Naive bayes classifier

# example code

|  |
| --- |
| # Import list  import numpy as np  from sklearn.datasets import fetch\_20newsgroups # Data set  from sklearn.feature\_extraction.text import CountVectorizer # for String data to bow (DTW)  from sklearn.feature\_extraction.text import TfidfTransformer# for bow to (TF-IDF)  from sklearn.naive\_bayes import MultinomialNB # navie\_bayes model  from sklearn.metrics import accuracy\_score # test accuracy\_score  # Load data from sklearn (Twenty Newsgroups)  news\_data = fetch\_20newsgroups(subset='train')  # The new's Data is String data so it need to translate to DTM or TF\_IDF  DTM\_translate = CountVectorizer()  TF\_IDF\_translate = TfidfTransformer()  # Translate String Data.  X\_train\_DTM = DTM\_translate.fit\_transform(news\_data.data)  X\_train\_TF\_IDF = TF\_IDF\_translate.fit\_transform(X\_train\_DTM)  # Make naive\_bayes model, and train X\_train\_TF\_IDF (Can also use X\_train\_DTM)  model = MultinomialNB(alpha=1.0,class\_prior=None,fit\_prior=True)  model.fit(X\_train\_TF\_IDF, news\_data.target)  # Load test\_dataset from sklearn  news\_data\_test = fetch\_20newsgroups(subset='test', shuffle=True)  # Translate test\_String Data.  X\_test\_DTM = DTM\_translate.transform(news\_data\_test.data)  X\_test\_TF\_IDF = TF\_IDF\_translate.transform(X\_test\_DTM)  # Make test String from real\_world car news. source : https://www.autocar.co.uk/car-review/ford/ranger-raptor/first-drives/ford-ranger-raptor-special-edition-2022-uk-review  test\_string = np.array(["Do you ever find yourself looking at the hulking, jacked-up Ford Ranger Raptor pick-up truck and thinking \“it’s nice, but it’s not quite lairy enough\”? Us neither: its outlandish off- road suspension-and-tyre package already give it more than enough presence on Britain’s cramped streets (not to mention agreeable levels of countryside competence). But this new Special Edition which arrives as the current-generation Ranger prepares to bow out-ups the ante with “extra badass as standard”.Roughly translated, that means it gains racing stripes, red accents inside and out and matt-black trim all round. If it didn’t stick out in the supermarket car park before, you can guarantee that it will now."])  # Translate test String Data.  test\_string\_DTM = DTM\_translate.transform(test\_string)  test\_string\_TF\_IDF = TF\_IDF\_translate.transform(test\_string\_DTM)  # Predicted test\_string  test\_predicted = model.predict(test\_string\_TF\_IDF)  # Print test\_string and predicted result.  print(test\_string)  print("\n{} : {}\n".format(test\_predicted[0],news\_data.target\_names[test\_predicted[0]]))  # Predicted test dataset  Predicted = model.predict(X\_test\_TF\_IDF)  # Test predicted dataset accuracy with orginal data  print("naive\_bayes model accuracy: ", accuracy\_score(news\_data\_test.target,Predicted)) |

# testing result

|  |
| --- |
| ['Do you ever find yourself looking at the hulking, jacked-up Ford Ranger Raptor pick-up truck and thinking \\“it’s nice, but it’s not quite lairy enough\\”?  Us neither: its outlandish off- road suspension-and-tyre package already give it more than enough presence on Britain’s cramped streets (not to mention agreeable levels of countryside competence).  But this new Special Edition which arrives as the current-generation Ranger prepares to bow out-ups the ante with “extra badass as standard”.  Roughly translated, that means it gains racing stripes, red accents inside and out and matt-black trim all round. If it didn’t stick out in the supermarket car park before, you can guarantee that it will now.']  7 : rec.autos  naive\_bayes model accuracy: 0.7738980350504514 |