Naive bayes classifier

# library link

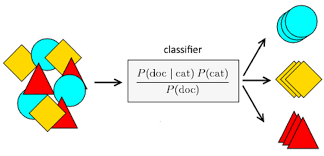
**install :**

<https://scikit-learn.org/stable/install.html>

**github :**

[https://github.com/scikit-learn/scikit-learn/blob/a9bf7f38d7d074962e8afe2d10c9e41bd8117f39/sklearn/naive\_bayes.py](https://github.com/scikit-learn/scikit-learn/blob/82df48934eba1df9a1ed3be98aaace8eada59e6e/sklearn/tree/_classes.py)

# basic description



naive Bayes classifiers are a family of simple “provavilistic classifiers” based on applying Bayes’ theorem with strong independence assumptions between the features.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables in a learning problem.

# version

* NumPy >= 1.14.6 (pip install numpy)
* Scipy >= 1.1.0 (pip install scipy)
* Joblib >= 0.11 (pip install joblib
* Threadpoolctl >= 2.0.0 (pip install threadpoolctl)
* threadpoolctl( >= 2.0.0 )

# dataset

* fetch\_20newsgroups: The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics split in two subsets: one for training (or development) and the other one for testing (or for performance evaluation). The split between the train and test set is based upon a messages posted before and after a specific date.
* Sources : <https://github.com/scikit-learn/scikit-learn/blob/534045dc582967261c0e67e2e63e28ecdac048cd/sklearn/datasets/descr/twenty_newsgroups.rst>

# code description

* Read fetch\_20newsgroups (option = train) dataset from sklearn and Translate news string to DTM and TF\_IDF and make naive bayes model (sklearn.naice\_bayes.MultinomialNB) for news string and news category. and train news TF\_IDF dataset to model.
* And test model with real\_world car news string, and fetch\_20newsgroups (option = test) dataset. And print the result of model prediction and accuracy\_score.

# validation

* Inside the code, the test dataset is loaded from sklearn sample dataset to verify this

naive bayes classifier model.

* Additionally, the fitness between the actual value and the predicted value is evaluated using Sklearn's accuracy\_score function.