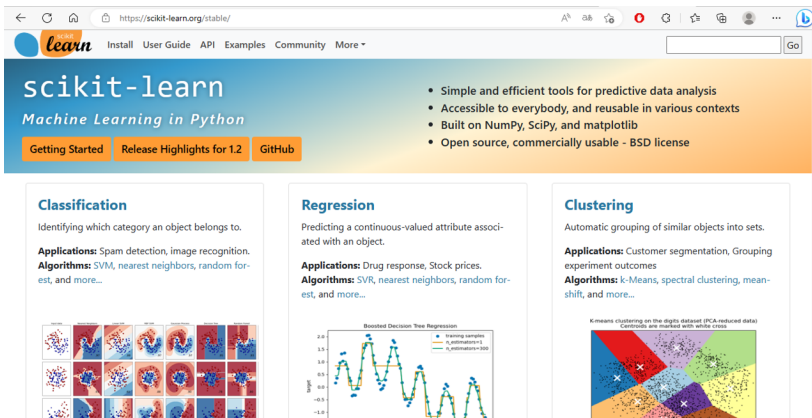


# Conhecendo a biblioteca Scikit-learn



## Conhecendo a biblioteca Scikit-learn



The screenshot shows the Scikit-learn website homepage. At the top, there's a navigation bar with links for 'Install', 'User Guide', 'API', 'Examples', 'Community', and 'More'. Below this, the 'scikit-learn' logo is prominently displayed, followed by the tagline 'Machine Learning in Python'. To the right of the logo, a list of features is provided: 'Simple and efficient tools for predictive data analysis', 'Accessible to everybody, and reusable in various contexts', 'Built on NumPy, SciPy, and matplotlib', and 'Open source, commercially usable - BSD license'. Below the main header, there are three buttons: 'Getting Started', 'Release Highlights for 1.2', and 'GitHub'. The page is divided into three main sections: 'Classification', 'Regression', and 'Clustering'. Each section includes a brief description, applications, and algorithms. The 'Classification' section features a grid of 18 small plots showing various data distributions and decision boundaries. The 'Regression' section includes a line plot titled 'Boosted Decision Tree Regression' showing training samples and two different estimators. The 'Clustering' section features a scatter plot titled 'K-means clustering on the digits dataset (PCA-reduced data)' showing data points grouped into clusters with centroids marked by white crosses.

**scikit-learn**  
*Machine Learning in Python*

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

[Getting Started](#) [Release Highlights for 1.2](#) [GitHub](#)

### Classification

Identifying which category an object belongs to.

**Applications:** Spam detection, image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, and more...

### Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, nearest neighbors, random forest, and more...

### Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, and more...

## Importante

**Scikit-learn** é hoje uma das ferramentas de Aprendizado de Máquina mais utilizadas.

- Mesmo em empresas, a biblioteca **Scikit-learn** é vastamente empregada
- É capaz de treinar desde modelos muito simples, até modelos muito complexos
- Geralmente, o modelo pode ser treinando usando apenas algumas poucas linhas de código (não é necessário implementar o Método do Gradiente na mão, por exemplo)
- Por outro lado, para que você tenha uma ideia sólida sobre todos esses algoritmos, é realmente importante que você seja capaz de implementar **Regressão Linear** na mão, sem fazer isso apenas chamando uma função do **Scikit-learn** como uma caixa-preta.
- Para conhecê-la melhor, acesse <https://scikit-learn.org/>

Vamos agora conhecer um pouco melhor a biblioteca **Scikit-learn** usando ela na prática no contexto de Regressão Linear.

Nome do arquivo que trabalharemos agora:

codigo - Regressão linear usando Scikit-Learn.ipynb

## Parte 1

Rode todo o “codigo - Regressão linear usando Scikit-Learn.ipynb” sem fazer qualquer tipo de alteração. Certifique-se de que você o compreendeu.

## Parte 2

- 1 Insira no código da Parte 1 o conjunto de medições que você já criou anteriormente para um resistor de 50  $\Omega$ , faça as adaptações necessárias e verifique os resultados.