Ensemble Techniques And Its Types Assignment - 4

March 12, 2024

#

Question 1

0.1 Question 1 : You are working on machine learning project where you have containing numerical and categorical features. You have identified that some features are highly correlated and there are missing values in some of the columns. You want to build a pipeline that automates feature engineering process and handles missing values.

0.1.1 Design a Pipeline that includes following steps:

- Use an automated method to identify important features of dataset
- Use a numerical pipeline that includes following steps:
  - Impute missing values in numeric columns with mean
  - Scale the numerical columns using standardisation
- Create a categorical pipeline that includes following steps:
  - Impute missing values in categorical columns with most frequent data
  - One Hot Encode the categorical columns
- Combine the numerical and categorical pipelines using a ColumnTransformer
- Use Random Forest Classifier to build final model
- Evaluate accuracy of model on the test dataset

0.2 ### Note : Your Solution should include code snippets for each step of pipeline and a brief explaination of each step. You should also provide an interpretation of results and suggest possible improvements for pipeline

0.3 Answer :

0.3.1 Used Employee Attrition dataset for performing above tasks

Dataset link : `https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset/`

Uncover the factors that lead to employee attrition and explore important questions such as 'show me a breakdown of distance from home by job role and attrition' or 'compare average monthly income by education and attrition'.

0.3.2 Read the dataset

[1]:
```python
import pandas as pd
df = pd.read_csv('Attrition.csv')
df.head()
```

[1]:
```
   Age Attrition     BusinessTravel  DailyRate              Department  \
0   41       Yes      Travel_Rarely       1102                   Sales
1   49        No  Travel_Frequently        279  Research & Development
2   37       Yes      Travel_Rarely       1373  Research & Development
3   33        No  Travel_Frequently       1392  Research & Development
4   27        No      Travel_Rarely        591  Research & Development

   DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber  \
0                 1          2  Life Sciences              1               1
1                 8          1  Life Sciences              1               2
2                 2          2          Other              1               4
3                 3          4  Life Sciences              1               5
4                 2          1        Medical              1               7

   …  RelationshipSatisfaction StandardHours  StockOptionLevel  \
0  …                         1            80                 0
1  …                         4            80                 1
2  …                         2            80                 0
3  …                         3            80                 0
4  …                         4            80                 1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0                  8                      0               1               6
1                 10                      3               3              10
2                  7                      3               3               0
3                  8                      3               3               8
4                  6                      3               3               2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

[2]:
```python
df.shape
```

[2]:
```
(1470, 35)
```

[3]:
```python
# Checking dataset info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

[4]:
```python
# Checking missing values in dataset
df.isnull().sum()
```

[4]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

0.3.3 No Missing values found in dataset

0.3.4 Seperate X and Y

[5]:
```python
X = df.drop(labels=['Attrition'],axis=1)
Y = df[['Attrition']]
```

[6]:
```
Y.head()
```

[6]:
```
  Attrition
0       Yes
1        No
2       Yes
3        No
4        No
```

[7]:
```
y_mapper = {'Yes':1,'No':0}
Y = Y.replace(y_mapper)
Y.head()
```

[7]:
```
   Attrition
0          1
1          0
2          1
3          0
4          0
```

[8]:
```
cat_cols = list(X.select_dtypes(include='object').columns)
num_cols = list(X.select_dtypes(exclude='object').columns)
```

[9]:
```
cat_cols
```

[9]:
```
['BusinessTravel',
 'Department',
 'EducationField',
 'Gender',
 'JobRole',
 'MaritalStatus',
 'Over18',
 'OverTime']
```

[10]:
```
num_cols
```

[10]:
```
['Age',
 'DailyRate',
 'DistanceFromHome',
 'Education',
 'EmployeeCount',
 'EmployeeNumber',
 'EnvironmentSatisfaction',
 'HourlyRate',
 'JobInvolvement',
 'JobLevel',
 'JobSatisfaction',
```

```
 'MonthlyIncome',
 'MonthlyRate',
 'NumCompaniesWorked',
 'PercentSalaryHike',
 'PerformanceRating',
 'RelationshipSatisfaction',
 'StandardHours',
 'StockOptionLevel',
 'TotalWorkingYears',
 'TrainingTimesLastYear',
 'WorkLifeBalance',
 'YearsAtCompany',
 'YearsInCurrentRole',
 'YearsSinceLastPromotion',
 'YearsWithCurrManager']
```

[11]:
```
len(num_cols)
```

[11]:
```
26
```

0.3.5 Feature selection for numerical columns

[12]:
```
from warnings import filterwarnings
filterwarnings('ignore')
```

[13]:
```
from sklearn.feature_selection import SelectKBest, f_classif
X_num = X[num_cols]
k_best_numerical = SelectKBest(f_classif,k=10)
k_best_numerical.fit_transform(X_num,Y)
selected_num_features = list(X_num.columns[k_best_numerical.get_support()])
selected_num_features
```

[13]:
```
['Age',
 'JobInvolvement',
 'JobLevel',
 'JobSatisfaction',
 'MonthlyIncome',
 'StockOptionLevel',
 'TotalWorkingYears',
 'YearsAtCompany',
 'YearsInCurrentRole',
 'YearsWithCurrManager']
```

0.3.6 Feature Selection for categorical variables

[14]:
```
from sklearn.feature_selection import SelectKBest, chi2
X_cat = X[cat_cols]
from sklearn.preprocessing import OrdinalEncoder
oe = OrdinalEncoder()
X_cat_encoded = pd.DataFrame(oe.fit_transform(X_cat),columns=oe.
 ↪get_feature_names_out())
k_best_categorical = SelectKBest(chi2,k=5)
k_best_categorical.fit_transform(X_cat_encoded,Y)
selected_cat_features = list(X_cat_encoded.columns[k_best_categorical.
 ↪get_support()])
selected_cat_features
```

[14]:
```
['Department', 'EducationField', 'JobRole', 'MaritalStatus', 'OverTime']
```

[15]:
```
selected_features = selected_num_features + selected_cat_features
selected_features
```

[15]:
```
['Age',
 'JobInvolvement',
 'JobLevel',
 'JobSatisfaction',
 'MonthlyIncome',
 'StockOptionLevel',
 'TotalWorkingYears',
 'YearsAtCompany',
 'YearsInCurrentRole',
 'YearsWithCurrManager',
 'Department',
 'EducationField',
 'JobRole',
 'MaritalStatus',
 'OverTime']
```

[16]:
```
X_selected = X[selected_features]
X_selected.head()
```

[16]:
```
   Age  JobInvolvement  JobLevel  JobSatisfaction  MonthlyIncome  \
0   41               3         2                4           5993
1   49               2         2                2           5130
2   37               2         1                3           2090
3   33               3         1                3           2909
4   27               3         1                2           3468

   StockOptionLevel  TotalWorkingYears  YearsAtCompany  YearsInCurrentRole  \
0                 0                  8               6                   4
```

```
1          1          10          10          7
2          0           7           0          0
3          0           8           8          7
4          1           6           2          2

   YearsWithCurrManager              Department EducationField  \
0                     5                   Sales  Life Sciences
1                     7  Research & Development  Life Sciences
2                     0  Research & Development          Other
3                     0  Research & Development  Life Sciences
4                     2  Research & Development        Medical

                JobRole MaritalStatus OverTime
0       Sales Executive        Single      Yes
1    Research Scientist       Married       No
2 Laboratory Technician        Single      Yes
3    Research Scientist       Married      Yes
4 Laboratory Technician       Married       No
```

[17]:
```
X_selected.shape
```

[17]:
```
(1470, 15)
```

0.3.7 Feature Selection is completed

0.3.8 Train Test Split of data

[18]:
```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(X_selected,Y,test_size=0.
↪2,random_state=42,stratify=Y)
```

[19]:
```
xtrain.shape
```

[19]:
```
(1176, 15)
```

[20]:
```
xtest.shape
```

[20]:
```
(294, 15)
```

[21]:
```
ytrain.value_counts()
```

[21]:
```
Attrition
0        986
1        190
dtype: int64
```

[22]:
```
ytest.value_counts()
```

[22]:
```
Attrition
0            247
1             47
dtype: int64
```

0.3.9 Creating numeric and categorical pipeline

[23]:
```
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder,StandardScaler

# Numeric Pipeline creation
num_pipeline = Pipeline(steps=[('imputer',SimpleImputer(strategy='mean')),
                               ('scaler',StandardScaler())])

# Categorical Pipeline creation
cat_pipeline =␣
 ↪Pipeline(steps=[('imputer',SimpleImputer(strategy='most_frequent')),
                               ('one_hot_encoder',OneHotEncoder()),
                               ('scaler',StandardScaler(with_mean=False))])
```

0.3.10 ColumnTransformer to combine numeric and Categorical pipelines

[24]:
```
selected_num_features
```

[24]:
```
['Age',
 'JobInvolvement',
 'JobLevel',
 'JobSatisfaction',
 'MonthlyIncome',
 'StockOptionLevel',
 'TotalWorkingYears',
 'YearsAtCompany',
 'YearsInCurrentRole',
 'YearsWithCurrManager']
```

[25]:
```
selected_cat_features
```

[25]:
```
['Department', 'EducationField', 'JobRole', 'MaritalStatus', 'OverTime']
```

[26]:
```
from sklearn.compose import ColumnTransformer
preprocessor =␣
 ↪ColumnTransformer([('num_pipeline',num_pipeline,selected_num_features),
                                 ␣
 ↪('cat_pipeline',cat_pipeline,selected_cat_features)])
```

0.3.11 Transform the dataset with ColumnTransformer

[27]:
```
xtrain_transformed = pd.DataFrame(preprocessor.
 ↪fit_transform(xtrain),columns=preprocessor.get_feature_names_out())
xtest_transformed = pd.DataFrame(preprocessor.
 ↪transform(xtest),columns=preprocessor.get_feature_names_out())
```

[28]:
```
preprocessor.get_feature_names_out()
```

[28]:
```
array(['num_pipeline__Age', 'num_pipeline__JobInvolvement',
       'num_pipeline__JobLevel', 'num_pipeline__JobSatisfaction',
       'num_pipeline__MonthlyIncome', 'num_pipeline__StockOptionLevel',
       'num_pipeline__TotalWorkingYears', 'num_pipeline__YearsAtCompany',
       'num_pipeline__YearsInCurrentRole',
       'num_pipeline__YearsWithCurrManager',
       'cat_pipeline__Department_Human Resources',
       'cat_pipeline__Department_Research & Development',
       'cat_pipeline__Department_Sales',
       'cat_pipeline__EducationField_Human Resources',
       'cat_pipeline__EducationField_Life Sciences',
       'cat_pipeline__EducationField_Marketing',
       'cat_pipeline__EducationField_Medical',
       'cat_pipeline__EducationField_Other',
       'cat_pipeline__EducationField_Technical Degree',
       'cat_pipeline__JobRole_Healthcare Representative',
       'cat_pipeline__JobRole_Human Resources',
       'cat_pipeline__JobRole_Laboratory Technician',
       'cat_pipeline__JobRole_Manager',
       'cat_pipeline__JobRole_Manufacturing Director',
       'cat_pipeline__JobRole_Research Director',
       'cat_pipeline__JobRole_Research Scientist',
       'cat_pipeline__JobRole_Sales Executive',
       'cat_pipeline__JobRole_Sales Representative',
       'cat_pipeline__MaritalStatus_Divorced',
       'cat_pipeline__MaritalStatus_Married',
       'cat_pipeline__MaritalStatus_Single', 'cat_pipeline__OverTime_No',
       'cat_pipeline__OverTime_Yes'], dtype=object)
```

[29]:
```
xtrain_transformed.head()
```

[29]:
```
   num_pipeline__Age  num_pipeline__JobInvolvement  num_pipeline__JobLevel  \
0           1.090194                      1.795282                1.762189
1          -1.634828                      0.373564               -0.986265
2           0.981193                      0.373564                1.762189
3          -1.307825                      0.373564               -0.986265
4           0.654191                      0.373564               -0.070114
```

```
   num_pipeline__JobSatisfaction  num_pipeline__MonthlyIncome  \
0                      -0.647997                     2.026752
1                       1.153526                    -0.864408
2                       0.252765                     2.347706
3                       0.252765                    -0.956202
4                       0.252765                    -0.185956

   num_pipeline__StockOptionLevel  num_pipeline__TotalWorkingYears  \
0                        2.613100                         2.261482
1                        0.247430                        -1.072675
2                        0.247430                         1.492061
3                       -0.935405                        -0.559727
4                        0.247430                        -0.175017

   num_pipeline__YearsAtCompany  num_pipeline__YearsInCurrentRole  \
0                     -0.665706                         -0.625365
1                     -0.830071                         -0.905635
2                      0.813578                          1.336527
3                     -0.008246                         -0.064824
4                      0.156119                          0.775986

   num_pipeline__YearsWithCurrManager  …  \
0                           -0.616406  …
1                           -0.897047  …
2                            1.348076  …
3                            0.506155  …
4                            0.786795  …

   cat_pipeline__JobRole_Manufacturing Director  \
0                                           0.0
1                                           0.0
2                                           0.0
3                                           0.0
4                                           0.0

   cat_pipeline__JobRole_Research Director  \
0                                      0.0
1                                      0.0
2                                      0.0
3                                      0.0
4                                      0.0

   cat_pipeline__JobRole_Research Scientist  \
0                                       0.0
1                                       0.0
2                                       0.0
3                                       0.0
```

```
4                                    0.0

   cat_pipeline__JobRole_Sales Executive  \
0                                   0.0
1                                   0.0
2                                   0.0
3                                   0.0
4                                   0.0

   cat_pipeline__JobRole_Sales Representative  \
0                                    0.000000
1                                    0.000000
2                                    0.000000
3                                    4.544641
4                                    0.000000

   cat_pipeline__MaritalStatus_Divorced  cat_pipeline__MaritalStatus_Married  \
0                              2.399905                             0.000000
1                              0.000000                             2.006697
2                              0.000000                             2.006697
3                              0.000000                             2.006697
4                              2.399905                             0.000000

   cat_pipeline__MaritalStatus_Single  cat_pipeline__OverTime_No  \
0                                 0.0                   2.205793
1                                 0.0                   2.205793
2                                 0.0                   2.205793
3                                 0.0                   2.205793
4                                 0.0                   0.000000

   cat_pipeline__OverTime_Yes
0                    0.000000
1                    0.000000
2                    0.000000
3                    0.000000
4                    2.205793

[5 rows x 33 columns]
```

[30]:
```
xtest_transformed.head()
```

[30]:
```
   num_pipeline__Age  num_pipeline__JobInvolvement  num_pipeline__JobLevel  \
0          -1.416826                      0.373564               -0.986265
1           0.763191                      1.795282               -0.986265
2          -0.653820                     -1.048155                0.846038
3           0.763191                      1.795282                2.678340
4          -0.108815                     -1.048155               -0.986265
```

```
   num_pipeline__JobSatisfaction  num_pipeline__MonthlyIncome  \
0                      -0.647997                    -0.969745
1                       1.153526                    -0.974474
2                       1.153526                     1.077650
3                      -1.548758                     2.718533
4                       0.252765                    -0.678457

   num_pipeline__StockOptionLevel  num_pipeline__TotalWorkingYears  \
0                        0.247430                        -1.329148
1                        0.247430                        -0.175017
2                        0.247430                        -0.175017
3                        1.430265                         1.876772
4                        0.247430                        -1.200911

   num_pipeline__YearsAtCompany  num_pipeline__YearsInCurrentRole  \
0                     -0.994436                         -1.185905
1                      0.484849                          0.215446
2                     -0.336976                         -0.064824
3                      2.950322                          1.336527
4                     -0.994436                         -1.185905

   num_pipeline__YearsWithCurrManager  …  \
0                           -1.177687  …
1                            0.786795  …
2                           -0.897047  …
3                            2.470637  …
4                           -1.177687  …

   cat_pipeline__JobRole_Manufacturing Director  \
0                                           0.0
1                                           0.0
2                                           0.0
3                                           0.0
4                                           0.0

   cat_pipeline__JobRole_Research Director  \
0                                      0.0
1                                      0.0
2                                      0.0
3                                      0.0
4                                      0.0

   cat_pipeline__JobRole_Research Scientist  \
0                                  0.000000
1                                  2.564289
2                                  0.000000
```

```
3                                  0.000000
4                                  2.564289

   cat_pipeline__JobRole_Sales Executive  \
0                                    0.0
1                                    0.0
2                                    0.0
3                                    0.0
4                                    0.0

   cat_pipeline__JobRole_Sales Representative  \
0                                    4.544641
1                                    0.000000
2                                    0.000000
3                                    0.000000
4                                    0.000000

   cat_pipeline__MaritalStatus_Divorced  cat_pipeline__MaritalStatus_Married  \
0                              0.000000                             2.006697
1                              0.000000                             2.006697
2                              2.399905                             0.000000
3                              2.399905                             0.000000
4                              0.000000                             2.006697

   cat_pipeline__MaritalStatus_Single  cat_pipeline__OverTime_No  \
0                                 0.0                   2.205793
1                                 0.0                   2.205793
2                                 0.0                   2.205793
3                                 0.0                   2.205793
4                                 0.0                   0.000000

   cat_pipeline__OverTime_Yes
0                    0.000000
1                    0.000000
2                    0.000000
3                    0.000000
4                    2.205793

[5 rows x 33 columns]
```

0.3.12 Training Random Forest

[31]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=100,max_depth=8,random_state=21)
rfc.fit(xtrain_transformed,ytrain)
```

[31]:
```
RandomForestClassifier(max_depth=8, random_state=21)
```

```python
[32]: from sklearn.model_selection import cross_val_score, StratifiedKFold
skf = StratifiedKFold(n_splits=5, shuffle=True)
scores =␣
 ↪cross_val_score(rfc,xtrain_transformed,ytrain,cv=skf,scoring='accuracy')
scores
```

```
[32]: array([0.84322034, 0.84255319, 0.85957447, 0.86382979, 0.86808511])
```

```python
[33]: import numpy as np
np.mean(scores)
```

```
[33]: 0.855452578434908
```

0.3.13 Hyperparameter Tuning

```python
[34]: parameters = {
    'n_estimators':[10,50,100,200,300,400],
    'max_depth':[4,5,6,7,8,9,10],
    'min_samples_leaf':[2,3,4,5,6],
    'min_samples_split':[2,5,10]
}
```

```python
[35]: from sklearn.model_selection import RandomizedSearchCV
rscv = RandomizedSearchCV(RandomForestClassifier(),
                          param_distributions=parameters,
                          cv=skf,
                          n_iter=50,
                          scoring='accuracy',
                          verbose=3)
```

```python
[36]: rscv.fit(xtrain_transformed,ytrain)
```

```
Fitting 5 folds for each of 50 candidates, totalling 250 fits
[CV 1/5] END max_depth=10, min_samples_leaf=5, min_samples_split=10,
n_estimators=300;, score=0.852 total time=   0.4s
[CV 2/5] END max_depth=10, min_samples_leaf=5, min_samples_split=10,
n_estimators=300;, score=0.864 total time=   0.4s
[CV 3/5] END max_depth=10, min_samples_leaf=5, min_samples_split=10,
n_estimators=300;, score=0.851 total time=   0.4s
[CV 4/5] END max_depth=10, min_samples_leaf=5, min_samples_split=10,
n_estimators=300;, score=0.860 total time=   0.4s
[CV 5/5] END max_depth=10, min_samples_leaf=5, min_samples_split=10,
n_estimators=300;, score=0.860 total time=   0.5s
[CV 1/5] END max_depth=8, min_samples_leaf=6, min_samples_split=10,
n_estimators=200;, score=0.864 total time=   0.3s
[CV 2/5] END max_depth=8, min_samples_leaf=6, min_samples_split=10,
n_estimators=200;, score=0.860 total time=   0.3s
```

```
[CV 3/5] END max_depth=8, min_samples_leaf=6, min_samples_split=10,
n_estimators=200;, score=0.855 total time=   0.3s
[CV 4/5] END max_depth=8, min_samples_leaf=6, min_samples_split=10,
n_estimators=200;, score=0.860 total time=   0.3s
[CV 5/5] END max_depth=8, min_samples_leaf=6, min_samples_split=10,
n_estimators=200;, score=0.860 total time=   0.2s
[CV 1/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.864 total time=   0.0s
[CV 2/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.860 total time=   0.0s
[CV 3/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.851 total time=   0.0s
[CV 4/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.847 total time=   0.0s
[CV 5/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.860 total time=   0.0s
[CV 1/5] END max_depth=6, min_samples_leaf=6, min_samples_split=2,
n_estimators=10;, score=0.869 total time=   0.0s
[CV 2/5] END max_depth=6, min_samples_leaf=6, min_samples_split=2,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 3/5] END max_depth=6, min_samples_leaf=6, min_samples_split=2,
n_estimators=10;, score=0.851 total time=   0.0s
[CV 4/5] END max_depth=6, min_samples_leaf=6, min_samples_split=2,
n_estimators=10;, score=0.860 total time=   0.0s
[CV 5/5] END max_depth=6, min_samples_leaf=6, min_samples_split=2,
n_estimators=10;, score=0.860 total time=   0.0s
[CV 1/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=200;, score=0.856 total time=   0.2s
[CV 2/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=200;, score=0.860 total time=   0.2s
[CV 3/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=200;, score=0.855 total time=   0.3s
[CV 4/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=200;, score=0.843 total time=   0.2s
[CV 5/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=200;, score=0.860 total time=   0.2s
[CV 1/5] END max_depth=10, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.869 total time=   0.4s
[CV 2/5] END max_depth=10, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.860 total time=   0.4s
[CV 3/5] END max_depth=10, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.851 total time=   0.5s
[CV 4/5] END max_depth=10, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.851 total time=   0.4s
[CV 5/5] END max_depth=10, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.864 total time=   0.5s
[CV 1/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.864 total time=   0.4s
```

```
[CV 2/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.864 total time=   0.4s
[CV 3/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.860 total time=   0.5s
[CV 4/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.851 total time=   0.4s
[CV 5/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=300;, score=0.860 total time=   0.4s
[CV 1/5] END max_depth=9, min_samples_leaf=3, min_samples_split=2,
n_estimators=100;, score=0.873 total time=   0.1s
[CV 2/5] END max_depth=9, min_samples_leaf=3, min_samples_split=2,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 3/5] END max_depth=9, min_samples_leaf=3, min_samples_split=2,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 4/5] END max_depth=9, min_samples_leaf=3, min_samples_split=2,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 5/5] END max_depth=9, min_samples_leaf=3, min_samples_split=2,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 1/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.864 total time=   0.2s
[CV 2/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.864 total time=   0.3s
[CV 3/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.855 total time=   0.3s
[CV 4/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.851 total time=   0.3s
[CV 5/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.860 total time=   0.3s
[CV 1/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 2/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 3/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 4/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=100;, score=0.851 total time=   0.2s
[CV 5/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 1/5] END max_depth=9, min_samples_leaf=5, min_samples_split=2,
n_estimators=50;, score=0.869 total time=   0.0s
[CV 2/5] END max_depth=9, min_samples_leaf=5, min_samples_split=2,
n_estimators=50;, score=0.855 total time=   0.0s
[CV 3/5] END max_depth=9, min_samples_leaf=5, min_samples_split=2,
n_estimators=50;, score=0.851 total time=   0.0s
[CV 4/5] END max_depth=9, min_samples_leaf=5, min_samples_split=2,
n_estimators=50;, score=0.843 total time=   0.0s
[CV 5/5] END max_depth=9, min_samples_leaf=5, min_samples_split=2,
n_estimators=50;, score=0.860 total time=   0.0s
```

```
[CV 1/5] END max_depth=5, min_samples_leaf=2, min_samples_split=10,
n_estimators=100;, score=0.856 total time=   0.1s
[CV 2/5] END max_depth=5, min_samples_leaf=2, min_samples_split=10,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 3/5] END max_depth=5, min_samples_leaf=2, min_samples_split=10,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 4/5] END max_depth=5, min_samples_leaf=2, min_samples_split=10,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 5/5] END max_depth=5, min_samples_leaf=2, min_samples_split=10,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 1/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.856 total time=   0.5s
[CV 2/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.860 total time=   0.5s
[CV 3/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.855 total time=   0.5s
[CV 4/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.847 total time=   0.5s
[CV 5/5] END max_depth=5, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.860 total time=   0.5s
[CV 1/5] END max_depth=10, min_samples_leaf=3, min_samples_split=2,
n_estimators=300;, score=0.864 total time=   0.5s
[CV 2/5] END max_depth=10, min_samples_leaf=3, min_samples_split=2,
n_estimators=300;, score=0.860 total time=   0.4s
[CV 3/5] END max_depth=10, min_samples_leaf=3, min_samples_split=2,
n_estimators=300;, score=0.847 total time=   0.5s
[CV 4/5] END max_depth=10, min_samples_leaf=3, min_samples_split=2,
n_estimators=300;, score=0.855 total time=   0.5s
[CV 5/5] END max_depth=10, min_samples_leaf=3, min_samples_split=2,
n_estimators=300;, score=0.860 total time=   0.5s
[CV 1/5] END max_depth=7, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.860 total time=   0.2s
[CV 2/5] END max_depth=7, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.855 total time=   0.2s
[CV 3/5] END max_depth=7, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.847 total time=   0.2s
[CV 4/5] END max_depth=7, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.860 total time=   0.3s
[CV 5/5] END max_depth=7, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.864 total time=   0.3s
[CV 1/5] END max_depth=9, min_samples_leaf=2, min_samples_split=10,
n_estimators=50;, score=0.869 total time=   0.0s
[CV 2/5] END max_depth=9, min_samples_leaf=2, min_samples_split=10,
n_estimators=50;, score=0.868 total time=   0.0s
[CV 3/5] END max_depth=9, min_samples_leaf=2, min_samples_split=10,
n_estimators=50;, score=0.847 total time=   0.0s
[CV 4/5] END max_depth=9, min_samples_leaf=2, min_samples_split=10,
n_estimators=50;, score=0.851 total time=   0.0s
```

```
[CV 5/5] END max_depth=9, min_samples_leaf=2, min_samples_split=10,
n_estimators=50;, score=0.864 total time=   0.0s
[CV 1/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=50;, score=0.869 total time=   0.0s
[CV 2/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=50;, score=0.855 total time=   0.0s
[CV 3/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=50;, score=0.855 total time=   0.0s
[CV 4/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=50;, score=0.847 total time=   0.0s
[CV 5/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=50;, score=0.864 total time=   0.0s
[CV 1/5] END max_depth=7, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.860 total time=   0.3s
[CV 2/5] END max_depth=7, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.864 total time=   0.2s
[CV 3/5] END max_depth=7, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.855 total time=   0.2s
[CV 4/5] END max_depth=7, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.851 total time=   0.3s
[CV 5/5] END max_depth=7, min_samples_leaf=5, min_samples_split=5,
n_estimators=200;, score=0.860 total time=   0.3s
[CV 1/5] END max_depth=6, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.864 total time=   0.4s
[CV 2/5] END max_depth=6, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.860 total time=   0.4s
[CV 3/5] END max_depth=6, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.855 total time=   0.4s
[CV 4/5] END max_depth=6, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.847 total time=   0.4s
[CV 5/5] END max_depth=6, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.864 total time=   0.4s
[CV 1/5] END max_depth=10, min_samples_leaf=3, min_samples_split=10,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 2/5] END max_depth=10, min_samples_leaf=3, min_samples_split=10,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 3/5] END max_depth=10, min_samples_leaf=3, min_samples_split=10,
n_estimators=100;, score=0.851 total time=   0.1s
[CV 4/5] END max_depth=10, min_samples_leaf=3, min_samples_split=10,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 5/5] END max_depth=10, min_samples_leaf=3, min_samples_split=10,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 1/5] END max_depth=9, min_samples_leaf=4, min_samples_split=5,
n_estimators=400;, score=0.864 total time=   0.6s
[CV 2/5] END max_depth=9, min_samples_leaf=4, min_samples_split=5,
n_estimators=400;, score=0.855 total time=   0.6s
[CV 3/5] END max_depth=9, min_samples_leaf=4, min_samples_split=5,
n_estimators=400;, score=0.851 total time=   0.6s
```

```
[CV 4/5] END max_depth=9, min_samples_leaf=4, min_samples_split=5,
n_estimators=400;, score=0.851 total time=   0.6s
[CV 5/5] END max_depth=9, min_samples_leaf=4, min_samples_split=5,
n_estimators=400;, score=0.864 total time=   0.6s
[CV 1/5] END max_depth=9, min_samples_leaf=3, min_samples_split=5,
n_estimators=10;, score=0.869 total time=   0.0s
[CV 2/5] END max_depth=9, min_samples_leaf=3, min_samples_split=5,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 3/5] END max_depth=9, min_samples_leaf=3, min_samples_split=5,
n_estimators=10;, score=0.847 total time=   0.0s
[CV 4/5] END max_depth=9, min_samples_leaf=3, min_samples_split=5,
n_estimators=10;, score=0.847 total time=   0.0s
[CV 5/5] END max_depth=9, min_samples_leaf=3, min_samples_split=5,
n_estimators=10;, score=0.872 total time=   0.0s
[CV 1/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=300;, score=0.856 total time=   0.4s
[CV 2/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=300;, score=0.855 total time=   0.4s
[CV 3/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=300;, score=0.851 total time=   0.4s
[CV 4/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=300;, score=0.851 total time=   0.4s
[CV 5/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=300;, score=0.864 total time=   0.4s
[CV 1/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 2/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 3/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=100;, score=0.851 total time=   0.1s
[CV 4/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=100;, score=0.851 total time=   0.1s
[CV 5/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 1/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=300;, score=0.864 total time=   0.5s
[CV 2/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=300;, score=0.864 total time=   0.5s
[CV 3/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=300;, score=0.847 total time=   0.5s
[CV 4/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=300;, score=0.847 total time=   0.5s
[CV 5/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=300;, score=0.860 total time=   0.5s
[CV 1/5] END max_depth=6, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.856 total time=   0.1s
[CV 2/5] END max_depth=6, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.864 total time=   0.1s
```

```
[CV 3/5] END max_depth=6, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 4/5] END max_depth=6, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 5/5] END max_depth=6, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 1/5] END max_depth=10, min_samples_leaf=6, min_samples_split=2,
n_estimators=400;, score=0.864 total time=   0.6s
[CV 2/5] END max_depth=10, min_samples_leaf=6, min_samples_split=2,
n_estimators=400;, score=0.864 total time=   0.6s
[CV 3/5] END max_depth=10, min_samples_leaf=6, min_samples_split=2,
n_estimators=400;, score=0.855 total time=   0.6s
[CV 4/5] END max_depth=10, min_samples_leaf=6, min_samples_split=2,
n_estimators=400;, score=0.851 total time=   0.6s
[CV 5/5] END max_depth=10, min_samples_leaf=6, min_samples_split=2,
n_estimators=400;, score=0.860 total time=   0.6s
[CV 1/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.856 total time=   0.4s
[CV 2/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.855 total time=   0.4s
[CV 3/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.855 total time=   0.4s
[CV 4/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.851 total time=   0.4s
[CV 5/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=300;, score=0.860 total time=   0.4s
[CV 1/5] END max_depth=4, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.864 total time=   0.0s
[CV 2/5] END max_depth=4, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.851 total time=   0.0s
[CV 3/5] END max_depth=4, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 4/5] END max_depth=4, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.843 total time=   0.0s
[CV 5/5] END max_depth=4, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.860 total time=   0.0s
[CV 1/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.852 total time=   0.0s
[CV 2/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.860 total time=   0.0s
[CV 3/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.851 total time=   0.0s
[CV 4/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.855 total time=   0.0s
[CV 5/5] END max_depth=6, min_samples_leaf=2, min_samples_split=5,
n_estimators=50;, score=0.860 total time=   0.0s
[CV 1/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=400;, score=0.869 total time=   0.5s
```

```
[CV 2/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=400;, score=0.855 total time=   0.5s
[CV 3/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=400;, score=0.855 total time=   0.5s
[CV 4/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=400;, score=0.843 total time=   0.5s
[CV 5/5] END max_depth=4, min_samples_leaf=3, min_samples_split=2,
n_estimators=400;, score=0.864 total time=   0.5s
[CV 1/5] END max_depth=7, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 2/5] END max_depth=7, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 3/5] END max_depth=7, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 4/5] END max_depth=7, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 5/5] END max_depth=7, min_samples_leaf=5, min_samples_split=10,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 1/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=100;, score=0.877 total time=   0.1s
[CV 2/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 3/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 4/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=100;, score=0.851 total time=   0.1s
[CV 5/5] END max_depth=5, min_samples_leaf=2, min_samples_split=5,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 1/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=100;, score=0.856 total time=   0.1s
[CV 2/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 3/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=100;, score=0.851 total time=   0.1s
[CV 4/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=100;, score=0.851 total time=   0.1s
[CV 5/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 1/5] END max_depth=4, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.864 total time=   0.0s
[CV 2/5] END max_depth=4, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.860 total time=   0.0s
[CV 3/5] END max_depth=4, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 4/5] END max_depth=4, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.855 total time=   0.0s
[CV 5/5] END max_depth=4, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.855 total time=   0.1s
```

```
[CV 1/5] END max_depth=4, min_samples_leaf=3, min_samples_split=10,
n_estimators=200;, score=0.864 total time=   0.2s
[CV 2/5] END max_depth=4, min_samples_leaf=3, min_samples_split=10,
n_estimators=200;, score=0.860 total time=   0.2s
[CV 3/5] END max_depth=4, min_samples_leaf=3, min_samples_split=10,
n_estimators=200;, score=0.851 total time=   0.2s
[CV 4/5] END max_depth=4, min_samples_leaf=3, min_samples_split=10,
n_estimators=200;, score=0.847 total time=   0.2s
[CV 5/5] END max_depth=4, min_samples_leaf=3, min_samples_split=10,
n_estimators=200;, score=0.855 total time=   0.2s
[CV 1/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=10;, score=0.873 total time=   0.0s
[CV 2/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 3/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=10;, score=0.860 total time=   0.0s
[CV 4/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=10;, score=0.834 total time=   0.0s
[CV 5/5] END max_depth=9, min_samples_leaf=4, min_samples_split=2,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 1/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=200;, score=0.856 total time=   0.2s
[CV 2/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=200;, score=0.864 total time=   0.2s
[CV 3/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=200;, score=0.855 total time=   0.2s
[CV 4/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=200;, score=0.851 total time=   0.2s
[CV 5/5] END max_depth=8, min_samples_leaf=4, min_samples_split=10,
n_estimators=200;, score=0.855 total time=   0.2s
[CV 1/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=400;, score=0.856 total time=   0.6s
[CV 2/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=400;, score=0.864 total time=   0.6s
[CV 3/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=400;, score=0.851 total time=   0.6s
[CV 4/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=400;, score=0.855 total time=   0.6s
[CV 5/5] END max_depth=10, min_samples_leaf=4, min_samples_split=2,
n_estimators=400;, score=0.864 total time=   0.6s
[CV 1/5] END max_depth=6, min_samples_leaf=2, min_samples_split=2,
n_estimators=10;, score=0.852 total time=   0.0s
[CV 2/5] END max_depth=6, min_samples_leaf=2, min_samples_split=2,
n_estimators=10;, score=0.851 total time=   0.0s
[CV 3/5] END max_depth=6, min_samples_leaf=2, min_samples_split=2,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 4/5] END max_depth=6, min_samples_leaf=2, min_samples_split=2,
n_estimators=10;, score=0.834 total time=   0.0s
```

```
[CV 5/5] END max_depth=6, min_samples_leaf=2, min_samples_split=2,
n_estimators=10;, score=0.872 total time=   0.0s
[CV 1/5] END max_depth=8, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.856 total time=   0.6s
[CV 2/5] END max_depth=8, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.864 total time=   0.6s
[CV 3/5] END max_depth=8, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.855 total time=   0.6s
[CV 4/5] END max_depth=8, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.855 total time=   0.6s
[CV 5/5] END max_depth=8, min_samples_leaf=5, min_samples_split=2,
n_estimators=400;, score=0.860 total time=   0.6s
[CV 1/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.856 total time=   0.0s
[CV 2/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.868 total time=   0.0s
[CV 3/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 4/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.864 total time=   0.0s
[CV 5/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 1/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=50;, score=0.873 total time=   0.0s
[CV 2/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=50;, score=0.855 total time=   0.0s
[CV 3/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=50;, score=0.855 total time=   0.0s
[CV 4/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=50;, score=0.860 total time=   0.0s
[CV 5/5] END max_depth=8, min_samples_leaf=4, min_samples_split=5,
n_estimators=50;, score=0.860 total time=   0.0s
[CV 1/5] END max_depth=5, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.856 total time=   0.2s
[CV 2/5] END max_depth=5, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.864 total time=   0.2s
[CV 3/5] END max_depth=5, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.855 total time=   0.2s
[CV 4/5] END max_depth=5, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.851 total time=   0.2s
[CV 5/5] END max_depth=5, min_samples_leaf=3, min_samples_split=2,
n_estimators=200;, score=0.864 total time=   0.2s
[CV 1/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=10;, score=0.864 total time=   0.0s
[CV 2/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=10;, score=0.864 total time=   0.0s
[CV 3/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=10;, score=0.855 total time=   0.0s
```

```
[CV 4/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=10;, score=0.847 total time=   0.0s
[CV 5/5] END max_depth=6, min_samples_leaf=5, min_samples_split=2,
n_estimators=10;, score=0.868 total time=   0.0s
[CV 1/5] END max_depth=4, min_samples_leaf=6, min_samples_split=10,
n_estimators=10;, score=0.869 total time=   0.0s
[CV 2/5] END max_depth=4, min_samples_leaf=6, min_samples_split=10,
n_estimators=10;, score=0.847 total time=   0.0s
[CV 3/5] END max_depth=4, min_samples_leaf=6, min_samples_split=10,
n_estimators=10;, score=0.851 total time=   0.0s
[CV 4/5] END max_depth=4, min_samples_leaf=6, min_samples_split=10,
n_estimators=10;, score=0.843 total time=   0.0s
[CV 5/5] END max_depth=4, min_samples_leaf=6, min_samples_split=10,
n_estimators=10;, score=0.860 total time=   0.0s
[CV 1/5] END max_depth=5, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.852 total time=   0.0s
[CV 2/5] END max_depth=5, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.847 total time=   0.0s
[CV 3/5] END max_depth=5, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.834 total time=   0.0s
[CV 4/5] END max_depth=5, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.843 total time=   0.0s
[CV 5/5] END max_depth=5, min_samples_leaf=4, min_samples_split=5,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 1/5] END max_depth=5, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.860 total time=   0.0s
[CV 2/5] END max_depth=5, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.864 total time=   0.0s
[CV 3/5] END max_depth=5, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 4/5] END max_depth=5, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.847 total time=   0.0s
[CV 5/5] END max_depth=5, min_samples_leaf=5, min_samples_split=5,
n_estimators=10;, score=0.855 total time=   0.0s
[CV 1/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 2/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 3/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.860 total time=   0.1s
[CV 4/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 5/5] END max_depth=8, min_samples_leaf=5, min_samples_split=5,
n_estimators=100;, score=0.864 total time=   0.1s
[CV 1/5] END max_depth=6, min_samples_leaf=4, min_samples_split=5,
n_estimators=100;, score=0.873 total time=   0.1s
[CV 2/5] END max_depth=6, min_samples_leaf=4, min_samples_split=5,
n_estimators=100;, score=0.864 total time=   0.1s
```

```
[CV 3/5] END max_depth=6, min_samples_leaf=4, min_samples_split=5,
n_estimators=100;, score=0.855 total time=   0.1s
[CV 4/5] END max_depth=6, min_samples_leaf=4, min_samples_split=5,
n_estimators=100;, score=0.851 total time=   0.1s
[CV 5/5] END max_depth=6, min_samples_leaf=4, min_samples_split=5,
n_estimators=100;, score=0.860 total time=   0.1s
```

[36]:
```
RandomizedSearchCV(cv=StratifiedKFold(n_splits=5, random_state=None,
shuffle=True),
                   estimator=RandomForestClassifier(), n_iter=50,
                   param_distributions={'max_depth': [4, 5, 6, 7, 8, 9, 10],
                                        'min_samples_leaf': [2, 3, 4, 5, 6],
                                        'min_samples_split': [2, 5, 10],
                                        'n_estimators': [10, 50, 100, 200, 300,
                                                         400]},
                   scoring='accuracy', verbose=3)
```

[37]:
```
rscv.best_params_
```

[37]:
```
{'n_estimators': 100,
 'min_samples_split': 2,
 'min_samples_leaf': 3,
 'max_depth': 9}
```

[38]:
```
rscv.best_score_
```

[38]:
```
0.8630869094843131
```

[39]:
```
best_rfc = rscv.best_estimator_
best_rfc
```

[39]:
```
RandomForestClassifier(max_depth=9, min_samples_leaf=3)
```

0.3.14 Evaluate model on test data

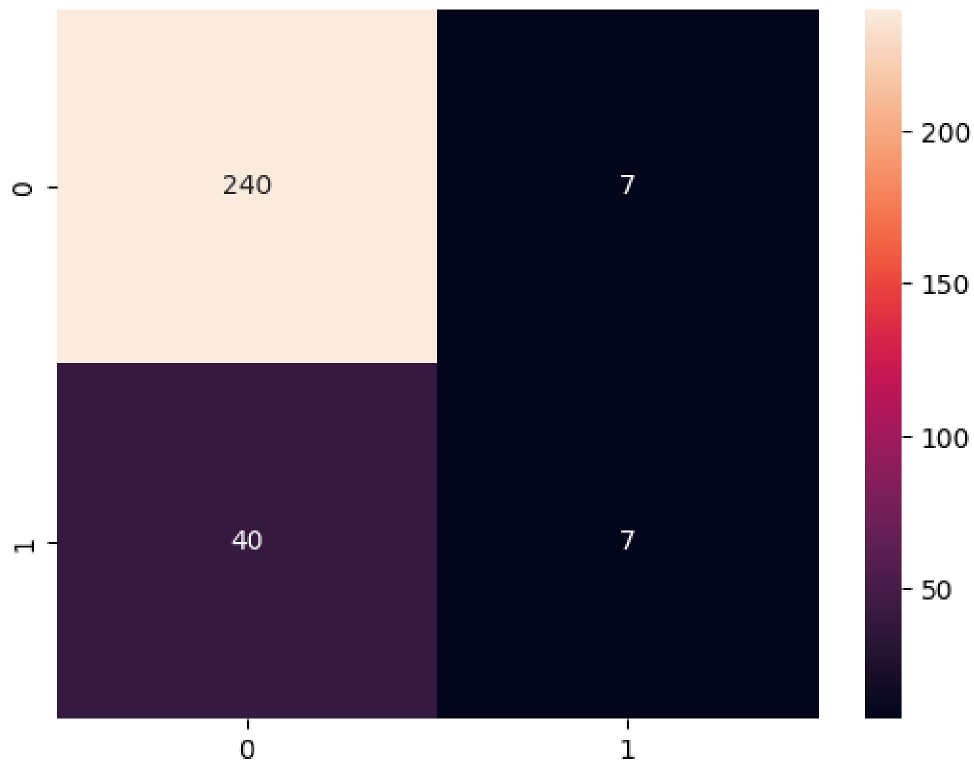
[40]:
```
ypred_test = best_rfc.predict(xtest_transformed)
ypred_test
```

[40]:
```
array([1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

[41]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns
cf = confusion_matrix(ytest, ypred_test)
sns.heatmap(cf,annot=True,fmt='d')
```

[41]: <Axes: >



[42]:
```
from sklearn.metrics import accuracy_score
acc = accuracy_score(ytest, ypred_test)
print(f'Accuracy on testing data is {acc:.4f}')
```

```
Accuracy on testing data is 0.8401
```

0.3.15 Insights

1. Above model has accuracy of 0.8503 on testing data
2. However above data has imbalance data on target

3. To deal with imbalanced techniques such as SMOTE (Synthetic Minority Oversampling TEchnique)
4. Feature selection can also be avoided in above data and performance should be checked with all features as well

#

Question 2

0.4 ## Question 2 : Build a pipeline that includes random forest classifier and a logistic regression classifier , and then voting classifier to combine their predictions. Train the pipeline on iris dataset and evaluate its accuracy

0.5 Answer :

0.5.1 Load Iris Dataset

[43]:
```
from sklearn.datasets import load_iris
X,Y = load_iris(return_X_y=True)
```

[44]:
```
X.shape
```

[44]: (150, 4)

[45]:
```
Y.shape
```

[45]: (150,)

0.5.2 Train Test Split

[46]:
```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(X,Y,test_size=0.
↪3,random_state=42)
```

[47]:
```
xtrain.shape
```

[47]: (105, 4)

[48]:
```
xtest.shape
```

[48]: (45, 4)

0.5.3 Standard Scaling

[49]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
xtrain = scaler.fit_transform(xtrain)
xtest = scaler.transform(xtest)
```

0.5.4 Create pipeline

[58]:
```
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import VotingClassifier
from sklearn.preprocessing import StandardScaler

# Defining Base models
rfc = RandomForestClassifier(n_estimators=100,max_depth=4)
lr = LogisticRegression(C=1.0)

# Voting Classifier Pipeline
vc = VotingClassifier(estimators=[('rfc',rfc),
                                  ('lr',lr)],
                                  voting='soft')
```

[59]:
```
vc.fit(xtrain,ytrain)
```

[59]:
```
VotingClassifier(estimators=[('rfc', RandomForestClassifier(max_depth=4)),
                             ('lr', LogisticRegression())],
                 voting='soft')
```

0.5.5 Predicting the test results

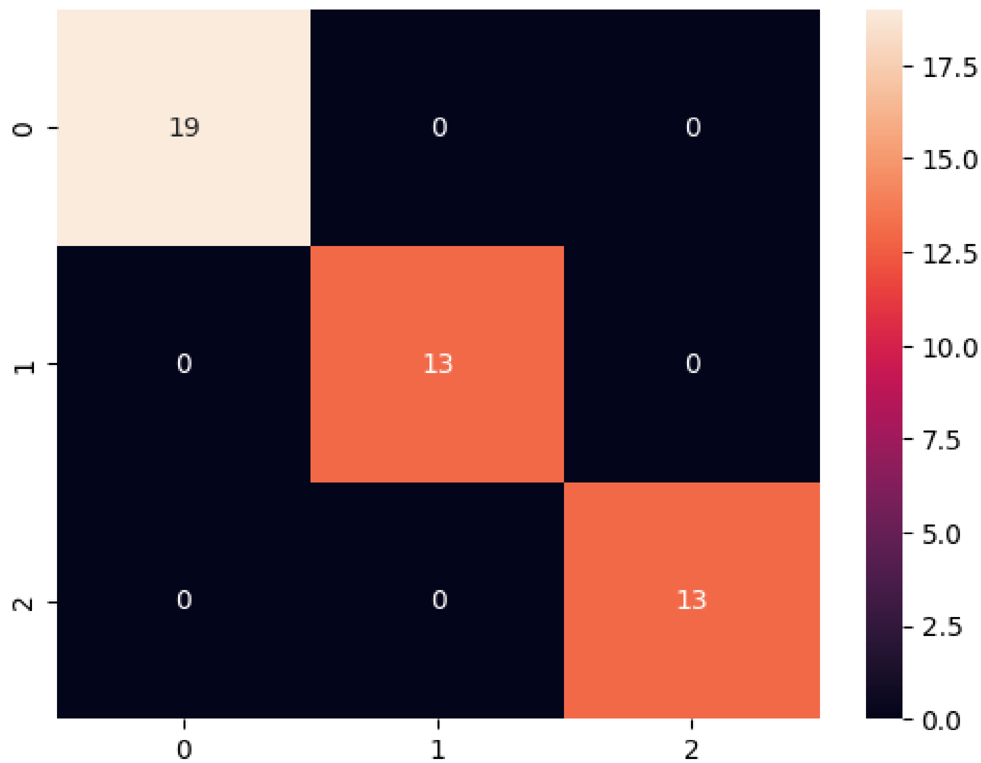
[61]:
```
ypred_test = vc.predict(xtest)
ypred_test
```

[61]:
```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0, 0, 0, 1, 0, 0, 2, 1, 0, 0, 0, 2, 1, 1, 0,
       0])
```

0.5.6 Evaluating the model on test data

[62]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns
cf = confusion_matrix(ytest,ypred_test)
sns.heatmap(cf,annot=True,fmt='d')
```

[62]:
```
<Axes: >
```



[63]:
```
# Classification Report
from sklearn.metrics import classification_report
print(classification_report(ytest,ypred_test))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       1.00      1.00      1.00        13
           2       1.00      1.00      1.00        13

    accuracy                           1.00        45
   macro avg       1.00      1.00      1.00        45
weighted avg       1.00      1.00      1.00        45
```

[64]:
```
# Accuracy Score
from sklearn.metrics import accuracy_score
acc = accuracy_score(ytest,ypred_test)
print(f'Accuracy on Final Voting Classifier model is {acc*100:.2f}%')
```

```
Accuracy on Final Voting Classifier model is 100.00%
```