```
!pip install fuzzy_c_means
```

```
Collecting fuzzy_c_means
  Downloading fuzzy_c_means-1.7.0-py3-none-any.whl (9.0 kB)
Requirement already satisfied: joblib<2.0.0,>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from fuzzy_c_means) (1.3.2)
Requirement already satisfied: numpy<2.0.0,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from fuzzy_c_means) (1.23.5)
Collecting pydantic<2.0.0,>=1.9.0 (from fuzzy_c_means)
  Downloading pydantic-1.10.14-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.1/3.1 MB 13.7 MB/s eta 0:00:00
Collecting tabulate<0.9.0,>=0.8.9 (from fuzzy_c_means)
  Downloading tabulate-0.8.10-py3-none-any.whl (29 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.64.1 in /usr/local/lib/python3.10/dist-packages (from fuzzy_c_means) (4.66.1)
Collecting typer<0.5.0,>=0.4.0 (from fuzzy_c_means)
  Downloading typer-0.4.2-py3-none-any.whl (27 kB)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<2.0.0,>=1.9.0->fuz
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.10/dist-packages (from typer<0.5.0,>=0.4.0->fuzzy_c_mea
Installing collected packages: typer, tabulate, pydantic, fuzzy_c_means
  Attempting uninstall: typer
    Found existing installation: typer 0.9.0
    Uninstalling typer-0.9.0:
      Successfully uninstalled typer-0.9.0
  Attempting uninstall: tabulate
    Found existing installation: tabulate 0.9.0
    Uninstalling tabulate-0.9.0:
      Successfully uninstalled tabulate-0.9.0
  Attempting uninstall: pydantic
    Found existing installation: pydantic 2.6.1
    Uninstalling pydantic-2.6.1:
      Successfully uninstalled pydantic-2.6.1
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
lida 0.0.10 requires fastapi, which is not installed.
lida 0.0.10 requires kaleido, which is not installed.
lida 0.0.10 requires python-multipart, which is not installed.
lida 0.0.10 requires uvicorn, which is not installed.
llmx 0.0.15a0 requires cohere, which is not installed.
llmx 0.0.15a0 requires openai, which is not installed.
llmx 0.0.15a0 requires tiktoken, which is not installed.
bigframes 0.20.1 requires tabulate>=0.9, but you have tabulate 0.8.10 which is incompatible.
Successfully installed fuzzy_c_means-1.7.0 pydantic-1.10.14 tabulate-0.8.10 typer-0.4.2
```

```
!pip install scipy
```

```
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.11.4)
Requirement already satisfied: numpy<1.28.0,>=1.21.6 in /usr/local/lib/python3.10/dist-packages (from scipy) (1.23.5)
```

```
!pip install -U scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Collecting scikit-learn
  Downloading scikit_learn-1.4.0-1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.1/12.1 MB 41.1 MB/s eta 0:00:00
Requirement already satisfied: numpy<2.0,>=1.19.5 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.23.5)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.2.0)
Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 1.2.2
    Uninstalling scikit-learn-1.2.2:
      Successfully uninstalled scikit-learn-1.2.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
bigframes 0.20.1 requires tabulate>=0.9, but you have tabulate 0.8.10 which is incompatible.
Successfully installed scikit-learn-1.4.0
```

```
import pandas as pd
import numpy as np
import sklearn as sk
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import make_classification
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import GradientBoostingClassifier
from fcmeans import FCM
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.metrics import mean_squared_error , precision_score,accuracy_score,f1_score,recall_score
from six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus


# load your data
data = pd.read_csv("heart.csv")
data.head(8)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1
6	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
7	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0

```
# Preprocess the data
data.replace('?',-99999, inplace=True)
print(data.axes)
```

```
[RangeIndex(start=0, stop=1025, step=1), Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')]
```

```
data.isnull().sum()
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

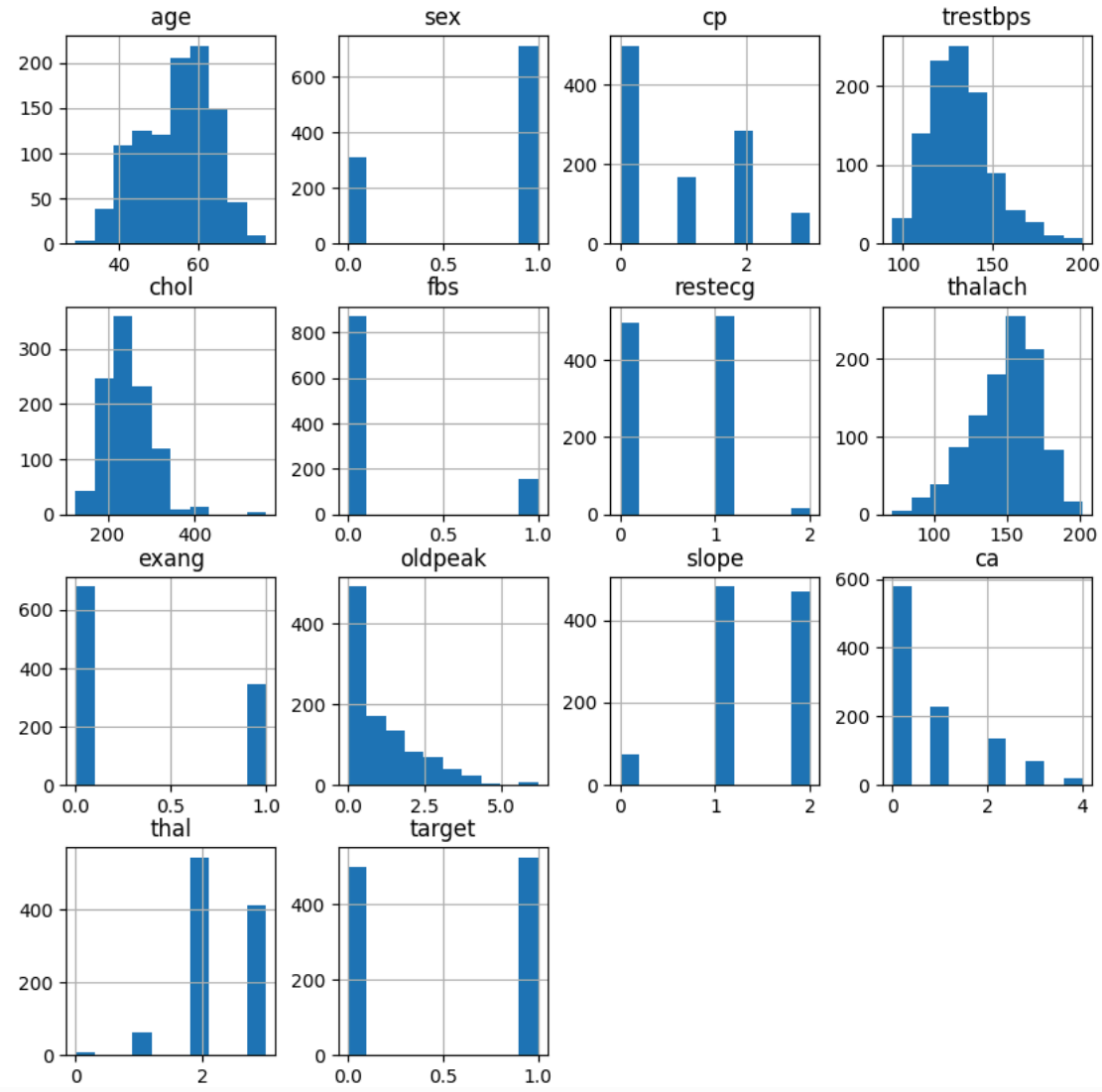
```
# Let explore the dataset and do a few visualizations
print(data.loc[10])
```

```
age          71.0
sex           0.0
cp            0.0
trestbps    112.0
chol        149.0
fbs           0.0
restecg       1.0
thalach     125.0
exang         0.0
oldpeak       1.6
slope         1.0
ca            0.0
thal          2.0
target        1.0
Name: 10, dtype: float64
```

```
data['target'].unique()
```

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

```
# Plot histograms for each variable
data.hist(figsize = (10, 10))
plt.show()
```




```
# Describe the dataset
print(data.describe())
print(data.info())
```

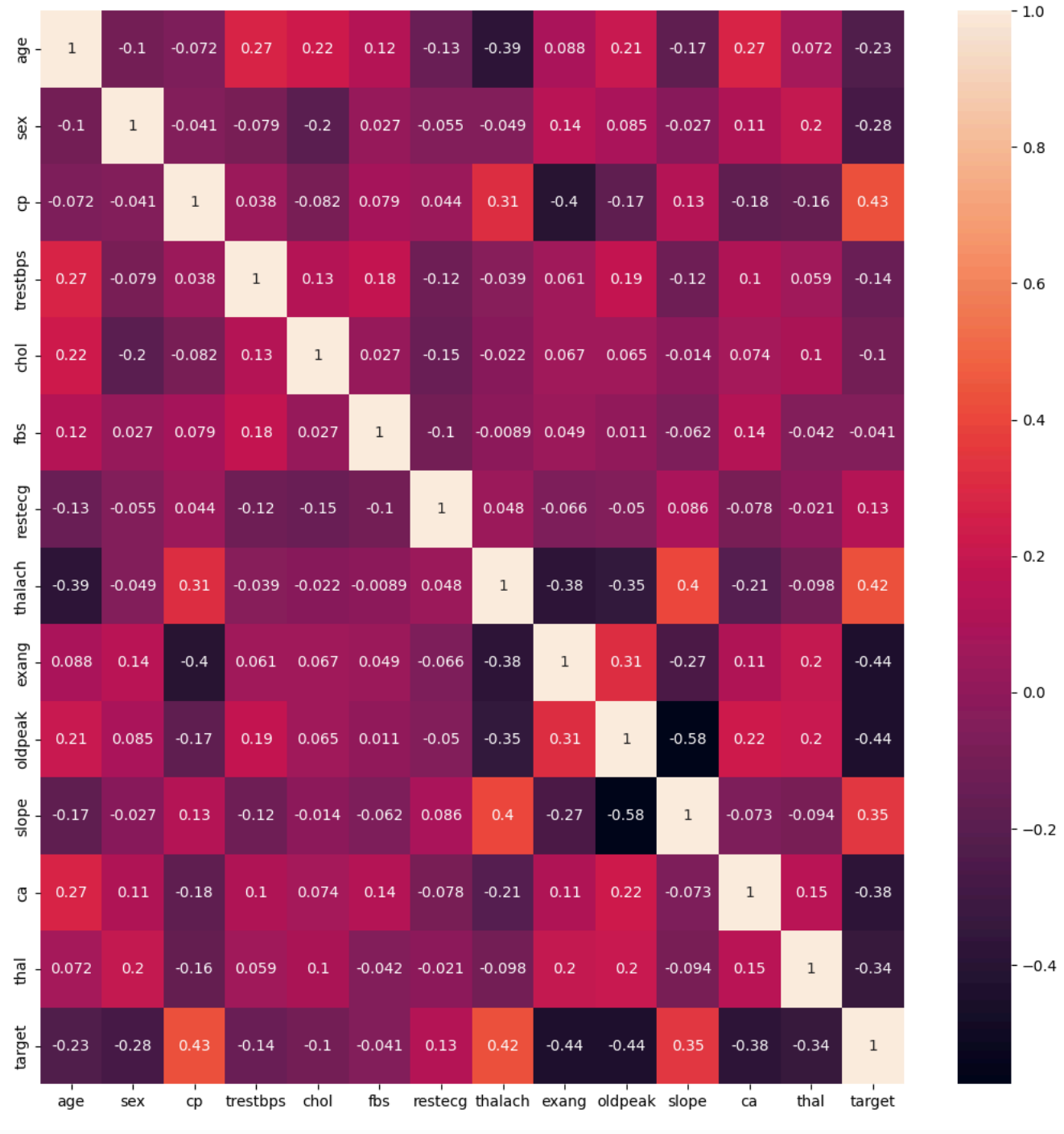
```
                age          sex           cp     trestbps        chol  \
count  1025.000000  1025.000000  1025.000000  1025.000000  1025.00000
mean     54.434146     0.695610     0.942439   131.611707   246.00000
std       9.072290     0.460373     1.029641    17.516718    51.59251
min      29.000000     0.000000     0.000000    94.000000   126.00000
25%      48.000000     0.000000     0.000000   120.000000   211.00000
50%      56.000000     1.000000     1.000000   130.000000   240.00000
75%      61.000000     1.000000     2.000000   140.000000   275.00000
max      77.000000     1.000000     3.000000   200.000000   564.00000

               fbs      restecg      thalach        exang      oldpeak  \
count  1025.000000  1025.000000  1025.000000  1025.000000  1025.000000
mean      0.149268     0.529756   149.114146     0.336585     1.071512
std       0.356527     0.527878    23.005724     0.472772     1.175053
min       0.000000     0.000000    71.000000     0.000000     0.000000
25%       0.000000     0.000000   132.000000     0.000000     0.000000
50%       0.000000     1.000000   152.000000     0.000000     0.800000
75%       0.000000     1.000000   166.000000     1.000000     1.800000
max       1.000000     2.000000   202.000000     1.000000     6.200000

             slope           ca         thal       target
count  1025.000000  1025.000000  1025.000000  1025.000000
mean      1.385366     0.754146     2.323902     0.513171
std       0.617755     1.030798     0.620660     0.500070
min       0.000000     0.000000     0.000000     0.000000
25%       1.000000     0.000000     2.000000     0.000000
50%       1.000000     0.000000     2.000000     1.000000
75%       2.000000     1.000000     3.000000     1.000000
max       2.000000     4.000000     3.000000     1.000000
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
```

```
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
 13  target    1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
None
```

```
#plt.figure(figsize=(100,81))
#sns.pairplot(data,palette="Set2",kind='hist')
plt.figure(figsize=(13,13))
sns.heatmap(data.corr(),annot=True)
```




```
X = data.drop("target", axis=1)
y = data["target"]
```

```
# split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)


sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)


#Decision Tree Classifier
DT_model = DecisionTreeClassifier(max_depth=3)
DT_model.fit(X_train_std,y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=3)
```

```
regressor = GradientBoostingClassifier(
    max_depth=3,
    n_estimators=3,
    learning_rate=1.0
)
regressor.fit(X_train, y_train)

errors = [mean_squared_error(y_test,y_pred) for y_pred in regressor.staged_predict(X_test)]
best_n_estimators = np.argmin(errors)

GBDT_best_regressor = GradientBoostingClassifier(
    max_depth =3 ,
    n_estimators=best_n_estimators,
    learning_rate = 1.0
)
GBDT_best_regressor.fit(X_train,y_train)
```

```
GradientBoostingClassifier
GradientBoostingClassifier(learning_rate=1.0, n_estimators=2)
```

```
# Bagging
gbdt = GradientBoostingClassifier()
Bag_GBDT = BaggingClassifier(base_estimator=gbdt, n_estimators=5, max_samples=0.8)
Bag_GBDT.fit(X_train, y_train)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-20-58b0080d037d> in <cell line: 3>()
      1 # Bagging
      2 gbdt = GradientBoostingClassifier()
----> 3 Bag_GBDT = BaggingClassifier(base_estimator=gbdt, n_estimators=5, max_samples=0.8)
      4 Bag_GBDT.fit(X_train, y_train)

TypeError: BaggingClassifier.__init__() got an unexpected keyword argument 'base_estimator'
```

```
df_tk=data
def lower_limit(data):
  col = data.columns
  data_lower_limit = []
  for i in col:
    df = data[i]
    x = (1*df[0] + 1*df[1] + 2*df[2] + 3*df[3] + 4*df[4]) / sum(df)
    data_lower_limit.append(x)
  return(data_lower_limit)

def middle_value(data):
  col = data.columns
  data_middle_value = []
  for i in col:
    df = data[i]
    x = (1*df[0] + 2*df[1] + 3*df[2] + 4*df[3] + 5*df[4]) / sum(df)
    data_middle_value.append(x)
  return(data_middle_value)

def upper_limit(data):
  col = data.columns
  data_upper_limit = []
  for i in col:
    df = data[i]
    x = (2*df[0] + 3*df[1] + 4*df[2] + 5*df[3] + 5*df[4]) / sum(df)
```

```
      data_upper_limit.append(x)
  return(data_upper_limit)

def fuzzi_work(data):
  clower_limit = lower_limit(data)
  dmiddle_value = middle_value(data)
  eupper_limit = upper_limit(data)
  df = pd.DataFrame(list(zip(clower_limit, dmiddle_value, eupper_limit)),columns =['batas bawah', 'nilai tengah', 'batas atas'])
  return(df)


fuzz_tk = fuzzi_work(df_tk)


regressor = GradientBoostingClassifier(
    max_depth=2,
    n_estimators=3,
    learning_rate=1.0
)
regressor.fit(X_train, y_train)

errors = [mean_squared_error(y_test,y_pred) for y_pred in regressor.staged_predict(X_test)]
best_n_estimators = np.argmin(errors)

Fuzzy_best_regressor = GradientBoostingClassifier(
    max_depth =2 ,
    n_estimators=best_n_estimators,
    learning_rate = 0.5
)
Fuzzy_best_regressor.fit(X_train,y_train)
```

```
GradientBoostingClassifier(learning_rate=0.5, max_depth=2, n_estimators=2)
```

```
regressor = GradientBoostingClassifier(
    max_depth=5,
    n_estimators=2,
    learning_rate=1.0
)
regressor.fit(X_train, y_train)

errors = [mean_squared_error(y_test,y_pred) for y_pred in regressor.staged_predict(X_test)]
best_n_estimators = np.argmin(errors)

best_regressor = GradientBoostingClassifier(
    max_depth =5,
    n_estimators=best_n_estimators,
    learning_rate = 1.0
)
best_regressor.fit(X_train,y_train)

Bag_Fuzzy_GBDT = BaggingClassifier(
    base_estimator= best_regressor,
    n_estimators=100,
    max_samples = 0.8,
    bootstrap=True,
    oob_score = True,
    random_state = 0
)
Bag_Fuzzy_GBDT.fit(X_train,y_train)
```

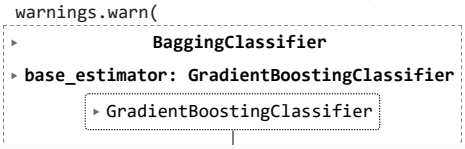
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-24-5956afa5d073> in <cell line: 18>()
     16 best_regressor.fit(X_train,y_train)
     17
---> 18 Bag_Fuzzy_GBDT = BaggingClassifier(
     19     base_estimator= best_regressor,
     20     n_estimators=100,

TypeError: BaggingClassifier.__init__() got an unexpected keyword argument 'base_estimator'
```

```
#Fuzzy-C-Means Clustering
fcm = FCM(n_clusters=8)
fcm.fit(X_train_std)
FX_train = fcm.predict(X_train_std)
FX_test = fcm.predict(X_test_std)
FX_train= FX_train.reshape(-1, 1)
FX_test = FX_test.reshape(-1, 1)
```

```
# Bagging
gbdt = GradientBoostingClassifier()
bag_fcm_gbdt = BaggingClassifier(base_estimator=gbdt, n_estimators=10, max_samples=0.8)
bag_fcm_gbdt.fit(FX_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/ensemble/_base.py:156: FutureWarning: `base_estimator` was renamed to `estimator` ir
  warnings.warn(
```




```
# Predict
DT_y_pred = DT_model.predict(X_test_std)
print(DT_y_pred)
```

```
[0 0 0 ... 0 0 0]
```

```
GBDT_y_pred = GBDT_best_regressor.predict(X_test)
print(GBDT_y_pred)
```

```
[0 0 1 1 1 0 0 0 0 1 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 1 1 0 1 1 1
 0 1 0 1 1 1 0 1 0 0 1 1 1 0 0 0 1 0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 1 1 0 1 1
 0 0 1 0 1 0 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1 0 0 1 0
 1 1 0 1 0 1 1 1 1 1 1 0 1 0 1 0 0 1 0 1 1 0 1 0 0 0 1 1 0 1 0 0 0 0 1 1 0
 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 1 1 0 1
 0 0 1 1 1 0 0 0 1 1 0 0 1 0 0 0 1 1 1 0 0 1 0 1 1 1 1 1 0 1 1 1 0 1 0 1 0
 1 0 1 0 1 0 1 1 1 0 0 1 1 1 0 0 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 0 0 0 1
 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 1
 1 0 1 1 0 0 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 1 1 0 0 0 0 0 1 1 0 0 1 1
 1 1 0 1 0 1]
```

```
# Predict
Bag_GBDT_y_pred = Bag_GBDT.predict(X_test)
print(Bag_GBDT_y_pred)
```

```
[0 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 1 1 0 0 1 0 1
 0 1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 0 1 1
 0 1 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 1 0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0
 1 1 0 1 0 1 1 1 1 0 1 0 1 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 1 0 0 0 0 1 1 1
 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 0 1
 0 0 1 1 1 0 0 0 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 1 1 1 0 0 1 0 1 1 1 1 1 0 1 1 1
 1 0 1 0 1 0 1 1 1 0 0 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1
 1 0 0 1 0 1 0 1 0 1 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 1 1 0 0 0 0 1 1 0 1 0 0
 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 1
 1 1 0 1 0 1]
```

```
Fuzzy_GBDT_y_pred = Fuzzy_best_regressor.predict(X_test)
print(Fuzzy_GBDT_y_pred)
```

```
[0 0 1 1 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1
 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 1 1 0 0 0 0 1 0 1 0 1 1 1 0 0 1
 0 1 1 0 1 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1 1 1 1 0
 0 1 0 1 0 1 1 1 0 0 1 0 1 0 0 0 0 1 1 1 1 0 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1
 1 1 0 0 0 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 1 0 1
 0 0 1 0 1 1 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 1 0 1 0 1 1 1 1 0 1 1 0 0 0 0 0
 1 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 1 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 1
 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
 0 0 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 0 1
 1 1 0 0 0 1]
```

```
Bag_Fuzzy_GBDT_y_pred = Bag_Fuzzy_GBDT.predict(X_test)
print(Bag_Fuzzy_GBDT_y_pred)
```

```
[0 0 1 0 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 1 1 0 1 1 1
 0 1 1 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 0 1 1
 0 1 1 0 1 0 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1 1 0 1 0
 1 1 0 1 0 1 1 1 1 1 1 0 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1
 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 0 1
 0 0 1 1 1 0 1 0 1 1 1 0 1 1 0 0 1 1 0 1 0 1 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 0
 1 0 1 0 1 0 1 1 1 0 0 1 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 1
 1 0 0 1 0 1 0 1 1 1 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0
 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 1 1
 1 1 0 1 0 1]
```

```
# Predict
bag_fcm_gbdt_y_pred = bag_fcm_gbdt.predict(FX_test)
print(bag_fcm_gbdt_y_pred)
```

```
[0 0 1 1 1 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 1 0 0 1 0 0 1 0 1 0 1 1 1 0 1 1 1
 0 1 1 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 1 1
```

```
 0 1 1 1 1 0 1 0 1 0 0 0 0 0 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 0 1 0 1 1 1 1 0
 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 0 0 0 0 1 1 0 1 1 0 0 1 1 1 1 0 1 0 0 1 0 1
 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 1 0 0 1 1 1 1 0 1
 0 0 1 1 1 1 1 0 1 1 1 0 1 0 0 0 1 1 1 1 1 0 1 0 1 1 1 0 0 1 1 1 1 0 1 1 0 1 0
 1 0 1 0 1 0 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 0 0 0 1
 1 0 0 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
 1 0 1 1 1 0 0 1 1 1 1 1 0 1 1 0 1 0 1 0 0 1 0 0 1 1 1 0 0 0 0 1 1 0 0 1 1
 1 1 0 0 0 1]
```

```
i = 0
print ("\n-------------------------------------------------------------------------")
print ('%-25s %-25s %-25s' % ('Original Label', 'Predicted Label', 'Correct/Wrong'))
print ("-------------------------------------------------------------------------")
for label in y_test:
    print ('%-25s %-25s' % (label, bag_fcm_gbdt_y_pred[i]), end="")
    if (label == bag_fcm_gbdt_y_pred[i]):
        print (' %-25s' % ('Correct'))
    else:
        print (' %-25s' % ('Wrong'))
    i = i + 1
print ("-------------------------------------------------------------------------")
print("\nConfusion Matrix:\n",metrics.confusion_matrix(y_test, bag_fcm_gbdt_y_pred))
print ("-------------------------------------------------------------------------")
print("\nClassification Report:\n",metrics.classification_report(y_test, bag_fcm_gbdt_y_pred))
print ("-------------------------------------------------------------------------")
print('Accuracy of the classifer is %0.2f' % metrics.accuracy_score(y_test, bag_fcm_gbdt_y_pred))
print ("-------------------------------------------------------------------------")
```

```
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         0                         Wrong
1                         1                         Correct
0                         1                         Wrong
1                         0                         Wrong
0                         1                         Wrong
0                         0                         Correct
0                         1                         Wrong
0                         0                         Correct
1                         0                         Wrong
1                         1                         Correct
1                         0                         Wrong
0                         0                         Correct
0                         1                         Wrong
1                         1                         Correct
0                         1                         Wrong
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
0                         0                         Correct
1                         0                         Wrong
0                         0                         Correct
1                         1                         Correct
-------------------------------------------------------------------------

Confusion Matrix:
 [[114  40]
 [ 25 160]]
-------------------------------------------------------------------------

Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.74      0.78       154
           1       0.80      0.86      0.83       185

    accuracy                           0.81       339
   macro avg       0.81      0.80      0.80       339
weighted avg       0.81      0.81      0.81       339

-------------------------------------------------------------------------
Accuracy of the classifer is 0.81
-------------------------------------------------------------------------
```

```
i = 0
print ("\n-------------------------------------------------------------------------------")
print ('%-25s %-25s %-25s' % ('Original Label', 'Predicted Label', 'Correct/Wrong'))
print ("-------------------------------------------------------------------------------")
for label in y_test:
    print ('%-25s %-25s' % (label, DT_y_pred[i]), end="")
    if (label == DT_y_pred[i]):
        print (' %-25s' % ('Correct'))
    else:
        print (' %-25s' % ('Wrong'))
    i = i + 1
print ("-------------------------------------------------------------------------------")
print("\nConfusion Matrix:\n",metrics.confusion_matrix(y_test, DT_y_pred))
print ("-------------------------------------------------------------------------------")
print("\nClassification Report:\n",metrics.classification_report(y_test, DT_y_pred))
print ("-------------------------------------------------------------------------------")
print('Accuracy of the classifer is %0.2f' % metrics.accuracy_score(y_test,DT_y_pred))
print ("-------------------------------------------------------------------------------")
```

```
-------------------------------------------------------------------------------
Original Label            Predicted Label           Correct/Wrong
-------------------------------------------------------------------------------
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         0                         Wrong
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         0                         Wrong
0                         0                         Correct
1                         0                         Wrong
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         0                         Wrong
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         0                         Wrong
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         0                         Wrong
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
```

```
i = 0
print ("\n-------------------------------------------------------------------------------")
print ('%-25s %-25s %-25s' % ('Original Label', 'Predicted Label', 'Correct/Wrong'))
print ("-------------------------------------------------------------------------------")
for label in y_test:
    print ('%-25s %-25s' % (label, GBDT_y_pred[i]), end="")
    if (label == GBDT_y_pred[i]):
        print (' %-25s' % ('Correct'))
    else:
```

```
            print (' %-25s' % ('Wrong'))
        i = i + 1
    print ("--------------------------------------------------------------------------")
    print("\nConfusion Matrix:\n",metrics.confusion_matrix(y_test, GBDT_y_pred))
    print ("--------------------------------------------------------------------------")
    print("\nClassification Report:\n",metrics.classification_report(y_test, GBDT_y_pred))
    print ("--------------------------------------------------------------------------")
    print('Accuracy of the classifer is %0.2f' % metrics.accuracy_score(y_test, GBDT_y_pred))
    print ("--------------------------------------------------------------------------")
```

```
    --------------------------------------------------------------------------
    Original Label            Predicted Label           Correct/Wrong
    --------------------------------------------------------------------------
    0                         0                         Correct
    1                         0                         Wrong
    0                         1                         Wrong
    0                         1                         Wrong
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         0                         Wrong
    1                         0                         Wrong
    0                         1                         Wrong
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
    1                         1                         Correct
    1                         1                         Correct
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    1                         1                         Correct
    1                         1                         Correct
    1                         1                         Correct
    1                         1                         Correct
    0                         1                         Wrong
    0                         0                         Correct
    1                         1                         Correct
    0                         1                         Wrong
    1                         1                         Correct
    0                         0                         Correct
    1                         1                         Correct
    1                         0                         Wrong
    1                         1                         Correct
    1                         1                         Correct
    1                         1                         Correct
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    0                         1                         Wrong
    1                         1                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
```

```
i = 0
print ("\n-------------------------------------------------------------------------------")
print ('%-25s %-25s %-25s' % ('Original Label', 'Predicted Label', 'Correct/Wrong'))
print ("-------------------------------------------------------------------------------")
for label in y_test:
    print ('%-25s %-25s' % (label, Bag_GBDT_y_pred[i]), end="")
    if (label == Bag_GBDT_y_pred[i]):
        print (' %-25s' % ('Correct'))
    else:
        print (' %-25s' % ('Wrong'))
    i = i + 1
print ("-------------------------------------------------------------------------------")
print("\nConfusion Matrix:\n",metrics.confusion_matrix(y_test, Bag_GBDT_y_pred))
print ("-------------------------------------------------------------------------------")
print("\nClassification Report:\n",metrics.classification_report(y_test, Bag_GBDT_y_pred))
print ("-------------------------------------------------------------------------------")
print('Accuracy of the classifer is %0.2f' % metrics.accuracy_score(y_test, Bag_GBDT_y_pred))
print ("-------------------------------------------------------------------------------")
```

```
-------------------------------------------------------------------------------
Original Label            Predicted Label           Correct/Wrong
-------------------------------------------------------------------------------
0                         0                         Correct
1                         0                         Wrong
0                         1                         Wrong
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
1                         0                         Wrong
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
```

```
i = 0
print ("\n-------------------------------------------------------------------------------")
print ('%-25s %-25s %-25s' % ('Original Label', 'Predicted Label', 'Correct/Wrong'))
print ("-------------------------------------------------------------------------------")
for label in y_test:
    print ('%-25s %-25s' % (label, Fuzzy_GBDT_y_pred[i]), end="")
    if (label == Fuzzy_GBDT_y_pred[i]):
        print (' %-25s' % ('Correct'))
    else:
```

```
            print (' %-25s' % ('Wrong'))
        i = i + 1
    print ("--------------------------------------------------------------------------------")
    print("\nConfusion Matrix:\n",metrics.confusion_matrix(y_test, Fuzzy_GBDT_y_pred))
    print ("--------------------------------------------------------------------------------")
    print("\nClassification Report:\n",metrics.classification_report(y_test, Fuzzy_GBDT_y_pred))
    print ("--------------------------------------------------------------------------------")
    print('Accuracy of the classifer is %0.2f' % metrics.accuracy_score(y_test, Fuzzy_GBDT_y_pred))
    print ("--------------------------------------------------------------------------------")
```

```
    --------------------------------------------------------------------------------
    Original Label            Predicted Label           Correct/Wrong
    --------------------------------------------------------------------------------
    0                         0                         Correct
    1                         0                         Wrong
    0                         1                         Wrong
    0                         1                         Wrong
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         0                         Wrong
    1                         0                         Wrong
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         0                         Wrong
    1                         1                         Correct
    1                         0                         Wrong
    0                         1                         Wrong
    1                         0                         Wrong
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    1                         1                         Correct
    1                         0                         Wrong
    1                         0                         Wrong
    1                         0                         Wrong
    0                         0                         Correct
    0                         0                         Correct
    1                         0                         Wrong
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    1                         1                         Correct
    1                         0                         Wrong
    1                         1                         Correct
    1                         1                         Correct
    1                         1                         Correct
    0                         0                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
    1                         1                         Correct
    0                         0                         Correct
    0                         0                         Correct
    1                         1                         Correct
```

```
i = 0
print ("\n-------------------------------------------------------------------------------")
print ('%-25s %-25s %-25s' % ('Original Label', 'Predicted Label', 'Correct/Wrong'))
print ("-------------------------------------------------------------------------------")
for label in y_test:
    print ('%-25s %-25s' % (label, Bag_Fuzzy_GBDT_y_pred[i]), end="")
    if (label == Bag_Fuzzy_GBDT_y_pred[i]):
        print (' %-25s' % ('Correct'))
    else:
        print (' %-25s' % ('Wrong'))
    i = i + 1
print ("-------------------------------------------------------------------------------")
print("\nConfusion Matrix:\n",metrics.confusion_matrix(y_test, Bag_Fuzzy_GBDT_y_pred))
print ("-------------------------------------------------------------------------------")
print("\nClassification Report:\n",metrics.classification_report(y_test, Bag_Fuzzy_GBDT_y_pred))
print ("-------------------------------------------------------------------------------")
print('Accuracy of the classifer is %0.2f' % metrics.accuracy_score(y_test,Bag_Fuzzy_GBDT_y_pred))
print ("-------------------------------------------------------------------------------")
```

```
-------------------------------------------------------------------------------
Original Label            Predicted Label           Correct/Wrong
-------------------------------------------------------------------------------
0                         0                         Correct
1                         0                         Wrong
0                         1                         Wrong
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
1                         0                         Wrong
0                         1                         Wrong
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
0                         1                         Wrong
0                         0                         Correct
1                         1                         Correct
0                         1                         Wrong
1                         1                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
1                         1                         Correct
0                         0                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
1                         1                         Correct
0                         0                         Correct
0                         0                         Correct
0                         0                         Correct
1                         1                         Correct
```

```
# Calculate the bias error
DT_bias_error = 1 - accuracy_score(y_test, DT_y_pred)

# Calculate the variance error
DT_y_pred_train = DT_model.predict(X_train)
DT_variance_error = accuracy_score(y_train, DT_y_pred_train) - accuracy_score(y_test, DT_y_pred)
```

```
# Calculate the bias error
GBDT_bias_error = 1 - accuracy_score(y_test, GBDT_y_pred)

# Calculate the variance error
GBDT_y_pred_train = GBDT_best_regressor.predict(X_train)
GBDT_variance_error = accuracy_score(y_train, GBDT_y_pred_train) - accuracy_score(y_test, GBDT_y_pred)

print("Bias error: ", GBDT_bias_error)
print("Variance error: ", GBDT_variance_error)
```

```
Bias error:  0.16519174041297935
Variance error:  0.05294684245379566
```

```
# Calculate the bias error
Bag_GBDT_bias_error = 1 - accuracy_score(y_test, Bag_GBDT_y_pred)

# Calculate the variance error
Bag_GBDT_y_pred_train = Bag_GBDT.predict(X_train)
Bag_GBDT_variance_error = accuracy_score(y_train, Bag_GBDT_y_pred_train) - accuracy_score(y_test, Bag_GBDT_y_pred)

print("Bias error: ", Bag_GBDT_bias_error)
print("Variance error: ", Bag_GBDT_variance_error)
```

```
Bias error:  0.0471976401179941
Variance error:  0.03845128443286294
```

```
# Calculate the bias error
Fuzzy_GBDT_bias_error = 1 - accuracy_score(y_test, Fuzzy_GBDT_y_pred)

# Calculate the variance error
```