



A case written by Sri Krishnamurthy, QuantUniversity

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 songs. Acknowledging there are new songs added to their catalog on a daily basis, they intend to build a machine learning based classifier service which would classify songs as (Happy/Sad).

K3G has three goals in mind. They want to prototype an app which would dynamically generate whether a song is happy/sad from a top-k list in the US. They intend to use Musicmatch [4] to prototype this. They also intend to use Musicmatch's API to get the top-K list of other countries (UK, Canada and Australia) and generate moods for the top lists there. Since the US market is crowded, K3G also is interested in releasing their service in India and China where they could potentially address a 3 Billion target market! In order to test feasibility, they would want to be able to build a classifier that would classify "any" Hindi/Chinese song as Happy/Sad. In addition, K3G researched and found 3 approaches to building models. A google tutorial leveraging Tensorflow and Keras[1], A research project for mood analysis[2], Using tone apis [8].

They have reached out to twelve consulting teams for each goal/model combination and they will review all submissions before selecting the goals and models for their final application. See the team allocations and the requirements below.

# PREPARATION

1. Work through tutorials [1, 2]. You will be using components from these in this exercise. Focus on the model part based on your team's allocation. If you are tasked with using tone analysis APIs for your models review [8]. Your team allocations are at the end of this document.
2. Review APIs [5, 6] for Google Translate and Music Match. You will need to use these when building out your REST APIs and calling services.
3. Review how to build a FLASK app here [7].

# DATA

For the purpose of this exercise, we will build models using these datasets:

- [https://raw.githubusercontent.com/rasbt/musicmood/master/dataset/training/train\\_lyrics\\_1000.csv](https://raw.githubusercontent.com/rasbt/musicmood/master/dataset/training/train_lyrics_1000.csv)
- [https://github.com/rasbt/musicmood/blob/master/dataset/validation/valid\\_lyrics\\_200.csv](https://github.com/rasbt/musicmood/blob/master/dataset/validation/valid_lyrics_200.csv)

# GOALS

## Goal 1: Mood classifier for a top-k list [Teams 1,2,3]

1. Retrieve the top k list from [4] and present them with a happy/sad icon
2. Use Flask to deploy the frontend.
3. Use the modeling approach allocated to you to compute the scores

## Goal 2: Mood classifier for country specific songs [Teams 4,5,7]

1. Have a dropdown for country (UK, Canada, Australia) and retrieve the top k list from the country [4] and present them with a happy/sad icon. See the MusicMatch api for details
2. Use Flask to deploy the frontend.
3. Use the modeling approach allocated to you to compute the scores

## Goal 3: Chinese song mood classifier [Teams 6, 8, 12]

有时快乐有时会伤心 music Co. will be launched in China that will take any Chinese songs in Simplified Chinese as input and will return whether the song was happy/sad.

1. Have a text box that can take Chinese songs as an input. See [10] on where to get Chinese songs.
2. On the backend, use Google translate API [5] to translate this to English.
3. Use the modeling approach allocated to you to compute the scores

## Goal 4: Hindi song mood classifier [Teams 9,10,11]

**Kabhi Kushi Kabhi Gham** music Co. will be launched in India that will take any Hindi songs as input and will return whether the song was happy/sad.

1. Have a text box that can take Hindi songs as an input. See [9] on where to get Hindi songs in Nagari format
2. On the backend, use Google translate API [5] to translate this to English.
3. Use the modeling approach allocated to you to compute the scores

## MODELING APPROACH

1. You are free to design the REST APIs to be invoked for the model backend. See the Flask tutorial [7] for example.
2. The model you invoke will be based on team allocations.
3. Use the data provided in the Data section for training your model if you are using approaches [1] or [2]
4. If you are allocated tone analyzer APIs, you won't have to build a model, but you will get tone Joy/sadness from the apis

## METRICS

- You will be expected to compute the confusion matrix for both training and testing data.
- You will also be computing the F1-score
- **Discuss how good your model is.**

## REPORT

As a consulting company, put together a CLAAT document discussing your project and how to use it.

## DELIVERABLES

1. Github link with fully functional code with documentation on how to use it.
2. A final report in CLAAT format
3. Host your website on Heroku or any other platform.
4. A 5 min video discussing your project loaded to Youtube. Embed the link in your CLAAT report.

## REFERENCE

1. <https://developers.google.com/machine-learning/guides/text-classification/>
2. <https://github.com/rasbt/musicmood>
3. <https://www.kaggle.com/theriley106/panic-at-the-dataset/activity>
4. <https://www.musixmatch.com/explore>
5. <https://cloud.google.com/translate/docs/>
6. <https://developer.musixmatch.com>
7. <https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask>
8. <https://tone-analyzer-demo.ng.bluemix.net/>
9. <http://www.giitaayan.com/>
10. <http://singklyrics.com/>

## MODELS & DATA TO USE:

Goal/Method	Models As described in [1]	Models As described in [2]	Tone Analysis APIs as described in [8]
1	1	2	3
2	4	5	7
3	6	8	12
4	9	10	11

Note: Numbers highlighted in Yellow are team number allocations