K3G Music Enterprises

Authors

1) Bhavya Haridas

2) Harshitha S Gaddadhar

3) Saumil Shah

4) Nikita Gawde

Introduction and Goal

- K3G Music enterprises is planning a new music station which would enable listeners to listen to music based on how they feel.

- As one of their consulting teams our goal is to build a machine learning based classifier service which would classify songs as (Happy/Sad) based on the country selected to list the songs.

- We have used Musixmatch’s API to get the top-K list of other countries (UK, Canada and Australia) and generate moods for the top lists there.

Data Collection and Preprocessing

- For our model we have obtained our training and testing data set from
    1) https://raw.githubusercontent.com/rasbt/musicmood/master/dataset/training/train_lyrics_1000.csv

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

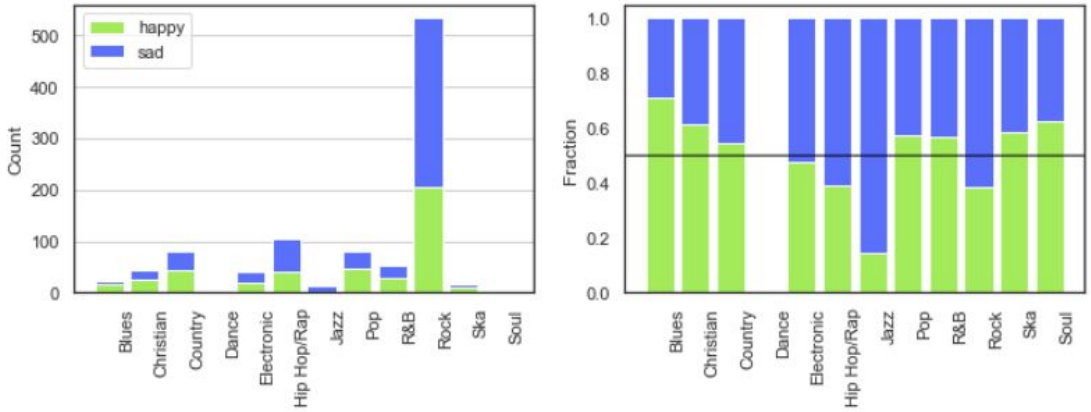
- Each song has a mood label assigned stating if the song is a happy or sad.

- The lyrics was converted to feature vectors. Further Python NLTK library was used to remove the non English words and Porter Stemmer algorithm was implemented for suffix removal.

- Different version of word count in the feature vectors for each song was implemented like Tf-IDF , binarization.

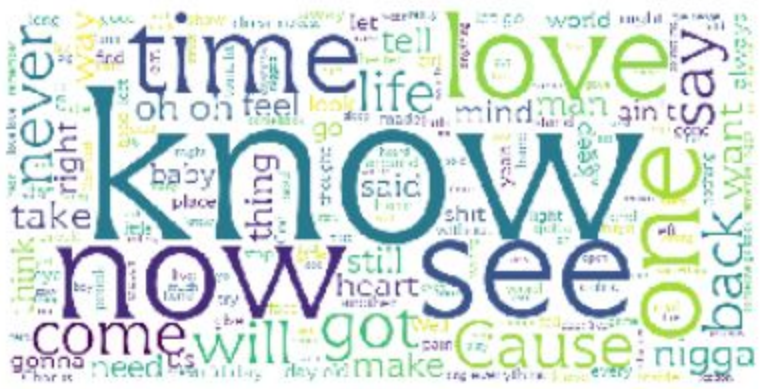
Genre v/s Mood

mood	happy	sad
genre		
Blues	17.0	7.0
Christian	27.0	17.0
Country	43.0	36.0
Dance	NaN	1.0
Electronic	19.0	21.0
Hip Hop/Rap	41.0	64.0
Jazz	2.0	12.0
Pop	47.0	35.0
R&B	30.0	23.0
Rock	205.0	328.0
Ska	10.0	7.0
Soul	5.0	3.0



Most Common happy words

Most Common Sad Words



ML Modelling

- We implemented Naive Bayes and K- nearest neighbours algorithm and noticed that the precision percentage is higher for Naive Bayes model.

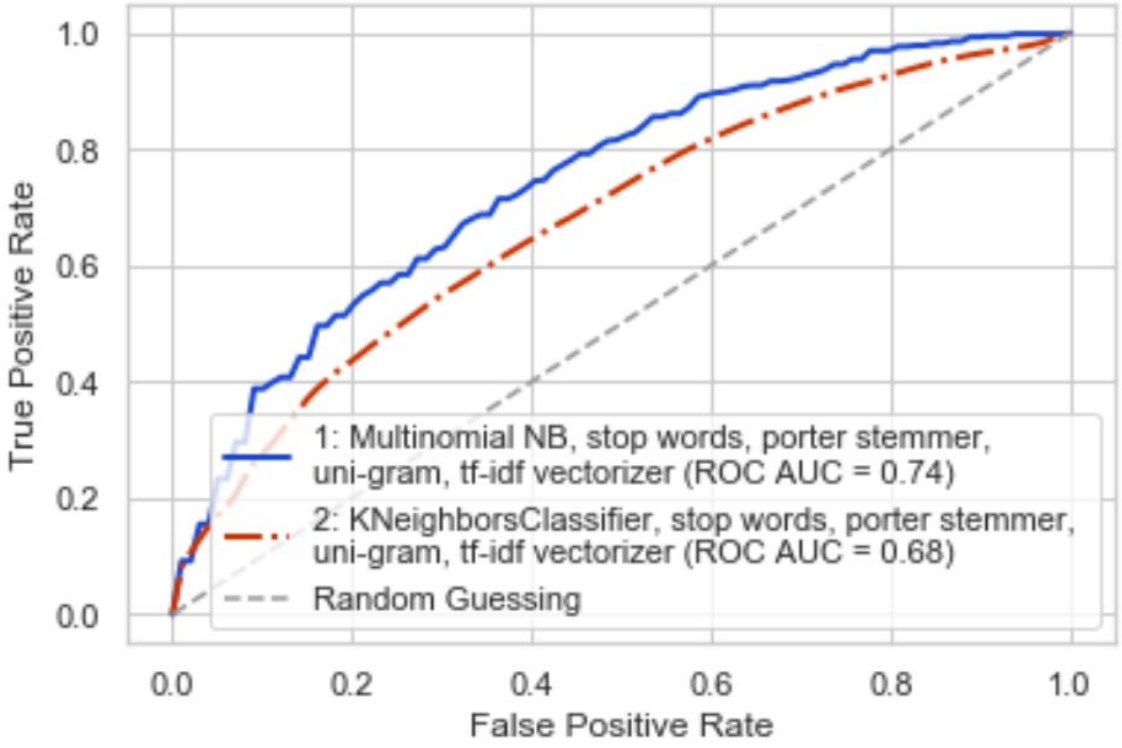
  For Naive Bayes Algorithm

	ACC (%)	PRE (%)	REC (%)	F1 (%)	ROC AUC (%)
Training	80.8	99.61	57.17	72.65	78.50
Validation	55.0	85.71	17.14	28.57	56.99

  For KNN Algorithm

	ACC (%)	PRE (%)	REC (%)	F1 (%)	ROC AUC (%)
Training	77.5	66.47	100.00	79.86	79.69
Validation	61.0	59.06	83.81	69.29	59.80

- The Naive Bayes model proves to give better prediction when compare to KNN model.
- Also from ROC auc curves we can see that the curve for Naive Bayes is much closer to the True predicted Rate.

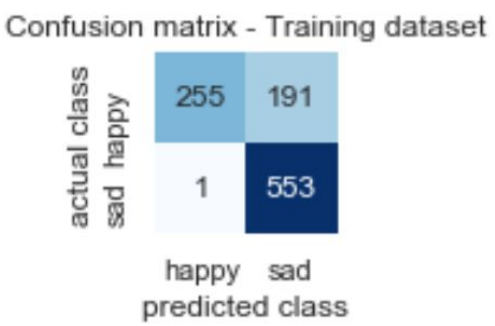


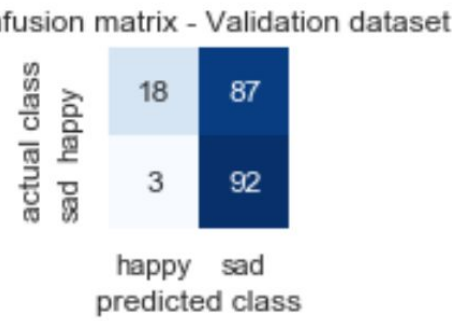
- Hence we choose to model our application using Naive Bayes.

- Three models of Naive Bayes were implemented : Multivariate Bernoulli Bayes with binary word counts as feature vectors, Multinomial Bayes with term frequency features, and Multinomial naive Bayes with tf-idf features .

- These 3 models were individually optimized via Grid Search and the ROC auc was evaluated for each and the best performing model was chosen for further hyper parameter tuning.

- The confusion matrix for the training and validation dataset as well the F1 score is obtained as



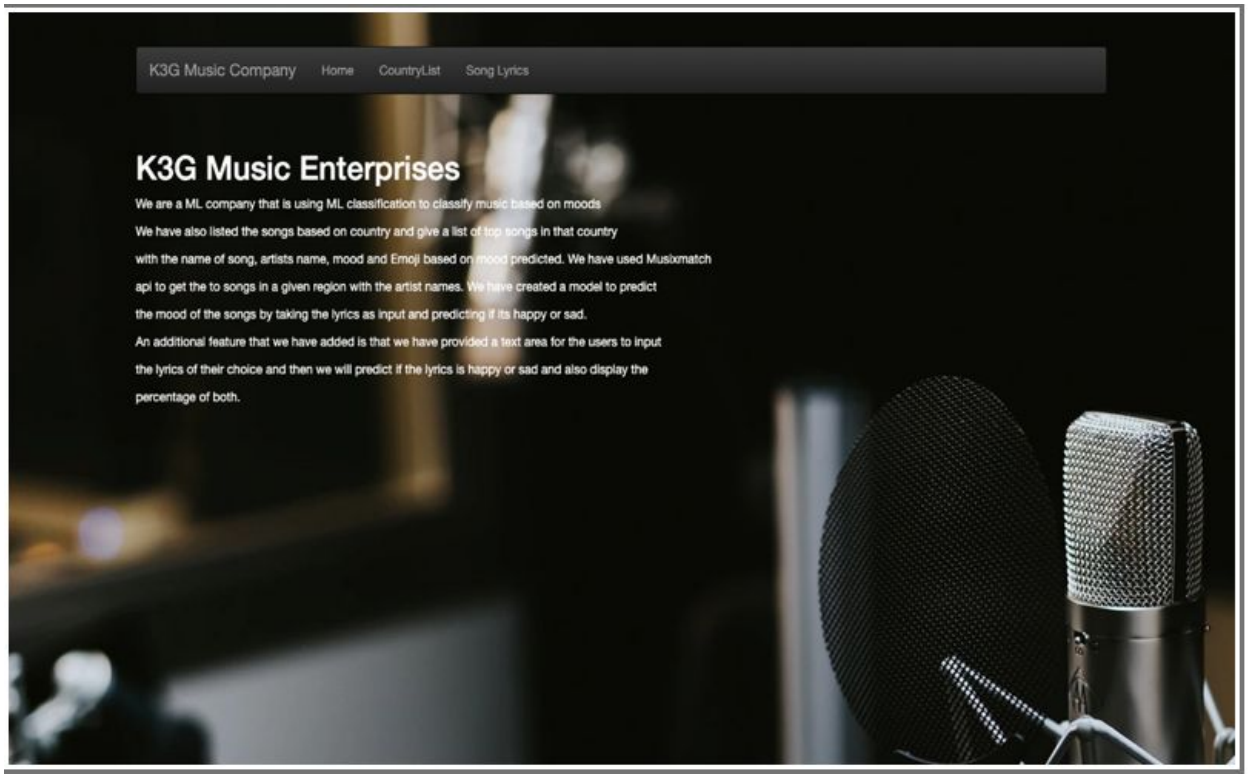


	ACC (%)	PRE (%)	REC (%)	F1 (%)	ROC AUC (%)
Training	80.8	99.61	57.17	72.65	78.50
Validation	55.0	85.71	17.14	28.57	56.99

Web-App

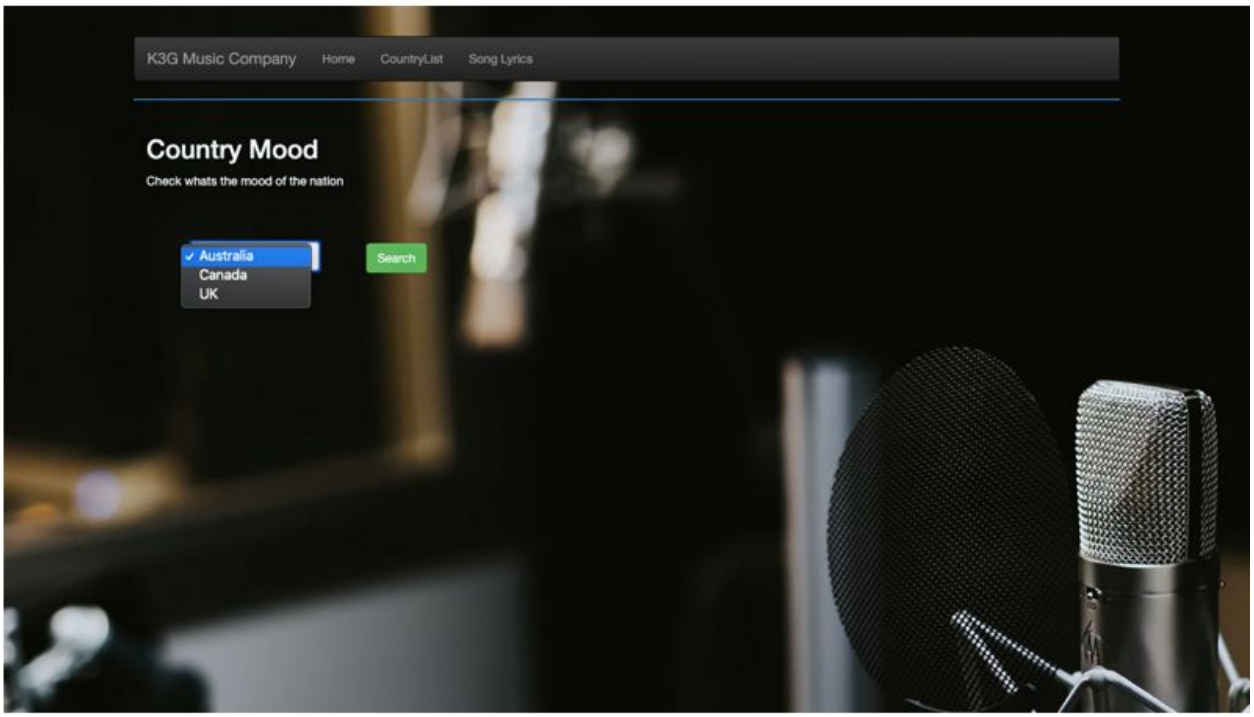
- Url - https://k3gmusic-team5.herokuapp.com/
- The top 10 music lyrics is retrieved from musixmatch API and run against the model.

- The Web application was designed in flask framework. The function **render_template** embeds the flask application to html pages which is stored in the templates folder and can be designed as per our requirements using html and bootstrap.

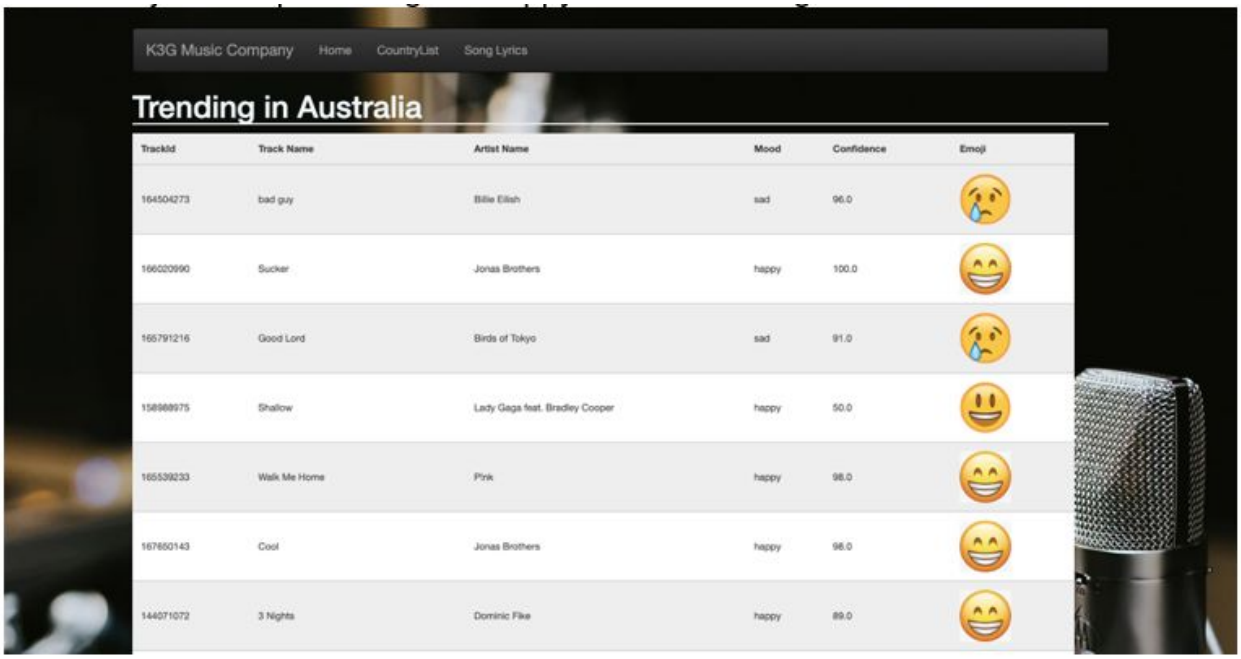


- The flask application is saved as .py files and executed on the command line and seen on local host server.

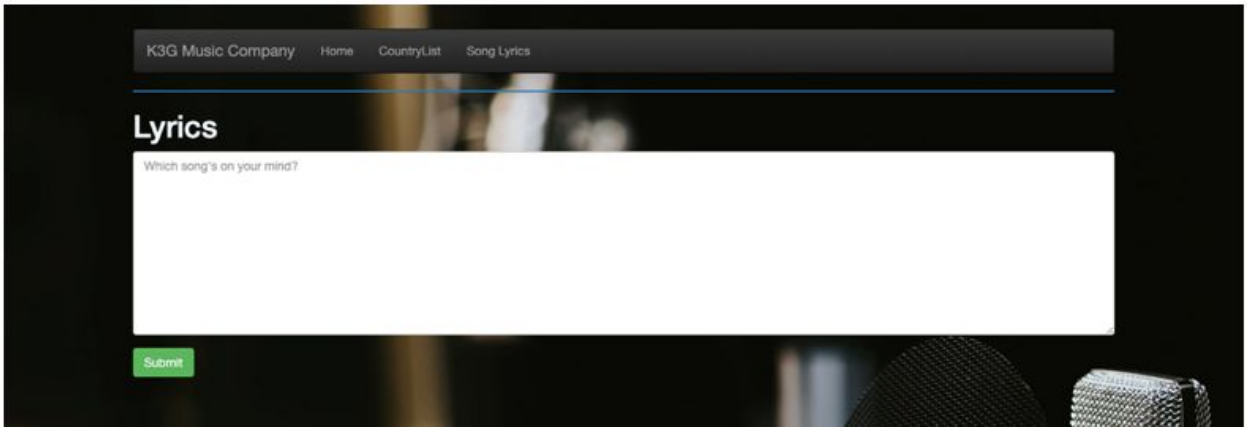
- @app.route(‘/’) function calls tells the flask application which URL to use. For our application we have the URLs from MusicMatch.
- The WTForms package helps us to validate the reusable forms which takes only one input from the users i.e. the country’s name.



- The function call **make_predictions** takes the input from the flask form and generates a json file which contains the trending songs of a country. These are then used to make predictions whether the song is happy or sad and assigns an emoji on the percentage of happy or sad the song is.



- In addition, we've also displayed another page which takes a snippet of lyrics from the users and displays the percentage of happy or sad the lyrics is. This is done using the api call made to MusicMatch.



CONCLUSION

- The analysis done on each model provides insights on the effectiveness of using each in such a scenario.
- Naive Bayes based model was used to build the web app that provides the interface to retrieve top 10 songs and input custom lyrics.
- Youtube Url - https://www.youtube.com/watch?v=rBByOJF6mwY&feature=youtu.be