MINOR FINAL REPORT

MUSIC APPLICATION

Submitted By:

Anant Agarwal (16103033, B3)

Prakhar Saxena (16103003, B3)

Rishabh Dhawan (16103213, B4)

Sonu Kumar Gupta (16103271, B4)

Submitted To:

Dr. Archana Purwar

Abstract :

Sentiment prediction of contemporary music can have a wide-range of applications in modern society, for instance, selecting music for public institutions such as hospitals or restaurants to potentially improve the emotional well-being of personnel, patients, and customers, respectively. In this project, music player based on facial emotions using CNN model and recommendation system built upon on a naive Bayes classifier, trained to predict the sentiment of songs based on song lyrics alone. The experimental results show that music corresponding to a happy mood (predicted on live camera capture) can be detected with high precision based on text features obtained from song lyrics followed by playing the same on Youtube.

Introduction :

With the rapid growth of digital music libraries as well as advancements in technology, music classification and recommendation has gained increased popularity in the music industry and among listeners. Many applications using machine learning algorithms have been developed to categorize music by instruments artist similarity emotion or genre

Psychological studies have shown that listening to music is one of the most popular activities in leisure time and that it has an enhancing effect on the social cohesion, emotional state, and mood of the listeners. The increasing number of song lyrics that are freely available on the Internet allow the effective training of machine learning algorithms to perform mood prediction and filtering for music that can be associated with positive or negative emotions.

Novelty of the Project:

For human-computer interaction facial expression makes a platform for non-verbal communication. The emotions are effectively changeable happings that are evoked as a result of impelling force.

In this project we presented the design of an artificially intelligent system capable of emotion recognition through facial expressions.
We further implement this emotion detector to a Music application in the real world which can help in various business and social media prospects. We further predict the song based on the emotion that occurred the most.

Techniques Used :

1.Convolutional Neural Networks

2.Haarcascade Face Detection Technology

3.OpenCV

4.Scikit-learn

5.Numpy

6.NLTK(natural language ToolKit)

7.Pandas

8.Beautiful Soup

9 .Natural Language Processing

10. Shutil

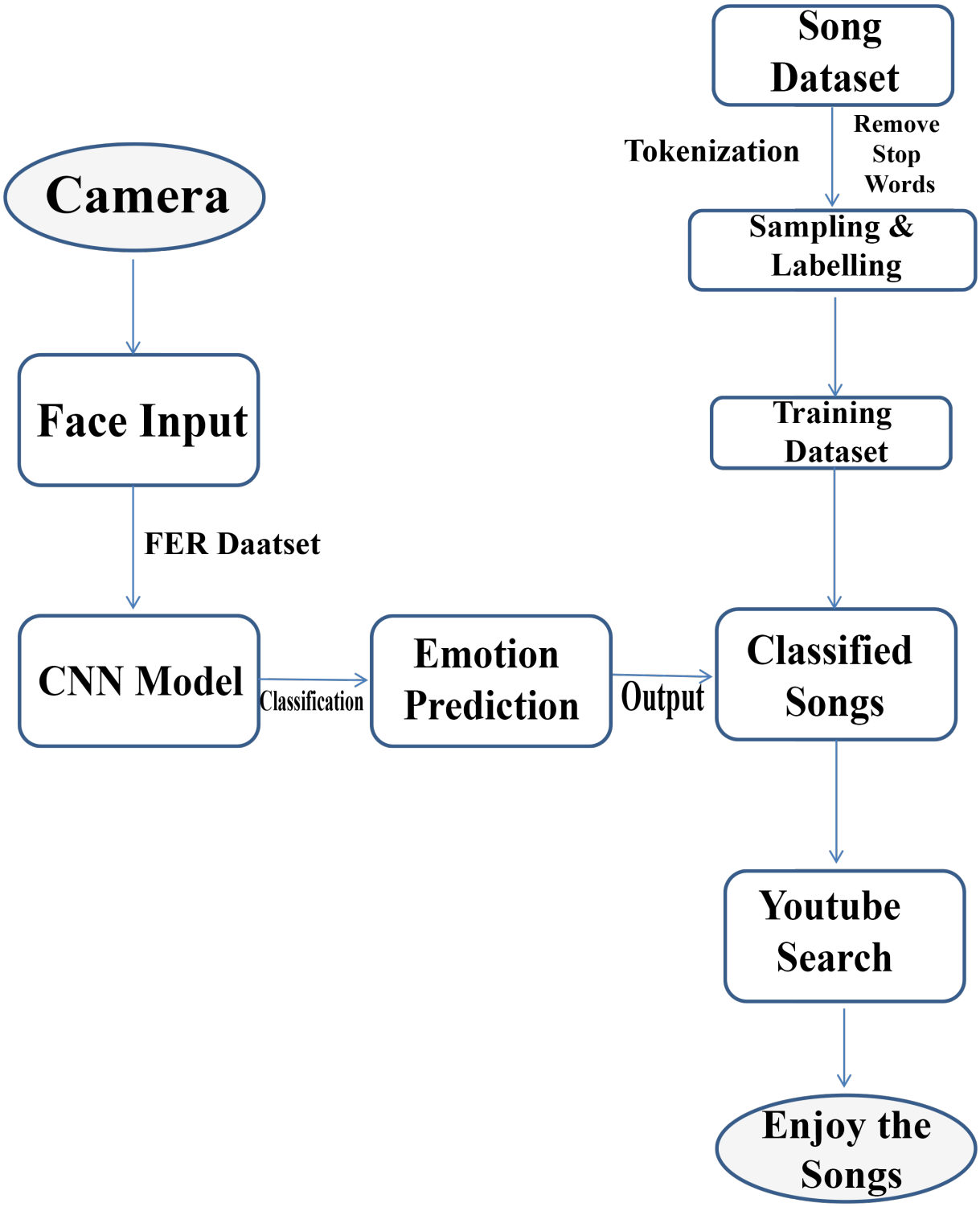
Dataset:

In an attempt to improve the final model even more, the network will be trained on a larger set than the one described previously. Instead of 9000 pictures, training will be done with 20000 pictures from the FER-2013 dataset. Newly composed validation (2000 images) and test sets (1000 images) from the FER-2013 dataset (Kaggle)are used.

The Million Song Dataset is a freely-available collection of audio features and metadata for a million contemporary popular music tracks.
Its purposes are:
To encourage research on algorithms that scale to commercial sizes
To provide a reference dataset for evaluating research
As a shortcut alternative to creating a large dataset with APIs (e.g. The Echo Nest's)
To help new researchers get started in the MIR field

The core of the dataset is the feature analysis and metadata for one million songs, provided by The Echo Nest. The dataset does not include any audio, only the derived features.
The Million Song Dataset is also a cluster of complementary datasets contributed by the community:
SecondHandSongs dataset -> cover songs
musiXmatch dataset -> lyrics
Last.fm dataset -> song-level tags and similarity
Taste Profile subset -> user data
thisismyjam-to-MSD mapping -> more user data
tagtraum genre annotations -> genre labels
Top MAGD dataset -> more genre labels

Flow Diagram



Approaches/Algorithms Used:

Convolutional Neural Networks:

Convolutional Neural Networks are very similar to ordinary Neural Networks: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other.

A simple ConvNet is a sequence of layers, and every layer of a ConvNet transforms one volume of activations to another through a differentiable function. We use three main types of layers to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer.

INPUT [48x48x3] will hold the raw pixel values of the image, in this case an image of width 48, height 48, and with three color channels R,G,B.

CONV layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. This may result in volume such as [48x48x12] if we decided to use 12 filters.

POOL layer will perform a downsampling operation along the spatial dimensions (width, height), resulting in volume such as [16x16x12].

FC (i.e. fully-connected) layer will compute the class scores, resulting in volume of size [1x1x7], where each of the 7 numbers correspond to a class score, such as among the 7 categories of Dataset. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume.

All networks are trained for 60 epochs with the dataset ,For network A, the final accuracy on the validation data is around 63%. Already after 10 epochs, the accuracy raised above 60%, indicating quick learning capabilities. the accuracy seems to still increase in the last epochs. We therefore will train the network for 100 epochs in the final run, to make sure the accuracy converges to the optimum. In an attempt to improve the final model even more, the network will be trained on a larger set than the one described previously. Instead of 9000 pictures, training will

be done with 20000 pictures from the FERC-2013 dataset. Newly composed validation (2000 images) and test sets (1000 images) from the FERC-2013 dataset are used.

OpenCV :

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

Haar Cascading:

We use the Haar Feature-Based Cascaded Classifier inside the OpenCV framework, all data is preprocessed. For every image, only the square part containing the face is taken, rescaled, and converted to an array with 48x48 grey-scale values. This data is then fed to the input of the neural network model, which in its turn returns the values of the output layer. These values represent the likelihood that the each emotion is depicted by the user. The output with the highest value is assumed to be the current emotion of the user, and is depicted by an emoticon on the left of the screen.

Multinomial Naïve Bayes:

Naive Bayes is a family of algorithms based on applying Bayes theorem with a strong(naive) assumption, that every feature is independent of the others, in order to predict the category of a given sample. They are probabilistic classifiers, therefore will calculate the probability of each category using Bayes theorem, and the category with the highest probability will be output. Naive Bayes classifiers have been successfully applied to many domains, particularly Natural Language Processing(NLP).

We use this to classify the comments in the twitter dataset using NLP and its methods such as:

Lemmatizing: This is grouping together of different inflections of the same word.

Removing Stopwords: These are common words that don't really add anything to the classification such as a, able, either, else, ever, etc.

Using n-gram: Instead of counting single words, we could count sequence of words.

Using TF-IDF: Instead of just counting frequence we could do something more advanced like the penalizing words that appear frequently in most of the samples.

NLTK:

We use this platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. It provides a practical introduction to programming for language processing. It is used in python to tokenize, lemmatize, stem, stop words, chunk, chink, text classification, part of speech tagging and name entity recognition to get the words that are mostly used in particular comments to further classify them using lexicon.

Literary Review:

Paper-1

Title	Analysis of Emotion Recognition using Facial Expressions and Speech
Authors	Carlos Busso, Zhigang Deng, Serdar Yildirim, Murtaza Bulut
Year of Publication	2005
Journal/ Publication Detail	Emotion Research Group , Department of Computer Science Viterbi School of Engineering, University of Southern California, Los Angeles
Summary	The interaction between human beings and computers will be more natural if computers are able to perceive and respond to human non-verbal communication such as emotions. Although several approaches have been proposed to recognize human emotions based on facial expressions or speech, relatively limited work has been done to fuse these two, and other, modalities to improve the accuracy and robustness of the emotion recognition system. This paper analyzes the strengths and the limitations of systems based only on facial expressions or acoustic information.This research analyzed the strengths and weaknesses of facial expression classifiers and acoustic emotion classifiers. In these unimodal systems, some pairs of emotions are usually misclassified. However, the results presented in this paper show that most of these confusions could be resolved by the use of another modality. Therefore, the performance of the bimodal emotion classifier was higher than

	each of the unimodal systems.
Web Link	http://www.cvauni.edu.vn/imgupload_dinhkem/file/Chuyen%20De%20HCI/3_%20Analysis%20of%20Emotion%20Recognition%20using%20facial%20expression-speech.pdf

Paper 2

Title	Facial Emotion Detection Using Convolutional Neural Networks
Authors	Prudhvi Raj Dachapally
Year of Publication	2014
Journal/ Publication Detail	School of Informatics and Computing Indiana University
Summary	Emotion being a subjective thing, leveraging knowledge and science behind labeled data and extracting the components that constitute it, has been a challenging problem in the industry for many years. With the evolution of deep learning in computer vision, emotion recognition has become a widely-tackled research problem. In this work, we propose two independent methods for this very task.The method used is an 8-layer convolutional neural network (CNN). These methods were trained on the posed-emotion dataset (JAFFE), and to test their robustness, both the models were also tested on 100 random images from the Labeled Faces in the Wild (LFW) dataset, which consists of images that are candid than posed. The results show that with more fine-tuning and depth, our CNN model can outperform the state-of-the-art methods for emotion recognition.

	We also propose some exciting ideas for expanding the concept of representational autoencoders to improve their performance.
Web Link	https://arxiv.org/ftp/arxiv/papers/1706/1706.01509.pdf

Paper 3

Title	Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction.
Authors	Marian Stewart Bartlett, Gwen Littlewort, Ian Fasel, Javier R. Movellan
Year of Publication	2003
Journal/ Publication Detail	2003 Conference on Computer Vision and Pattern Recognition Workshop.
Summary	This paper presents some first steps towards the development of a system that finds faces in a video stream and codes facial expression dynamics in real time. The system was trained and tested on Cohn and Kanade's DFAT-504 dataset. This dataset consists of 100 university students ranging in age from 18 to 30 years. The system consists of a cascade of classifiers, each of which contains a subset of filters reminiscent of Haar Basis functions, which can be computed very fast at any location and scale in constant time. A combination approach, in which the Gabor Features chosen by Adaboost were used as a reduced representation for training

	SVM's (AdaSVM's) outperformed Adaboost and SVM. The system has been deployed on a wide variety of platforms including Sony's Aibo pet robot, ATR's RoboVie, and CU animator.
Web Link	https://ieeexplore.ieee.org/document/4624313

Paper 4

Title	Automatic Emotion Recognition Using Facial Expression: A Review
Authors	Karaboga, Dervis Basturk, Bahriye
Year of Publication	2016
Journal/ Publication Detail	International Research Journal of Engineering and Technology (IRJET)
Summary	This paper objective is to introduce needs and applications of facial expression recognition. Between Verbal & Non-Verbal form of communication facial expression is form of non-verbal communication but it plays pivotal role. It express human perspective or filling & his or her mental situation. A big research has been addressed to enhance Human Computer Interaction (HCI) over two decades. This paper includes introduction of facial emotion recognition system, Application, comparative study of popular face expression recognition techniques & phases of automatic facial expression recognition system.

	Emotional aspects have huge impact on Social intelligence like communication understanding, decision making and also helps in understanding behavioral aspect of human. Emotion play pivotal role during communication. Emotion recognition is carried out in diverse way, it may be verbal or non-verbal .Voice (Audible) is verbal form of communication & Facial expression, action, body postures and gesture is non-verbal form of communication. [1] While communicating only 7% effect of message is contributes by verbal part as a whole, 38% by vocal part and 55% effect of the speaker's message is contributed by facial expression. For that reason automated & real time facial expression would play important role in human and machine interaction. Facial expression recognition would be useful from human facilities to clinical practices. Analysis of facial expression plays fundamental roles for applications which are based on emotion recognition like Human Computer Interaction (HCI), Social Robot, Animation, Alert System & Pain monitoring for patients.
Web Link	https://www.irjet.net/archives/V3/i2/IRJET-V3I284.pdf

Paper-5

Title	Comparing SVM and Naive Bayes Classifiers for Text Categorization with Wikitology as knowledge Enrichment
Authors	Sundus Hassan Muhammad Rafi

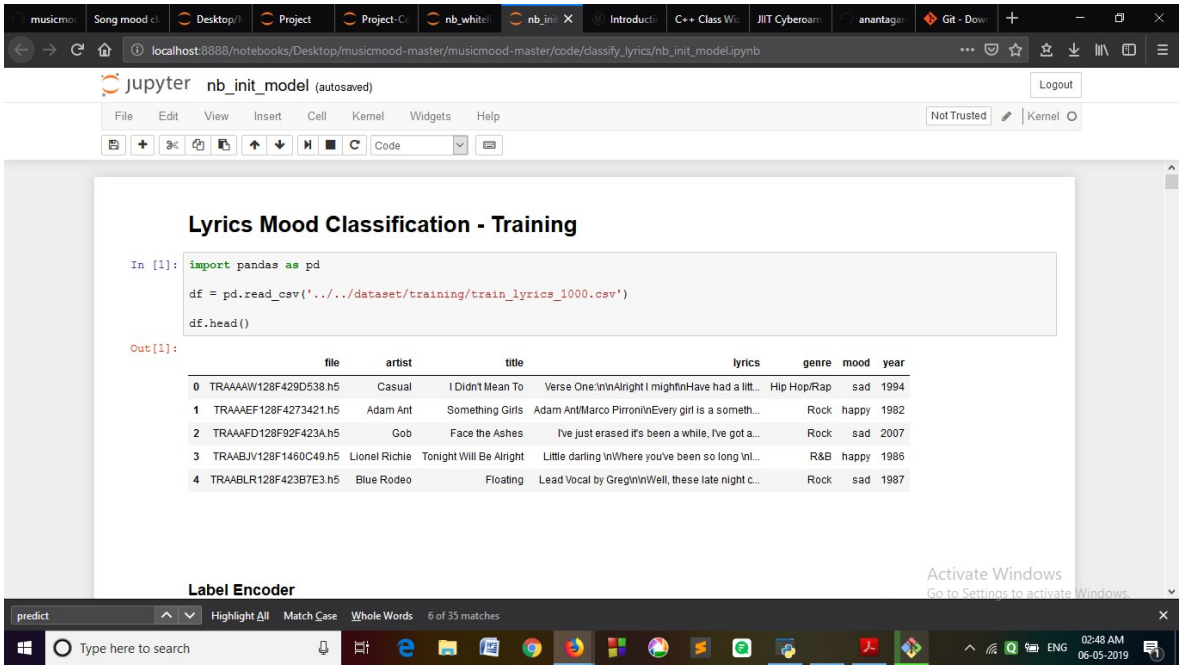
	Muhammad Shahid Shaikh
Year of Publication	2011
Journal/ Publication Detail	Published in IEEE 14th International Multitopic Conference 2011
Summary	The activity of labeling of documents according to their content is known as text categorization. Many experiments have been carried out to enhance text categorization by adding background knowledge to the document using knowledge repositories like Word Net, Open Project Directory (OPD), Wikipedia and Wikitology. In our previous work, we have carried out intensive experiments by extracting knowledge from Wikitology and evaluating the experiment on Support Vector Machine with 10- fold cross-validatious. The results clearly indicate Wikitology is far better than other knowledge bases. In this paper we are comparing Support Vector Machine (SVM) and Naive Bayes (NB) classifiers under text enrichment through Wikitology. We validated results with IO-fold cross validation and shown that NB gives an improvement of +28.78%, on the other hand SVM gives an improvement of +6.36% when compared with baseline results. Naive Bayes classifier is better choice when external enriching is used through any external knowledge base
Web Link	https://www.semanticscholar.org/paper/Comparing-SVM-and-na%C3%AFve-Bayes-classifiers-for-text-Hassan-

	Rafi/501cfa39b2d94443ddbf4bfb7bb82c9845fa789c

Paper-6

Title	Genre Classification for Million Song Dataset Using Confidence-Based Classifiers Combination
Authors	Yajie Hu, Mitsunori Ogihara
Year of Publication	2012
Journal/ Publication Detail	School of Informatics and Computing Indiana University
Summary	We proposed a method to classify songs in the Million Song Dataset according to song genre. Since songs have several data types, we trained sub-classifiers by different types of data. These sub-classifiers are combined using both classifier authority and classification confidence for a particular instance. In the experiments, the combined classifier surpasses all of these sub-classifiers and the SVM classifier using concatenated vectors from all data types. Finally, the genre labels for the Million Song Dataset are provided.
Web Link	https://www.semanticscholar.org/paper/Genre-classification-for-million-song-dataset-using-Ogihara/d12c2f65d3792cd8aa3b9a80046f52fd77744403

Results



jupyter nb_init_model (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Kernel

Code

Lyrics Mood Classification - Training

In [1]:
```
import pandas as pd

df = pd.read_csv('../../dataset/training/train_lyrics_1000.csv')

df.head()
```

Out[1]:

	file	artist	title	lyrics	genre	mood	year
0	TRAAAAW128F429D538.h5	Casual	I Didn't Mean To	Verse One:\n\nAlright I might\nHave had a litt...	Hip Hop/Rap	sad	1994
1	TRAAAEF128F4273421.h5	Adam Ant	Something Girls	Adam Ant/Marco Pirroni\nEvery girl is a someth...	Rock	happy	1982
2	TRAAAFD128F92F423A.h5	Gob	Face the Ashes	I've just erased it's been a while, I've got a...	Rock	sad	2007
3	TRAABJV128F1460C49.h5	Lionel Richie	Tonight Will Be Alright	Little darling \nWhere you've been so long \nI...	R&B	happy	1986
4	TRAABLR128F423B7E3.h5	Blue Rodeo	Floating	Lead Vocal by Greg\n\nWell, these late night c...	Rock	sad	1987

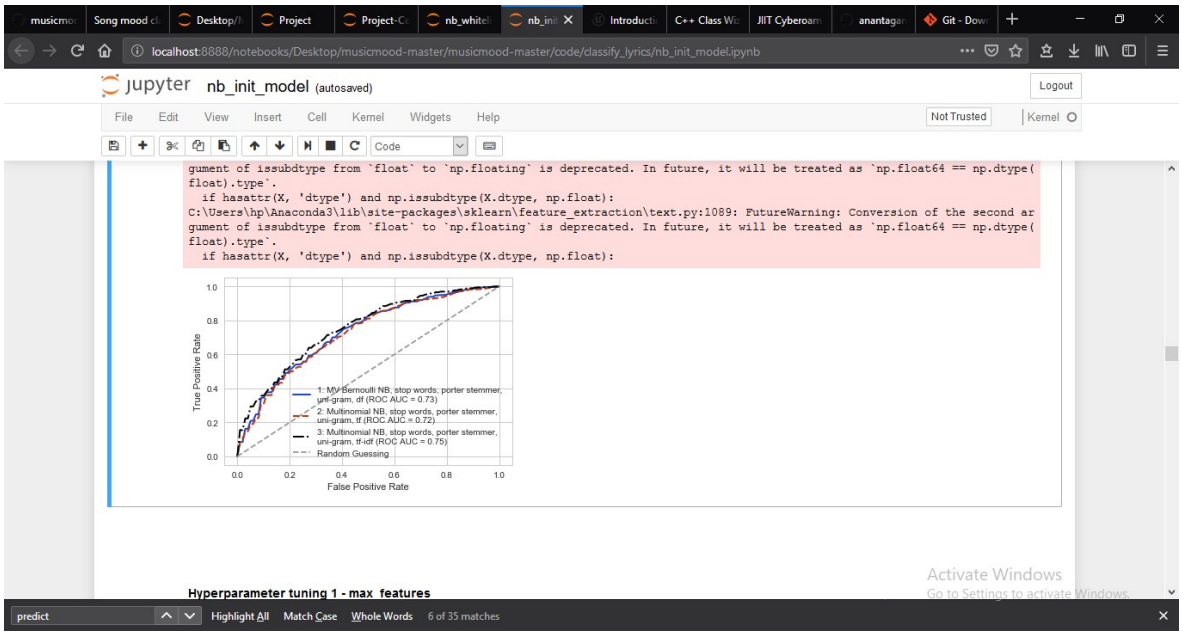
Activate Windows
Go to Settings to activate Windows.

Label Encoder

predict Highlight All Match Case Whole Words 6 of 35 matches

Type here to search

ENG 02:48 AM 06-05-2019



jupyter nb_init_model (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Kernel

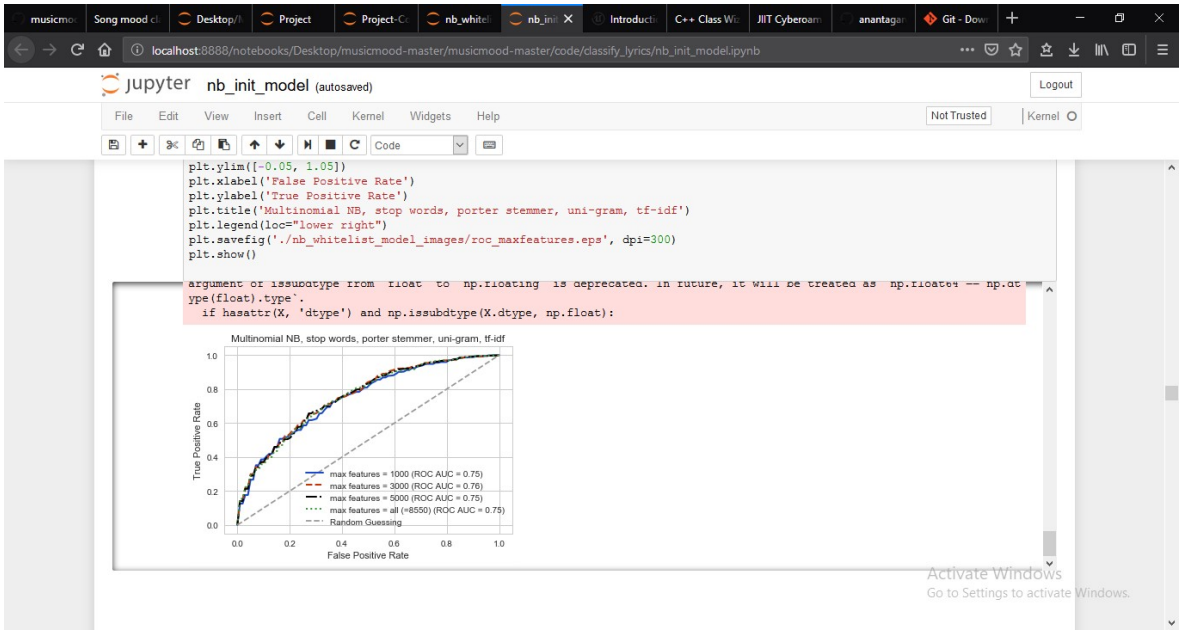
Code

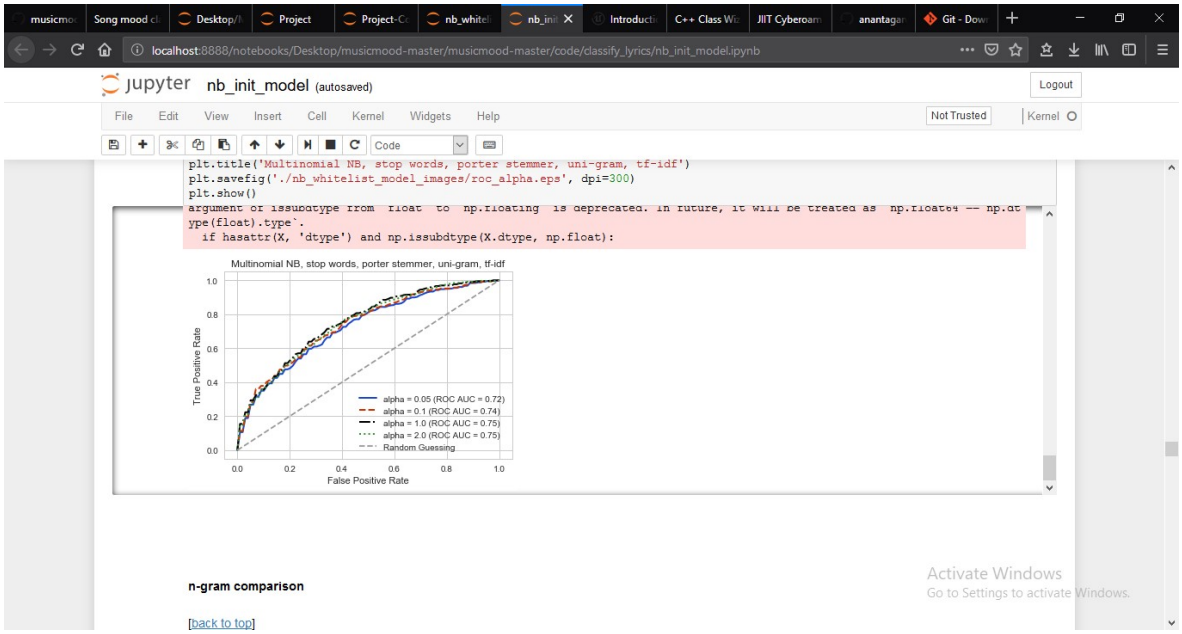
```
gument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(
float).type`.
  if hasattr(X, 'dtype') and np.issubdtype(X.dtype, np.float):
C:\Users\hp\Anaconda3\lib\site-packages\sklearn\feature_extraction\text.py:1089: FutureWarning: Conversion of the second ar
gument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(
float).type`.
  if hasattr(X, 'dtype') and np.issubdtype(X.dtype, np.float):
```


Activate Windows
Go to Settings to activate Windows.

Hyperparameter tuning 1 - max_features

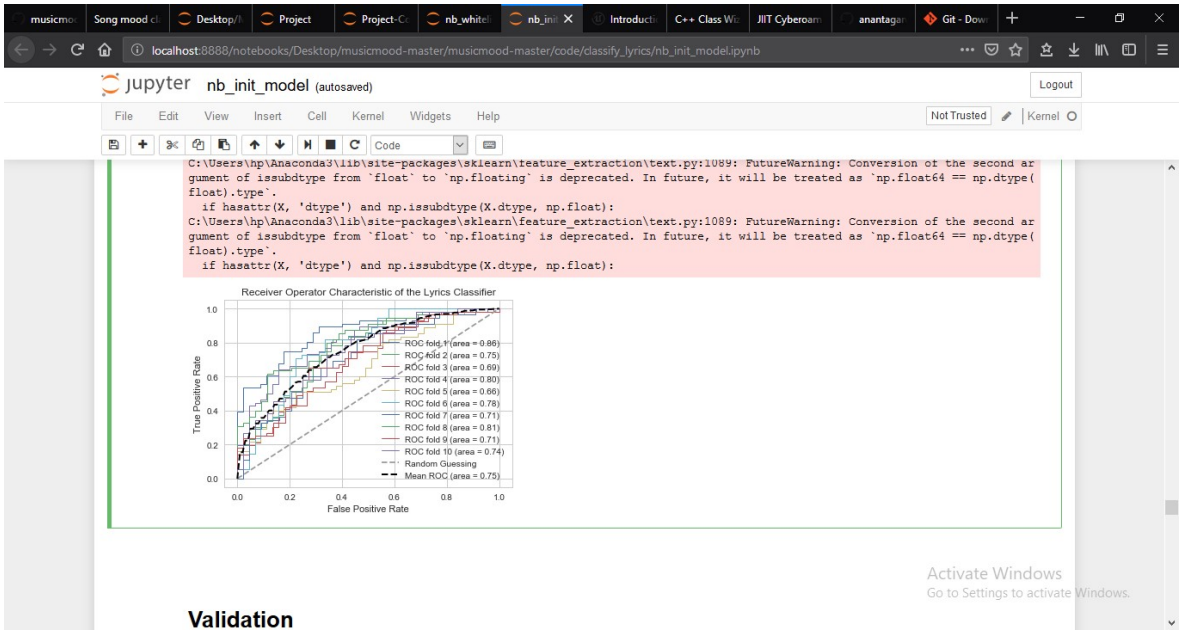
predict Highlight All Match Case Whole Words 6 of 35 matches



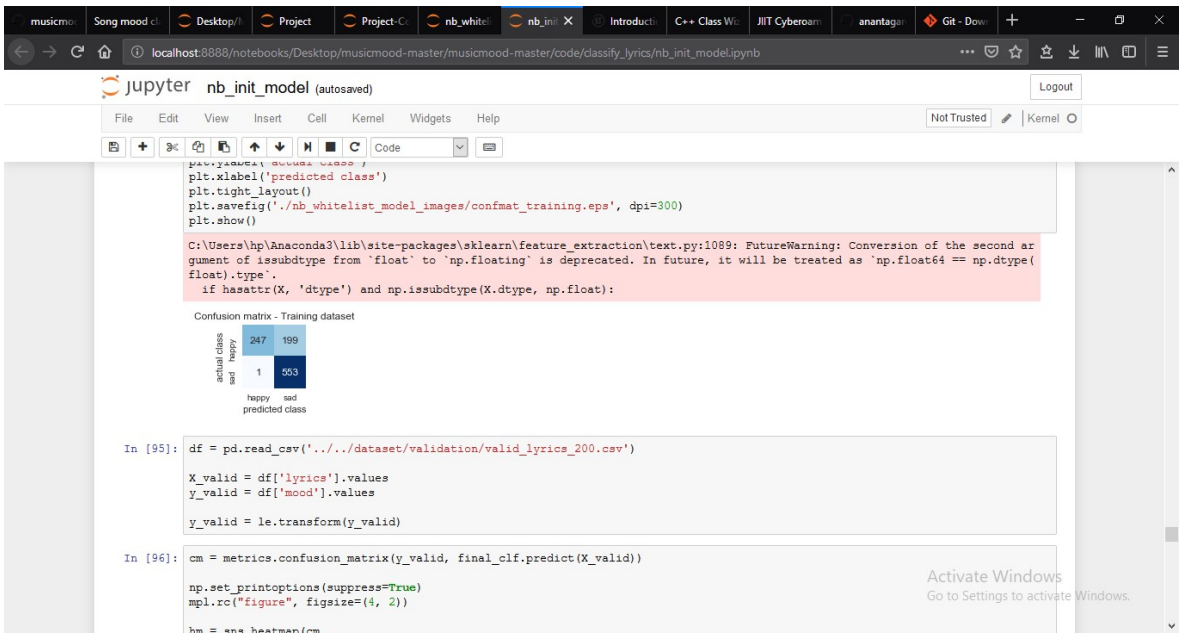


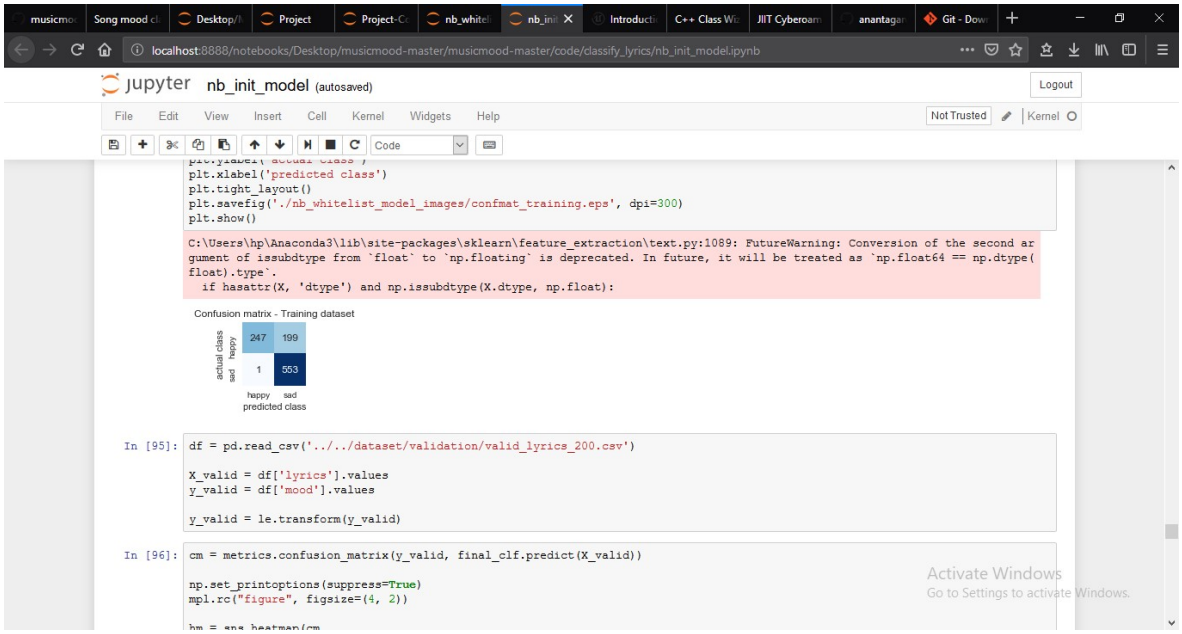
n-gram comparison

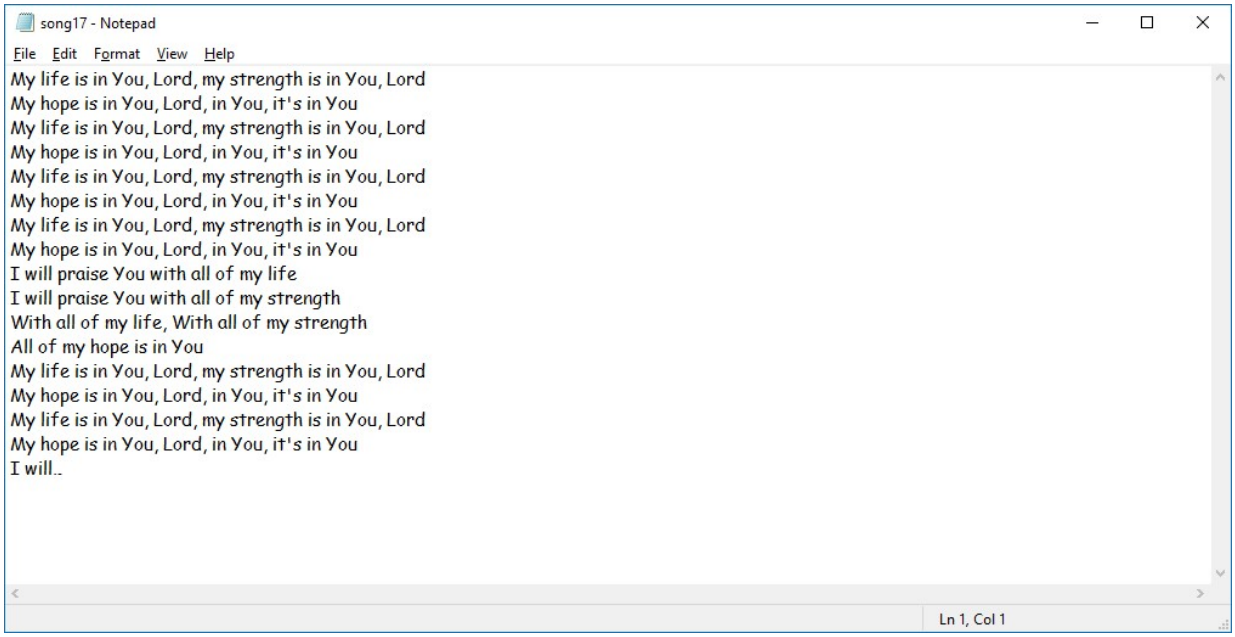
[back to top]

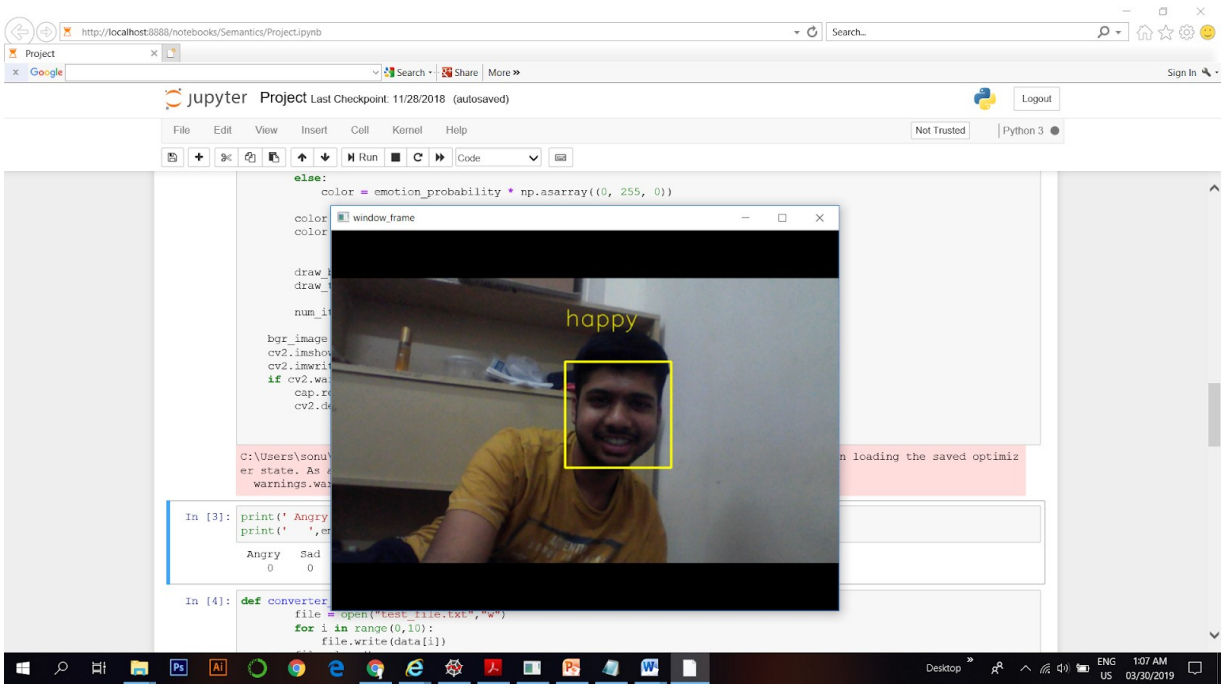

jupyter nb_init_model (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Kernel
Code
C:\Users\hp\Anaconda3\lib\site-packages\sklearn\feature_extraction\text.py:1089: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
if hasattr(X, 'dtype') and np.issubdtype(X.dtype, np.float):
Receiver Operator Characteristic of the Lyrics Classifier
True Positive Rate
False Positive Rate
ROC fold 1 (area = 0.86)
ROC fold 2 (area = 0.75)
ROC fold 3 (area = 0.69)
ROC fold 4 (area = 0.80)
ROC fold 5 (area = 0.66)
ROC fold 6 (area = 0.78)
ROC fold 7 (area = 0.71)
ROC fold 8 (area = 0.81)
ROC fold 9 (area = 0.71)
ROC fold 10 (area = 0.74)
Random Guessing
Mean ROC (area = 0.75)
Activate Windows
Go to Settings to activate Windows.

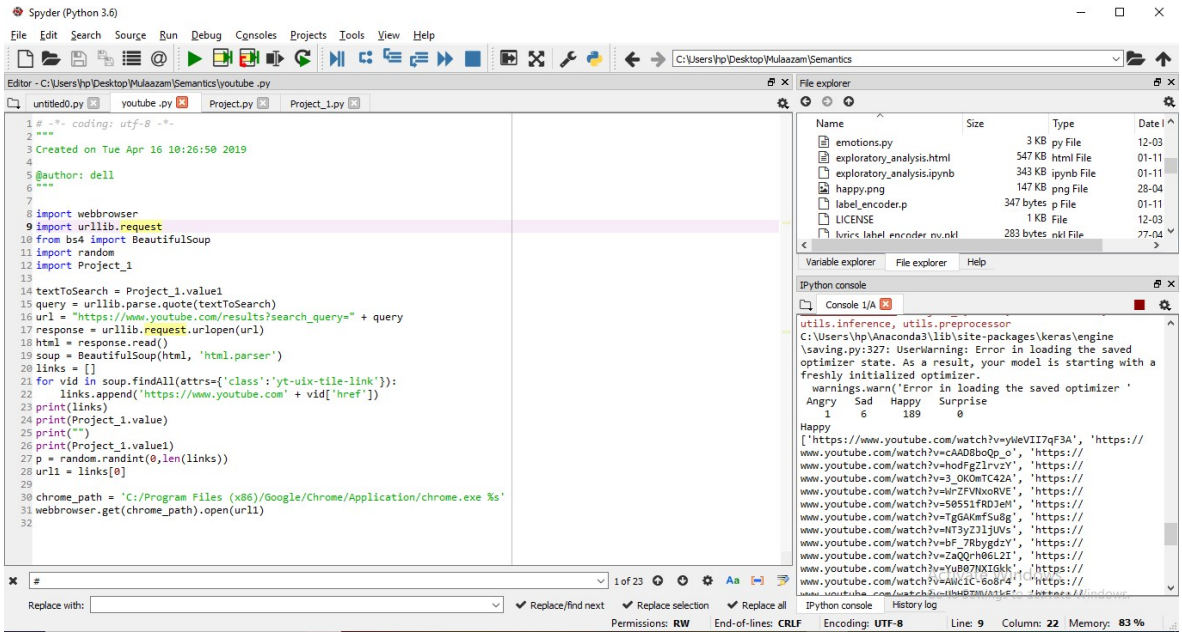
Validation

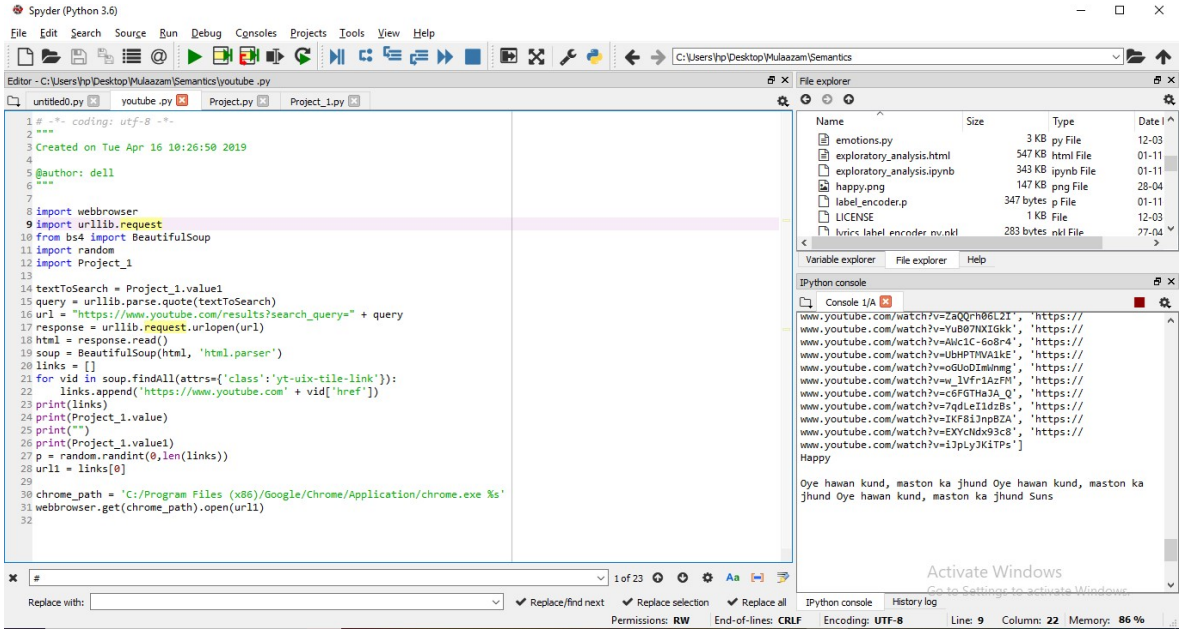

jupyter nb_init_model (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Kernel
Code
plt.xlabel('predicted class')
plt.tight_layout()
plt.savefig('./nb_whitelist_model_images/confmat_training.eps', dpi=300)
plt.show()
C:\Users\hp\Anaconda3\lib\site-packages\sklearn\feature_extraction\text.py:1089: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
if hasattr(X, 'dtype') and np.issubdtype(X.dtype, np.float):
Confusion matrix - Training dataset
actual class
happy
sad
247
199
1
553
happy
sad
predicted class
In [95]:
df = pd.read_csv('../../dataset/validation/valid_lyrics_200.csv')
X_valid = df['lyrics'].values
y_valid = df['mood'].values
y_valid = le.transform(y_valid)
In [96]:
cm = metrics.confusion_matrix(y_valid, final_clf.predict(X_valid))
np.set_printoptions(suppress=True)
mpl.rc("figure", figsize=(4, 2))
Activate Windows
Go to Settings to activate Windows.

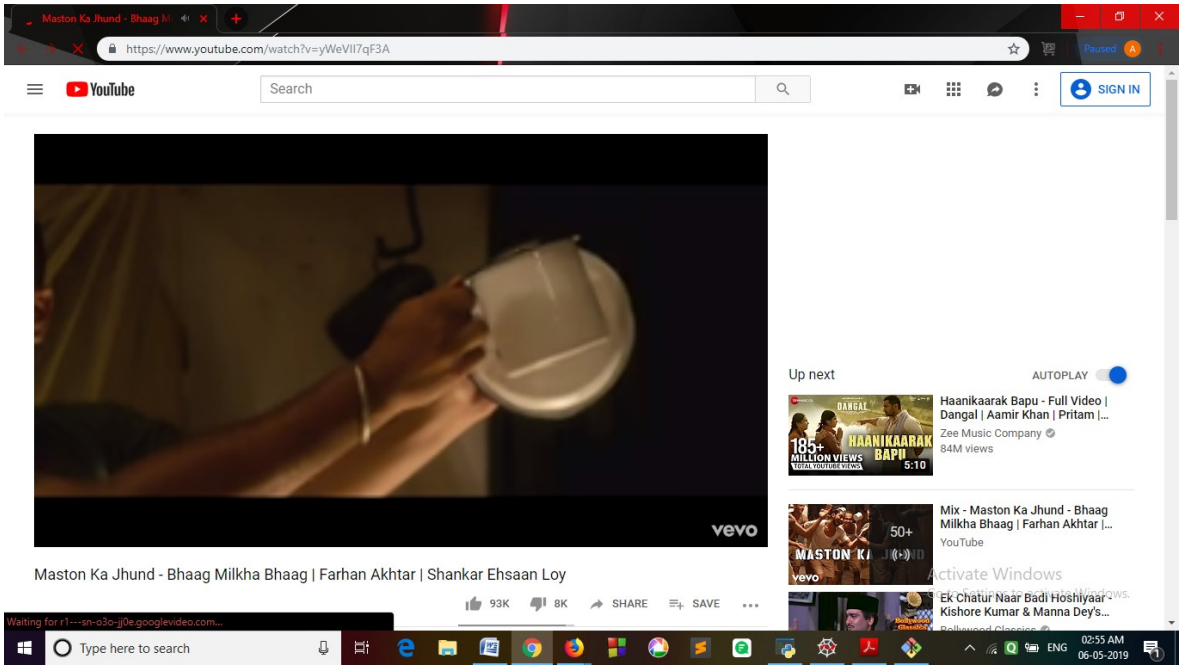

jupyter nb_init_model (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Kernel
Code
plt.xlabel('predicted class')
plt.tight_layout()
plt.savefig('./nb_whitelist_model_images/confmat_training.eps', dpi=300)
plt.show()
C:\Users\hp\Anaconda3\lib\site-packages\sklearn\feature_extraction\text.py:1089: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
if hasattr(X, 'dtype') and np.issubdtype(X.dtype, np.float):
Confusion matrix - Training dataset
actual class
happy
sad
247
199
1
553
happy
sad
predicted class
In [95]:
df = pd.read_csv('../../dataset/validation/valid_lyrics_200.csv')
X_valid = df['lyrics'].values
y_valid = df['mood'].values
y_valid = le.transform(y_valid)
In [96]:
cm = metrics.confusion_matrix(y_valid, final_clf.predict(X_valid))
np.set_printoptions(suppress=True)
mpl.rc("figure", figsize=(4, 2))
Activate Windows
Go to Settings to activate Windows.


song17 - Notepad
File Edit Format View Help
My life is in You, Lord, my strength is in You, Lord
My hope is in You, Lord, in You, it's in You
My life is in You, Lord, my strength is in You, Lord
My hope is in You, Lord, in You, it's in You
My life is in You, Lord, my strength is in You, Lord
My hope is in You, Lord, in You, it's in You
My life is in You, Lord, my strength is in You, Lord
My hope is in You, Lord, in You, it's in You
I will praise You with all of my life
I will praise You with all of my strength
With all of my life, With all of my strength
All of my hope is in You
My life is in You, Lord, my strength is in You, Lord
My hope is in You, Lord, in You, it's in You
My life is in You, Lord, my strength is in You, Lord
My hope is in You, Lord, in You, it's in You
I will..
Ln 1, Col 1


http://localhost:8888/notebooks/Semantics/Project.ipynb
jupyter Project Last Checkpoint: 11/28/2018 (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Not Trusted
Python 3
Run
Code
window_frame
happy
In [3]:
Angry Sad
0 0
In [4]:
def converter
file = open("test_file.txt","w")
for i in range(0,10):
file.write(data[i])
Desktop
ENG
US
1:07 AM
03/30/2019


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\hp\Desktop\Mulaazam\Semantics
Editor - C:\Users\hp\Desktop\Mulaazam\Semantics\youtube .py
untitled0.py
youtube .py
Project.py
Project_1.py
-*- coding: utf-8 -*-
"""
Created on Tue Apr 16 10:26:50 2019
@author: dell
"""
import webbrowser
import urllib.request
from bs4 import BeautifulSoup
import random
import Project_1
textToSearch = Project_1.value1
query = urllib.parse.quote(textToSearch)
url = "https://www.youtube.com/results?search_query=" + query
response = urllib.request.urlopen(url)
html = response.read()
soup = BeautifulSoup(html, 'html.parser')
links = []
for vid in soup.findAll(attrs={'class':'yt-uix-tile-link'}):
links.append('https://www.youtube.com' + vid['href'])
print(links)
print(Project_1.value)
print("")
print(Project_1.value1)
p = random.randint(0,len(links))
url1 = links[0]
chrome_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'
webbrowser.get(chrome_path).open(url1)
File explorer
Name Size Type Date
emotions.py 3 KB py File 12-03
exploratory_analysis.html 547 KB html File 01-11
exploratory_analysis.ipynb 343 KB ipynb File 01-11
happy.png 147 KB png File 28-04
label_encoder.p 347 bytes p File 01-11
LICENSE 1 KB File 12-03
Variable explorer File explorer Help
IPython console
Console 1/A
utils.inference, utils.preprocessor
C:\Users\hp\Anaconda3\lib\site-packages\keras\engine
\saving.py:327: UserWarning: Error in loading the saved
optimizer state. As a result, your model is starting with a
freshly initialized optimizer.
warnings.warn('Error in loading the saved optimizer '
Angry Sad Happy Surprise
1 6 189 0
Happy
['https://www.youtube.com/watch?v=yWeVII7qF3A', 'https://
www.youtube.com/watch?v=cAAD8boQp_o', 'https://
www.youtube.com/watch?v=hodFgZlrvzY', 'https://
www.youtube.com/watch?v=3_OKOmTC42A', 'https://
www.youtube.com/watch?v=WrZFVNxoRVE', 'https://
www.youtube.com/watch?v=50551fRDJeM', 'https://
www.youtube.com/watch?v=TgGAKmfSu8g', 'https://
www.youtube.com/watch?v=NT3yZ7ljUVs', 'https://
www.youtube.com/watch?v=bF_7RbygdzY', 'https://
www.youtube.com/watch?v=ZaQQrh06L2I', 'https://
www.youtube.com/watch?v=YuB07NXIGkk', 'https://
www.youtube.com/watch?v=AWc1C-6o8r4', 'https://
Replace with:
Replace/find next
Replace selection
Replace all
IPython console
History log
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 9
Column: 22
Memory: 83 %


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:\Users\hp\Desktop\Mulaazam\Semantics\youtube .py
untitled0.py
youtube .py
Project.py
Project_1.py
-*- coding: utf-8 -*-
"""
Created on Tue Apr 16 10:26:50 2019
@author: dell
"""
import webbrowser
import urllib.request
from bs4 import BeautifulSoup
import random
import Project_1
textToSearch = Project_1.value1
query = urllib.parse.quote(textToSearch)
url = "https://www.youtube.com/results?search_query=" + query
response = urllib.request.urlopen(url)
html = response.read()
soup = BeautifulSoup(html, 'html.parser')
links = []
for vid in soup.findAll(attrs={'class':'yt-uix-tile-link'}):
links.append('https://www.youtube.com' + vid['href'])
print(links)
print(Project_1.value)
print("")
print(Project_1.value1)
p = random.randint(0,len(links))
url1 = links[0]
chrome_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'
webbrowser.get(chrome_path).open(url1)
File explorer
Name Size Type
emotions.py 3 KB py File
exploratory_analysis.html 547 KB html File
exploratory_analysis.ipynb 343 KB ipynb File
happy.png 147 KB png File
label_encoder.p 347 bytes p File
LICENSE 1 KB File
Variable explorer File explorer Help
IPython console
Console 1/A
www.youtube.com/watch?v=ZaQQrh06L2I', 'https://
www.youtube.com/watch?v=YuB07NXIGkk', 'https://
www.youtube.com/watch?v=AWc1C-6o8r4', 'https://
www.youtube.com/watch?v=UbHPTMVA1kE', 'https://
www.youtube.com/watch?v=oGUoDImWnmg', 'https://
www.youtube.com/watch?v=w_lVfr1AzFM', 'https://
www.youtube.com/watch?v=c6F6TGTHaJA_Q', 'https://
www.youtube.com/watch?v=7qdLeI1dzBs', 'https://
www.youtube.com/watch?v=IKF8iJnpBZA', 'https://
www.youtube.com/watch?v=EXYcNdx93c8', 'https://
www.youtube.com/watch?v=iJpLyJKiTPs']
Happy
Oye hawan kund, maston ka jhund Oye hawan kund, maston ka jhund Oye hawan kund, maston ka jhund Suns
Activate Windows
Go to Settings to activate Windows.
Replace with:
Replace/find next
Replace selection
Replace all
IPython console
History log
Permissions: RW
End-of-lines: CRLF
Encoding: UTF-8
Line: 9
Column: 22
Memory: 86 %


Maston Ka Jhund - Bhaag M
https://www.youtube.com/watch?v=yWeVII7qF3A
Paused
YouTube
Search
SIGN IN
vevo
Maston Ka Jhund - Bhaag Milkha Bhaag | Farhan Akhtar | Shankar Ehsaan Loy
93K
8K
SHARE
SAVE
Up next
AUTOPLAY
Haanikaarak Bapu - Full Video | Dangal | Aamir Khan | Pritam |...
Zee Music Company
84M views
185+ MILLION VIEWS
5:10
Mix - Maston Ka Jhund - Bhaag Milkha Bhaag | Farhan Akhtar |...
YouTube
50+
Ek Chatur Naar Badi Hoshiyaar - Kishore Kumar & Manna Dey's...
Activate Windows
Waiting for r1---sn-o3o-jj0e.googlevideo.com...
Type here to search
ENG
02:55 AM
06-05-2019

Reference

1. http://www.cvauni.edu.vn/imguplоad_dinhkem/file/Chuyen%20De%20HCI/3_%20Analysis%20of%20Emotion%20Recognition%20using%20facial%20expression-speech.pdf

2. https://arxiv.org/ftp/arxiv/papers/1706/1706.01509.pdf

3. https://ieeexplore.ieee.org/document/4624313

4. https://www.irjet.net/archives/V3/i2/IRJET-V3I284.pdf

5. https://www.semanticscholar.org/paper/Comparing-SVM-and-na%C3%AFve-Bayes-classifiers-for-text-Hassan-Rafi/501cfa39b2d94443ddbf4bfb7bb82c9845fa789c

6. https://www.semanticscholar.org/paper/Genre-classification-for-million-song-dataset-using-Ogihara/d12c2f65d3792cd8aa3b9a80046f52fd77744403

7. https://www.semanticscholar.org/paper/Genre-classification-for-million-song-dataset-using-Hu-Ogihara/d12c2f65d3792cd8aa3b9a80046f52fd77744403

8. https://www.semanticscholar.org/paper/A-Bayesian-Approach-to-Filtering-Junk-E-Mail-Sahami-Dumais/4b7d5d9203e4ec2f03c1ffeff97fe28ead9d9db1

9. https://www.semanticscholar.org/paper/Automatic-Mood-Classification-Using-TF*IDF-Based-on-Zaanen-Kanters/2086ead3dc0f4c9d9e8b193aa2871c8f8c782fa3

10. https://www.semanticscholar.org/paper/Comparing-SVM-and-na%C3%AFve-Bayes-classifiers-for-text-Hassan-Rafi/501cfa39b2d94443ddbf4bfb7bb82c9845fa789c