```
import numpy as np
```

text classification

['alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'comp.windows.x', 'misc.forsale', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.electronics', 'sci.med', 'sci.space', 'soc.religion.christian', 'talk.politics.guns', 'talk.politics.mideast', 'talk.politics.misc', 'talk.religion.misc']

```
import numpy as np
```

```
from sklearn.datasets import fetch_20newsgroups
```

```
database = fetch_20newsgroups(data_home='/content/',
                                                         subset='all',
                                                         categories=['talk.politics.misc', 'talk.religion.misc'],
                                                         shuffle=True,
                                                         random_state=42,
                                                         remove={'headers', 'footers', 'quotes'},
                                                         download_if_missing=True,
                                                         return_X_y=False)
```

```
X = database.data
```

```
len(X)

     1403
```

Saved successfully!

```
len(y)

     1403
```

```
y

     array([0, 0, 1, ..., 0, 0, 0])
```

```
X[0]

     '\n\n      Perhaps I failed to make myself clear:  Minorities in the U.S.\n*correlate* with poverty.
     This isn\'t good and we should address it,\nbut we shouldnt\' ignore that minorities and poverty *do*
     tend to go\ntogether.\n\n      *Does* Vancouver have a consistantly poor population drawn along\nracia
     l lines?  If it doesn\'t, then assumptions of being able to compare\nminority vs. majority in both cit
     ies is questionable at best.\n\n\n       If the *rate* of increase over a period of several years rema
     ins\nunchanged, or increases, I think it\'s not a far jump to say that the laws\nare not effective.  N
     o, you can\'t sit down and say that things wouldn\'t\nhave been worse.  I don\'t have a crystal ball a
     nd neither do you.  However,\nthat road leads us to a place where it is impossible to critique *any*\n
```

```
X[2]

     'So we have this highly Christian religious order that put fire\non their house, killing most of the p
     eople inside.\n\nI'm not that annoyed about the adults, they knew supposedly what\nthey were doing, an
     d it's their own actions.\n\nWhat I mostly are angry about is the fact that the people inside,\ninclud
     ing mothers, let the children suffer and die during awful\nconditions.\n\nIf this is considered religi
     ous following to the end, I'm proud\nthat I don't follow such fanatical and non-compassionate religion
     s.\n\nYou might want to die for whatever purpose, but please spare\nthe innocent young ones that has n
     othing to do with this all.\n\nI have a hard time just now understanding that Christianity\nknows abou
```

```
database.target_names

     ['talk.politics.misc', 'talk.religion.misc']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,stratify=y)
```

```
X_train, X_val, y_train, y_val = train_test_split(X_train,y_train,test_size=0.2,stratify=y_train)
```

Data preprocessing

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_selection import SelectKBest, f_classif
```

```
vectorizer = TfidfVectorizer(
                             strip_accents='unicode',
                             decode_error='replace',
                             dtype='int32',
                             analyzer="word",
                             ngram_range=(1, 2),
                             min_df=2)
```

```
X_train = vectorizer.fit_transform(X_train)
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/feature_extraction/text.py:2029: UserWarning: Only (<class 'numpy.float64'>, <class
  warnings.warn(
```

```
X_test = vectorizer.transform(X_test)
```

```
X_val = vectorizer.transform(X_val)
```

```
selector = SelectKBest(f_classif, k = min(20000, X_train.shape[1]))
selector.fit(X_train, y_train)
```

```
SelectKBest(k=20000)
```

```
X_train = selector.transform(X_train).astype('float32')
[illegible].astype('float32')
[illegible]stype('float32')
```

Saved successfully!

```
X_train.shape
```

```
(897, 20000)
```

```
X_test.shape
```

```
(281, 20000)
```

```
#X_trainoriginal = selector.inverse_transform(X_train)
```

```
#X_trainoriginal.shape
```

```
y_train = np.array(y_train)
y_test = np.array(y_test)
y_val = np.array(y_val)
```

```
X_train = X_train.toarray()
X_test = X_test.toarray()
X_val = X_val.toarray()
```

```
X_train.shape
```

```
(897, 20000)
```

ANN

```
from keras.models import Sequential, load_model
from keras.layers import Dense, Dropout
```

```
newsANN = Sequential()
```

```
newsANN.add(Dense(units=512, activation='relu', input_dim=20000))
newsANN.add(Dense(units=1, activation='sigmoid'))
newsANN.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
from keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
es = EarlyStopping(monitor='val_accuracy', min_delta=0, patience=20, verbose=1, mode='auto', baseline=None, restore_best_weights=False)
mc = ModelCheckpoint(filepath='bestweights.h5', monitor='val_accuracy', verbose=1, save_best_only=True)
rd = ReduceLROnPlateau(monitor='val_accuracy', factor=0.1, patience=10, verbose=1, mode='auto')
```

```
history = newsANN.fit(X_train, y_train, epochs=20, callbacks=[es,rd,mc], validation_split=0.25)
```

```
    21/21 [==============================] - ETA: 0s - loss: 0.0372 - accuracy: 0.9851
    Epoch 7: val_accuracy did not improve from 0.97333
    21/21 [==============================] - 4s 178ms/step - loss: 0.0372 - accuracy: 0.9851 - val_loss: 0.1147 - val_accuracy: 0.97
    Epoch 8/20
    21/21 [==============================] - ETA: 0s - loss: 0.0342 - accuracy: 0.9851
    Epoch 8: val_accuracy did not improve from 0.97333
    21/21 [==============================] - 5s 237ms/step - loss: 0.0342 - accuracy: 0.9851 - val_loss: 0.1091 - val_accuracy: 0.97
    Epoch 9/20
    21/21 [==============================] - ETA: 0s - loss: 0.0314 - accuracy: 0.9851
    Epoch 9: val_accuracy improved from 0.97333 to 0.97778, saving model to bestweights.h5
    21/21 [==============================] - 4s 191ms/step - loss: 0.0314 - accuracy: 0.9851 - val_loss: 0.1050 - val_accuracy: 0.97
    Epoch 10/20
    21/21 [==============================] - ETA: 0s - loss: 0.0296 - accuracy: 0.9866
    Epoch 10: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 4s 178ms/step - loss: 0.0296 - accuracy: 0.9866 - val_loss: 0.1007 - val_accuracy: 0.97
    Epoch 11/20
    21/21 [==============================] - ETA: 0s - loss: 0.0284 - accuracy: 0.9866
    Epoch 11: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 5s 238ms/step - loss: 0.0284 - accuracy: 0.9866 - val_loss: 0.0979 - val_accuracy: 0.97
    Epoch 12/20
    21/21 [==============================] - ETA: 0s - loss: 0.0276 - accuracy: 0.9866
    Epoch 12: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 4s 177ms/step - loss: 0.0276 - accuracy: 0.9866 - val_loss: 0.0957 - val_accuracy: 0.97
    Epoch 13/20
    21/21 [==============================] - ETA: 0s - loss: 0.0267 - accuracy: 0.9866
    Epoch 13: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 4s 185ms/step - loss: 0.0267 - accuracy: 0.9866 - val_loss: 0.0935 - val_accuracy: 0.97
    Epoch 14/20
    21/21 [==============================] - ETA: 0s - loss: 0.0271 - accuracy: 0.9866
Saved successfully!           improve from 0.97778
                              =======] - 5s 228ms/step - loss: 0.0271 - accuracy: 0.9866 - val_loss: 0.0920 - val_accuracy: 0.97
    21/21 [==============================] - ETA: 0s - loss: 0.0257 - accuracy: 0.9866
    Epoch 15: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 4s 177ms/step - loss: 0.0257 - accuracy: 0.9866 - val_loss: 0.0899 - val_accuracy: 0.97
    Epoch 16/20
    21/21 [==============================] - ETA: 0s - loss: 0.0254 - accuracy: 0.9866
    Epoch 16: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 4s 178ms/step - loss: 0.0254 - accuracy: 0.9866 - val_loss: 0.0888 - val_accuracy: 0.97
    Epoch 17/20
    21/21 [==============================] - ETA: 0s - loss: 0.0253 - accuracy: 0.9866
    Epoch 17: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 11s 551ms/step - loss: 0.0253 - accuracy: 0.9866 - val_loss: 0.0878 - val_accuracy: 0.9
    Epoch 18/20
    21/21 [==============================] - ETA: 0s - loss: 0.0249 - accuracy: 0.9866
    Epoch 18: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 9s 421ms/step - loss: 0.0249 - accuracy: 0.9866 - val_loss: 0.0864 - val_accuracy: 0.97
    Epoch 19/20
    21/21 [==============================] - ETA: 0s - loss: 0.0244 - accuracy: 0.9866
    Epoch 19: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.

    Epoch 19: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 4s 190ms/step - loss: 0.0244 - accuracy: 0.9866 - val_loss: 0.0855 - val_accuracy: 0.97
    Epoch 20/20
    21/21 [==============================] - ETA: 0s - loss: 0.0242 - accuracy: 0.9866
    Epoch 20: val_accuracy did not improve from 0.97778
    21/21 [==============================] - 4s 196ms/step - loss: 0.0242 - accuracy: 0.9866 - val_loss: 0.0854 - val_accuracy: 0.97
```

```
newmodel = load_model('bestweights.h5')
```

```
newmodel.evaluate(X_test, y_test)
```

```
    9/9 [==============================] - 0s 21ms/step - loss: 0.3478 - accuracy: 0.8149
    [0.34780314564704895, 0.8149465918540955]
```

Deployment

```
comment = ['we are celebrating all festivals']
```

```
X_deployment = vectorizer.transform(comment)
```

```
X_deployment_best = selector.transform(X_deployment)
```

```
X_deployment_best_array = X_deployment_best.toarray()
```

```
newmodel.predict(X_deployment_best_array)
```

```
    1/1 [==============================] - 0s 115ms/step
    array([[0.46869773]], dtype=float32)
```

Deployment in web server

```
!pip install flask gevent requests pillow flask-ngrok pyngrok
```

```
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Requirement already satisfied: flask in /usr/local/lib/python3.8/dist-packages (1.1.4)
    Collecting gevent
      Downloading gevent-22.10.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.5 MB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.5/6.5 MB 39.8 MB/s eta 0:00:00
    Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-packages (2.25.1)
    Requirement already satisfied: pillow in /usr/local/lib/python3.8/dist-packages (7.1.2)
    Collecting flask-ngrok
      Downloading flask_ngrok-0.0.25-py3-none-any.whl (3.1 kB)
    Collecting pyngrok
      Downloading pyngrok-5.2.1.tar.gz (761 kB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 761.3/761.3 KB 48.3 MB/s eta 0:00:00
      Preparing metadata (setup.py) ... done
    Requirement already satisfied: click<8.0,>=5.1 in /usr/local/lib/python3.8/dist-packages (from flask) (7.1.2)
    Requirement already satisfied: itsdangerous<2.0,>=0.24 in /usr/local/lib/python3.8/dist-packages (from flask) (1.1.0)
    Requirement already satisfied: Jinja2<3.0,>=2.10.1 in /usr/local/lib/python3.8/dist-packages (from flask) (2.11.3)
    Requirement already satisfied: Werkzeug<2.0,>=0.15 in /usr/local/lib/python3.8/dist-packages (from flask) (1.0.1)
    Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-packages (from gevent) (57.4.0)
Saved successfully!                           y2.py3-none-any.whl (6.8 kB)
    Collecting zope.interface
      Downloading zope.interface-5.5.2-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 261.4/261.4 KB 24.3 MB/s eta 0:00:00
    Requirement already satisfied: greenlet>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from gevent) (2.0.2)
    Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests) (2.10)
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests) (2022.12.7)
    Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests) (4.0.0)
    Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests) (1.24.3)
    Requirement already satisfied: PyYAML in /usr/local/lib/python3.8/dist-packages (from pyngrok) (6.0)
    Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.8/dist-packages (from Jinja2<3.0,>=2.10.1->flask) (2.0.1)
    Building wheels for collected packages: pyngrok
      Building wheel for pyngrok (setup.py) ... done
      Created wheel for pyngrok: filename=pyngrok-5.2.1-py3-none-any.whl size=19792 sha256=40ab31804ebafd8bb7b99a50c41aae8ffb973e9407d
      Stored in directory: /root/.cache/pip/wheels/5d/f2/70/526da675d32f17577ec47ac4c663084efe39d47c826b6c3bb1
    Successfully built pyngrok
    Installing collected packages: zope.interface, zope.event, pyngrok, gevent, flask-ngrok
    Successfully installed flask-ngrok-0.0.25 gevent-22.10.2 pyngrok-5.2.1 zope.event-4.6 zope.interface-5.5.2
```

```
from flask_ngrok import run_with_ngrok
from flask import Flask, render_template, request
```

```
procfile = 'web: gunicorn app:app'
procfiles = open('/content/Procfile', 'w')
```

```
procfiles.write(procfile)
```

```
    21
```

```
procfiles.close()
```

```
!mkdir '/content/templates'
```

```
'''
index.html

<!doctype html>
<html lang="en">
<head>
 <title>Image Recognition Server</title>
```

```
  <body>

      <form  action="" method="post" enctype=multipart/form-data>
        <input type="text" name="comment" placeholder="Movie review" required="required" />
        <input type=submit value=Upload>
    </form>

    <h3>Prediction is</h3>
    {{label}}


  </body>
</html>
'''
```

Connecting webpage with ANN

```
import pyngrok
```

```
!ngrok authtoken
```

```
     NAME:
        authtoken - save authtoken to configuration file

     USAGE:
        ngrok authtoken [command options] [arguments...]

     DESCRIPTION:
        The authtoken command modifies your configuration file to include
        the specified authtoken. By default, this configuration file is located
        at $HOME/.ngrok2/ngrok.yml

        The ngrok.com service requires that you sign up for an account to use
        many advanced service features. In order to associate your client with
                                    secret token to the ngrok.com service when it
                                    ng this authtoken on every invocation, you may
                                     into your configuration file so that your
        client always authenticates you properly.

     EXAMPLE:
         ngrok authtoken BDZIXnhJt2HNWLXyQ5PM_qCaBq0W2sNFcCa0rfTZd

     OPTIONS:
        --config              save in this config file, default: ~/.ngrok2/ngrok.yml
        --log "false"         path to log file, 'stdout', 'stderr' or 'false'
        --log-format "term"   log record format: 'term', 'logfmt', 'json'
        --log-level "info"    logging level

     ERROR:  You must pass a single argument, the authtoken to save to configuration file.
```

Saved successfully!

```
app = Flask(__name__)
run_with_ngrok(app)

@app.route('/')
def home():
  return render_template('index.html')

@app.route('/', methods=['POST'])
def prediction():
  data = request.form['textbox']
  features = [data]
  X_deployment = vectorizer.transform(features)
  X_deployment_best = selector.transform(X_deployment)
  X_deployment_best_array = X_deployment_best.toarray()
  preds = newmodel.predict(X_deployment_best_array)

  if preds >=0.5:
    label = 'Religious news'
  else:
    label = 'Political news'

  return render_template('index.html', prediction=label)

if __name__=='__main__':
  app.run()
```

```
      * Serving Flask app "__main__" (lazy loading)
      * Environment: production
        WARNING: This is a development server. Do not use it in a production deployment.
        Use a production WSGI server instead.
      * Debug mode: off
```

```
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Running on http://540b-35-221-167-105.ngrok.io
 * Traffic stats available on http://127.0.0.1:4040
```

✓ 16m 44s completed at 3:23 PM

Saved successfully!