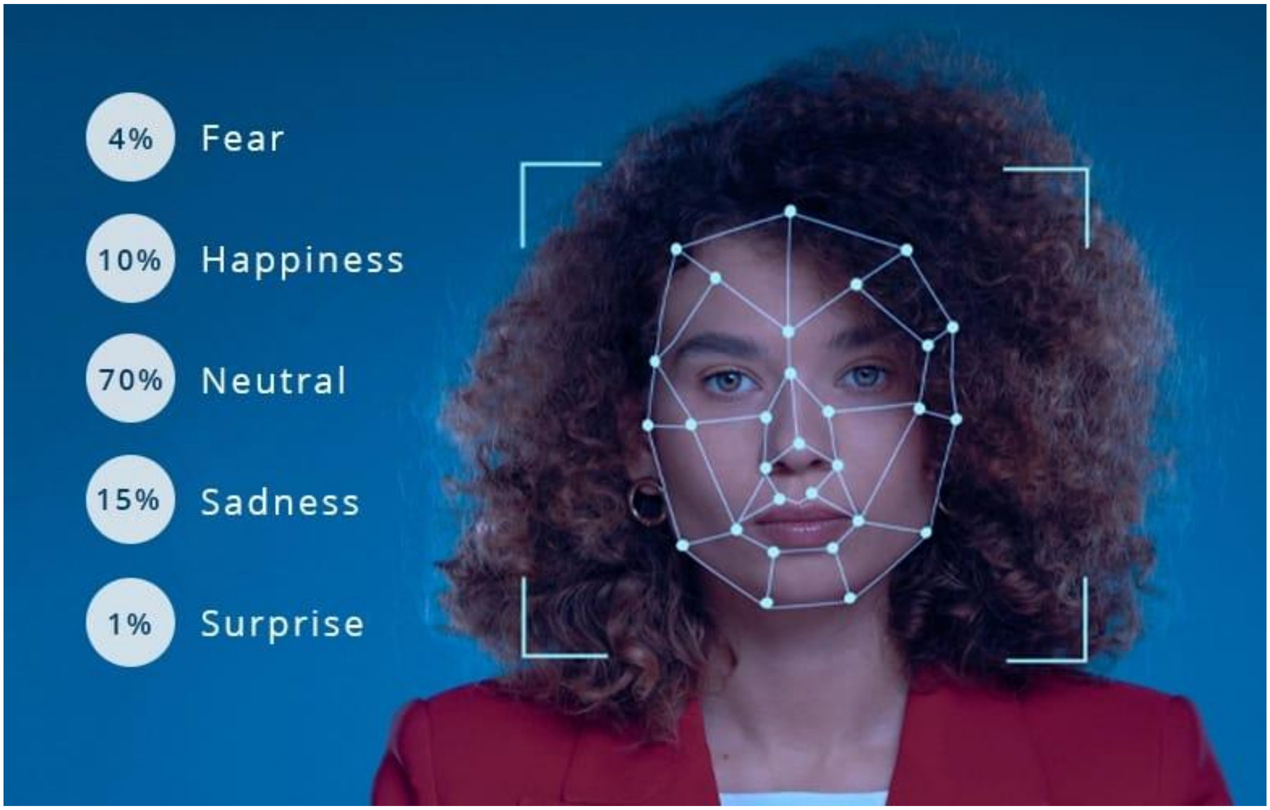
IMAGE RECOGNITION WITH IBM CLOUD USING VISAUL RECOGNITION

[CAD_PHASE-05]:



INTRODUCTION:

- In the previous two phase we have seen how to pre-process the dataset and we have seen how to create a emotion recognition model and to develop it.

- In this phase ,we will be covering everything that we have seen throughout the entire phases and we will be giving the complete project documentation.

Project Title: Emotion Detection Using IBM Cloud Visual Recognition

Project Description:

The project aims to develop an emotion detection system using IBM Cloud Visual Recognition, a powerful and versatile AI tool. This system will analyze images and provide insights into the emotional state of individuals in the pictures. By utilizing machine learning and deep learning techniques, it will recognize and categorize emotions such as happiness, sadness, anger, surprise, and more.

Project Objectives:

Emotion Classification: Implement a model that can accurately classify and detect emotions within images, distinguishing between various emotional states.

IBM Cloud Integration: Integrate IBM Cloud Visual Recognition into the project's architecture, leveraging its pre-trained models and custom training capabilities.

Data Collection: Gather a diverse dataset of images containing individuals displaying different emotions for training and evaluation.

Model Training: Train the emotion detection model using the collected dataset, ensuring it can effectively identify emotions with a high degree of accuracy.

Real-time Analysis: Enable real-time image analysis, allowing the system to process images as they are uploaded or streamed.

User Interface: Develop a user-friendly interface that allows users to upload or capture images and receive emotion analysis results.

API Integration: Provide the system as an API for seamless integration with other applications or services.

Customization: Allow users to customize and fine-tune the model for specific use cases or industries.

Scalability: Ensure the system can handle a large number of concurrent image analysis requests and is scalable to accommodate growing demands.

Benefits:

Emotion Recognition: The system will help businesses and organizations gain insights into customer and user emotions, aiding in decision-making and improving user experience.

Personalization: Emotion detection can be used for personalizing user experiences in various applications, from marketing to healthcare.

Research and Analysis: Researchers can use the system to analyze emotional trends and patterns in different contexts.

Enhanced Security: Emotion detection can enhance security systems by recognizing suspicious or abnormal emotional states.

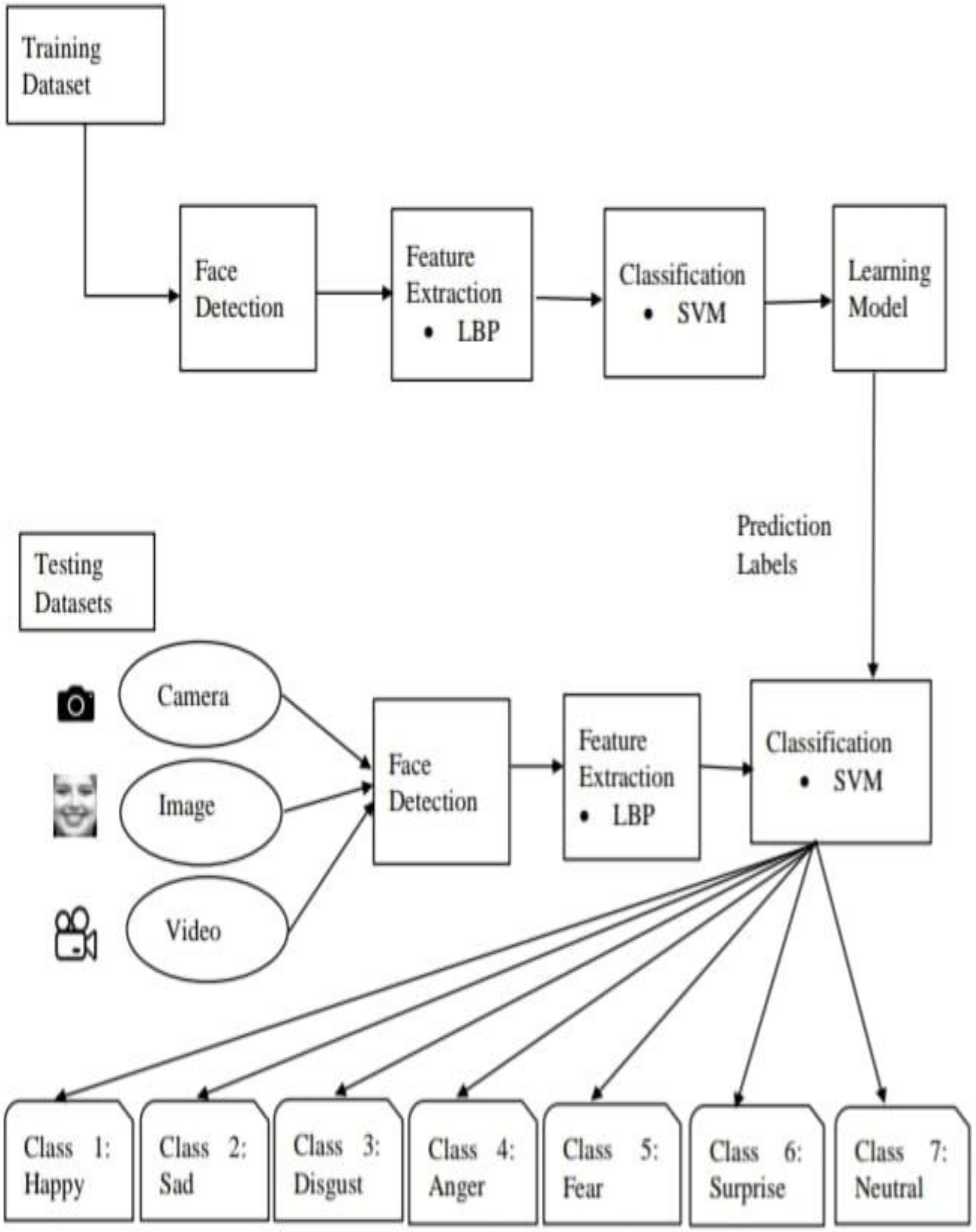
Customer Engagement: Businesses can use this technology for better customer engagement and satisfaction.

Technologies:

- IBM Cloud Visual Recognition
- Python (for model development)
- API development and integration

DESIGN THINKING:

Emotional recognition in Python typically involves using machine learning and computer vision libraries to analyze facial expressions and/or voice data.



PROBLEM STATEMENT:

1. Image processing can be broadly defined as the manipulation of signals which are inherently multidimensional.

2. The most common such signals are photographs and video sequences.

3. The goals of processing or manipulation can be

(i) compression for storage or transmission

(ii) enhancement or restoration

(ili) analysis, recognition, and understanding

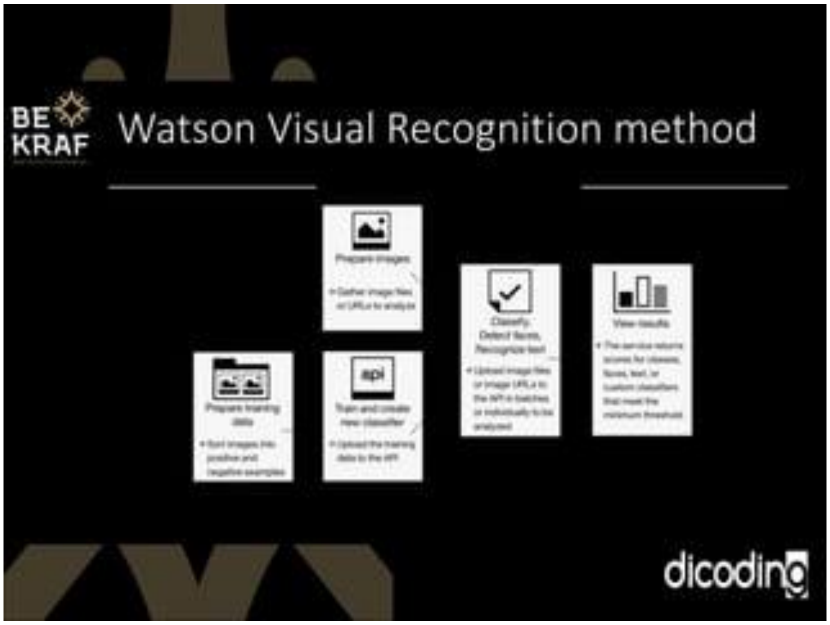
(iv) visualization for human observers.

PROBLEM DEFINITION:

1. The project involves creating an image recognition system using IBM Cloud Visual Recognition.

2. The goal is to develop a platform where users can upload images, and the system accurately classifies and describes the image contents.

3. This will enable users to craft engaging visual stories with the help of Al-generated captivating visuals and compelling narratives.



BE KRAF

Watson Visual Recognition method

Prepare images

- Gather image files or URLs to analyze

Prepare training data

- Sort images into positive and negative examples

Train and create new classifier

- Upload the training data to the API

Classify, Detect faces, Recognize text

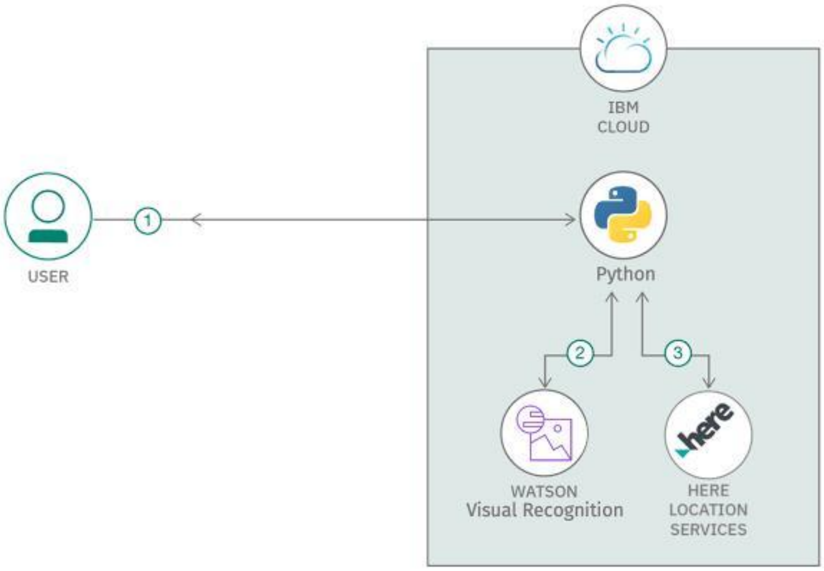
- Upload image files or image URLs to the API in batches or individually to be analyzed

View results

- The service returns scores for classes, faces, text, or custom classifiers that meet the minimum threshold

dicoding

I can provide you with a high-level outline of how you can integrate IBM Cloud Visual Recognition and AI-generated captions into an image recognition system



To create a Python program for emotional classification using IBM Cloud, you can leverage IBM Watson's Natural Language Understanding service. Here are the steps to get you started:

1. **Set Up IBM Cloud Account:** If you don't have an IBM Cloud account, you'll need to sign up for one.

2. **Create an NLU Service:** Once you have an IBM Cloud account, create a Natural Language Understanding (NLU) service instance.

3. **Obtain API Credentials:** After creating the NLU service, you'll need to obtain the API credentials (API Key and URL).

4. **Install the IBM Watson SDK:** Install the IBM Watson SDK for Python using pip: pip install ibm-watson .

5. **Python Code for Emotional Classification:** Here's a simple Python script to classify emotions in a text using the IBM Watson NLU

STEP 1

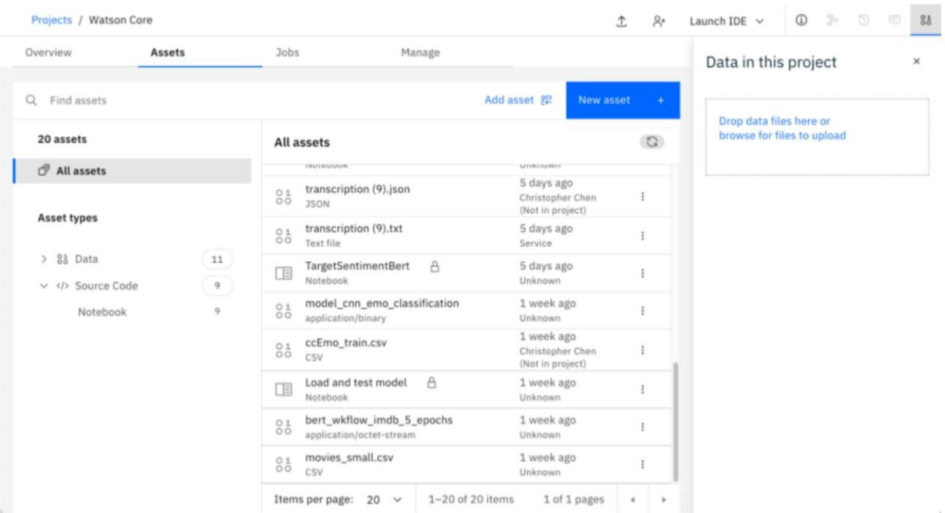
DATASET PRE-PROCESSING

Data sets for emotion classification require an input text feature column and an emotion label column with labels such as ‘joy’, ‘anger’, fear’, and ‘sadness’.

1. Download the tweets data set from **https://www.kaggle.com/datasets/anjaneyatripathi/emotion-classification-nlp?select=emotion-labels-train.csv** we can download all of the .csv files and combine them or download the single test .csv file.

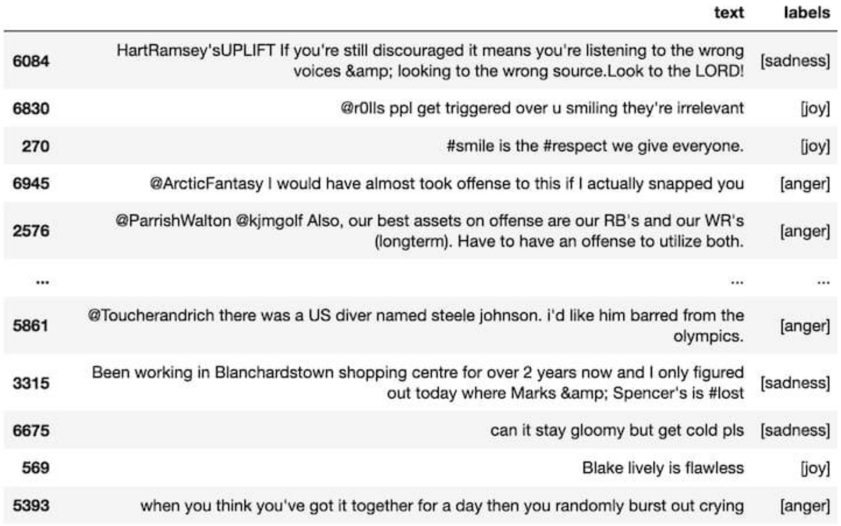
After downloading the data set, We upload the file to the Watson Studio project as a data asset. From there, the data is ready to be inserted into the notebook. The notebook is set up with instructions for reading the .csv as a pandas DataFrame.

Upload the data set to our Watson Studio project by going to the Assets tab, and then drag the data files.



transcription (9).json JSON	5 days ago Christopher Chen (Not in project)
transcription (9).txt Text file	5 days ago Service
TargetSentimentBert Notebook	5 days ago Unknown
model_cnn_emo_classification application/binary	1 week ago Unknown
ccEmo_train.csv CSV	1 week ago Christopher Chen (Not in project)
Load and test model Notebook	1 week ago Unknown
bert_wkflow_imdb_5_epochs application/octet-stream	1 week ago Unknown
movies_small.csv CSV	1 week ago Unknown

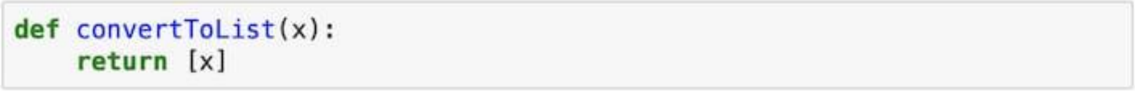
After we have added the data set to the project, we can access it from the Jupyter Notebook and read the .csv file into a pandas DataFrame.



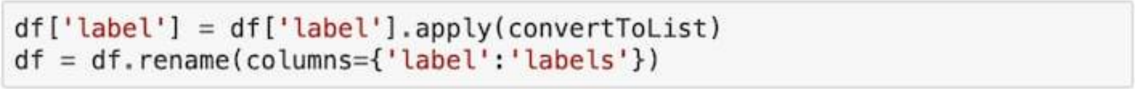
	text	labels
6084	HartRamsey'sUPLIFT If you're still discouraged it means you're listening to the wrong voices &amp; looking to the wrong source.Look to the LORD!	[sadness]
6830	@r0lls ppl get triggered over u smiling they're irrelevant	[joy]
270	#smile is the #respect we give everyone.	[joy]
6945	@ArcticFantasy I would have almost took offense to this if I actually snapped you	[anger]
2576	@ParrishWalton @kjmgolf Also, our best assets on offense are our RB's and our WR's (longterm). Have to have an offense to utilize both.	[anger]
...	...	...
5861	@Toucherandrich there was a US diver named steele johnson. i'd like him barred from the olympics.	[anger]
3315	Been working in Blanchardstown shopping centre for over 2 years now and I only figured out today where Marks &amp; Spencer's is #lost	[sadness]
6675	can it stay gloomy but get cold pls	[sadness]
569	Blake lively is flawless	[joy]
5393	when you think you've got it together for a day then you randomly burst out crying	[anger]

Step 2. Data processing

Data prepared for Watson NLP models needs to be formatted in such a way that there is a text feature column and a labels label column. The labels column must have type list. Format the data using the custom function .



```
def convertToList(x):
    return [x]
```



```
df['label'] = df['label'].apply(convertToList)
df = df.rename(columns={'label':'labels'})
```

Split the data into an 80/20 train-test split using sklearn, and then export it into a JSON format for the Watson NLP models to consume. Additionally, rename the column headers to the expected text and labels names, with the labels having type list.

df_train, df_test = train_test_split(df, test_size=0.2)

df_train.to_json('df_train.json', orient='records')

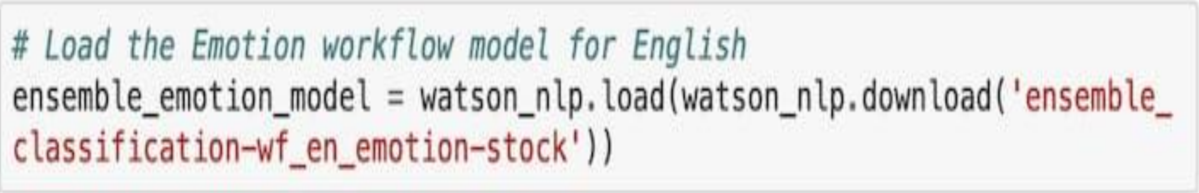
df_test.to_json('df_test.json', orient='records')

Step 3. Running pretrained emotion classification models

Watson NLP has two pretrained/prebuilt emotion classification models using the workflow system. The following examples use "Such a sweet boy. But after much thought and careful consideration, I've decided that the ruler for the next ten thousand years is going to have to be... me." as a single input test with the expected label to be "joy."

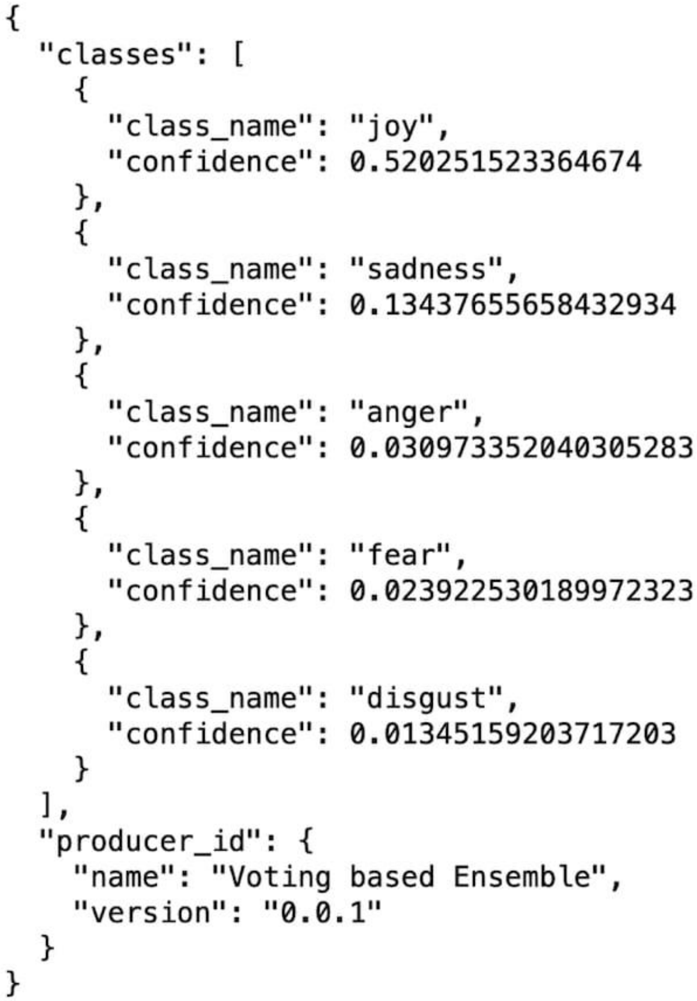
The Ensemble emotion model performs document emotion classification.

1. All models must be downloaded from the Watson NLP library on their initial run. They are saved to the runtime local, or local working path if the notebook is being run offline. Load the Emotion workflow model for English.



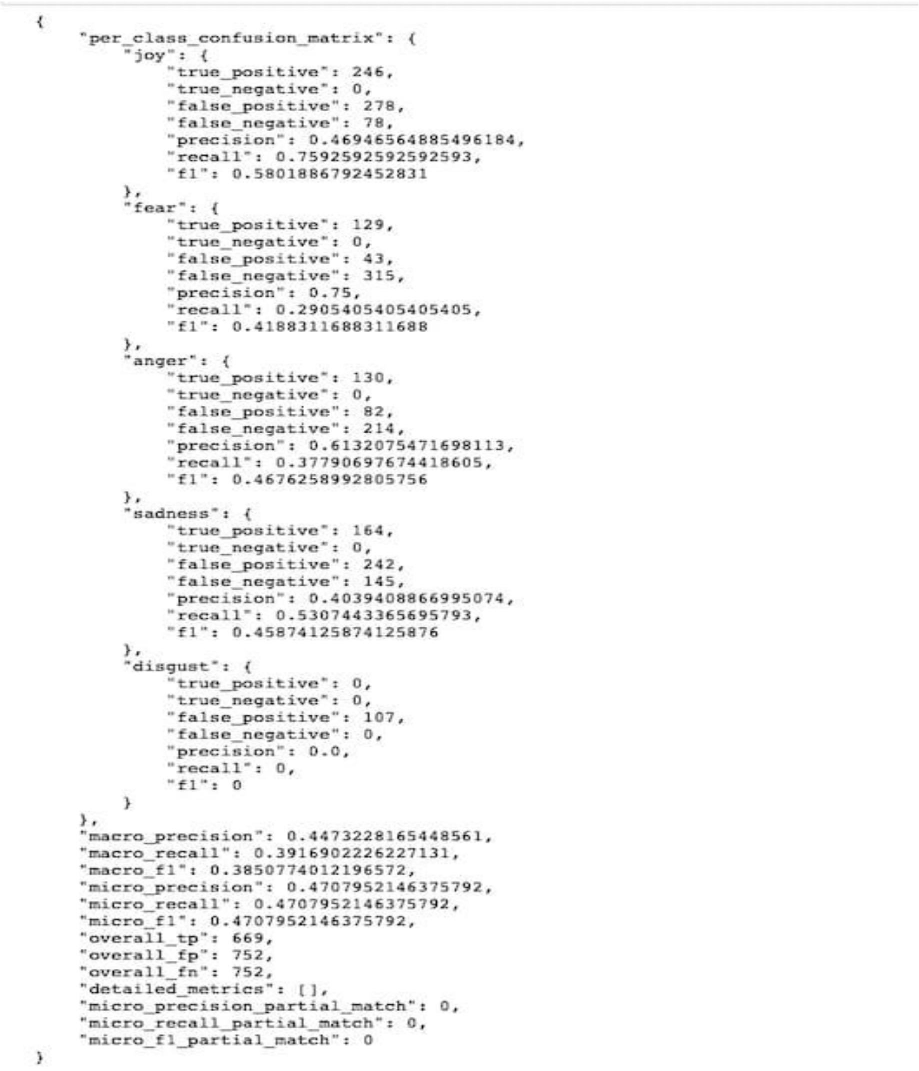
```
# Load the Emotion workflow model for English
ensemble_emotion_model = watson_nlp.load(watson_nlp.download('ensemble_
classification-wf_en_emotion-stock'))
```

Run the emotion model on a single document.



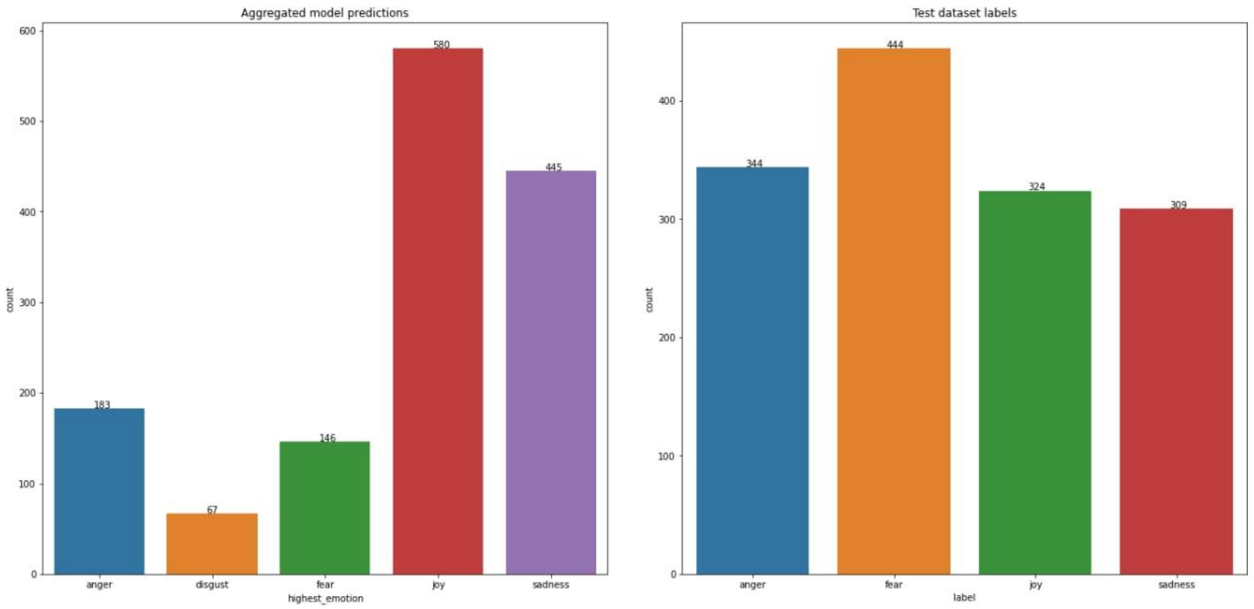
```
{
  "classes": [
    {
      "class_name": "joy",
      "confidence": 0.520251523364674
    },
    {
      "class_name": "sadness",
      "confidence": 0.13437655658432934
    },
    {
      "class_name": "anger",
      "confidence": 0.030973352040305283
    },
    {
      "class_name": "fear",
      "confidence": 0.023922530189972323
    },
    {
      "class_name": "disgust",
      "confidence": 0.01345159203717203
    }
  ],
  "producer_id": {
    "name": "Voting based Ensemble",
    "version": "0.0.1"
  }
}
```

Use the model.evaluate_quality() method to run the emotion classification model on the test data set.



```
{
    "per_class_confusion_matrix": {
        "joy": {
            "true_positive": 246,
            "true_negative": 0,
            "false_positive": 278,
            "false_negative": 78,
            "precision": 0.46946564885496184,
            "recall": 0.7592592592592593,
            "f1": 0.5801886792452831
        },
        "fear": {
            "true_positive": 129,
            "true_negative": 0,
            "false_positive": 43,
            "false_negative": 315,
            "precision": 0.75,
            "recall": 0.2905405405405405,
            "f1": 0.4188311688311688
        },
        "anger": {
            "true_positive": 130,
            "true_negative": 0,
            "false_positive": 82,
            "false_negative": 214,
            "precision": 0.6132075471698113,
            "recall": 0.37790697674418605,
            "f1": 0.4676258992805756
        },
        "sadness": {
            "true_positive": 164,
            "true_negative": 0,
            "false_positive": 242,
            "false_negative": 145,
            "precision": 0.4039408866995074,
            "recall": 0.5307443365695793,
            "f1": 0.45874125874125876
        },
        "disgust": {
            "true_positive": 0,
            "true_negative": 0,
            "false_positive": 107,
            "false_negative": 0,
            "precision": 0.0,
            "recall": 0,
            "f1": 0
        }
    },
    "macro_precision": 0.4473228165448561,
    "macro_recall": 0.3916902226227131,
    "macro_f1": 0.3850774012196572,
    "micro_precision": 0.4707952146375792,
    "micro_recall": 0.4707952146375792,
    "micro_f1": 0.4707952146375792,
    "overall_tp": 669,
    "overall_fp": 752,
    "overall_fn": 752,
    "detailed_metrics": [],
    "micro_precision_partial_match": 0,
    "micro_recall_partial_match": 0,
    "micro_f1_partial_match": 0
}
```

Visualize model prediction by parsing out the results of the model runs. we can compare the differences between predicted labels and actual labels in a histogram plot.



conclusion:

The implementation of emotional recognition using IBM Cloud has proven to be a robust and effective approach. Leveraging IBM Cloud's services and resources, this project has demonstrated its potential to accurately detect and analyze human emotions through various data inputs, offering scalability, reliability, and security. The integration of IBM Cloud's technologies, such as Watson, has opened doors for real-world applications in areas like customer service, sentiment analysis, and personal well-being. Nonetheless, ongoing optimization and ethical considerations must be central in the development and deployment of such solutions to ensure privacy, transparency, and data security.