Machine Learning

By EvolkAI

Model Name: RidgeClassifier

The Ridge Classifier is a machine learning algorithm used for classification tasks. It is a variant of the linear classifier that incorporates a regularization term called ridge regression, which helps prevent overfitting.
In simple terms, the Ridge Classifier works by finding a linear decision boundary that separates different classes of data. It assigns weights to the input features and combines them to make predictions. However, unlike the standard linear classifier, the Ridge Classifier adds a penalty term to the loss function during training.

This penalty term, controlled by a regularization parameter, helps shrink the weights of less informative features, reducing their impact on the decision boundary. This regularization prevents the model from becoming overly complex and helps generalize better to unseen data.
The Ridge Classifier is known for its ability to handle high-dimensional data and mitigate the effects of multicollinearity (correlation between input features). It is widely used in applications such as text classification, sentiment analysis, and medical diagnostics.

```
import pandas as pd
data = pd.read_csv('iris_dataset.csv')
data.info()
```

In First line we Import pandas library as pd, then we read iris_dataset.csv file using the read_csv() function, and prints information about the data using the info() method.

```
feature = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
predection_class = ['species']
X = data[feature].values
y = data[predection_class].values
```

This defines the ‘feature’ and ‘predection_class’ variables, which specify the columns of the data to use as ‘features’ and the column to use as the prediction target.
The code then creates ‘X’ and ‘y’ arrays containing the values of these columns from the data DataFrame.

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size=0.30)
```

This imports the train_test_split function from the sklearn.model_selection module and uses it to split the data into training and testing sets. The test_size parameter specifies that 30% of the data should be used for testing.

```
print(f"Shape of X_test is {X_test.shape}")
print(f"Shape of X_train is {X_train.shape}")
print(f"Shape of Y_test is {Y_test.shape}")
print(f"Shape of Y_train is {Y_train.shape}")
```

These lines print the shapes of the training and testing data arrays. This output totally depends on the test size we took while train_test_split.

OUTPUT

```
Shape of X_test is (45, 4)
Shape of X_train is (105, 4)
Shape of Y_test is (45, 1)
Shape of Y_train is (105, 1)
```

```
from sklearn.linear_model import RidgeClassifier
clf = RidgeClassifier()
clf.fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
```

This imports the *RidgeClassifier* module from the sklearn linear_model library, creates a *RidgeClassifier* classifier object, fits the classifier with the training data using the *fit()* method, and use the *predict()* method to generate predictions for the testing data.

```
from sklearn import metrics
print("Accuracy", metrics.accuracy_score(Y_test, Y_pred)*100)
```

This imports the metrics module from sklearn and uses the accuracy_score() function to calculate the accuracy of the model on the testing data. The result is printed in the console.

OUTPUT

```
Accuracy 75.55555555555556
```

Conclusion

In conclusion, the Ridge Classifier is a valuable machine learning algorithm for classification tasks. By incorporating a regularization term, it strikes a balance between model complexity and generalization, preventing overfitting and improving performance on unseen data. The Ridge Classifier is particularly beneficial in scenarios with high-dimensional data and multicollinearity.

Its ability to handle these challenges makes it applicable in various domains, including text classification, sentiment analysis, and medical diagnostics. With its simplicity, efficiency, and robustness, the Ridge Classifier offers a reliable solution for classification tasks, providing accurate predictions and aiding decision-making processes. It is a useful tool for researchers and practitioners seeking to tackle classification problems with complex datasets.

Thank You