

```
#Program-10
"""
Implement AdaBoost ensemble method on a given dataset.(Iris dataset)
"""
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import AdaBoostClassifier
#import warnings warnings.filterwarnings("ignore")
```

```
# Reading the dataset from the csv file # separator is a vertical
line, as seen in the dataset
data = pd.read_csv("Iris.csv")
# Printing the shape of the dataset
print(data.shape)
```

```
(150, 6)
```

```
data.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	
Species						
0	1	5.1	3.5	1.4	0.2	Iris-
						setosa
1	2	4.9	3.0	1.4	0.2	Iris-
						setosa
2	3	4.7	3.2	1.3	0.2	Iris-
						setosa
3	4	4.6	3.1	1.5	0.2	Iris-
						setosa
4	5	5.0	3.6	1.4	0.2	Iris-
						setosa

```
data = data.drop('Id',axis=1)
X = data.iloc[:, :-1]
y = data.iloc[:, -1]
print("Shape of X is %s and shape of y is %s"%(X.shape,y.shape))
```

```
Shape of X is (150, 4) and shape of y is (150,)
```

```
total_classes = y.nunique()
print("Number of unique species in dataset are: ",total_classes)
```

```
Number of unique species in dataset are: 3
```

```
distribution = y.value_counts()
print(distribution)
```

```
Species
Iris-setosa      50
Iris-versicolor  50
```

```
Iris-virginica      50
Name: count, dtype: int64
```

```
X_train, X_val, Y_train, Y_val = train_test_split( X, y,
test_size=0.25, random_state=42)
```

```
# Creating adaboost classifier model
```

```
adb = AdaBoostClassifier()
adb_model = adb.fit(X_train,Y_train)
```

```
print("The accuracy of the model on validation set is",
adb_model.score(X_val,Y_val))
```

The accuracy of the model on validation set is 1.0

```
from sklearn.metrics import accuracy_score
```

```
# Make predictions on the testing data
```

```
y_pred = adb_model.predict(X_val)
```

```
# Calculate the accuracy of the model
```

```
accuracy = accuracy_score(Y_val, y_pred)
```

```
print(f"The Accuracy of Prediction on Iris Flower is: {accuracy}")
```

The Accuracy of Prediction on Iris Flower is: 1.0

```
# Create a DataFrame to display actual and predicted values
```

```
df = pd.DataFrame({'Actual': Y_val, 'Predicted': y_pred})
```

```
# Print the table
```

```
print(df)
```

	Actual	Predicted
73	Iris-versicolor	Iris-versicolor
18	Iris-setosa	Iris-setosa
118	Iris-virginica	Iris-virginica
78	Iris-versicolor	Iris-versicolor
76	Iris-versicolor	Iris-versicolor
31	Iris-setosa	Iris-setosa
64	Iris-versicolor	Iris-versicolor
141	Iris-virginica	Iris-virginica
68	Iris-versicolor	Iris-versicolor
82	Iris-versicolor	Iris-versicolor
110	Iris-virginica	Iris-virginica
12	Iris-setosa	Iris-setosa
36	Iris-setosa	Iris-setosa
9	Iris-setosa	Iris-setosa
19	Iris-setosa	Iris-setosa
56	Iris-versicolor	Iris-versicolor
104	Iris-virginica	Iris-virginica
69	Iris-versicolor	Iris-versicolor

55	Iris-versicolor	Iris-versicolor
132	Iris-virginica	Iris-virginica
29	Iris-setosa	Iris-setosa
127	Iris-virginica	Iris-virginica
26	Iris-setosa	Iris-setosa
128	Iris-virginica	Iris-virginica
131	Iris-virginica	Iris-virginica
145	Iris-virginica	Iris-virginica
108	Iris-virginica	Iris-virginica
143	Iris-virginica	Iris-virginica
45	Iris-setosa	Iris-setosa
30	Iris-setosa	Iris-setosa
22	Iris-setosa	Iris-setosa
15	Iris-setosa	Iris-setosa
65	Iris-versicolor	Iris-versicolor
11	Iris-setosa	Iris-setosa
42	Iris-setosa	Iris-setosa
146	Iris-virginica	Iris-virginica
51	Iris-versicolor	Iris-versicolor
27	Iris-setosa	Iris-setosa