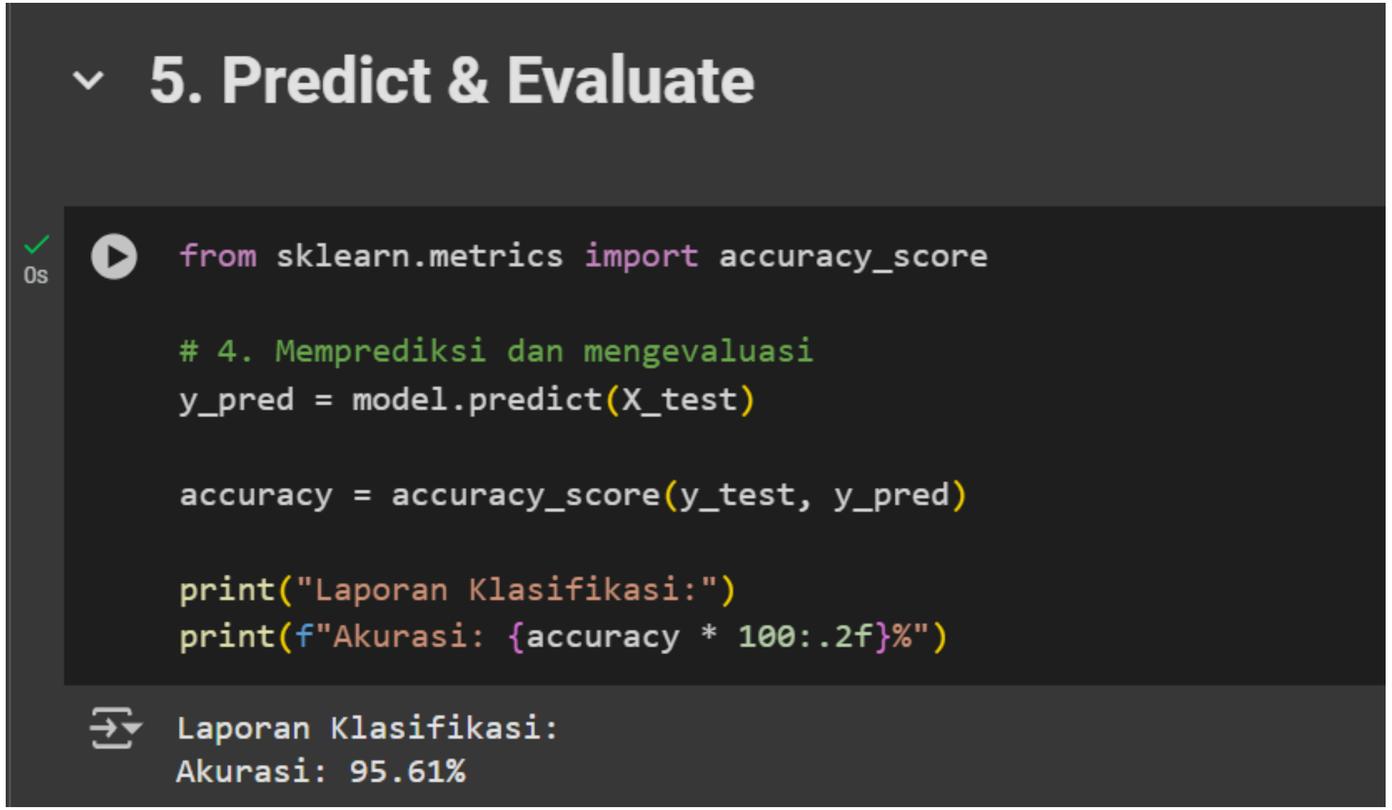
```
from sklearn.tree import DecisionTreeClassifier

# Membuat dan melatih model Decision Tree
model = DecisionTreeClassifier(random_state=70)
model.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=70)
```

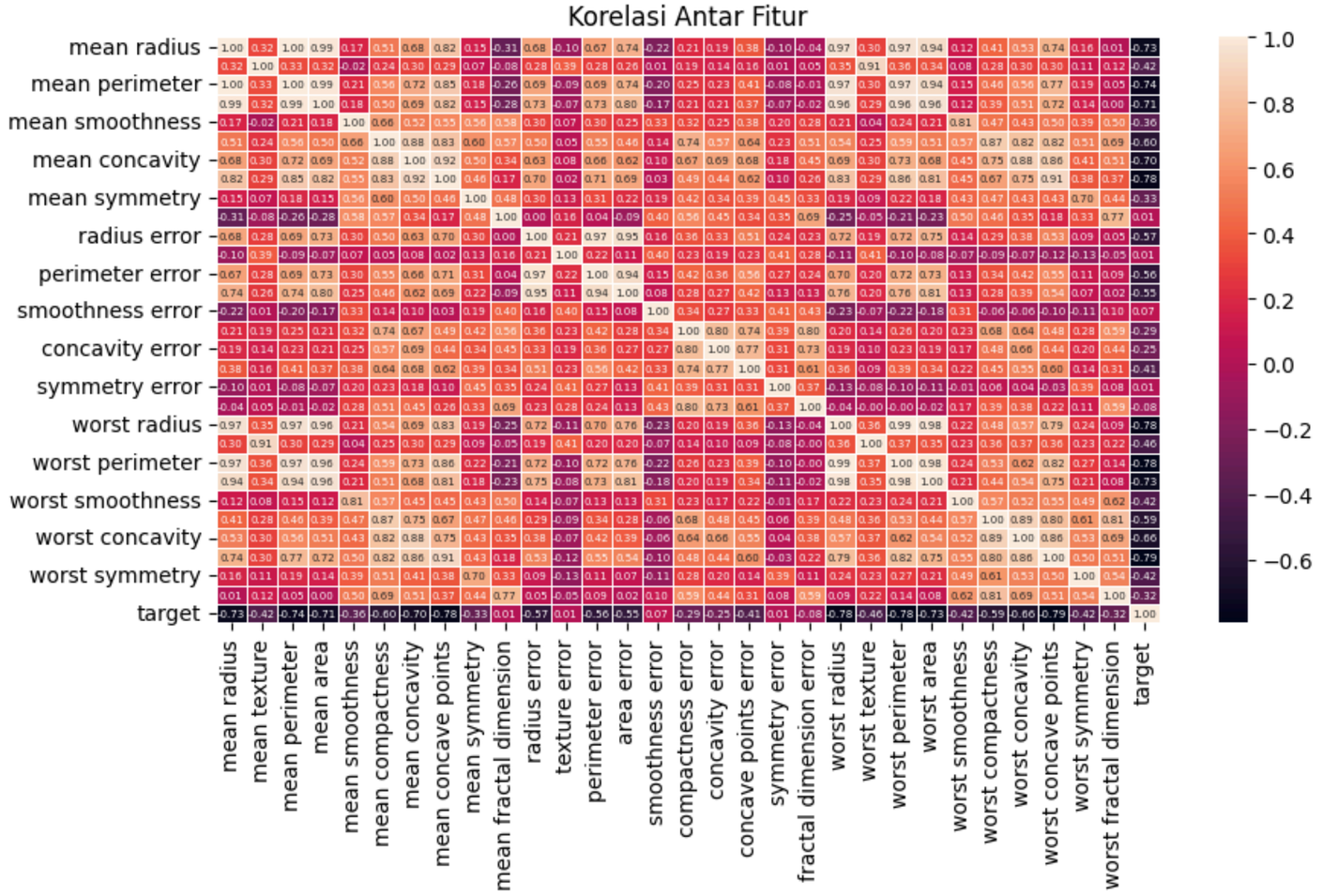
# Predict and Evaluate

Dilakukan untuk memprediksi akurasi model dengan menggunakan penerapan algoritma decision tree yang sudah di buat, dapat dilihat bahwa hasil akurasi yang dilakukan mencapai 95.61% yang dimana pemodelan yang dilakukan sudah sangat baik



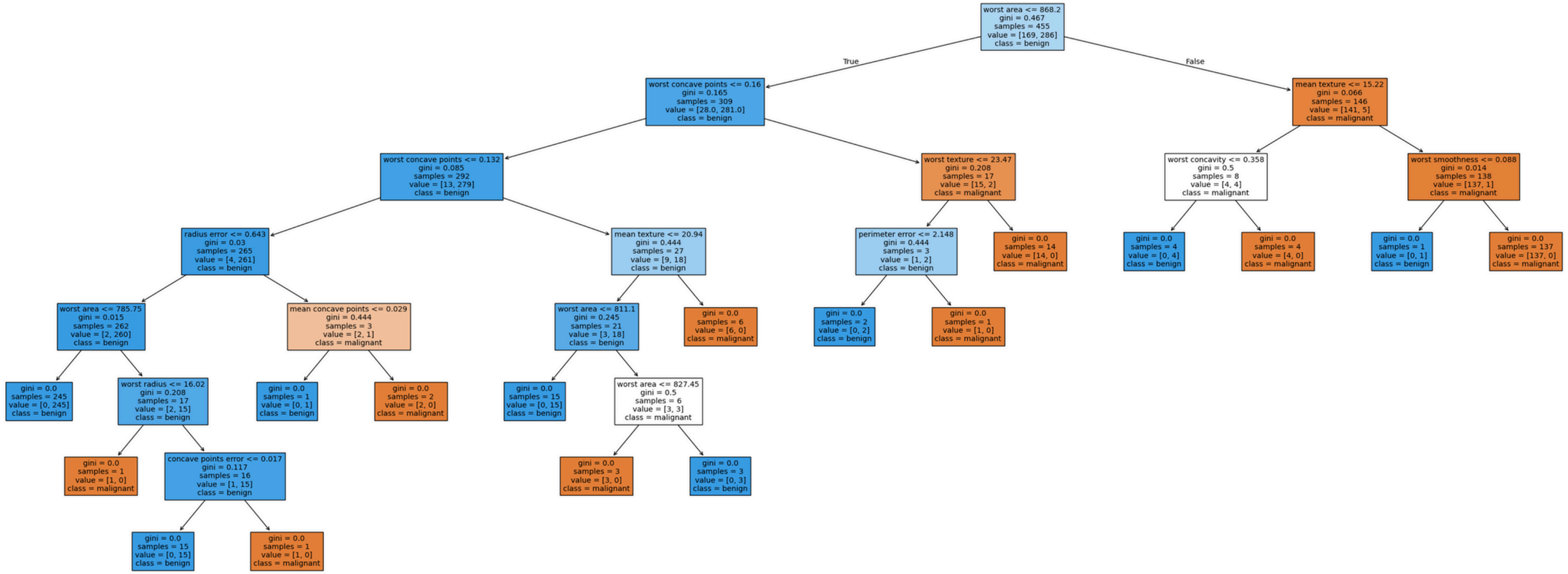
# Visualization/Visualisasi Data

Hasil Visualisasi Data Breast Cancer Classification dengan menggunakan pemodelan Corelation Matrix



# Visualization/Visualisasi Data

Hasil Visualisasi Data Breast Cancer Classification dengan Algoritma Decision Tree pada pemodelan Machine Learning



# Terima Kasih

https://github.com/Aksaa17

ig : sptaa_sa