classification of newswires a multiclass classification reuteras dataset

```
import numpy as np
import pandas as pd
import warnings
import tensorflow as tf # import tensor flow
import numpy as np
import keras
from tensorflow.keras import layers
import matplotlib.pyplot as plt
from tensorflow.keras.datasets import reuters
(train_data, train_labels), (test_data, test_labels) = reuters.load_data(num_words=10000)

word_index = reuters.get_word_index()
reverse_word_index = dict([(value, key) for (key, value) in word_index.items()])
decoded_newswire = ' '.join([reverse_word_index.get(i - 3, '?') for i in
   train_data[0]])
def vectorize_sequences(sequences, dimension=10000):
   results = np.zeros((len(sequences), dimension))
   for i, sequence in enumerate(sequences):
      results[i, sequence] = 1.
   return results
x_train = vectorize_sequences(train_data)#1
x_test = vectorize_sequences(test_data)#2
def to_one_hot(labels, dimension=46):
   results = np.zeros((len(labels), dimension))
   for i, label in enumerate(labels):
      results[i, label] = 1.
   return results
one_hot_train_labels = to_one_hot(train_labels)#1
one_hot_test_labels = to_one_hot(test_labels)#2
from tensorflow.keras.utils import to_categorical
```

```
one_hot_train_labels = to_categorical(train_labels)
one_hot_test_labels = to_categorical(test_labels)
model = keras.Sequential([
  layers.Dense(64, activation='relu'),
  layers.Dense(64, activation='relu'),
  layers.Dense(46, activation='softmax')
])
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
x_val = x_train[:1000]
partial_x_train = x_train[1000:]
y_val = one_hot_train_labels[:1000]
partial_y_train = one_hot_train_labels[1000:]
history = model.fit(partial_x_train,
                    partial_y_train,
                    epochs=25,
                    batch_size=512,
                    validation_data=(x_val, y_val))
results = model.evaluate(x_test, one_hot_test_labels)
```