[COM4513-6513] Assignment 1: Text Classification with Logistic Regression

Instructor: Nikos Aletras

The goal of this assignment is to develop and test two text classification systems:

- **Task 1:** sentiment analysis, in particular to predict the sentiment of movie review, i.e. positive or negative (binary classification).
- **Task 2:** topic classification, to predict whether a news article is about International issues, Sports or Business (multiclass classification).

For that purpose, you will implement:

- Text processing methods for extracting Bag-Of-Word features, using (1) unigrams, bigrams and trigrams to obtain vector representations of documents. Two vector weighting schemes should be tested: (1) raw frequencies (**3 marks; 1 for each ngram type**); (2) tf.idf (**1 marks**).
- Binary Logistic Regression classifiers that will be able to accurately classify movie reviews trained with (1) BOW-count (raw frequencies); and (2) BOW-tfidf (tf.idf weighted) for Task 1.
- Multiclass Logistic Regression classifiers that will be able to accurately classify news articles trained with (1) BOW-count (raw frequencies); and (2) BOW-tfidf (tf.idf weighted) for Task 2.
- The Stochastic Gradient Descent (SGD) algorithm to estimate the parameters of your Logistic Regression models. Your SGD algorithm should:
  - Minimise the Binary Cross-entropy loss function for Task 1 (**3 marks**)
  - Minimise the Categorical Cross-entropy loss function for Task 2 (**3 marks**)
  - Use L2 regularisation (both tasks) (**1 mark**)
  - Perform multiple passes (epochs) over the training data (**1 mark**)
  - Randomise the order of training data after each pass (**1 mark**)
  - Stop training if the difference between the current and previous validation loss is smaller than a threshold (**1 mark**)
  - After each epoch print the training and development loss (**1 mark**)
- Discuss how did you choose hyperparameters (e.g. learning rate and regularisation strength)? (**2 marks; 0.5 for each model in each task**).
- After training the LR models, plot the learning process (i.e. training and validation loss in each epoch) using a line plot (**1 mark; 0.5 for both BOW-count and BOW-tfidf LR models in each task**) and discuss if your model overfits/underfits/is about right.
- Model interpretability by showing the most important features for each class (i.e. most positive/negative weights). Give the top 10 for each class and comment on whether they make sense (if they don't you might have a bug!). If we were to apply the classifier we've learned into a different domain such laptop reviews or restaurant reviews, do you think these features would generalise well? Can you propose what features the classifier could pick up as important in the new domain? (**2 marks; 0.5 for BOW-count and BOW-tfidf LR models respectively in each task**)

Data - Task 1

The data you will use for Task 1 are taken from here: http://www.cs.cornell.edu/people/pabo/movie-review-data/ (http://www.cs.cornell.edu/people/pabo/movie-review-data/) and you can find it in the `./data_sentiment` folder in CSV format:

- `data_sentiment/train.csv` : contains 1,400 reviews, 700 positive (label: 1) and 700 negative (label: 0) to be used for training.
- `data_sentiment/dev.csv` : contains 200 reviews, 100 positive and 100 negative to be used for hyperparameter selection and monitoring the training process.
- `data_sentiment/test.csv` : contains 400 reviews, 200 positive and 200 negative to be used for testing.

Data - Task 2

The data you will use for Task 2 is a subset of the AG News Corpus (http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html) and you can find it in the `./data_topic` folder in CSV format:

- `data_topic/train.csv` : contains 2,400 news articles, 800 for each class to be used for training.
- `data_topic/dev.csv` : contains 150 news articles, 50 for each class to be used for hyperparameter selection and monitoring the training process.
- `data_topic/test.csv` : contains 900 news articles, 300 for each class to be used for testing.

Submission Instructions

You should submit a Jupyter Notebook file (assignment1.ipynb) and an exported PDF version (you can do it from Jupyter: `File->Download as->PDF via Latex`).

You are advised to follow the code structure given in this notebook by completing all given funtions. You can also write any auxilliary/helper functions (and arguments for the functions) that you might need but note that you can provide a full solution without any such functions. Similarly, you can just use only the packages imported below but you are free to use any functionality from the Python Standard Library (https://docs.python.org/2/library/index.html), NumPy, SciPy and Pandas. You are not allowed to use any third-party library such as Scikit-learn (apart from metric functions already provided), NLTK, Spacy, Keras etc..

Please make sure to comment your code. You should also mention if you've used Windows (not recommended) to write and test your code. There is no single correct answer on what your accuracy should be, but correct implementations usually achieve F1-scores around 80% or higher. The quality of the analysis of the results is as important as the accuracy itself.

This assignment will be marked out of 20. It is worth 20% of your final grade in the module.

The deadline for this assignment is **23:59 on Fri, 20 Mar 2020** and it needs to be submitted via MOLE. Standard departmental penalties for lateness will be applied. We use a range of strategies to detect unfair means (https://www.sheffield.ac.uk/ssid/unfair-means/index), including Turnitin which helps detect plagiarism, so make sure you do not plagiarise.

In [1]:

```
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import random

# fixing random seed for reproducibility
random.seed(123)
np.random.seed(123)
```

Load Raw texts and labels into arrays

First, you need to load the training, development and test sets from their corresponding CSV files (tip: you can use Pandas dataframes).

In [2]:

```
data_tr=pd.read_csv('./data_sentiment/train.csv',names=["text","label"])
data_dev = pd.read_csv('./data_sentiment/dev.csv', names=["text","label"])
data_te = pd.read_csv('./data_sentiment/test.csv', names=["text","label"])
```

If you use Pandas you can see a sample of the data.

In [3]:

```
data_tr.head()
```

Out[3]:

	text	label
0	note : some may consider portions of the follo...	1
1	note : some may consider portions of the follo...	1
2	every once in a while you see a film that is s...	1
3	when i was growing up in 1970s , boys in my sc...	1
4	the muppet movie is the first , and the best m...	1

The next step is to put the raw texts into Python lists and their corresponding labels into NumPy arrays:

In [4]:

```
# fill in your code...
# put all 3 file into lists and array
X_tr_raw = data_tr['text'].tolist() #DataFrame to python list
Y_tr = data_tr['label'].values # DataFrame to numpy arrays

X_dev_raw = data_dev['text'].tolist()
Y_dev = data_dev['label'].values

X_test_raw = data_te['text'].tolist()
Y_te = data_te['label'].values
```

Bag-of-Words Representation

To train and test Logisitc Regression models, you first need to obtain vector representations for all documents given a vocabulary of features (unigrams, bigrams, trigrams).

Text Pre-Processing Pipeline

To obtain a vocabulary of features, you should:

- tokenise all texts into a list of unigrams (tip: using a regular expression)
- remove stop words (using the one provided or one of your preference)
- compute bigrams, trigrams given the remaining unigrams
- remove ngrams appearing in less than K documents
- use the remaining to create a vocabulary of unigrams, bigrams and trigrams (you can keep top N if you encounter memory issues).

In [5]:

```
stop_words = ['a','in','on','at','and','or',
              'to', 'the', 'of', 'an', 'by',
              'as', 'is', 'was', 'were', 'been', 'be',
              'are','for', 'this', 'that', 'these', 'those', 'you', 'i',
             'it', 'he', 'she', 'we', 'they' 'will', 'have', 'has',
              'do', 'did', 'can', 'could', 'who', 'which', 'what',
             'his', 'her', 'they', 'them', 'from', 'with', 'its']
```

N-gram extraction from a document

You first need to implement the `extract_ngrams` function. It takes as input:

- `x_raw` : a string corresponding to the raw text of a document
- `ngram_range` : a tuple of two integers denoting the type of ngrams you want to extract, e.g. (1,2) denotes extracting unigrams and bigrams.
- `token_pattern` : a string to be used within a regular expression to extract all tokens. Note that data is already tokenised so you could opt for a simple white space tokenisation.
- `stop_words` : a list of stop words
- `vocab` : a given vocabulary. It should be used to extract specific features.

and returns:

- a list of all extracted features.

See the examples below to see how this function should work.

In [6]:

```
def extract_ngrams(x_raw, ngram_range=(1,3), token_pattern=r'\b[A-Za-z][A-Za-z]+\b', stop_words=[],
    x = []
    # Regular expression
    pattern = re.compile(token_pattern).findall(x_raw.lower())
    # Get words without stop word
    unigram = [word for word in pattern if word not in stop_words and re.match(token_pattern, wor
    # generate 3 types ngrams
    for n in range(ngram_range[0], ngram_range[1]+1):
        if n == 1:
            x.extend(unigram)
        elif n == 2:
            x.extend([(word1, word2) for word1, word2 in zip(unigram[:-1], unigram[1:])])
        elif n == 3:
            x.extend([(word1, word2, word3) for word1, word2, word3 in zip(unigram[:-2], unigram[1
        if len(vocab) != 0:
            x = [w for w in x if w in vocab]
    return x

    # fill in your code...
```

In [7]:

```
extract_ngrams("this is a great movie to watch",
               ngram_range=(1,3),
               stop_words=stop_words)
```

Out[7]:

```
['great',
 'movie',
 'watch',
 ('great', 'movie'),
 ('movie', 'watch'),
 ('great', 'movie', 'watch')]
```

In [8]:

```
extract_ngrams("this is a great movie to watch",
               ngram_range=(1,2),
               stop_words=stop_words,
               vocab=set(['great',  ('great','movie')]))
```

Out[8]:

```
['great', ('great', 'movie')]
```

Note that it is OK to represent n-grams using lists instead of tuples: e.g. `['great', ['great', 'movie']]`

Create a vocabulary of n-grams

Then the `get_vocab` function will be used to (1) create a vocabulary of ngrams; (2) count the document frequencies of ngrams; (3) their raw frequency. It takes as input:

- `X_raw` : a list of strings each corresponding to the raw text of a document
- `ngram_range` : a tuple of two integers denoting the type of ngrams you want to extract, e.g. (1,2) denotes extracting unigrams and bigrams.
- `token_pattern` : a string to be used within a regular expression to extract all tokens. Note that data is already tokenised so you could opt for a simple white space tokenisation.
- `stop_words` : a list of stop words
- `vocab` : a given vocabulary. It should be used to extract specific features.
- `min_df` : keep ngrams with a minimum document frequency.
- `keep_topN` : keep top-N more frequent ngrams.

and returns:

- `vocab` : a set of the n-grams that will be used as features.
- `df` : a Counter (or dict) that contains ngrams as keys and their corresponding document frequency as values.
- `ngram_counts` : counts of each ngram in vocab

Hint: it should make use of the `extract_ngrams` function.

In [9]:

```
from collections import Counter

def get_vocab(X_raw, ngram_range=(1,3), token_pattern=r'\b[A-Za-z][A-Za-z]+\b', min_df=0, keep_topl
    grams = []
    for x in X_raw:
          grams.extend(set(extract_ngrams(x, ngram_range, token_pattern, stop_words))) # get n-gram
    df_grams = Counter(grams)
    for key in df_grams.keys(): # if df value is smaller than required value, delete the element
        if df_grams[key] < min_df:
            del df_grams[key]
    # get top N occur times in df
    ngram_counts = df_grams.most_common(keep_topN)
    # get word in ngram counts
    vocab = set([word[0] for word in ngram_counts])

    # fill in your code...


    return vocab, df_grams, ngram_counts
```

Now you should use `get_vocab` to create your vocabulary and get document and raw frequencies of n-grams:

In [10]:

```
vocab, df, ngram_counts = get_vocab(X_tr_raw, ngram_range=(1,3), keep_topN=5000, stop_words=stop_wo
print(len(vocab))
print()
print(list(vocab)[:100])
print()
print(df.most_common()[:10])
```

```
5000

[('might', 'well'), ('not', 'so'), ('few', 'years'), 'cheating', 'ups', 'getting',
'after', 'join', 'ask', 'gross', 'surrounding', ('comic', 'relief'), 'glenn', ('litt
le', 'too'), ('beginning', 'end'), ('after', 'being'), ('know', 'but'), 'drew', 'iss
ues', 'travel', 'path', 'awake', 'president', ('their', 'characters'), 'notice', 'co
mplex', 'built', 'groups', ('both', 'films'), 'plausible', 'entertaining', 'elaborat
e', 'reputation', 'efforts', ('more', 'more'), ('most', 'notably'), 'really', 'point
less', 'destroy', 'alike', 'hanks', 'remake', 'connected', 'walks', 'fell', 'done',
('one', 'their'), 'room', 'list', ('however', 'when'), 'dedicated', 'progress', 'car
ds', 'martin', 'special', ('so', 'there'), 'sky', 'ridiculous', 'object', 'section',
'boredom', 'dreams', 'blockbuster', 'miscast', 've', 'appreciated', ('small', 'tow
n'), 'average', 'shannon', 'terrifying', 'contrived', 'beautiful', 'twice', 'hostag
e', 'allen', 'bringing', ('all', 'too'), 'develops', 'devil', 'slightly', 'kong', 'l
anguage', 'security', 'promised', ('good', 'time'), 'help', 'comedy', 'uplifting',
'subplots', 'viewers', 'laughable', ('film', 'had'), 'investigation', 'written', ('o
ut', 'film'), ('martial', 'arts'), 'moments', ('there', 'but'), 'weapons', 'disappoi
ntment']

[('but', 1334), ('one', 1247), ('film', 1231), ('not', 1170), ('all', 1117), ('movi
e', 1095), ('out', 1080), ('so', 1047), ('there', 1046), ('like', 1043)]
```

Then, you need to create vocabulary id -> word and id -> word dictionaries for reference:

In [11]:

```
# fill in your code...
# generate word and corresponding id
word_id = {}
i = 0
for i, w in enumerate(list(vocab)):
    word_id[w] = i
```

In [12]:

```
word_id
```

Out[12]:

```
{('might', 'well'): 0,
 ('not', 'so'): 1,
 ('few', 'years'): 2,
 'cheating': 3,
 'ups': 4,
 'getting': 5,
 'after': 6,
 'join': 7,
 'ask': 8,
 'gross': 9,
 'surrounding': 10,
 ('comic', 'relief'): 11,
 'glenn': 12,
 ('little', 'too'): 13,
 ('beginning', 'end'): 14,
 ('after', 'being'): 15,
 ('know', 'but'): 16,
```

Now you should be able to extract n-grams for each text in the training, development and test sets:

In [13]:

```
# fill in your code...
X_train_ngram = []
X_dev_ngram = []
X_test_ngram = []

# generate 3 file's ngrams list
for i in X_tr_raw:
    X_train_ngram.append(extract_ngrams(i, ngram_range=(1,3), token_pattern=r'\b[A-Za-z][A-Za-z]+\b
for i in X_dev_raw:
    X_dev_ngram.append(extract_ngrams(i, ngram_range=(1,3), token_pattern=r'\b[A-Za-z][A-Za-z]+\b',
for i in X_test_raw:
    X_test_ngram.append(extract_ngrams(i, ngram_range=(1,3), token_pattern=r'\b[A-Za-z][A-Za-z]+\b'
```

In [14]:

```
#X_train_ngram
#X_dev_ngram
#X_test_ngram
```

Vectorise documents

Next, write a function `vectoriser` to obtain Bag-of-ngram representations for a list of documents. The function should take as input:

- `X_ngram` : a list of texts (documents), where each text is represented as list of n-grams in the `vocab`
- `vocab` : a set of n-grams to be used for representing the documents

and return:

- `X_vec` : an array with dimensionality Nx|vocab| where N is the number of documents and |vocab| is the size of the vocabulary. Each element of the array should represent the frequency of a given n-gram in a document.

In [15]:

```
def vectorise(X_ngram, vocab):

    # fill in your code...
    X_vec = []
    for i in X_ngram:
        # creater counter object
        counter = Counter(i)
        # create zero lists
        vector = list(0 for j in range(len(vocab)))
        for j in counter.keys():
            # add number into list correspond to the position
            vector[word_id[j]] = vector[word_id[j]]+ counter[j]
        X_vec.append(vector)
    # change to np array format
    X_vec = np.array(X_vec)


    return X_vec
```

Finally, use `vectorise` to obtain document vectors for each document in the train, development and test set. You should extract both count and tf.idf vectors respectively:

Count vectors

In [16]:

```
# fill in your code...
X_tr_count = vectorise(X_train_ngram, vocab)
X_dev_count = vectorise(X_dev_ngram, vocab)
X_test_count = vectorise(X_test_ngram, vocab)
```

In [17]:

```
X_tr_count.shape
```

Out[17]:

```
(1400, 5000)
```

In [18]:

```
X_tr_count[:2,:50]
```

Out[18]:

```
array([[0, 0, 1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0],
       [0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0]])
```

TF.IDF vectors

First compute `idfs` an array containing inverted document frequencies (Note: its elements should correspond to your `vocab`)

In [19]:

```
idfs = []
for i in list(vocab):
    # set a variable to record occurance times
    df_count = 0
    # if it occur, variable + 1
    for j in X_train_ngram:
        if i in j:
            df_count = df_count + 1
    idfs.append(np.log10(X_tr_count.shape[0]/(df_count+1)))
# change to numpy array
idfs = np.array(idfs)
```

Then transform your count vectors to tf.idf vectors:

In [20]:

```
# fill in your code...
X_tr_tfidf = X_tr_count * idfs
X_dev_tfidf = X_dev_count * idfs
X_test_tfidf = X_test_count * idfs
```

In [21]:

```
X_tr_tfidf[1,:50]
```

Out[21]:

```
array([0.        , 1.27689632, 0.        , 0.        , 2.89431606,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       1.73115469, 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       1.16840443, 0.        , 1.84509804, 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.37675071, 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ])
```

Binary Logistic Regression

After obtaining vector representations of the data, now you are ready to implement Binary Logistic Regression for classifying sentiment.

First, you need to implement the `sigmoid` function. It takes as input:

- `z` : a real number or an array of real numbers

and returns:

- `sig` : the sigmoid of `z`

In [22]:

```
def sigmoid(z):
    # fill in your code...
    z = 1/(z+np.exp(-z))
    return z
```

In [23]:

```
print(sigmoid(0))
print(sigmoid(np.array([-5., 1.2])))
```

```
1.0
[0.00697286 0.66613633]
```

Then, implement the `predict_proba` function to obtain prediction probabilities. It takes as input:

- `X` : an array of inputs, i.e. documents represented by bag-of-ngram vectors ($N \times |vocab|$)
- `weights` : a 1-D array of the model's weights $(1, |vocab|)$

and returns:

- `preds_proba` : the prediction probabilities of X given the weights

In [24]:

```
def predict_proba(X, weights):
    preds_proba = sigmoid(np.matmul(X, weights.T))

    return preds_proba
```

Then, implement the `predict_class` function to obtain the most probable class for each vector in an array of input vectors. It takes as input:

- `X` : an array of documents represented by bag-of-ngram vectors ($N \times |vocab|$)
- `weights` : a 1-D array of the model's weights $(1, |vocab|)$

and returns:

- `preds_class` : the predicted class for each x in X given the weights

In [25]:

```
def predict_class(X, weights):
    pre_pro = predict_proba(X, weights)
    preds_class = (pre_pro > 0.5)

    return preds_class
```

To learn the weights from data, we need to minimise the binary cross-entropy loss. Implement `binary_loss` that takes as input:

- `X` : input vectors
- `Y` : labels
- `weights` : model weights
- `alpha` : regularisation strength

and return:

- `l` : the loss score

In [26]:

```
def binary_loss(X, Y, weights, alpha=0.00001):

    l = - np.mean(np.reshape(Y,[-1,1]) * np.log(predict_proba(X, weights)) + (1 - np.reshape(Y,[-1

    return l
```

Now, you can implement Stochastic Gradient Descent to learn the weights of your sentiment classifier. The `SGD` function takes as input:

- `X_tr` : array of training data (vectors)
- `Y_tr` : labels of `X_tr`
- `X_dev` : array of development (i.e. validation) data (vectors)
- `Y_dev` : labels of `X_dev`
- `lr` : learning rate
- `alpha` : regularisation strength
- `epochs` : number of full passes over the training data
- `tolerance` : stop training if the difference between the current and previous validation loss is smaller than a threshold
- `print_progress` : flag for printing the training progress (train/validation loss)

and returns:

- `weights` : the weights learned
- `training_loss_history` : an array with the average losses of the whole training set after each epoch
- `validation_loss_history` : an array with the average losses of the whole development set after each epoch

In [27]:

```
def SGD(X_tr, Y_tr, X_dev=[], Y_dev=[], loss="binary", lr=0.1, alpha=0.00001, epochs=5, tolerance=

    cur_loss_tr = 1.
    cur_loss_dev = 1.
    training_loss_history = []
    validation_loss_history = []

    # fill in your code...
    weights = np.zeros([1, X_tr.shape[1]])
    for i in range(epochs):
        # generate random order of Dtrain
        ran_order = np.random.permutation(X_tr.shape[0])
        for j in ran_order:
            Y_pred = predict_proba(X_tr[j:j+1,:], weights)
            weights = weights - lr * ((Y_pred - Y_tr[j])*X_tr[j:j+1,:] + 2*alpha*weights)
        training_loss_history.append(binary_loss(X_tr, Y_tr, weights, alpha))
        validation_loss_history.append(binary_loss(X_dev, Y_dev, weights, alpha))

        if print_progress:
            print('Epoch times: {}, Training loss value: {}, Validation loss value: {}'.format(i, t


        if len(validation_loss_history)>=2 and np.abs(validation_loss_history[-1] - validation_los
            break

    return weights, training_loss_history, validation_loss_history
```

Train and Evaluate Logistic Regression with Count vectors

First train the model using SGD:

In [28]:

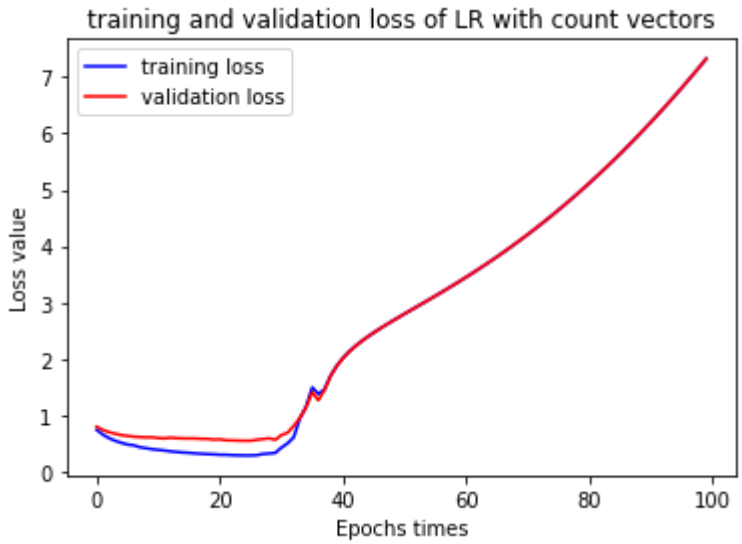
```
w_count, loss_tr_count, dev_loss_count = SGD(X_tr_count, Y_tr,
                                             X_dev=X_dev_count,
                                             Y_dev=Y_dev,
                                             lr=0.0001,
                                             alpha=0.001,
                                             epochs=100)
```

```
Epoch times: 0, Training loss value: 0.749131362931535, Validation loss value:
0.8046820309458966
Epoch times: 1, Training loss value: 0.6632104780039288, Validation loss value:
0.744169092692292
Epoch times: 2, Training loss value: 0.6012626358432159, Validation loss value:
0.7082495173510467
Epoch times: 3, Training loss value: 0.554256859573442, Validation loss value:
0.6792630611601946
Epoch times: 4, Training loss value: 0.5189381109621553, Validation loss value:
0.6554419213407493
Epoch times: 5, Training loss value: 0.4889336856158846, Validation loss value:
0.640525346232427
Epoch times: 6, Training loss value: 0.4746281930232947, Validation loss value:
0.6270564165245279
Epoch times: 7, Training loss value: 0.44159661547162926, Validation loss valu
e: 0.621170878026744
Epoch times: 8, Training loss value: 0.4233945730531655, Validation loss value:
0.6144345526848554
Epoch times: 9, Training loss value: 0.406143203261203, Validation loss value:
```

Now plot the training and validation history per epoch. Does your model underfit, overfit or is it about right? Explain why.

In [29]:

```
plt.plot(range(len(loss_tr_count)), loss_tr_count, c = 'blue', label='training loss')
plt.plot(range(len(dev_loss_count)), dev_loss_count,c = 'red', label = 'validation loss')
plt.xlabel('Epochs times')
plt.ylabel('Loss value')
plt.legend()
plt.title('training and validation loss of LR with count vectors')
plt.show()
```



From the graph above, we can know my model is overfitting, because the training loss is lower than the validation loss.

Compute accuracy, precision, recall and F1-scores:

In [30]:

```
preds_te_count = predict_class(X_test_count, w_count)
print('Accuracy:', accuracy_score(Y_te,preds_te_count))
print('Precision:', precision_score(Y_te,preds_te_count))
print('Recall:', recall_score(Y_te,preds_te_count))
print('F1-Score:', f1_score(Y_te,preds_te_count))
```

```
Accuracy: 0.5
Precision: 0.0
Recall: 0.0
F1-Score: 0.0

C:\software_program\anaconda3\lib\site-packages\sklearn\metrics\classification.py:14
37: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 due to no
predicted samples.
  'precision', 'predicted', average, warn_for)
C:\software_program\anaconda3\lib\site-packages\sklearn\metrics\classification.py:14
37: UndefinedMetricWarning: F-score is ill-defined and being set to 0.0 due to no pr
edicted samples.
  'precision', 'predicted', average, warn_for)
```

Finally, print the top-10 words for the negative and positive class respectively.

In [31]:

```
# fill in your code...
pre_word = predict_proba(X_test_count, w_count)
index = np.argsort(pre_word,axis = 0)
print('top 10 postive, 1-10')
for i in range(1,11):
    print(X_test_raw[index[-i,0]])
```

```
top 10 postive, 1-10
e tough souls who stay with it can marvel at the sleepy-eyed schwimmer , a houn
d dog with a head cold , who can go for over an hour without ever changing his
expression .
kolya is one of the richest films i've seen in some time .  zdenek sverak plays
a confirmed old bachelor ( who's likely to remain so ) , who finds his life as
a czech cellist increasingly impacted by the five-year old boy that he's taking
care of .  though it ends rather abruptly-- and i'm whining , 'cause i wanted t
o spend more time with these characters-- the acting , writing , and production
values are as high as , if not higher than , comparable american dramas .  this
father-and-son delight-- sverak also wrote the script , while his son , jan , d
irected-- won a golden globe for best foreign language film and , a couple days
after i saw it , walked away an oscar .  in czech and russian , with english su
btitles .
the second serial-killer thriller of the month is just awful .  oh , it starts
deceptively okay , with a handful of intriguing characters and some solid locat
ion work .  after a baby-sitter gets gutted in the suit- ably spooky someone's-
in-the-house prologue , parallel stories unfold , the first involving a texas s
heriff ( r . lee emery ) , a gruesome double murder , and the arrival of a moro
se fbi agent ( dennis quaid ) on the eve of voting for the local lawman's reele
ction .  the second pairs a hitch- hiker ( jared leto ) with a friendly former
```

In [32]:

```
# fill in your code...
print('top 10 negative 1-10')
for i in range(1,11):
    print(X_test_raw[index[i-1,0]])
```

```
top 10 negative 1-10
much ballyhoo has been made over this new version of " lolita , " made in a tim
e when one would think that a faithful adaptation of the infamous novel could b
e made , over its use of pedophilia , and as such , it's important to address i
t straight-forwardly , before any other ideals such as goodness and themes can
be discussed , as this film has been in film limbo for a number of years , lyin
g around in vaults san a distributor , and having critics waiting to either hai
l it a masterpiece or call it anticlimactic horseshit .  when seeing this film
, after all the hoopla , keeping in mind that there are people , namely me , wh
o are fans of the novel , who have been eagerly awaiting this flick since its c
reation . . . well ,  you just have to wonder why no one really picked it up fo
r distribution .  what's even worse is that seeing this " lolita , " especially
the first time and if you're familiar with anything " lolita , " is admittingly
very anticlimactic .  this is a real pity because when you really sit down to w
atch this film , ignoring all the crap that has preceeded it , it's really quit
e a film , perhaps the best film by director adrian lyne ( although , really ,
look at its competition : " flashdance , " " fatal attraction , " and " indecen
t proposal " ) , at least besides " jacob's ladder . "  i've seen this " lolita
" twice : the first time , i wasn't so blown away .  it seemed overly dramatic
```

If we were to apply the classifier we've learned into a different domain such laptop reviews or restaurant reviews, do you think these features would generalise well? Can you propose what features the classifier could pick up as important in the new domain?

I think it might be a little difficult to apply this classifier,because this classifier is about movies, emotional vocabulary will account for the vast majority. However, laptops may have comparisons such as numerical values, comparison of objective attributes, such as comparison of CPU and GPU models, and comparison of such attributes.

Train and Evaluate Logistic Regression with TF.IDF vectors

Follow the same steps as above (i.e. evaluating count n-gram representations).

In [33]:

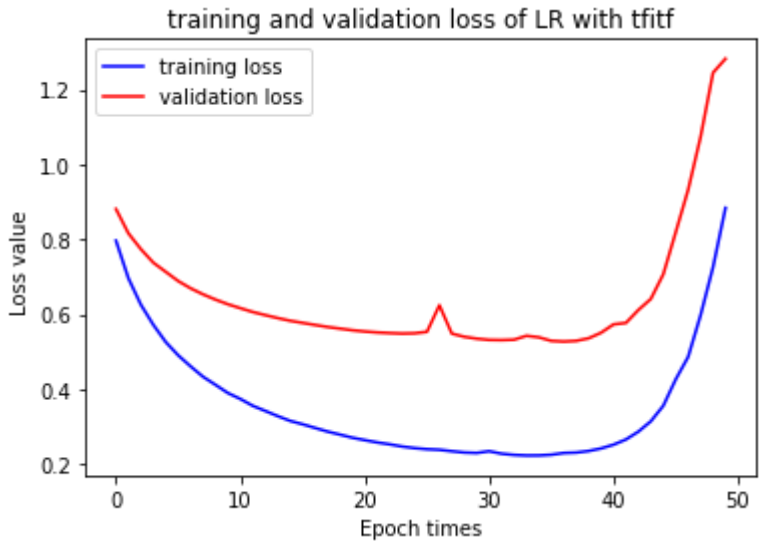
```
w_tfidf, trl, devl = SGD(X_tr_tfidf, Y_tr,
                         X_dev=X_dev_tfidf,
                         Y_dev=Y_dev,
                         lr=0.0001,
                         alpha=0.00001,
                         epochs=50)
```

```
Epoch times: 0, Training loss value: 0.7961803324262857, Validation loss value:
0.8805536715692955
Epoch times: 1, Training loss value: 0.695720397715392, Validation loss value:
0.8157106366431993
Epoch times: 2, Training loss value: 0.625428781520306, Validation loss value:
0.7736591046570265
Epoch times: 3, Training loss value: 0.5717758283666531, Validation loss value:
0.7375531347311709
Epoch times: 4, Training loss value: 0.5258114869056711, Validation loss value:
0.7130212236252071
Epoch times: 5, Training loss value: 0.4903877821693669, Validation loss value:
0.6884843956364922
Epoch times: 6, Training loss value: 0.460763008736669, Validation loss value:
0.6690819661256523
Epoch times: 7, Training loss value: 0.4330799677798458, Validation loss value:
0.6529878232147064
Epoch times: 8, Training loss value: 0.41158260673321195, Validation loss valu
e: 0.6389297020287383
Epoch times: 9, Training loss value: 0.38965605207868853, Validation loss valu
```

Now plot the training and validation history per epoch. Does your model underfit, overfit or is it about right? Explain why.

In [34]:

```
# fill in your code...
plt.plot(range(len(trl)), trl, c = 'blue', label='training loss')
plt.plot(range(len(devl)), devl, c = 'red',label = 'validation loss')
plt.xlabel('Epoch times')
plt.ylabel('Loss value ')
plt.legend()
plt.title('training and validation loss of LR with tfitf')
plt.show()
```



Answer:this model is also overfit, because the training loss is lower than the validation loss.

Compute accuracy, precision, recall and F1-scores:

In [35]:

```
# fill in your code...
preds_te = predict_class(X_test_tfidf, w_tfidf)
print('Accuracy:', accuracy_score(Y_te,preds_te))
print('Precision:', precision_score(Y_te,preds_te))
print('Recall:', recall_score(Y_te,preds_te))
print('F1-Score:', f1_score(Y_te,preds_te))
```

```
Accuracy: 0.4925
Precision: 0.4944237918215613
Recall: 0.665
F1-Score: 0.5671641791044776
```

Print top-10 most positive and negative words:

In [36]:

```
# fill in your code...
pred_te_tfidf = predict_proba(X_test_tfidf,w_tfidf)
# get index by ascend order
index = np.argsort(pred_te_tfidf,axis = 0)
print('Top10 positive:1-10')
for i in range(1,11):
    print(X_test_raw[index[-i,0]])
```

```
Top10 positive:1-10
[that the word "bitch" is their favorite word, as it seemed to crop up in ever
y sentence they used .  it makes for a film with no surprises or nothing fresh
to say , but if one has a sense a humor for what their lifestyle is about , the
n some of what they rap about might seem amusing .  they seem to all want to be
thought of as businessmen , in the business because it is the easiest way for t
hem to make big money .  it's also a power trip , accomplished by manipulating
the girls to work for them , mostly by humiliating them and keeping them in pla
ce .  this sleazy pic , soon became grating and wore out its welcome to my unre
ceptive ears .  these verbose pimps had a smart answer for everything and never
knew when to shut their face .  the hughes brothers used as their pimp role mod
els , the feather-hatted , fur-coated , diamond ring-wearing , gold chain weari
ng , flashy cadillac-cruising pimp of the late '70s blaxploitation movies--like
the mack and willie dynamite .  also used as reference , was iceberg slim's bes
t-seller pimp , the story of my life .  we meet pimps such as : fillmore slim ,
c-note , charm , k-red , gorgeous dre , bishop don magic juan , and rosebudd .
they readily discuss their business arrangements : including percentages , life
styles , knockin ( stealing another pimp's ho ) , and the thrill they get from
women giving them money .  these dudes needed no prompting to talk , as they ju
st love to brag about about themselves .  " priests need nuns , '' yaps c-note
, a san francisco pimp .  " doctors need nurses .  so ho's need pimps . ''  as
[that the word "bitch" is their favorite word, as it seemed to crop up in ever
```

In [37]:

```
# fill in your code...
print('Top10 negative: 1 - 10')
for i in range(1,11):
    print(X_test_raw[index[i-1,0]])
```

```
Top10 negative: 1 - 10
[a batman feature film, casually kicking at a pathetic looking rubber shar
k attached to his leg  i had never thought that an entry in the modern incarnat
ion of the batman feature film would approach this level of campiness , but in
many instances batman and robin nears , and at some point even exceeds this sta
ndard .  this is a disasterously bad film , easily the worst in the series to d
ate , and fairly epitomizes a cinematic definition of the word excessive - it's
loud , garish , and obnoxious , with pointless , gratuitous action sequences an
d set pieces which clutter up the screen with elaborate production design to th
e point of overkill .  batman and robin features the caped crusaders ( george c
looney debuting as batman , with chris o'donnell returing as robin ) squaring o
ff against another bevy of chemically-induced villains - the nefarious ice-cold
mr . freeze ( arnold schwarzenegger ) , armed with a weapon which freezes every
thing in its sights , and the slinky poison ivy ( uma thurman ) , who has the a
bility to blow powerful love dust into the faces of men in order so that they w
ill fall helplessly in love with her ( not that the dust is really necessary to
accomplish this result , but whatever ) , and then dispatch them with a poisone

d kiss .  by ivy's side is the giant steroid monster bane ( jeep swanson ) , a
grunting hulk of a beast .  the villains' goals are noble ones - freeze steals
diamonds to power his climate suit ( in order to keep his body temperature at z
ero degrees )  so that he can survive in order to devise a cure for his belove
[a batman feature film, casually kicking at a pathetic looking rubber shar
```

Discuss how did you choose model hyperparameters (e.g. learning rate and regularisation strength)? What is the relation between training epochs and learning rate? How the regularisation strength affects performance?

Enter your answer here... Hyperparameter is usually selected very carefully, such as the learning rate, if the learning rate is too large, then the number of epochs will be reduced, which will lead to underfit, and if the learning rate is too small, then the model running time will increase significantly, the number of times Will increase, and easily lead to overfitting. And regularisation strength can prevent overfitting

Full Results

Add here your results:

LR	Precision	Recall	F1-Score
BOW-count	0.7741935483870968	0.84	0.8057553956834531
BOW-tfidf	0.7658536585365854	0.785	0.7753086419753088

Multi-class Logistic Regression

Now you need to train a Multiclass Logistic Regression (MLR) Classifier by extending the Binary model you developed above. You will use the MLR model to perform topic classification on the AG news dataset consisting of three classes:

- Class 1: World
- Class 2: Sports
- Class 3: Business

You need to follow the same process as in Task 1 for data processing and feature extraction by reusing the functions you wrote.

In [38]:

```
# fill in your code...
data_tr = pd.read_csv('./data_topic/train.csv', names=["class","article"])
data_dev = pd.read_csv('./data_topic/dev.csv',  names=["class","article"])
data_te = pd.read_csv('./data_topic/test.csv',  names=["class","article"])
```

In [39]:

```
data_tr.head()
```

Out[39]:

	class	article
0	1	Reuters - Venezuelans turned out early\and in ...
1	1	Reuters - South Korean police used water canno...
2	1	Reuters - Thousands of Palestinian\prisoners i...
3	1	AFP - Sporadic gunfire and shelling took place...
4	1	AP - Dozens of Rwandan soldiers flew into Suda...

In [40]:

```
# fill in your code...
X_tr_raw = []
Y_tr = []
X_dev_raw = []
Y_dev = []
X_test_raw = []
Y_te = []

# add all raw data into list
for i in range(len(data_tr)):
    X_tr_raw.append(data_tr.iat[i, 1])
    Y_tr.append(data_tr.iat[i, 0])

for i in range(len(data_dev)):
    X_dev_raw.append(data_dev.iat[i, 1])
    Y_dev.append(data_dev.iat[i, 0])

for i in range(len(data_te)):
    X_test_raw.append(data_te.iat[i, 1])
    Y_te.append(data_te.iat[i, 0])
```

In [41]:

```
vocab, df, ngram_counts = get_vocab(X_tr_raw, ngram_range=(1,3), keep_topN=5000, stop_words=stop_wo
print(len(vocab))
print()
print(list(vocab)[:100])
print()
print(df.most_common()[:10])
```

```
5000

[('legg', 'mason', 'tennis'), ('investor', 'reuters'), ('bj', 'wholesale'), ('out',
'had', 'added'), 'getting', 'after', 'join', 'ask', ('region', 'south', 'ossetia'),
'herat', 'virginia', ('target', 'stocks'), ('third', 'round'), ('fell', 'their', 'lo
west'), ('fell', 'more'), 'tellers', ('tomas', 'berdych'), 'pettitte', ('world', 'se
cond', 'largest'), ('dream', 'team'), ('satellite', 'operator', 'said'), 'fueled',
'ongoing', ('commission', 'expected', 'declare'), 'wholesale', ('out', 'prospectiv
e'), 'payments', ('sunday', 'killing', 'least'), 'drew', 'issues', 'path', 'travel',
'florida', 'climbed', 'president', ('expected', 'declare', 'initial'), ('exporters',
'such', 'toyota'), ('miss', 'season', 'after'), ('third', 'straight'), 'groups', 'rb
i', ('financial', 'services'), ('beijing', 'reuters'), ('washington', 'reuters'),
('retailer', 'behind'), 'efforts', ('blue', 'jays'), ('more', 'than', 'doubled'), 's
ettler', 'income', ('al', 'sadr'), 'really', 'slump', ('government', 'reported'),
('had', 'decided'), 'fell', 'mariel', 'done', ('routines', 'career', 'win'), ('elect
ion', 'year'), 'list', 'progress', ('oil', 'exporter'), 'special', ('concern', 'ove
r'), ('jewish', 'settlements', 'west'), ('way', 'playoff', 'sunday'), 'customer', 'k
mart', ('barrel', 'wednesday'), 'regional', 've', ('beijing', 'reuters', 'china'),
'average', 'arafat', 'twice', 'quote', 'allen', 'bringing', ('start', 'trading', 'na
sdaq'), 'victories', 'reliever', 'kong', ('corey', 'koskie'), ('crashed', 'out'),
('economic', 'reports', 'showed'), 'security', 'promised', ('strained', 'right'), 'h
elp', ('shi', 'ite', 'cleric'), ('york', 'san', 'francisco'), 'dug', ('three', 'year
s'), ('billion', 'wednesday'), 'investigation', 'assets', ('season', 'after', 'teari
ng'), 'trailed', ('reuters', 'venezuela', 'left')]

[('reuters', 631), ('said', 432), ('tuesday', 413), ('wednesday', 344), ('new', 32
5), ('after', 295), ('ap', 275), ('athens', 245), ('monday', 221), ('first', 210)]
```

In [42]:

```
# fill in your code...
word_id = {}
i = 0
for i, w in enumerate(list(vocab)):
    word_id[w] = i
# generate 3 file ngram list
X_train_ngram = []
X_dev_ngram = []
X_test_ngram = []

for i in X_tr_raw:
    X_train_ngram.append(extract_ngrams(i, ngram_range=(1,3), token_pattern=r'\b[A-Za-z][A-Za-z]+\b
for i in X_dev_raw:
    X_dev_ngram.append(extract_ngrams(i, ngram_range=(1,3), token_pattern=r'\b[A-Za-z][A-Za-z]+\b',
for i in X_test_raw:
    X_test_ngram.append(extract_ngrams(i, ngram_range=(1,3), token_pattern=r'\b[A-Za-z][A-Za-z]+\b'

    # vectorise 3 list
X_tr_count = vectorise(X_train_ngram, vocab)
X_dev_count = vectorise(X_dev_ngram, vocab)
X_test_count = vectorise(X_test_ngram, vocab)
```

In [43]:

```
idfs = []
for i in list(vocab):
    # set a variable to record occurance times
    df_count = 0
    # if it occur, variable + 1
    for j in X_train_ngram:
        if i in j:
            df_count = df_count + 1
    idfs.append(np.log10(X_tr_count.shape[0]/(df_count+1)))
# change to numpy array
idfs = np.array(idfs)

X_tr_tfidf = X_tr_count * idfs
X_dev_tfidf = X_dev_count * idfs
X_test_tfidf = X_test_count * idfs
```

Now you need to change `SGD` to support multiclass datasets. First you need to develop a `softmax` function. It takes as input:

- `z` : array of real numbers

and returns:

- `smax` : the softmax of `z`

In [44]:

```
def softmax(z):

    # fill in your code...
    smax = np.exp(z) / np.sum(np.exp(z),1,keepdims=True)
    return smax
```

Then modify `predict_proba` and `predict_class` functions for the multiclass case:

In [45]:

```
def predict_proba(X, weights):

    # fill in your code...
    preds_proba = softmax(np.matmul(X,weights.T))
    return preds_proba
```

In [46]:

```
def predict_class(X, weights):

    # fill in your code...
    pre_pro = predict_proba(X, weights)
    preds_class = np.argmax(pre_pro, 1)

    return preds_class+1
```

Toy example and expected functionality of the functions above:

In [47]:

```
X = np.array([[0.1,0.2],[0.2,0.1],[0.1,-0.2]])
w = np.array([[2,-5],[-5,2]])
```

In [48]:

```
predict_proba(X, w)
```

Out[48]:

```
array([[0.33181223, 0.66818777],
       [0.66818777, 0.33181223],
       [0.89090318, 0.10909682]])
```

In [49]:

```
predict_class(X, w)
```

Out[49]:

```
array([2, 1, 1], dtype=int64)
```

Now you need to compute the categorical cross entropy loss (extending the binary loss to support multiple classes).

In [50]:

```
def categorical_loss(X, Y, weights, num_classes=5, alpha=0.00001):
    # fill in your code...
    l = 0
    log_pb = np.log(predict_proba(X, weights))
    for i in range(X.shape[0]):
        l = l + log_pb[i, Y[i]-1]
    l = l/X.shape[0]
    l = alpha * np.sum(weights**2) - l
    return l
```

Finally you need to modify SGD to support the categorical cross entropy loss:

In [51]:

```
def SGD(X_tr, Y_tr, X_dev=[], Y_dev=[], num_classes=5, lr=0.01, alpha=0.00001, epochs=5, tolerance

    # fill in your code...
    training_loss_history = []
    validation_loss_history = []

    weights = np.zeros([num_classes, X_tr.shape[1]])
    for i in range(epochs):
        #  generate random order of Dtrain
        ran_order = np.random.permutation(X_tr.shape[0])
        for j in ran_order:
            Z_tr = predict_proba(X_tr[j:j+1,:], weights)
            Z_tr[0, Y_tr[j]-1] -= 1
            weights -=  lr * ( Z_tr.T * X_tr[j:j+1,:] + 2*alpha*weights)

        # append every loss results into training and validation loss
        training_loss_history.append(categorical_loss(X_tr, Y_tr, weights, alpha))
        validation_loss_history.append(categorical_loss(X_dev, Y_dev, weights, alpha))

        if print_progress:
            print('Epoch times: {}, Training loss value: {}, Validation loss: {}'.format(i, trainin
        if len(validation_loss_history)>=2 and np.abs(validation_loss_history[-1] - validation_los
            break

    return weights, training_loss_history, validation_loss_history
```

Now you are ready to train and evaluate you MLR following the same steps as in Task 1 for both Count and tfidf features:

In [52]:

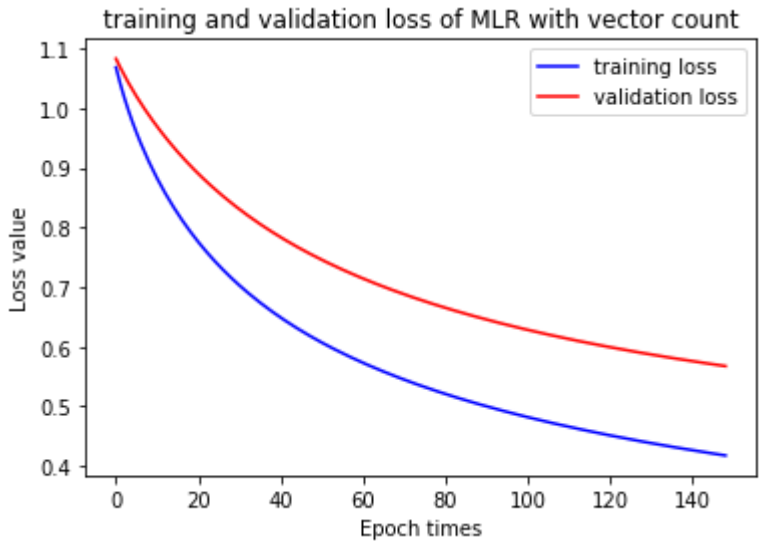
```
w_count, loss_tr_count, dev_loss_count = SGD(X_tr_count, Y_tr,
                                             X_dev=X_dev_count,
                                             Y_dev=Y_dev,
                                             num_classes=3,
                                             lr=0.0001,
                                             alpha=0.001,
                                             epochs=200)
```

```
Epoch times: 9, Training loss value: 0.8958815880622502, Validation loss: 0.978
6220761981399
Epoch times: 10, Training loss value: 0.8822551254168536, Validation loss: 0.96
92374200144914
Epoch times: 11, Training loss value: 0.8693004022194268, Validation loss: 0.96
01669939751173
Epoch times: 12, Training loss value: 0.8569665192873845, Validation loss: 0.95
13940135091511
Epoch times: 13, Training loss value: 0.8451930157003291, Validation loss: 0.94
2899140386813
Epoch times: 14, Training loss value: 0.8339432220070775, Validation loss: 0.93
46682889910447
Epoch times: 15, Training loss value: 0.8231818321222316, Validation loss: 0.92
66875478339072
Epoch times: 16, Training loss value: 0.812867766895967, Validation loss: 0.918
9461539837518
Epoch times: 17, Training loss value: 0.8029745539649112, Validation loss: 0.91
14303909053281
Epoch times: 18, Training loss value: 0.7934690361574589, Validation loss: 0.90
41320951317005
```

Plot training and validation process and explain if your model overfit, underfit or is about right:

In [53]:

```
# fill in your code...
plt.plot(range(len(loss_tr_count)), loss_tr_count, c = 'blue', label='training loss')
plt.plot(range(len(dev_loss_count)), dev_loss_count,c= 'red', label = 'validation loss', )
plt.xlabel('Epoch times')
plt.ylabel('Loss value')
plt.legend()
plt.title('training and validation loss of MLR with vector count')
plt.show()
```



Answer:this model is also overfit, because the training loss is lower than the validation loss.

Compute accuracy, precision, recall and F1-scores:

In [54]:

```
# fill in your code...

preds_te = predict_class(X_test_count, w_count)
print('Accuracy:', accuracy_score(Y_te,preds_te))
print('Precision:', precision_score(Y_te,preds_te,average='macro'))
print('Recall:', recall_score(Y_te,preds_te,average='macro'))
print('F1-Score:', f1_score(Y_te,preds_te,average='macro'))
```

```
Accuracy: 0.8522222222222222
Precision: 0.8545592790763165
Recall: 0.8522222222222222
F1-Score: 0.8514022251643323
```

Print the top-10 words for each class respectively.

In [55]:

```
# fill in your code...

classes = ['1', '2', '3']
preds_test_count = predict_proba(X_test_count, w_count)

for i, j in enumerate(classes):
    print(j)
    loc = np.argsort(preds_test_count[:,i])
    for i in range(10):
        print('Top10 ',i+1, X_test_raw[loc[-1-i]])
```

```
1
Top10  1 NAJAF, Iraq - Iraq's most powerful Shiite cleric returned home from Britain
on Wednesday to help broker an end to nearly three weeks of fighting in Najaf and is
calling on his followers to join him in a march to reclaim the holy city, his spokes
men and witnesses said.    Grand Ayatollah Ali Husseini al-Sistani return came as he
avy fighting persisted in Najaf's Old City...
Top10  2 BAGHDAD, Iraq - Rebel Shiite cleric Muqtada al-Sadr called for his follower
s across Iraq to end fighting against U.S. and Iraqi forces and is planning to join
the political process in the coming days, an al-Sadr aide said Monday...
Top10  3 NAJAF, Iraq - Militants loyal to radical Shiite cleric Muqtada al-Sadr kept
their hold on a revered shrine, and clashes flared in Najaf on Saturday, raising fea
rs that a resolution to the crisis in the holy city could collapse amid bickering be
tween Shiite leaders.    The clashes between U.S...
Top10  4 BAGHDAD, Iraq - Delegates at Iraq's National Conference called on radical S
hiite cleric Muqtada al-Sadr to abandon his uprising against U.S. and Iraqi troops a
nd pull his fighters out of a holy shrine in Najaf...
Top10  5  NAJAF, Iraq (Reuters) - A mortar attack on a packed mosque  in the town of
Kufa on Thursday killed at least 25 people as  Iraq's most influential Shi'ite cleri
c headed to the nearby  holy city of Najaf to try to end a bloody three-week uprisin
g.
Top10  6  NAJAF, Iraq (Reuters) - Rebel Shi'ite fighters appeared  still to be in co
ntrol of the Imam Ali mosque in the Iraqi city  Najaf early on Saturday, but the whe
reabouts of their leader,  the fiery cleric Moqtada al-Sadr, were unknown.
Top10  7 A top aide to Iraq #39;s rebel Shi #39;ite leader Muqtada al-Sadr Monday ca
lled on the Mehdi Army militia to cease fire across Iraq and said Sadr was preparing
to announce plans for a major political program.
Top10  8 NAJAF, Iraq - Explosions and gunfire rattled through the city of Najaf as
U.S. troops in armored vehicles and tanks rolled back into the streets here Sunday,
a day after the collapse of talks - and with them a temporary cease-fire - intended
to end the fighting in this holy city...
Top10  9 US and Iraqi forces battled militants in Najaf on Tuesday and Iraqi Nationa
l Guardsmen advanced to within 200 yards of the holy city #39;s Imam Ali Shrine comp
ound, where insurgents loyal to radical cleric Muqtada al-Sadr have been holed up fo
r weeks.
Top10  10  NAJAF, Iraq (Reuters) - Rebel Iraqi cleric Moqtada al-Sadr  on Friday ord
ered his men inside Najaf's Imam Ali mosque to lay  down their weapons and join thou
sands of Shi'ite pilgrims  outside the shrine.
2
Top10  1  ATHENS (Reuters) - The U.S. women's basketball team showed  their men how
to win gold Saturday as around 70,000 spectators  flocked to the Olympic stadium for
a hectic athletics program  on the penultimate night of the Athens Games.
Top10  2  ATHENS (Reuters) - The U.S. men's basketball team was  beaten by Argentina
Friday, denying it an Olympic gold medal  for the first time since 1992 when NBA pla
yers started  competing.
Top10  3 ATHENS, Greece - Right now, the Americans aren't just a Dream Team - they'r
e more like the Perfect Team. Lisa Fernandez pitched a three-hitter Sunday and Cryst
```

l Bustos drove in two runs as the Americans rolled to their eighth shutout in eight days, 5-0 over Australia, putting them into the gold medal game...
Top10 4 ATHENS (Reuters) - Carly Patterson upstaged Russian diva Svetlana Khorkin a to become the first American in 20 years to win the women's Olympic gymnastics al l-round gold medal on Thursday.
Top10 5 ATHENS (Reuters) - Greek sprinters Costas Kenteris and Katerina Thanou have arrived at an Athens hotel for an International Olympic Committee (IOC) hearing into their missed doped tests, a saga that has shamed and angered the Olympic host ...
Top10 6 Athens, Greece (Sports Network) - Wednesday night it was Paul Hamm #39;s tu rn to shine for the United States, as he won the gold medal in the men #39;s all-aro und competition. Will Thursday produce a sweep for the US at the Olympics? ...
Top10 7 ATHENS (Reuters) - Aaron Peirsol won his second gold medal at the Athens Olympics Thursday after winning an appeal against his disqualification from the me n's 200 meter backstroke.
Top10 8 ATHENS (Reuters) - An exhausted Nicolas Massu reeled in Mardy Fish in fiv e tortuous sets on Sunday to win Chile their second gold medal at an Olympic Games less than 24 hours after helping them to their first.
Top10 9 ATHENS, Aug. 19 (Xinhuanet) -- Chinese Hercules Liu Chunhong Thursday lifte d three world records on her way to winning the women #39;s 69kg gold medal at the A thens Olympics, the fourth of the power sport competition for China.
Top10 10 ATHENS (Reuters) - World 100 meters champion Torri Edwards will miss the Athens Olympics after her appeal against a two-year drugs ban was dismissed on Tues day, a source told Reuters.
3
Top10 1 NEW YORK (Reuters) - Retailer Kmart Holdings Corp. <A HREF="http://www. investor.reuters.com/FullQuote.aspx?ticker=KMRT.O target=/stocks/quickinfo/fullquot e">KMRT.O</A> on Monday said it finalized a deal to sell 18 of its stores to Home Depot Inc. <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticke r=HD.N target=/stocks/quickinfo/fullquote">HD.N</A> for \$271 million.
Top10 2 BRUSSELS/SAO PAULO (Reuters) - Shareholders gave their blessing on Friday for Belgium's Interbrew <A HREF="http://www.investor.reuters.com/FullQuote.aspx?t icker=INTB.BR target=/stocks/quickinfo/fullquote">INTB.BR</A> to buy Brazi l's AmBev <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=AMBV4.SA target=/stocks/quickinfo/fullquote">AMBV4.SA</A><ABV.N> in a \$9.7 bi llion deal that will create the world's largest brewer.
Top10 3 <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=RSE.N ta rget=/stocks/quickinfo/fullquote">RSE.N</A> jumped before the bell after Ge neral Growth Properties Inc. <A HREF="http://www.investor.reuters.com/FullQuote. aspx?ticker=GGP.N target=/stocks/quickinfo/fullquote">GGP.N</A>, the No. 2 U.S. shopping mall owner, on Friday said it would buy Rouse for \$7.2 billion.
Top10 4 NEW YORK (Reuters) - Staples Inc. <A HREF="http://www.investor.reuters. com/FullQuote.aspx?ticker=SPLS.O target=/stocks/quickinfo/fullquote">SPLS.O</A >, the top U.S. office products retailer, on Tuesday reported a 39 percent jump in quarterly profit, raised its full-year forecast and said it plans to enter the f ast-growing Chinese market, sending its shares higher.
Top10 5 NEW YORK (Reuters) - BlackRock Inc. <A HREF="http://www.investor.reuter s.com/FullQuote.aspx?ticker=BLK.N target=/stocks/quickinfo/fullquote">BLK.N</A >, one of the largest U.S. fixed income managers, on Thursday said it will buy its far smaller competitor State Street Research Management Co., marking the bigg est takeover in the asset management business this year.
Top10 6 NEW YORK (Reuters) - Colgate-Palmolive Co. <A HREF="http://www.investo r.reuters.com/FullQuote.aspx?ticker=CL.N target=/stocks/quickinfo/fullquote">CL.N </A> will cut about 4,400 jobs, or 12 percent of its work force, and close n early a third of its factories under a restructuring, the consumer products company said on Tuesday.
Top10 7 NEW YORK (Reuters) - U.S. blue chips were near the unchanged mark on Mond ay as a disappointing sales forecast from retailer Wal-Mart Stores Inc. <A HREF ="http://www.investor.reuters.com/FullQuote.aspx?ticker=WMT.N target=/stocks/quickin fo/fullquote">WMT.N</A> dampened sentiment, offsetting the benefit of easi ng oil prices.
Top10 8 NEW YORK (Reuters) - Verizon Communications Inc. <A HREF="http://www.in

```
vestor.reuters.com/FullQuote.aspx?ticker=VZ.N target=/stocks/quickinfo/fullquote"&g
t;VZ.N&lt;/A&gt; is  near an agreement to sell its Canadian telephone directory  bus
iness to private equity firm Bain Capital, the New York Post  said on Wednesday.
Top10  9  NEW YORK (Reuters) - Hartford Financial Services Group Inc.  &lt;A HREF="h
ttp://www.investor.reuters.com/FullQuote.aspx?ticker=HIG.N target=/stocks/quickinfo/
fullquote"&gt;HIG.N&lt;/A&gt; on Tuesday became the latest insurer to issue a profit
warning tied to Hurricane Charley, the strongest storm to hit  Florida in a dozen ye
ars.
Top10  10  NEW YORK (Reuters) - Intel Corp's &lt;A HREF="http://www.investor.reuter
s.com/FullQuote.aspx?ticker=INTC.O target=/stocks/quickinfo/fullquote"&gt;INTC.O&l
t;/A&gt; sharp cut in its  revenue outlook dragged down shares of personal computer
makers  on Friday, on fears that the chipmaker's problems could signal  weak PC mark
ets, analysts said.
```

Discuss how did you choose model hyperparameters (e.g. learning rate and regularisation strength)? What is the relation between training epochs and learning rate? How the regularisation strength affects performance?

Hyperparameter is usually selected very carefully, such as the learning rate, if the learning rate is too large, then the number of epochs will be reduced, which will lead to underfit, and if the learning rate is too small, then the model running time will increase significantly, the number of times Will increase, and easily lead to overfitting. And regularisation strength can prevent overfitting

Now evaluate BOW-tfidf...

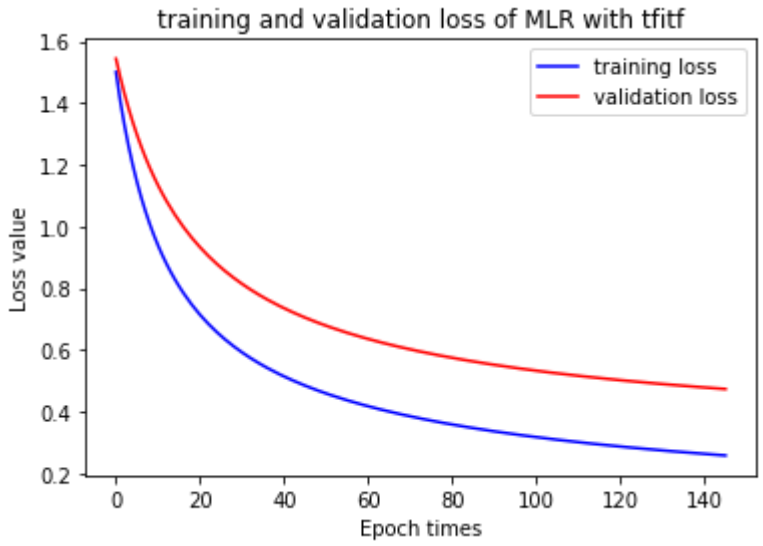
In [56]:

```
w_tfidf, trl, devl = SGD(X_tr_tfidf, Y_tr,
                         X_dev=X_dev_tfidf,
                         Y_dev=Y_dev,
                         lr=0.0001,
                         alpha=0.001,
                         epochs=200)
```

```
Epoch times: 2, Training loss value: 1.3289855538609154, Validation loss: 1.430
841683674241
Epoch times: 3, Training loss value: 1.2597672788385388, Validation loss: 1.382
1813732710295
Epoch times: 4, Training loss value: 1.1983043592892662, Validation loss: 1.337
5757170599305
Epoch times: 5, Training loss value: 1.1433543600325324, Validation loss: 1.296
5669466810905
Epoch times: 6, Training loss value: 1.0939411745771432, Validation loss: 1.258
731305284789
Epoch times: 7, Training loss value: 1.0493390764071575, Validation loss: 1.223
7643777197835
Epoch times: 8, Training loss value: 1.0088688292211838, Validation loss: 1.191
335719087575
Epoch times: 9, Training loss value: 0.9720292648799856, Validation loss: 1.161
2182279390817
Epoch times: 10, Training loss value: 0.9383503236564024, Validation loss: 1.13
31544270100045
Epoch times: 11, Training loss value: 0.9074732981303807, Validation loss: 1.10
```

In [57]:

```
plt.plot(range(len(trl)), trl, c = 'blue', label='training loss')
plt.plot(range(len(devl)), devl,c = 'red', label = 'validation loss' )
plt.xlabel('Epoch times')
plt.ylabel('Loss value')
plt.legend()
plt.title('training and validation loss of MLR with tfitf')
plt.show()
```



this model is also overfit, because the training loss is lower than the validation loss

In [58]:

```
preds_te = predict_class(X_test_tfidf, w_tfidf)
print('Accuracy:', accuracy_score(Y_te,preds_te))
print('Precision:', precision_score(Y_te,preds_te,average='macro'))
print('Recall:', recall_score(Y_te,preds_te,average='macro'))
print('F1-Score:', f1_score(Y_te,preds_te,average='macro'))
```

```
Accuracy: 0.88
Precision: 0.881547680555811
Recall: 0.8799999999999999
F1-Score: 0.8793734034496704
```

In [59]:

```
classes = ['1', '2', '3']
pred_test_tfidf = predict_proba(X_test_tfidf, w_tfidf)

for i, j in enumerate(classes):
    print(j)
    loc = np.argsort(pred_test_tfidf[:,i])
    for i in range(10):
        print('Top10 ',i+1, X_test_raw[loc[-1-i]])
```

```
1
Top10  1 NAJAF, Iraq - Iraq's most powerful Shiite cleric returned home from Britain
on Wednesday to help broker an end to nearly three weeks of fighting in Najaf and is
calling on his followers to join him in a march to reclaim the holy city, his spokes
men and witnesses said.    Grand Ayatollah Ali Husseini al-Sistani return came as he
avy fighting persisted in Najaf's Old City...
Top10  2 BAGHDAD, Iraq - Rebel Shiite cleric Muqtada al-Sadr called for his follower
s across Iraq to end fighting against U.S. and Iraqi forces and is planning to join
the political process in the coming days, an al-Sadr aide said Monday...
Top10  3 NAJAF, Iraq - Militants loyal to radical Shiite cleric Muqtada al-Sadr kept
their hold on a revered shrine, and clashes flared in Najaf on Saturday, raising fea
rs that a resolution to the crisis in the holy city could collapse amid bickering be
tween Shiite leaders.    The clashes between U.S...
Top10  4 BAGHDAD, Iraq - Delegates at Iraq's National Conference called on radical S
hiite cleric Muqtada al-Sadr to abandon his uprising against U.S. and Iraqi troops a
nd pull his fighters out of a holy shrine in Najaf...
Top10  5  NAJAF, Iraq (Reuters) - A mortar attack on a packed mosque  in the town of
Kufa on Thursday killed at least 25 people as  Iraq's most influential Shi'ite cleri
c headed to the nearby  holy city of Najaf to try to end a bloody three-week uprisin
g.
Top10  6 AFP - Democratic White House hopeful Senator John Kerry warned that Preside
nt George W. Bush's plan to withdraw 70,000 troops from Europe and Asia would hinder
the war on terrorism and embolden North Korea.
Top10  7 US forces and radical Shiite cleric Muqtada al-Sadr #39;s militia battled S
aturday in Baghdad even as the truce that ended the bloody fighting between US-Iraqi
troops and the militia forces in Najaf held for a second day.
Top10  8  NAJAF, Iraq (Reuters) - Rebel Shi'ite fighters appeared  still to be in co
ntrol of the Imam Ali mosque in the Iraqi city  Najaf early on Saturday, but the whe
reabouts of their leader,  the fiery cleric Moqtada al-Sadr, were unknown.
Top10  9 US and Iraqi forces battled militants in Najaf on Tuesday and Iraqi Nationa
l Guardsmen advanced to within 200 yards of the holy city #39;s Imam Ali Shrine comp
ound, where insurgents loyal to radical cleric Muqtada al-Sadr have been holed up fo
r weeks.
Top10  10 AFP - Georgian and South Ossetian forces overnight accused each other of t
rying to storm the other side's positions in Georgia's breakaway region of South Oss
etia, as four Georgian soldiers were reported to be wounded.
2
Top10  1  ATHENS (Reuters) - The U.S. women's basketball team showed  their men how
to win gold Saturday as around 70,000 spectators  flocked to the Olympic stadium for
a hectic athletics program  on the penultimate night of the Athens Games.
Top10  2 Athens, Greece (Sports Network) - Wednesday night it was Paul Hamm #39;s tu
rn to shine for the United States, as he won the gold medal in the men #39;s all-aro
und competition. Will Thursday produce a sweep for the US at the Olympics? ...
Top10  3  ATHENS (Reuters) - The U.S. men's basketball team was  beaten by Argentina
Friday, denying it an Olympic gold medal  for the first time since 1992 when NBA pla
yers started  competing.
Top10  4 Andruw Jones hit a two-run homer off Trevor Hoffman in the ninth inning and
the Atlanta Braves threw out the potential tying run at the plate for the final out
```

Wednesday night, preserving a 6-5 come-from-behind win over the San Diego Padres.
Top10 5 ATHENS (Reuters) - World 100 meters champion Torri Edwards will miss the Athens Olympics after her appeal against a two-year drugs ban was dismissed on Tuesday, a source told Reuters.
Top10 6 Heather O #39;Reilly, minutes after missing a wide open net, scored in the ninth minute of overtime Monday to give the United States a 2-1 victory over World Cup champion Germany and a place in Thursday #39;s gold-medal game.
Top10 7 ATHENS, Greece - Right now, the Americans aren't just a Dream Team - they're more like the Perfect Team. Lisa Fernandez pitched a three-hitter Sunday and Crystl Bustos drove in two runs as the Americans rolled to their eighth shutout in eight days, 5-0 over Australia, putting them into the gold medal game...
Top10 8 -- The United States men #39;s basketball team capped off a big day for the USA by fighting off Greece for a vital win, 77-71. quot;They played with heart, quot; said Coach Larry Brown. quot;That #39;s all you can ask. quot; ...
Top10 9 AP - Manny Ramirez and David Ortiz homered on consecutive pitches to start the eighth inning Sunday night and the streaking Boston Red Sox beat the Chicago White Sox 6-5 for their sixth straight win.
Top10 10 ATHENS (Reuters) - Hungarian Olympic discus champion Robert Fazekas will lose his gold medal and be expelled from the Games after breaking doping rules, the International Olympic Committee (IOC) said Tuesday.
3
Top10 1 NEW YORK (Reuters) - Retailer Kmart Holdings Corp. <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=KMRT.O target=/stocks/quickinfo/fullquote">KMRT.O</A> on Monday said it finalized a deal to sell 18 of its stores to Home Depot Inc. <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=HD.N target=/stocks/quickinfo/fullquote">HD.N</A> for \$271 million.
Top10 2 BRUSSELS/SAO PAULO (Reuters) - Shareholders gave their blessing on Friday for Belgium's Interbrew <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=INTB.BR target=/stocks/quickinfo/fullquote">INTB.BR</A> to buy Brazil's AmBev <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=AMBV4.SA target=/stocks/quickinfo/fullquote">AMBV4.SA</A><ABV.N> in a \$9.7 billion deal that will create the world's largest brewer.
Top10 3 <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=RSE.N target=/stocks/quickinfo/fullquote">RSE.N</A> jumped before the bell after General Growth Properties Inc. <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=GGP.N target=/stocks/quickinfo/fullquote">GGP.N</A>, the No. 2 U.S. shopping mall owner, on Friday said it would buy Rouse for \$7.2 billion.
Top10 4 NEW YORK (Reuters) - BlackRock Inc. <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=BLK.N target=/stocks/quickinfo/fullquote">BLK.N</A>, one of the largest U.S. fixed income managers, on Thursday said it will buy its far smaller competitor State Street Research Management Co., marking the biggest takeover in the asset management business this year.
Top10 5 NEW YORK (Reuters) - Staples Inc. <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=SPLS.O target=/stocks/quickinfo/fullquote">SPLS.O</A>, the top U.S. office products retailer, on Tuesday reported a 39 percent jump in quarterly profit, raised its full-year forecast and said it plans to enter the fast-growing Chinese market, sending its shares higher.
Top10 6 NEW YORK (Reuters) - Colgate-Palmolive Co. <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=CL.N target=/stocks/quickinfo/fullquote">CL.N</A> will cut about 4,400 jobs, or 12 percent of its work force, and close nearly a third of its factories under a restructuring, the consumer products company said on Tuesday.
Top10 7 CHICAGO (Reuters) - Medtronic Inc. <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=MDT.N target=/stocks/quickinfo/fullquote">MDT.N</A> on Wednesday said its quarterly earnings rose amid brisk demand for devices that manage irregular heart beats and products used to treat the spine.
Top10 8 CHICAGO (Reuters) - Medtronic Inc. <A HREF="http://www.investor.reuters.com/FullQuote.aspx?ticker=MDT.N target=/stocks/quickinfo/fullquote">MDT.N</A> on Wednesday said its quarterly earnings rose on brisk demand for devices that manage irregular heart beats and products used to treat the spine.
Top10 9 NEW YORK (Reuters) - Verizon Communications Inc. <A HREF="http://www.in

vestor.reuters.com/FullQuote.aspx?ticker=VZ.N target=/stocks/quickinfo/fullquote">VZ.N</A> is near an agreement to sell its Canadian telephone directory business to private equity firm Bain Capital, the New York Post said on Wednesday. Top10 10 NEW YORK, August 26 (New Ratings) - BlackRock Inc (BLK.NYS), a leading US-based fixed-income asset management company, has reportedly agreed to buy State Street Research amp; Management Company, a unit of MetLife Inc, for \$375 million in a cash and stock

Full Results

Add here your results:

LR	Precision	Recall	F1-Score
BOW-count	0.8545592790763165	0.8522222222222222	0.8514022251643323
BOW-tfidf	0.881547680555811	0.8799999999999999	0.8793734034496704