Telco Customer churn dataset

Code explanation:

This code is an implementation of a **Random Forest Classifier** model to predict customer churn in a Telco company. We use the **pandas** library to load the dataset from an CSV file and preprocess it. The **train\_test\_split** function from **sklearn.model\_selection** is used to split the dataset into training and testing sets. The **RandomForestClassifier** class from **sklearn.ensemble** is used to initialize the model with 100 estimators. The model is then trained on the training set using the **fit** method. The **predict** method is used to make predictions on the testing set. Finally, the **accuracy\_score**, **confusion\_matrix**, and **classification\_report** functions from **sklearn.metrics** are used to evaluate the model’s performance. The **confusion matrix** is plotted using the **heatmap** function from **seaborn** and **matplotlib** libraries.

The dataset used contains information about customers who left within the last month. The dataset has 21 columns and 7,043 rows. The target variable is the **Churn Label** column, which indicates whether a customer has left or not. The features used in the model are all columns except for **Churn Label**, **Churn Value**, **Churn Score**, **CLTV**, and **Churn Reason**. The categorical variables in the dataset are converted to dummy/indicator variables using the **get\_dummies** function from **pandas**.

The model’s accuracy is **0.7877, or 79%**. The **confusion matrix** shows the number of true positives, false positives, true negatives, and false negatives. The **classification report** shows the precision, recall, f1-score, and support for each class. The **heatmap** shows the confusion matrix in a graphical format.

The model’s performance was improved by adding two more components to the code: **feature importance plot** and **ROC curve**.

The **feature importance** plot shows the relative importance of the top 10 features in the model. The **ROC curve** shows the trade-off between the true positive rate and the false positive rate for different threshold values. The **AUC score** is a measure of the model’s performance, with a higher score indicating better performance. The AUC score for this model is **0.8536, or 85%**.

Une image contenant capture d’écran, texte, Rectangle, diagramme

Description générée automatiquement**Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquementUne image contenant texte, diagramme, ligne, Tracé

Description générée automatiquement**

Iris

Code explanation:

This code demonstrates a common approach to solve a multi-class classification problem using the Random Forest Classifier. It begins by loading the Iris dataset and preprocessing it.

Next, it splits the dataset into training and testing sets using the train\_test\_split function from sklearn.

After that, it creates a Random Forest Classifier with 100 trees and trains it using the training data.

Then, it makes predictions on the testing data and calculates the accuracy score by comparing the predictions with the actual values.

It also prints the confusion matrix, which is a table that describes the performance of the classifier by comparing predicted values with actual values.

The code further plots the confusion matrix and the feature importance, which represents the value of each feature in improving the classifier's performance.

Here is the breakdown of the code:

1.Load the Iris dataset using the pandas library.

2.Convert the 'species' column into numerical values using the LabelEncoder class from sklearn.preprocessing.

3.Split the dataset into training and testing sets using the train\_test\_split function from sklearn.model\_selection.

4.Create a Random Forest Classifier with 100 trees using the RandomForestClassifier class from sklearn.ensemble.

5.Train the classifier using the training data.

6.Make predictions on the testing data.

7.Calculate the accuracy score of the classifier using the accuracy\_score function from sklearn.metrics.

8.Print the confusion matrix and accuracy score.

9.Plot the confusion matrix using the seaborn library.

10.Plot the feature importance of the Random Forest Classifier.

Une image contenant capture d’écran, Rectangle, diagramme, carré

Description générée automatiquement

Une image contenant capture d’écran, texte, ligne, diagramme

Description générée automatiquement