1. Import ke Data ke google Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Load data menggunakan link Gdrive

Ambil file ID https://drive.google.com/file/d/10nE6gPlK409gmtGRxjucc-oK9jbFtiIi/view?usp=sharing

Ubah Ke URL DOWNLOAD /uc?export=download&id= https://drive.google.com/uc?export=download&id=10nE6gPlK409gmtGRxjucc-oK9jbFtiIi

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.datasets import load_breast_cancer
from sklearn.linear_model import LogisticRegression

dataset = pd.read_csv("https://drive.google.com/uc?export=download&id=10nE6gPlK409gmtGRxjucc-oK9jbFtiIi")
dataset
```

```
{"type":"dataframe","variable_name":"dataset"}
```

2. Info Data

Dataset information:

- Dataset Characteristics: Multivariate
- Attribute Characteristics: Real
- Attribute Characteristics: Classification
- Number of Instances: 569
- Number of Attributes: 32
- Missing Values: No

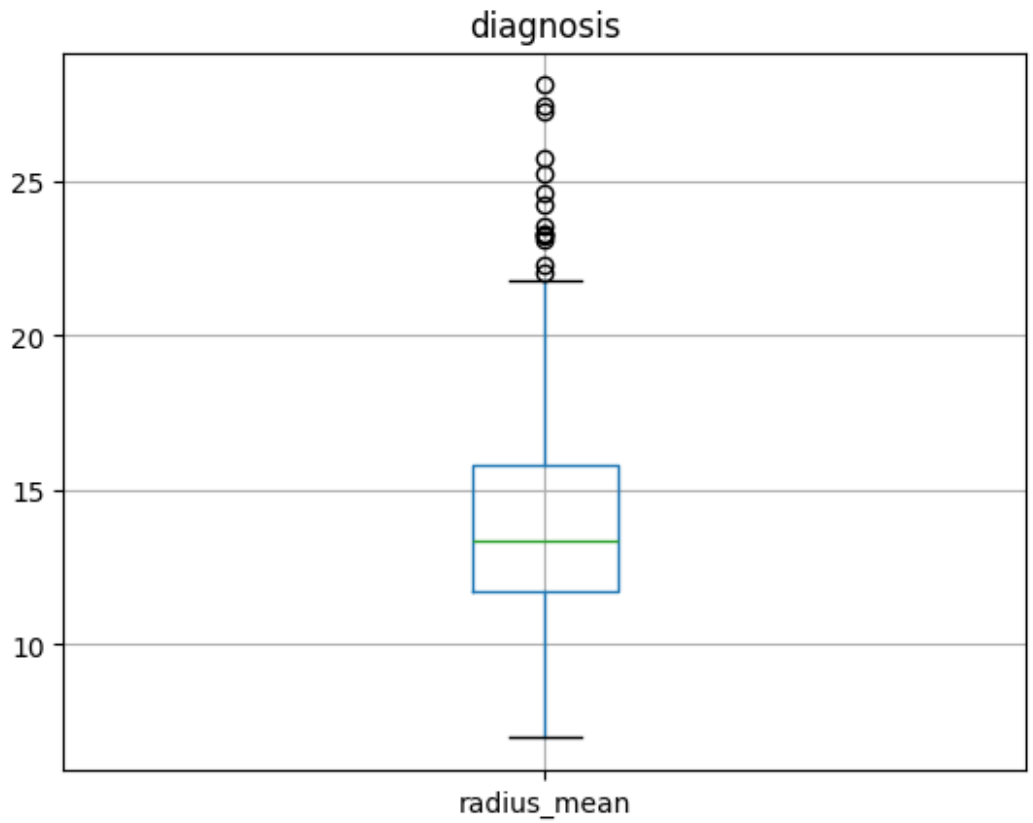
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

```
dataset.head()
```


```
{"type":"dataframe","variable_name":"dataset"}
```


```
dataset.boxplot(column=['radius_mean'])
plt.title('diagnosis')
plt.show()
```




```
dataset.columns
```

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean',
'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean',
'concavity_mean',
       'concave points_mean', 'symmetry_mean',
'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se',
'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se',
'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
```

```
dataset = dataset.drop(["id"], axis = 1)
```

```
dataset = dataset.drop(["Unnamed: 32"], axis = 1)
```

```
dataset.head(3)
```

{"type":"dataframe","variable_name":"dataset"}

```
X = dataset.drop(columns=['diagnosis'])
y = dataset['diagnosis']

dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   diagnosis                569 non-null    object 
 1   radius_mean              569 non-null    float64
 2   texture_mean             569 non-null    float64
 3   perimeter_mean           569 non-null    float64
 4   area_mean                569 non-null    float64
 5   smoothness_mean          569 non-null    float64
 6   compactness_mean         569 non-null    float64
 7   concavity_mean           569 non-null    float64
 8   concave points_mean      569 non-null    float64
 9   symmetry_mean            569 non-null    float64
 10  fractal_dimension_mean   569 non-null    float64
 11  radius_se                569 non-null    float64
 12  texture_se               569 non-null    float64
 13  perimeter_se             569 non-null    float64
 14  area_se                  569 non-null    float64
 15  smoothness_se            569 non-null    float64
 16  compactness_se           569 non-null    float64
 17  concavity_se             569 non-null    float64
 18  concave points_se        569 non-null    float64
 19  symmetry_se              569 non-null    float64
 20  fractal_dimension_se     569 non-null    float64
 21  radius_worst             569 non-null    float64
 22  texture_worst            569 non-null    float64
 23  perimeter_worst          569 non-null    float64
 24  area_worst               569 non-null    float64
 25  smoothness_worst         569 non-null    float64
 26  compactness_worst        569 non-null    float64
 27  concavity_worst          569 non-null    float64
 28  concave points_worst     569 non-null    float64
 29  symmetry_worst           569 non-null    float64
 30  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), object(1)
memory usage: 137.9+ KB
```

```
M = dataset[dataset.diagnosis == "M"]

M.head(5)
```

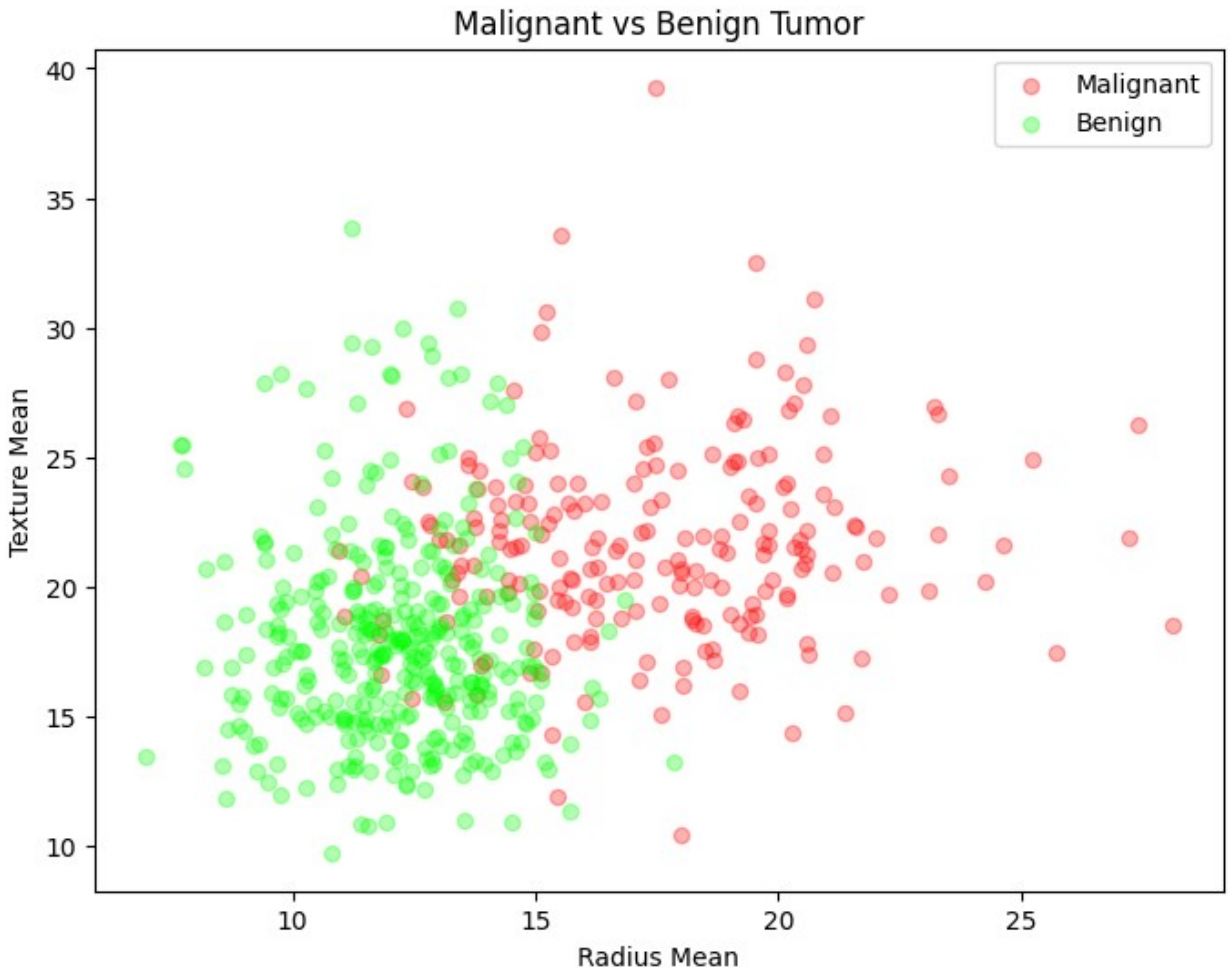
{"type":"dataframe","variable_name":"M"}

```
B = dataset[dataset.diagnosis == "B"]

B.head(5)
```

{"type":"dataframe","variable_name":"B"}

Visualization

```
plt.figure(figsize=(8,6))
plt.title("Malignant vs Benign Tumor")
plt.xlabel("Radius Mean")
plt.ylabel("Texture Mean")

plt.scatter(M.radius_mean, M.texture_mean, color="red",
label="Malignant", alpha=0.3)
plt.scatter(B.radius_mean, B.texture_mean, color="lime",
label="Benign", alpha=0.3)

plt.legend()
plt.show()
```




```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

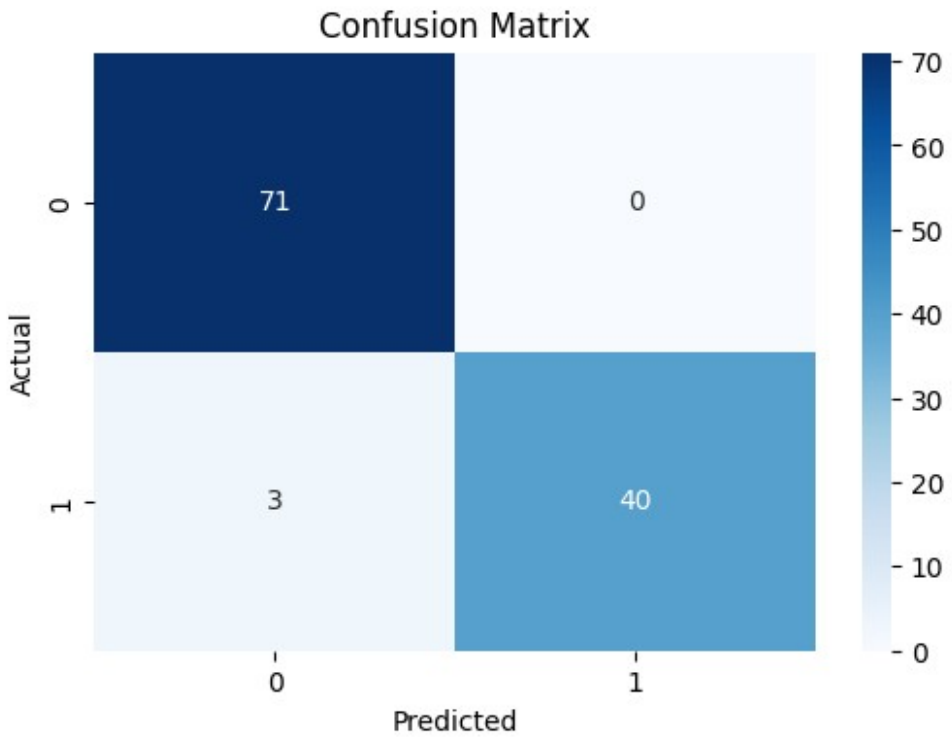
Evaluasi Model

```
# Akurasi
acc = accuracy_score(y_test, y_pred)
print("Akurasi Model:", acc)

# Confusion matrix
cm = confusion_matrix(y_test, y_pred)
```

```
print("\nConfusion Matrix:\n", cm)

# Classification report
report = classification_report(y_test, y_pred)
print("\nClassification Report:\n", report)
```

```
Akurasi Model: 0.9736842105263158

Confusion Matrix:
 [[71  0]
 [ 3 40]]

Classification Report:
              precision    recall  f1-score   support

           B       0.96      1.00      0.98        71
           M       1.00      0.93      0.96        43

    accuracy                           0.97       114
   macro avg       0.98      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114
```

```
import seaborn as sns

plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```


Confusion Matrix
71
0
3
40
Actual
0
1
Predicted
70
60
50
40
30
20
10
0