Scikit Learn

Introducción al aprendizaje automático

CertiDevs

Índice de contenidos

1. Introducción

Scikit-learn es una biblioteca de **aprendizaje automático** de código abierto para Python.

Sitio web oficial

Proporciona herramientas simples y eficientes para el **análisis de datos** y la **minería de datos**.

Scikit-learn se basa en otras bibliotecas de Python como **NumPy**, **SciPy** y **Matplotlib**, lo que le permite integrarse fácilmente con otros proyectos científicos de Python.

2. Historia

Scikit-learn se inició en **2007** como parte del proyecto Google Summer of Code por David Cournapeau.

Posteriormente, en **2010**, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort y Vincent Michel, todos ellos investigadores del INRIA (Instituto Nacional de Investigación en Informática y Automática de Francia), tomaron la iniciativa y lanzaron la **primera versión estable** de Scikit-learn.

Desde entonces, ha ganado popularidad y se ha convertido en una de las bibliotecas de aprendizaje automático más utilizadas en la comunidad de Python.

3. Funcionalidades

Scikit-learn ofrece una amplia gama de **algoritmos** de **aprendizaje supervisado** y **no supervisado**, herramientas para la selección y evaluación de modelos, así como herramientas para la extracción de características y la reducción de la dimensionalidad. Algunas de las funcionalidades principales incluyen:

- **Clasificación**: algoritmos para clasificar objetos en distintas categorías, como SVM, árboles de decisión, k-vecinos más cercanos y regresión logística.
- **Regresión**: algoritmos para predecir variables continuas, como la regresión lineal y la regresión de soporte vectorial.
- **Clustering**: algoritmos para agrupar objetos similares, como k-means, clustering jerárquico y DBSCAN.
- **Reducción de dimensionalidad**: algoritmos para reducir la cantidad de características en los datos, como PCA y t-SNE.
- **Selección de modelos**: herramientas para evaluar y seleccionar modelos, como la validación cruzada y la búsqueda en cuadrícula.
- **Preprocesamiento de datos**: herramientas para limpiar y transformar datos, como la estandarización, la normalización y la codificación one-hot.

4. Instalación

Antes de instalar Scikit-learn, es necesario tener instaladas las bibliotecas **NumPy** y **SciPy**.

Para instalar Scikit-learn, puede utilizar `pip` o `conda`.

4.1. Instalación con pip

Para instalar Scikit-learn con pip, ejecute el siguiente comando en una terminal o ventana de comandos:

```
pip install scikit-learn
```

4.2. Instalación con conda

Si está utilizando la distribución Anaconda ejecuta siguiente comando en una terminal o ventana de comandos:

```
conda install scikit-learn
```

5. Comprobar la versión

Para verificar que Scikit-learn se ha instalado correctamente y conocer la versión instalada, ejecute el siguiente código en un intérprete de Python o en un archivo .py:

```
import sklearn
print("Versión de scikit-learn:", sklearn.__version__)
```

Al ejecutar este código, debería verse la versión de scikit-learn que ha instalado.

6. Carga de datos de Scikit-learn

Scikit-learn proporciona varios **conjuntos de datos** de demostración que se pueden utilizar para practicar y aprender.

Vamos a cargar el conjunto de datos de "Iris" como ejemplo.

```
from sklearn import datasets

# Cargar el conjunto de datos de Iris
iris = datasets.load_iris()

# Ver una descripción del conjunto de datos de Iris
```

```
print(iris.DESCR)
```

7. Conversión del conjunto de datos a un DataFrame de Pandas

Después de cargar el **conjunto de datos**, podemos convertirlo en un **DataFrame** de **Pandas** para facilitar la exploración y manipulación de los datos.

```
import pandas as pd

# Convertir el conjunto de datos Iris a un DataFrame de Pandas
iris_df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

# Agregar la columna objetivo (especies de Iris) al DataFrame
iris_df['species'] = iris.target

# Ver las primeras filas del DataFrame
print(iris_df.head())
```