**CS 5588 Data Science Capstone**

**Hands-On Session (1/30/2025)**

**Multi-Modal Sentiment & Emotion Analytics**

**with GitHub & Hugging Face**

**Objective**

This session integrates sentiment analysis on text with emotion classification from human face images using pre-trained models from Hugging Face in Jupyter Notebook. Additionally, participants will learn to manage version control with GitHub, including cloning repositories, creating branches, committing changes, and merging contributions.

At the end of this session, participants are required to submit a survey at:  
📌 [Survey Submission Link](https://forms.gle/FGus9sGeaNvsMMjo6)

**Session Breakdown (45 minutes)**

| Time | Activity |
| --- | --- |
| 0 - 5 min | Setup & Load Data (Dataset guidelines, extract & verify paths, GitHub setup) |
| 5 - 15 min | Data Preprocessing (Clean text, preprocess face images, GitHub branch creation) |
| 15 - 25 min | Model Inference (Text sentiment using BERT, image emotion using CNN) |
| 25 - 35 min | Multi-Modal Analysis & Visualization (Combine text sentiment with facial emotion) |
| 35 - 40 min | GitHub Collaboration (Commit changes, merge branches, review model results) |
| 40 - 45 min | Survey Submission & Wrap-Up (Discussion, next steps) |

**1. Setup & Load Data**

Dataset Guidelines

* Sentiment140 Dataset (Text Sentiment Analysis)
  + Large-scale Twitter dataset with labeled sentiments.
  + Available at: Sentiment140 Dataset
  + Sentiment Labels:
    - 0 → Negative
    - 4 → Positive
* FER2013 Dataset (Facial Emotion Recognition)
  + Contains images categorized into 7 emotions: Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise.
  + Available at: [FER2013 Dataset](https://www.kaggle.com/datasets/msambare/fer2013)
  + Folder Structure:

markdown

CopyEdit

fer2013/

├── images/

├── train/

├── angry/

├── disgust/

├── fear/

├── happy/

├── neutral/

├── sad/

├── surprise/

**Install & Import Libraries**

pip install transformers datasets torch torchvision pandas matplotlib seaborn plotly wordcloud pillow opencv-python zipfile36

**GitHub Setup**

1. **Clone the Repository**

git clone https://github.com/YOUR\_GITHUB\_USERNAME/YOUR\_PROJECT\_REPO.git

cd YOUR\_PROJECT\_REPO

1. **Extract and Verify Data Paths**

import os

import zipfile

# Extract dataset

zip\_path = "fer2013.zip" # Ensure correct uploaded filename

with zipfile.ZipFile(zip\_path, 'r') as zip\_ref:

zip\_ref.extractall("fer2013")

# Confirm extraction

print("Extracted files:", os.listdir("fer2013"))

1. **Data Preprocessing**

**Text Preprocessing for Sentiment Analysis**

|  |
| --- |
| import pandas as pd |
| import re |
| import string |
|  |
| # Load Sentiment140 dataset |
| csv\_path = "fer2013/sentiment140/training.1600000.processed.noemoticon.csv" |
| df = pd.read\_csv(csv\_path, encoding='ISO-8859-1', header=None) |
| df.columns = ["sentiment", "id", "date", "query", "user", "text"] |
| df = df[["sentiment", "text"]] # Keep relevant columns |
|  |
| # Convert sentiment labels (0 → Negative, 4 → Positive) |
| df["sentiment"] = df["sentiment"].replace({0: "Negative", 4: "Positive"}) |
|  |
| # Clean text |
| def clean\_text(text): |
| text = text.lower() |
| text = re.sub(r"http\S+", "", text) |
| text = text.translate(str.maketrans("", "", string.punctuation)) |
| return text |
|  |
| df["clean\_text"] = df["text"].apply(clean\_text) |
| df.head() |

**Image Preprocessing for Emotion Recognition**

|  |
| --- |
| import os |
| from torchvision import transforms |
| from PIL import Image |
| import matplotlib.pyplot as plt |
|  |
| # Define image path |
| image\_folder = "fer2013/images/train/happy" |
|  |
| # Confirm image path |
| if os.path.exists(image\_folder): |
| print("Available images:", os.listdir(image\_folder)[:5]) # Show some file names |
| else: |
| print("Error: Image folder not found!") |
|  |
| # Load images |
| image\_files = [os.path.join(image\_folder, f) for f in os.listdir(image\_folder) if f.endswith(('.jpg', '.png'))][:10] |
|  |
| # Define transformations |
| transform = transforms.Compose([ |
| transforms.Grayscale(num\_output\_channels=3), |
| transforms.Resize((224, 224)), |
| transforms.ToTensor(), |
| ]) |
|  |
| # Apply transformation |
| images = [transform(Image.open(img)) for img in image\_files] |
|  |
| # Display images |
| fig, axes = plt.subplots(1, 5, figsize=(10, 2)) |
| for i, img in enumerate(images[:5]): |
| axes[i].imshow(img.permute(1, 2, 0).numpy()) |
| axes[i].axis("off") |
| plt.show() |

1. **Model Inference**

**Text Sentiment Analysis with Hugging Face**

|  |
| --- |
| from transformers import pipeline |
|  |
| classifier = pipeline("sentiment-analysis") |
| df["predicted\_sentiment"] = df["clean\_text"].apply(lambda x: classifier(x)[0]["label"]) |
| df.head() |

**Emotion Classification Using Pre-Trained CNN**

|  |
| --- |
| import torch |
| from torchvision import models |
|  |
| # Load pre-trained ResNet model |
| model = models.resnet18(pretrained=True) |
| model.eval() |
|  |
| # Perform inference |
| with torch.no\_grad(): |
| predictions = [model(img.unsqueeze(0)) for img in images] |
|  |
| predictions = [torch.argmax(pred, dim=1).item() for pred in predictions] |
| emotions = ["Angry", "Disgust", "Fear", "Happy", "Sad", "Surprise", "Neutral"] |
| predicted\_emotions = [emotions[p] for p in predictions] |
|  |
| print(predicted\_emotions) |

1. **GitHub Collaboration**

**1. Create a New Branch & Make Changes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| git checkout | b feature | multi | modal | analysis |

**2. Add & Commit Changes**

git add .

git commit -m "Added multi-modal sentiment analysis"

**3. Push Branch to GitHub**

git push origin feature-multi-modal-analysis

**4. Create a Pull Request & Merge**

* Open GitHub Repository
* Click on Pull Requests → New Pull Request
* Compare with main branch
* Click Merge after review

**5. Survey Submission & Wrap-Up**

**Interactive Sentiment vs. Emotion Analysis**

import plotly.express as px

fig = px.scatter(df, x="predicted\_sentiment", y="predicted\_emotion", color="predicted\_emotion",

title="Sentiment vs Facial Emotion", labels={"predicted\_emotion": "Facial Emotion"})

fig.show()

Final Step: Submit the Survey

✅ [Submit Your Feedback Here](https://forms.gle/FGus9sGeaNvsMMjo6)

Discussion & Next Steps

* Review model results
* Discuss improvements for sentiment-emotion integration
* Explore real-time deployment (Flask/Streamlit)