```
import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
%matplotlib inline
from IPython.display import display_html
import statsmodels.formula.api as smf
import statsmodels.api as sm

from sklearn import model_selection
from sklearn import metrics

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression

from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, accuracy_score
import tensorflow as tf
import matplotlib.gridspec as gridspec
from sklearn.metrics import roc_curve, auc

blue_red = ['#74a09e','#86c1b2','#98e2c6','#f3c969','#f2a553', '#d96548', '#c14953']
sns.palplot(sns.color_palette(blue_red))

# Set Style
sns.set_style("whitegrid")
sns.despine(left=True, bottom=True)
```

```
from google.colab import files
uploded = files.upload()
```

Pilih File Tidak ada file yang dipilih Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving CADdataset (1).csv to CADdataset (1) (1).csv
```

```
# Load the dataset
df = pd.read_csv('CADdataset (1).csv')
df.head()
```

| | Age | Weight | Height | Sex | BMI | DM | HTN | Current Smoker | Obesity | DLP | ... | LVH | PRI |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 53 | 90 | 175 | Male | 29.387755 | 0 | 1 | 1 | Y | Y | ... | N | N |
| **1** | 67 | 70 | 157 | Fmale | 28.398718 | 0 | 1 | 0 | Y | N | ... | N | N |
| **2** | 54 | 54 | 164 | Male | 20.077335 | 0 | 0 | 1 | N | N | ... | N | N |
| **3** | 66 | 67 | 158 | Fmale | 26.838648 | 0 | 1 | 0 | Y | N | ... | N | N |
| **4** | 50 | 87 | 153 | Fmale | 37.165193 | 0 | 1 | 0 | Y | N | ... | N | N |

```
df1 = df.drop(columns=['Age','Sex'])
df1.shape
```

```
(303, 25)
```

```
import matplotlib.pyplot as plt

plt.boxplot(df3['Age'], vert=False)
plt.ylabel('Sex')
plt.xlabel('Age')
plt.title('Weight')
plt.show()

# Calculate summary statistics
```

```
mean = df3['Age'].mean()
std = df3['Age'].std()

# Calculate the lower and upper bounds
lower_bound = mean - std*2
upper_bound = mean + std*2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-11-ddd4c72d94df> in <cell line: 3>()
      1 import matplotlib.pyplot as plt
      2 
----> 3 plt.boxplot(df3['Age'], vert=False)
      4 plt.ylabel('Sex')
      5 plt.xlabel('Age')

NameError: name 'df3' is not defined
```

TELUSURI STACK OVERFLOW

```
# import libraries
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import numpy as np

# load the iris dataset
iris = load_iris()
x = iris.data
y = iris.target

# split the data into training and testing sets
x_train, x_test,\
y_train, y_test = train_test_split(x, y,
                                   test_size=0.2,
                                   random_state=42)

# create a Multinomial logistic regression model
multi_logreg = LogisticRegression(multi_class='Age',
                                  solver='Sex')
multi_logreg.fit(x_train, y_train)

# create a One-vs-Rest logistic regression model
ovr_logreg = LogisticRegression(multi_class='Age',
                                solver='Sex')
ovr_logreg.fit(x_train, y_train)
```

```
---------------------------------------------------------------------------
InvalidParameterError                     Traceback (most recent call last)
<ipython-input-12-16abc864c221> in <cell line: 23>()
     21 multi_logreg = LogisticRegression(multi_class='Age',
     22                                   solver='Sex')
---> 23 multi_logreg.fit(x_train, y_train)
     24 
     25 # create a One-vs-Rest logistic regression model

                          2 frames

/usr/local/lib/python3.10/dist-packages/sklearn/utils/_param_validation.py in
validate_parameter_constraints(parameter_constraints, params, caller_name)
     95                 )
     96 
---> 97             raise InvalidParameterError(
     98                 f"The {param_name!r} parameter of {caller_name} must be"
     99                 f" {constraints_str}. Got {param_val!r} instead."

InvalidParameterError: The 'multi_class' parameter of LogisticRegression must be a
str among {'multinomial', 'ovr', 'auto'}. Got 'Age' instead.
```

TELUSURI STACK OVERFLOW

```
LogisticRegression(multi_class='Age', solver='Sex')
```

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression

iris=load_iris()
```

```
X=iris.data[:,:2]
y=iris.target

clf=LogisticRegression(random_state=0,
                       multi_class='Age',
                       solver='Sex')

clf.fit(X,y)

x_min, x_max=X[:,0].min()-.5,X[:,0].max()+.5
y_min, y_max=X[:,1].min()-.5,X[:,1].max()+.5
xx,yy=np.meshgrid(np.arange(x_min,x_max,0.2),
                  np.arange(y_min,y_max,0.2))
Z=clf.predict(np.c_[xx.ravel(),yy.ravel()])
Z=Z.reshape(xx.shape)
plt.figure(1,figsize=(4,3))
plt.pcolormesh(xx,yy,Z,cmap=plt.cm.Paired)
plt.scatter(X[:,0],X[:,1],c=y,edgecolors='k',
            cmap=plt.cm.Paired)
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.title('One-vs-Rest Logistic regression')
plt.show()
```