

K3G Music Report:

Using Machine Learning to classify songs as



HAPPY

OR

SAD



Team Member:

Jixiao Yang / Cuicui Feng / Miner Yang / Yuning Song

Using Machine Learning to classify songs as **Happy / Sad**

MAIN SOURCE:

K3G Music enterprises is planning a new music station which would enable listeners to listen to music based on how they feel. They plan to use Machine learning to classify Chinese songs.

Using Machine Learning to classify songs as **Happy / Sad**

Requirements:

- ★ Translate Chinese lyrics to English
 - ★ Classify English songs as Happy or Sad
 - ★ Build a APP and host it on platform
-

Using Machine Learning to classify songs as Happy / Sad

Project Overview:

- ★ Using Tone Analyzer to classify songs
 - ★ Using Google Translate API to translate lyrics
 - ★ Create Web APIs with FLASK
-

Using Tone Analyzer to classify songs

What is Tone Analyzer:

- ★ Tone Analyzer service uses linguistic analysis to detect joy, fear, sadness, anger, analytical, confident and tentative tones found in text. You can use the service to understand how your written communications are perceived and then to improve the tone of your communications.
-

Using Tone Analyzer to classify songs

What is Tone Analyzer:

Sample use cases

Choose an example to learn how you can adjust the tone of your content to change people's perceptions, or improve its effectiveness.
[Learn more.](#)

☐ Tweets ☐ Online Review ☐ Email message ☐ Product Review in French ☒ Your own text

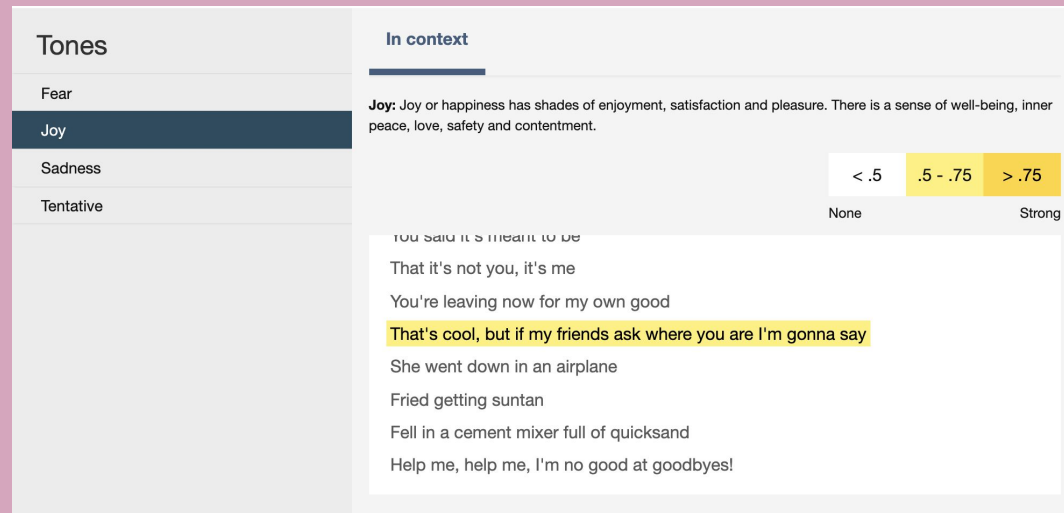
Analyzing Customer Engagement Data? Try out the [Tone Analyzer Customer Engagement Endpoint](#).

Choose Language: ☒ English ☐ French

Analyze

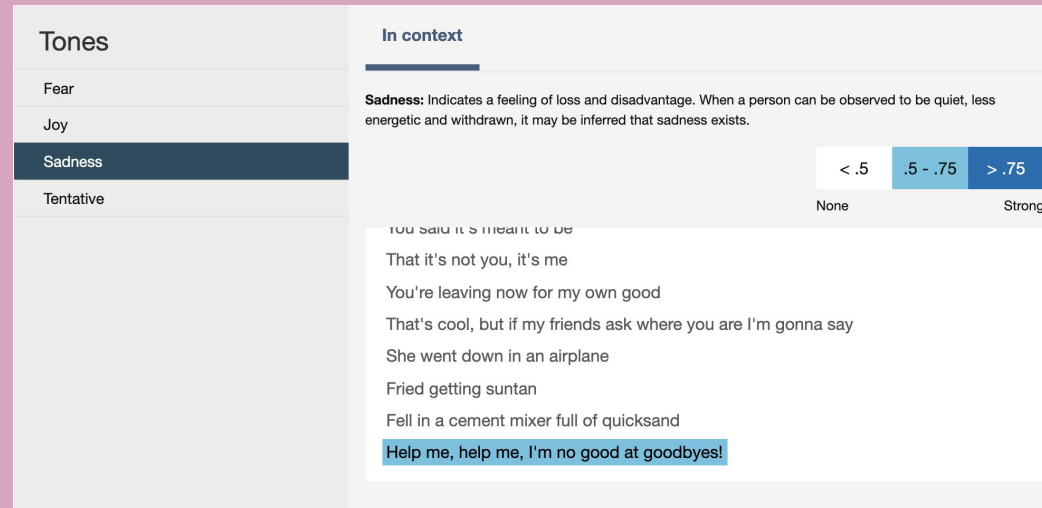
Using Tone Analyzer to classify songs

What is Tone Analyzer:



Using Tone Analyzer to classify songs

What is Tone Analyzer:



Using Tone Analyzer to classify songs

How to use Tone Analyzer:

- ★ The Python installation source as follows:

```
pip install --upgrade "watson-developer-cloud>=2.8.0"
```

- ★ The Python basic authentication source as follows:

```
from watson_developer_cloud import ToneAnalyzerV3
tone_analyzer = ToneAnalyzerV3(
    version='{version}',
    username='{username}',
    password='{password}',
    url='{url}'
)
```

Using Tone Analyzer to classify songs

Tone Analyzer 's Data Handling:

1) Additional headers

Some Watson services accept special parameters in headers that are passed with the request. You can pass request header parameters in all requests or in a single request to the service.

2) Response details

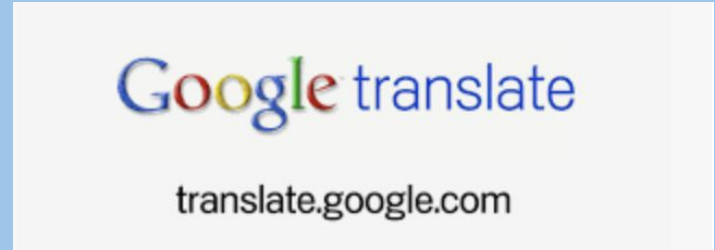
The Tone Analyzer service might return information to the application in response headers.

3) Data collection

Request logging is disabled for the Tone Analyzer service.

Using Google Translate API to translate lyrics

What is Google Translate:



- ★ The Cloud Translation API dynamically translates between thousands of language pairs. With the Cloud Translation API, we can programmatically integrate websites and programs with translation services.
-

Using Google Translate API to translate lyrics

How to use Google Translate:

The Python source for install client library as follows:

```
pip install --upgrade google-cloud-translate
```

The Python source for run the first example as follows:

```
# Imports the Google Cloud client library
```

```
from google.cloud import translate
```

```
# Instantiates a client
```

```
translate_client = translate.Client()
```

```
# The text to translate
```

```
text = u'Hello, world!'
```

```
# The target language
```

```
target = 'ru'
```

```
# Translates some text into Russian
```

```
translation =
```

```
translate_client.translate(
```

```
    text,
```

```
    target language=target)
```

```
print(u'Text: {}'.format(text))
```

```
print(u'Translation:{}'.format(
```

```
translation['translatedText']))
```

Create Web APIs with FLASK



Basic Knowledge of Web APIs:

- ★ What is an API?
 - ★ When to Create an API?
 - ★ What is JSON?
 - ★ What is REST?
 - ★ What is FLASK?
-

Create Web APIs with FLASK



★ What is an API?

A web API allows for information or functionality to be manipulated by other programs via the internet. For example, with Twitter's web API, you can write a program in a language like Python or Javascript that can perform tasks such as favoriting tweets or collecting tweet metadata.

Create Web APIs with FLASK

★ When to Create an API?

In general, consider an API if:

1. Your data set is large, making download via FTP unwieldy or resource-intensive.
 2. Your users will need to access your data in real time, such as for display on another website or as part of an application.
 3. Your data changes or is updated frequently.
 4. Your users only need access to a part of the data at any one time.
 5. Your users will need to perform actions other than retrieve data, such as contributing, updating, or deleting data.
-

Create Web APIs with FLASK



★ What is JSON?

JavaScript Object Notation is a text-based data storage format that is designed to be easy to read for both humans and machines. JSON is generally the most common format for returning data through an API, XML being the second most common.

Create Web APIs with FLASK

★ What is REST?

REpresentational State Transfer is a philosophy that describes some best practices for implementing APIs. APIs designed with some or all of these principles in mind are called REST APIs. While the API outlined in this lesson uses some REST principles, there is a great deal of disagreement around this term. For this reason, I do not describe the example APIs here as REST APIs, but instead as web or HTTP APIs.

Create Web APIs with FLASK



★ What is FLASK?

Flask is a web framework for Python, meaning that it provides functionality for building web applications, including managing HTTP requests and rendering templates. In this section, we will create a basic Flask application. In later sections, we'll add to this application to create our API.

Host Website on Heroku



Welcome to Heroku

Now that your account has been set up, here's how to get started.

Hide next steps



Create a new app

Create another app and deploy your code to a running dyno.

Create new app



Create a team

Create teams to collaborate on your apps and pipelines.

Create a team

Host Website on Heroku

<https://chinese-song-tone-analyzer.herokuapp.com>

Chinese Song Tone Analysis

Analyze Lyrics

Help

Input your lyrics:

Submit

Reset

Host Website on Heroku

<https://chinese-song-tone-analyzer.herokuapp.com>

Lyric Tones

Tone Name	Score
Sadness	0.702883
Tentative	0.544371
Analytical	0.624611

Sentence Tones

1. It was a sigh that was gently sighed at sunset.

Tone Name	Score
Sadness	0.857056

Confusion Matrix of Tone Analyzer API

Overview

Because we do not need to train our model, we can directly use a part of the raw data as the training dataset. We do not need to split that set to training set or validation set. Our training data is based on the Chinese lyrics scraping from the website [10].

We cannot make the raw emotional tag for the dataset. So, we choose a small scaled dataset to add the tag manually. As for the testing data set, I just choose the dataset posted on the gituhub:

https://github.com/rasbt/musicmood/blob/master/dataset/validation/valid_lyrics_200.csv

Confusion Matrix

Confusion Matrix of our data sets:

52	26
24	51

Accuracy Score: 0.673202614379085

Other metric:

	precision	recall	F1-score	support
happy	0.68	0.68	0.67	78
sad	0.66	0.68	0.67	75
avg/total	0.67	0.67	0.67	153

Confusion Matrix of test data

- According to the above matrix, our accuracy is 0.67. Average F1 score is 0.67. The total valid data is 153 rows but not 200 rows. That's because our API will come out with a multi-emotional result that contains not only happy and sad, but also "fear", "confidence", "analytical" and "tentative". It is hard to just classify the result into happy and sad. I just choose useful results for the processes.
- The results of the both datasets show that the training process is better than the testing process. Because the accuracy of training process is 0.7. But the testing accuracy is around 0.67. But I think this difference is within a reasonable range.

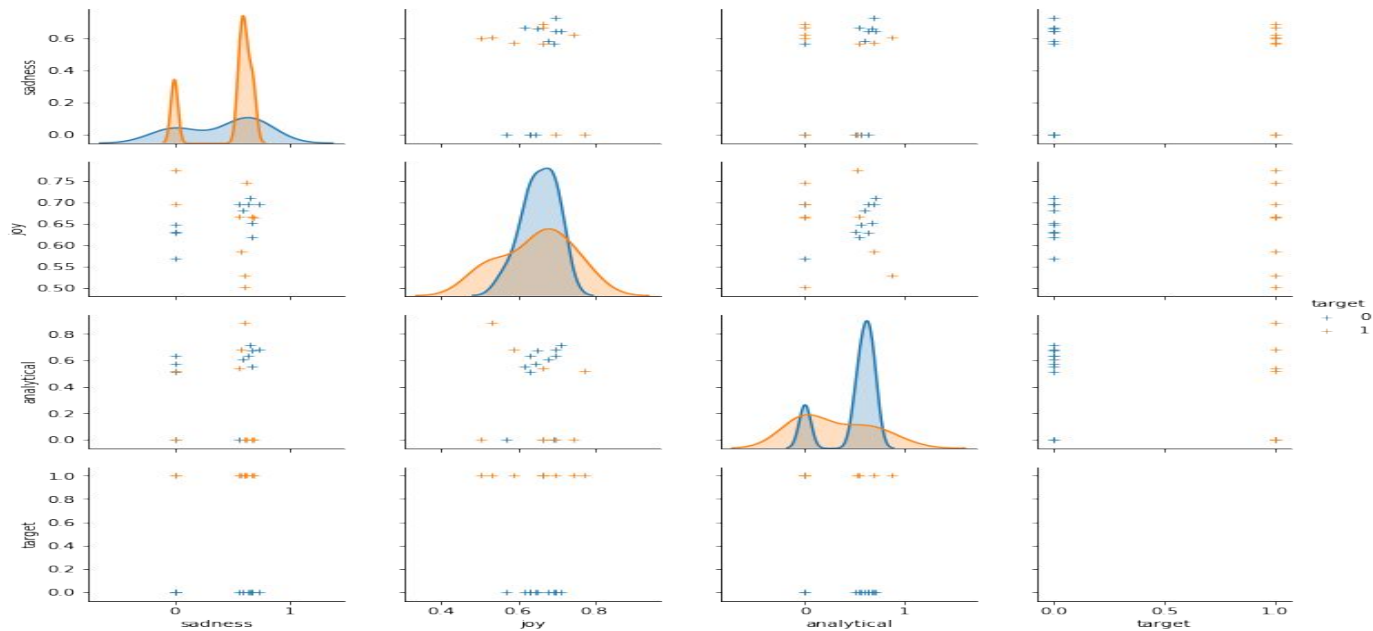
Confusion Matrix of test data

➤ The F-1 Score of training process is a little bit higher than the testing process.

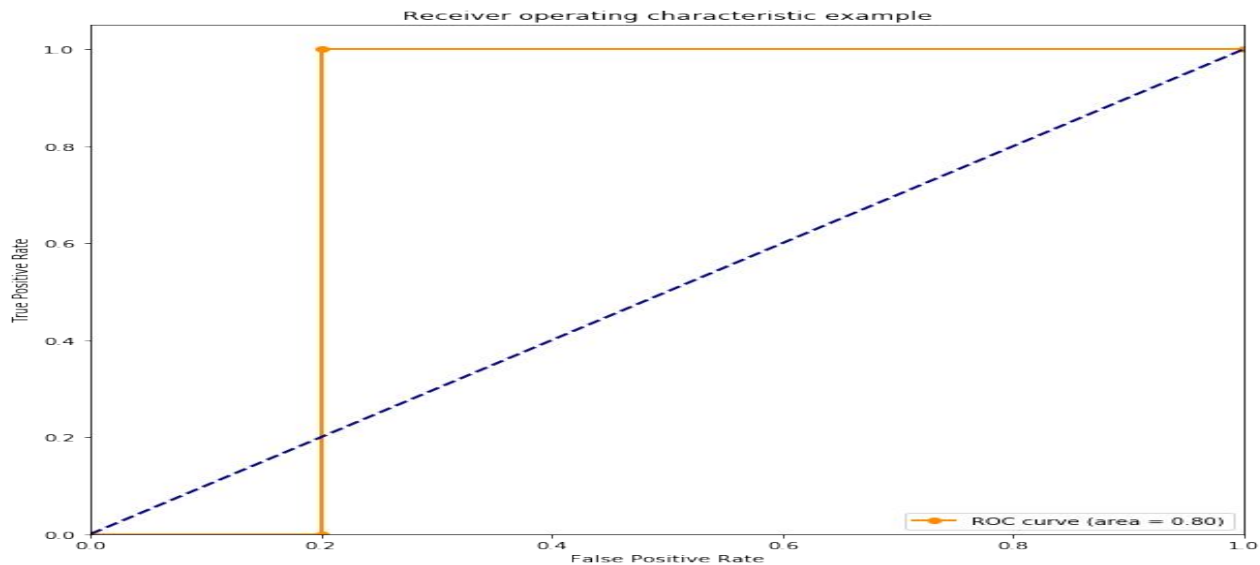
Firstly, I expect the testing score will be higher than the training score. Because the API was built based on English. In other words, we will obviously get a higher score if we use English song to do the prediction. But the truth is that our lyrics in training set has been put through out a translate API. The emotion and the tone of the lyrics may change because of that process. But our training data is so small. I think if the dataset is big enough, the above influence is more visible.

Secondly, it is really confused that to classify the 6-parameter output to a single happy-sad output. There might be some error classification and some misleading of tags. Finally, I think not only the F1 score but also the performance of the both processes are pretty similar.

Simple Analysis



Simple Analysis



Confusion Matrix of test data

Conclusion

The tone analyzer API can classify the input text to different emotion like joy, fear, sadness, anger, analytical, confident and tentative. The performance of the API is obviously higher than the result of our experiment. We just classify the emotion to sad and happy, although the emotion like analytical is a really tricky part and hard to define. In other words, the API is a really precise and effective process while testing. If we see each score of their prediction inside the results.

We can see if there is a prediction, the score must be higher than 85%. If we set the method into a status that is more suitable for the method, I think it will output a high accuracy and F1 score about the predictions.
