K-Memes Final Proposal

1. Introduction

The rise of social media has significantly impacted youth mental health, making it essential to moderate harmful content. Platforms like Instagram have implemented features such as “Teen Accounts” to protect younger users. However, memes present a unique challenge because they convey messages through a combination of text and images, often with hidden or subtextual meanings. Traditional content moderation techniques struggle to analyze such complex media. Our project aims to develop a machine learning model capable of classifying memes into positive, negative, and neutral sentiments using a labeled dataset from Kaggle [1].

Problem Definition: The primary goal of the K-Memes project is to develop a machine learning model that can accurately classify memes into simplified sentiment categories (positive, negative, and neutral) to aid in content moderation. This tool aims to enhance mental well-being by identifying potentially harmful content, especially on platforms frequented by younger users.

Motivation: Given the unique, subtextual communication style of memes, traditional moderation techniques are often ineffective. By utilizing both image and text encoding through a multimodal model, this project seeks to bridge that gap, providing a more sophisticated approach to meme sentiment analysis.

Key Objectives:

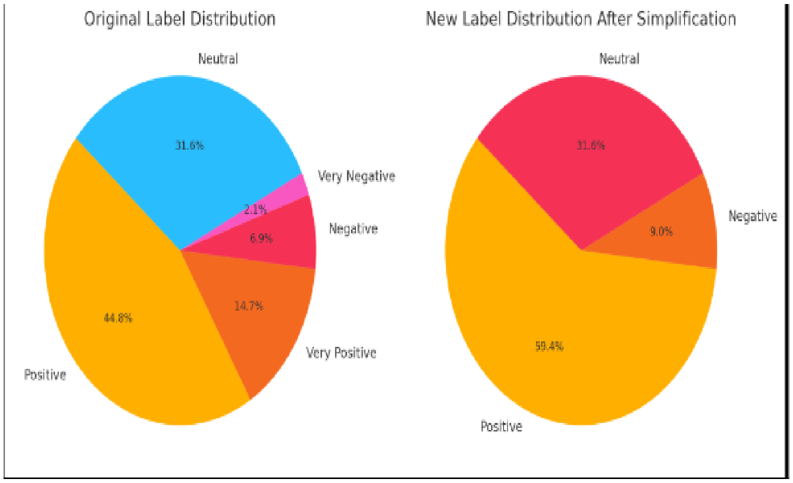
- Develop and implement a model architecture that integrates both image and text encoding.
- Process a labeled meme dataset to produce three sentiment classes for analysis.
- Evaluate the model’s ability to classify meme sentiment with an accuracy target of 85%.

2. Dataset and Preprocessing

Dataset: The dataset for this project is sourced from Kaggle, containing 6,992 labeled meme images [4]. Each meme image includes both image data and text data, which provides the basis for sentiment classification.

Preprocessing Steps:

- **Column Dropping:** Removed irrelevant columns, such as text_ocr, which are unnecessary for classification because another column, text_corrected, was already provided in the dataset
- **Handling Missing Values:** Replaced all NaN values with empty strings to ensure no interruptions in processing.
- **Label Simplification:** Reduced the sentiment labels from more granular categories to a simplified scale:
  - Combined "very positive" and "positive" into a single "positive" category.
  - Combined "very negative" and "negative" into a single "negative" category.



3. Model Development and Methodology

Overview: The K-Memes project implements a combined approach for sentiment analysis of memes, leveraging both text and image features. Currently, the model focuses on processing textual data with BERT embeddings and employs a classification model to categorize memes as positive, negative, or neutral.

Text Encoder (BERT):

We use BERT (Bidirectional Encoder Representations from Transformers) to handle the text data within memes. The model is loaded using the bert-base-uncased pretrained model from Hugging Face's Transformers library, ensuring it can capture the sentiment nuances in meme text.

Embedding Extraction: To extract embeddings from meme text, we pass each sample through BERT, taking the mean of the last hidden layer's outputs. This averaged representation serves as a compact and context-rich vector for each meme's text, capturing sentiment and contextual information.

Efficiency: For computational efficiency, we first check if precomputed embeddings are available (saved as bert_embeddings.pkl). If not, embeddings are generated and stored for future runs.

Label Preprocessing:

The overall_sentiment labels in the dataset are simplified by mapping "very_positive" to "positive" and "very_negative" to "negative." This preprocessing reduces class complexity and improves model interpretability. The labels are then encoded into numerical form using LabelEncoder to prepare them for classification.

Handling Class Imbalance:

To address potential class imbalance, we employ RandomOverSampler to increase the representation of minority classes within the training data. This technique generates a more balanced dataset, enhancing the model's ability to generalize across all sentiment classes.

Classification Model (XGBoost):

Choice of Model: XGBoost was selected for its robustness in handling high-dimensional data and its ability to model complex, non-linear relationships.

Model Parameters: The classifier is configured with 100 estimators, a learning rate of 0.1, and a maximum tree depth of 3. These settings balance training time with performance, given the available resources and complexity of the embeddings. Additionally, XGBoost is also more robust under imbalance classes compared to Random Forest and most models.

Training Process: The XGBoost classifier is trained on the resampled training dataset to optimize sentiment classification accuracy. Training is followed by evaluation on a test set to measure its effectiveness.

Cross-Validation: The code includes placeholders for experimenting with other classifiers (e.g., RandomForest) and conducting cross-validation to further refine the classification process.

Classification Model (Random Forest):

Choice of Model: Random Forest was chosen for its ensemble learning capabilities, combining multiple decision trees to improve classification accuracy. It is particularly effective in handling datasets with high dimensionality and categorical features, making it robust against overfitting.

Model Parameters: The classifier uses 100 estimators with a maximum depth of 10. These parameters were selected to balance model complexity and computation time, ensuring effective performance given the dataset size and structure.

Training Process: The Random Forest model is trained on the preprocessed training dataset to optimize sentiment classification performance. The trained model is then evaluated on a separate test dataset to measure its accuracy and robustness.

Cross-Validation: The approach includes cross-validation to fine-tune the number of trees and maximum depth, ensuring the model generalizes well across unseen data. The flexibility of this method allows for easy experimentation with alternative classifiers.

Clustering Model (K-Means):

Choice of Model: KMeans was selected for its simplicity and efficiency in unsupervised learning tasks. It is effective for partitioning data into distinct clusters, making it ideal for exploratory analysis of meme sentiment distributions.

Model Parameters: The algorithm was configured to identify 3 clusters (positive, neutral, negative sentiments) with a maximum of 300 iterations and an initialization via the k-means++ method. These settings ensure balanced convergence while reducing the likelihood of suboptimal clustering.

Training Process: KMeans was applied to the normalized embeddings of text and images. The clustering results were analyzed to identify the natural groupings within the data and to uncover latent sentiment patterns.

Cross-Validation: While cross-validation is less common in unsupervised models, experiments with varying cluster counts (e.g., 2, 4, 5) were conducted to validate the appropriateness of three sentiment categories. Metrics such as silhouette score and inertia guided the evaluation process.

4. Results and Discussion

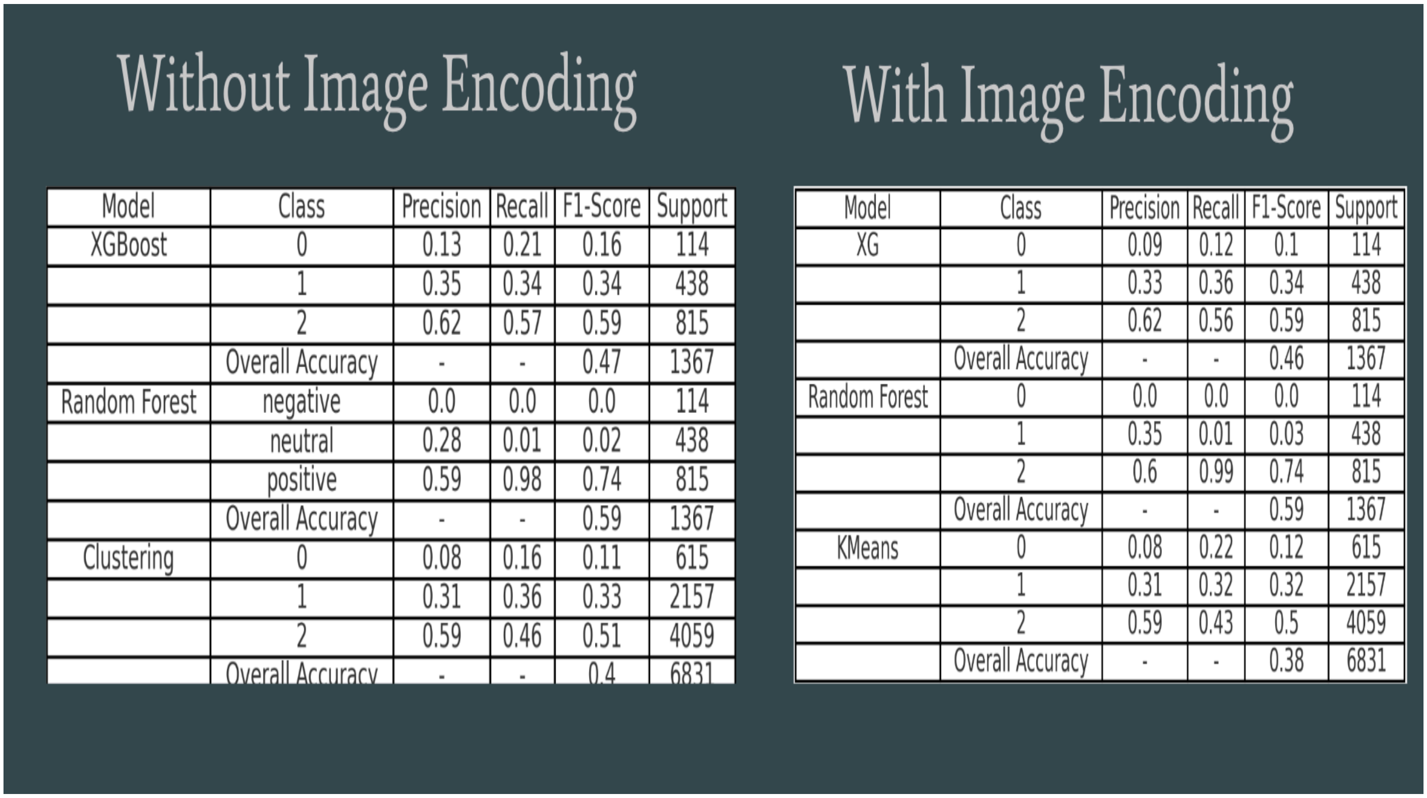
Without Image Encoding: The models generally performed better, as the inclusion of image encodings did not improve results, likely due to inefficiencies in feature fusion or overfitting caused by the dataset's size and quality. With Image Encoding: Performance metrics indicated hindrance in some model types, suggesting that the image encoding process did not add significant predictive power to the model. Overall Trends: The dataset's imbalance (skewed toward positive sentiment) contributed to the model bias, as the metrics heavily favored predicting positive sentiments.

- Random Forest: Performed the best due to its ensemble learning nature, robustness against overfitting, and ability to handle high-dimensional data. Its performance was aided by the categorical nature of the meme sentiment classification.
- XGBoost: Ranked second in performance. Its gradient boosting mechanism allows for modeling complex relationships, but it was slightly less effective than Random Forest due to overfitting challenges on the small dataset.

- KMeans: Performed poorly as it is an unsupervised algorithm and not inherently designed for classification tasks. Its clusters did not align well with the sentiment labels, likely due to the lack of well-defined boundaries in the feature space.

Random Forest > XGBoost > KMeans:

- Random Forest outperformed other models, achieving the highest accuracy and generalizability due to its ensemble learning approach.
- XGBoost closely followed but suffered from overfitting on a smaller dataset.
- KMeans was the least effective as it lacked the supervised framework necessary for sentiment classification.
- Impact of Image Encoding: The inclusion of image encodings negatively affected the performance across models, indicating that the additional image features were either redundant or poorly integrated into the feature space.

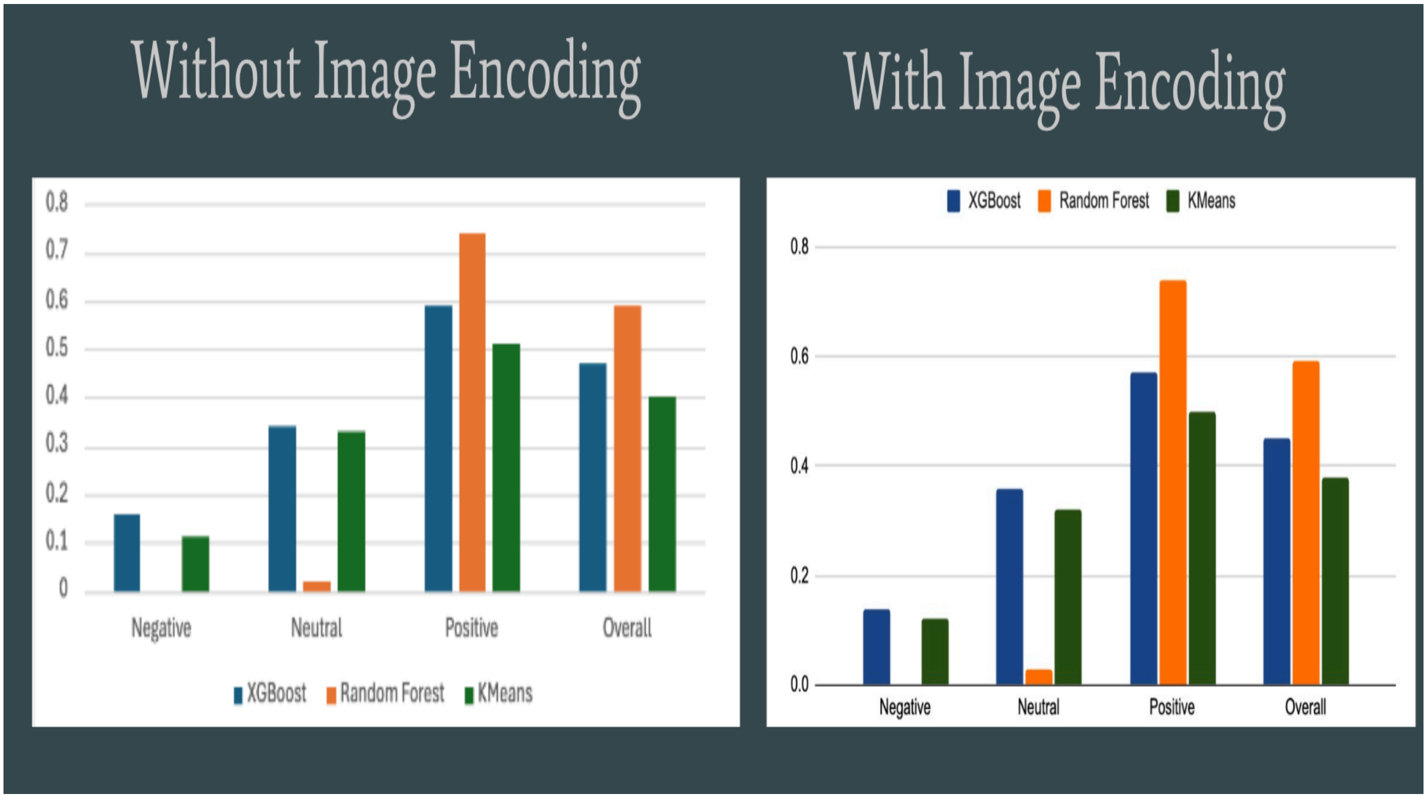


Without Image Encoding

Model	Class	Precision	Recall	F1-Score	Support
XGBoost	0	0.13	0.21	0.16	114
	1	0.35	0.34	0.34	438
	2	0.62	0.57	0.59	815
	Overall Accuracy	-	-	0.47	1367
Random Forest	negative	0.0	0.0	0.0	114
	neutral	0.28	0.01	0.02	438
	positive	0.59	0.98	0.74	815
	Overall Accuracy	-	-	0.59	1367
Clustering	0	0.08	0.16	0.11	615
	1	0.31	0.36	0.33	2157
	2	0.59	0.46	0.51	4059
	Overall Accuracy	-	-	0.4	6831

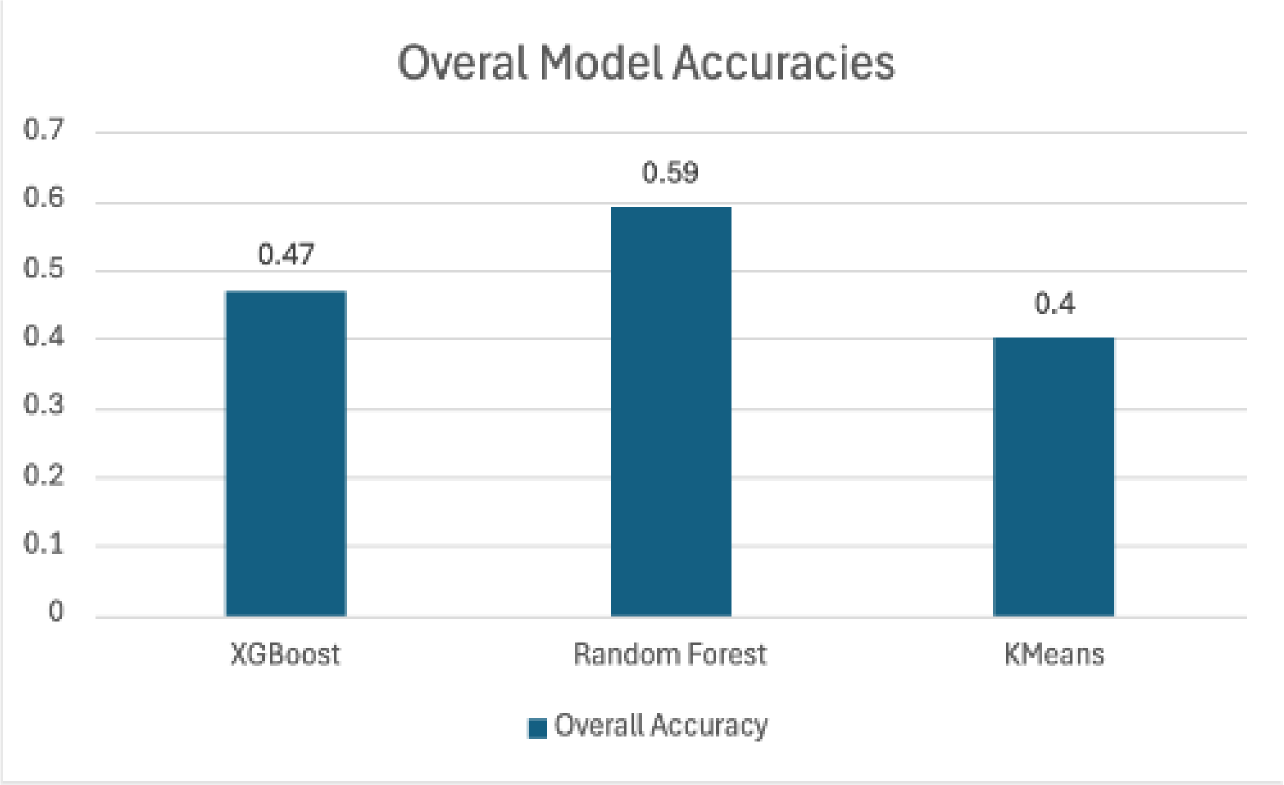
With Image Encoding

Model	Class	Precision	Recall	F1-Score	Support
XG	0	0.09	0.12	0.1	114
	1	0.33	0.36	0.34	438
	2	0.62	0.56	0.59	815
	Overall Accuracy	-	-	0.46	1367
Random Forest	0	0.0	0.0	0.0	114
	1	0.35	0.01	0.03	438
	2	0.6	0.99	0.74	815
	Overall Accuracy	-	-	0.59	1367
KMeans	0	0.08	0.22	0.12	615
	1	0.31	0.32	0.32	2157
	2	0.59	0.43	0.5	4059
	Overall Accuracy	-	-	0.38	6831



Without Image Encoding

With Image Encoding



5. Team Contributions

Name	Proposal Contributions
Aravinth	Model1, Model Improvement, Visualization
Ethan	Model2, Report Results and Discussion, Visualization, Overall Review
Jay	Model3, Report, Contribution Table, GitHub Repository, Pages
Shrey	Gantt Chart, Report, Model Verification, References, Repo Cleanup
Siddhant	Dataset and Metric Research, Model Development, Video Recording

6. Gantt Chart

Gantt Chart Link

7. References

- Yang, X., Lyu, T., Li, Q., Lee, C. Y., Ren, J., & Zhao, W. X. (2023). A survey on deep multimodal learning for social media: Models, applications and challenges. arXiv preprint arXiv:2302.09719.
- Sharma, P., Ding, N., Goodman, S., & Soricut, R. (2018). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2556-2565).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- Poria, S., Hazarika, D., Majumder, N., & Mihalcea, R. (2020). Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research. IEEE Transactions on Affective Computing.
- Xu, L., Lin, H., Pan, Y., Ren, H., & Chen, J. (2008). Constructing the affective lexicon ontology. Journal of the China Society for Scientific and Technical Information, 27(2), 180-185.
- Baltrušaitis, T., Ahuja, C., & Morency, L. P. (2018). Multimodal machine learning: A survey and taxonomy. IEEE transactions on pattern analysis and machine intelligence, 41(2), 423-443.
- Soleymani, M., Garcia, D., Jou, B., Schuller, B., Chang, S. F., & Pantic, M. (2017). A survey of multimodal sentiment analysis. Image and Vision Computing, 65, 3-14.
- French, A. M., Guo, C., & Shim, J. P. (2014). Current status, issues, and future of bring your own device (BYOD). Communications of the Association for Information Systems, 35(1), 10.

K-Memes Midterm Proposal

1. Project Objectives

Goal: The primary goal of the K-Memes project is to develop a machine learning model that can accurately classify memes into simplified sentiment categories (positive, negative, and neutral) to aid in content moderation. This tool aims to enhance mental well-being by identifying potentially harmful content, especially on platforms frequented by younger users.

Motivation: Given the unique, subtextual communication style of memes, traditional moderation techniques are often ineffective. By utilizing both image and text encoding through a multimodal model, this project seeks to bridge that gap, providing a more sophisticated approach to meme sentiment analysis.

Key Objectives:

- Develop and implement a model architecture that integrates both image and text encoding.
- Process a labeled meme dataset to produce three sentiment classes for analysis.
- Evaluate the model’s ability to classify meme sentiment with an accuracy target of 85%.

Summary of Progress:

For this checkpoint, we only used the textual data that was given in the dataset for each meme. We wanted to have a working model for the textual data and get a baseline accuracy, before we incorporated the image data as well. A summary of the work we did with respect to the checkpoint’s requirements are:

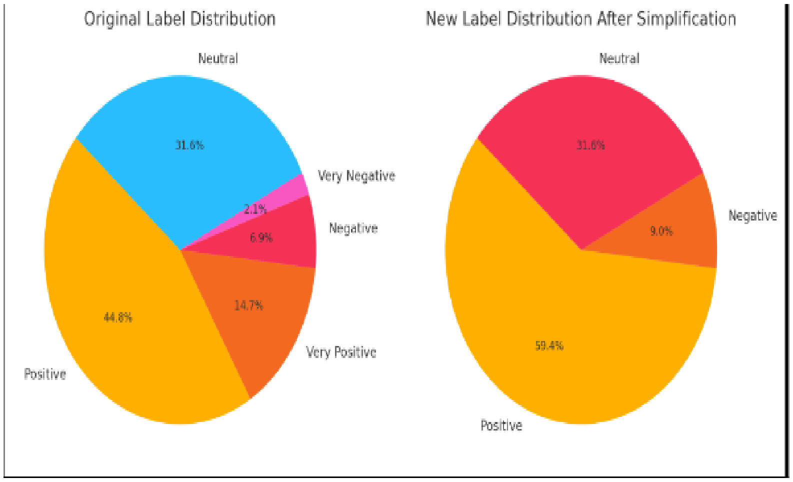
- Data Preprocessing: dropped unnecessary columns, replaced NaN values with empty strings, Reduced number of classes
- Data Preprocessing: BERT Embeddings for text. We converted the text for each of the images into BERT vector embeddings, that can be used by our two models
- Oversampling:
- Model selection: XGBoost (Supervised), Random Forest (Supervised) and KMeans (Unsupervised). We have evaluated and compared the results for these models below

2. Dataset and Preprocessing

Dataset: The dataset for this project is sourced from Kaggle, containing 6,992 labeled meme images [4]. Each meme image includes both image data and text data, which provides the basis for sentiment classification.

Preprocessing Steps:

- **Column Dropping:** Removed irrelevant columns, such as text_ocr, which are unnecessary for classification because another column, text_corrected, was already provided in the dataset
- **Handling Missing Values:** Replaced all NaN values with empty strings to ensure no interruptions in processing.
- **Label Simplification:** Reduced the sentiment labels from more granular categories to a simplified scale:
  - Combined "very positive" and "positive" into a single "positive" category.
  - Combined "very negative" and "negative" into a single "negative" category.



3. Model Development and Methodology

Overview: The K-Memes project implements a combined approach for sentiment analysis of memes, leveraging both text and image features. Currently, the model focuses on processing textual data with BERT embeddings and employs a classification model to categorize memes as positive, negative, or neutral.

Text Encoder (BERT):

We use BERT (Bidirectional Encoder Representations from Transformers) to handle the text data within memes. The model is loaded using the bert-base-uncased pretrained model from Hugging Face’s Transformers library, ensuring it can capture the sentiment nuances in meme text.

Embedding Extraction: To extract embeddings from meme text, we pass each sample through BERT, taking the mean of the last hidden layer’s outputs. This averaged representation serves as a compact and context-rich vector for each meme’s text, capturing sentiment and contextual information.

Efficiency: For computational efficiency, we first check if precomputed embeddings are available (saved as bert_embeddings.pkl). If not, embeddings are generated and stored for future runs.

Label Preprocessing:

The overall_sentiment labels in the dataset are simplified by mapping "very_positive" to "positive" and "very_negative" to "negative." This preprocessing reduces class complexity and improves model interpretability. The labels are then encoded into numerical form using LabelEncoder to prepare them for classification.

Handling Class Imbalance:

To address potential class imbalance, we employ RandomOverSampler to increase the representation of minority classes within the training data. This technique generates a more balanced dataset, enhancing the model’s ability to generalize across all sentiment classes.

Classification Model (XGBoost):

Choice of Model: XGBoost was selected for its robustness in handling high-dimensional data and its ability to model complex, non-linear relationships.

Model Parameters: The classifier is configured with 100 estimators, a learning rate of 0.1, and a maximum tree depth of 3. These settings balance training time with performance, given the available resources and complexity of the embeddings. Additionally, XGBoost is also more robust under imbalance classes compared to Random Forest and most models.

Training Process: The XGBoost classifier is trained on the resampled training dataset to optimize sentiment classification accuracy. Training is followed by evaluation on a test set to measure its effectiveness.

Cross-Validation: The code includes placeholders for experimenting with other classifiers (e.g., RandomForest) and conducting cross-validation to further refine the classification process.

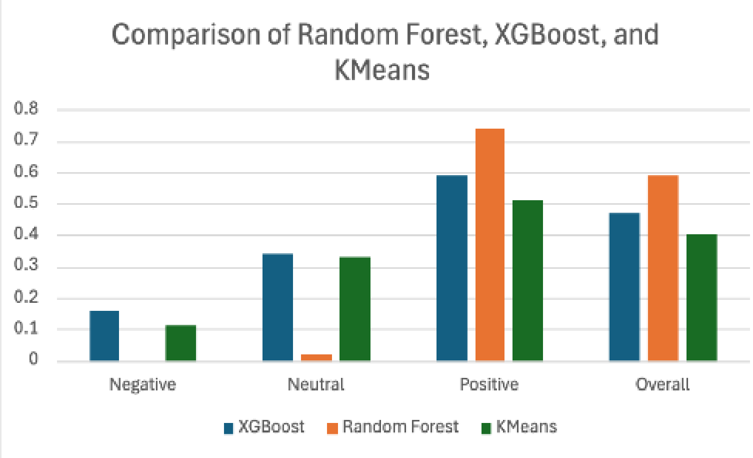
Model Evaluation:

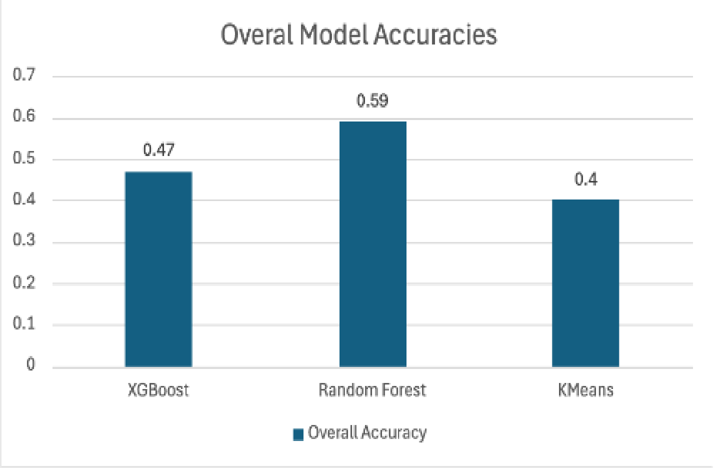
After training, the model generates predictions on the test set. Key performance metrics, such as precision, recall, and F1-Score, are then computed to assess the classifier's performance.

Preliminary Results: These results will be documented and analyzed to determine the model's ability to classify meme sentiment accurately.

4. Results and Discussion

When visualizing the models' capabilities through their testing accuracy, it becomes abundantly clear from the Random Forest data that much of the input data is considered positive, resulting in the model skewing towards simply marking no meme as negative. This resulted in the largest accuracy of 0.59 across the three models and we will need to determine a method to more address this limitation. One way we can push the models to be more selective is through adding more classifications to the data, as we had lowered the groupings from 5 (very positive, positive, neutral, negative, very negative) to 3 (positive, neutral, negative). This process may have resulted in positive becoming too large of a section of the data and further splitting the data may serve to reduce model reliance on biases in the data. Additionally, adding image encoders will serve to add more classification methods, allowing for more accurate representations of the data. Finally, adding penalizations for falsely indicating positives or promoting true indications of negatives/neutrals may push the model to more readily indicate neutral and negative classifications. Kmeans also has a better accuracy on the minority classes so it seems that the BERT embeddings do tend to cluster some amount. The silhouette score of 0.05 however indicates that our clustering is still not that good and has room for improvement. We will be trying to use other clustering methods such as DBSCAN and HDBSCAN to see if we can find better results with unsupervised learning.





5. Team Contributions

Name	Proposal Contributions
Aravinth	Model Improvement, Visualization
Ethan	Report Results and Discussion, Visualization, Overall Review
Jay	Report, Contribution Table, GitHub Repository, Pages
Shrey	Gantt Chart, Report, Model Verification, References, Repo Cleanup
Siddhant	Dataset and Metric Research, Model Development

6. Gantt Chart

Gantt Chart Link

7. References

- Yang, X., Lyu, T., Li, Q., Lee, C. Y., Ren, J., & Zhao, W. X. (2023). A survey on deep multimodal learning for social media: Models, applications and challenges. arXiv preprint arXiv:2302.09719.
- Sharma, P., Ding, N., Goodman, S., & Soricut, R. (2018). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2556-2565).

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- Poria, S., Hazarika, D., Majumder, N., & Mihalcea, R. (2020). Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research. IEEE Transactions on Affective Computing.
- Xu, L., Lin, H., Pan, Y., Ren, H., & Chen, J. (2008). Constructing the affective lexicon ontology. Journal of the China Society for Scientific and Technical Information, 27(2), 180-185.
- Baltrušaitis, T., Ahuja, C., & Morency, L. P. (2018). Multimodal machine learning: A survey and taxonomy. IEEE transactions on pattern analysis and machine intelligence, 41(2), 423-443.
- Soleymani, M., Garcia, D., Jou, B., Schuller, B., Chang, S. F., & Pantic, M. (2017). A survey of multimodal sentiment analysis. Image and Vision Computing, 65, 3-14.
- French, A. M., Guo, C., & Shim, J. P. (2014). Current status, issues, and future of bring your own device (BYOD). Communications of the Association for Information Systems, 35(1), 10.

K-Memes: Proposal Report

1. Introduction

The rise of social media has significantly impacted youth mental health, making it essential to moderate harmful content. Platforms like Instagram have implemented features such as “Teen Accounts” to protect younger users. However, memes present a unique challenge because they convey messages through a combination of text and images, often with hidden or subtextual meanings. Traditional content moderation techniques struggle to analyze such complex media. Our project aims to develop a machine learning model capable of classifying memes into positive, negative, and neutral sentiments using a labeled dataset from Kaggle [1].

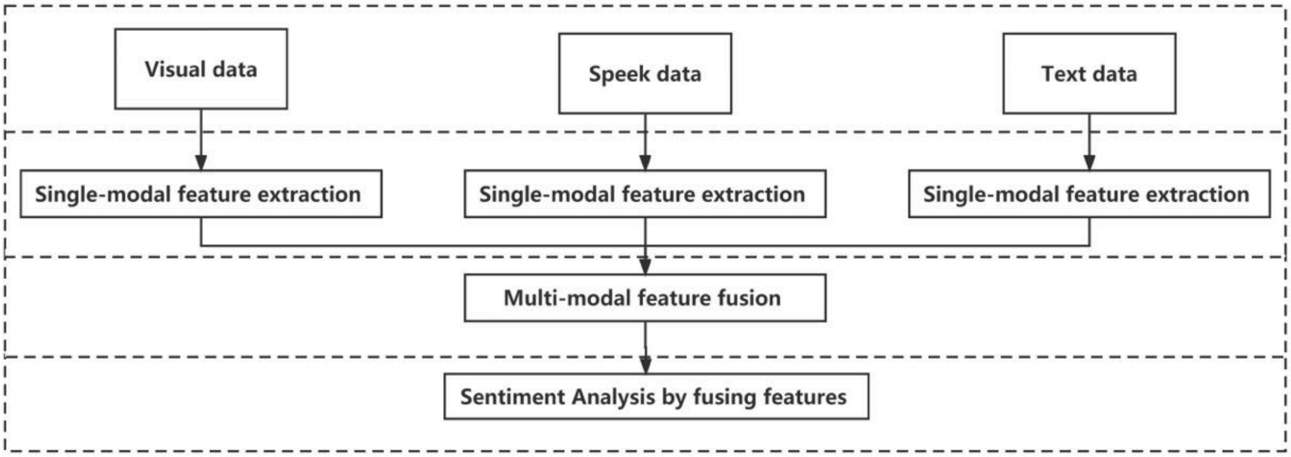
2. Problem Definition

Moderating memes is difficult because they often include irony, humor, or subtext, making it challenging to detect harmful messages. The primary objective of this project is to create a machine learning model that can accurately categorize memes into three simplified sentiment categories: positive, negative, or neutral. This tool can help improve online spaces by filtering harmful content and ensuring a safer environment for younger audiences.

3. Model Architecture Overview

Our model will analyze memes by encoding both image and text data before combining these encodings into a single vector, which will then be used for classification.

- **Image Encoder:** We will use **ResNet50**, a deep convolutional neural network pretrained on the ImageNet dataset [2]. ResNet50 excels at extracting meaningful visual features from images, making it ideal for our meme classification task.
- **Text Encoder:** We will employ **BERT** (Bidirectional Encoder Representations from Transformers) to handle the textual content of memes. BERT is capable of understanding the context of text and will generate feature vectors representing the sentiment expressed in the text [3].
- **Feature Concatenation:** After encoding both the image and text features, we will concatenate them into a single vector, capturing all the relevant information for sentiment classification.


Visual data
Speek data
Text data
Single-modal feature extraction
Single-modal feature extraction
Single-modal feature extraction
Multi-modal feature fusion
Sentiment Analysis by fusing features

4. Data Preprocessing Methods

- **Label Simplification:** Converting the original sentiment scale into three categories: positive, negative, and neutral.
- **Image Resizing:** Using OpenCV and PIL to resize all images to 224x224 pixels, ensuring compatibility with ResNet50 [2].
- **Text/Image Normalization:** Case normalization for text data, removal of stop words using NLP libraries like nltk or spaCy, and normalization of image pixel values.
- **Tokenization:** Breaking down text into tokens (words) for analysis and sentiment extraction.

5. Models for Classification

- **Supervised Learning (Decision Tree):** A Decision Tree classifier will categorize memes into positive, negative, or neutral categories. This model is non-linear and suited for complex, small to medium-sized datasets.
- **Supervised Learning (Support Vector Machine - SVM):** SVM is well-suited for high-dimensional data like the concatenated BERT and ResNet50 embeddings. It can capture non-linear relationships through different kernel choices and is robust to outliers. More about SVM in Scikit-learn can be found here [4].
- **Unsupervised Learning (KMeans):** We will also implement KMeans clustering to group memes based on their similarities. This approach is useful for exploring hidden patterns within the dataset.

6. Metrics for Evaluation

- **Accuracy:** Measures the percentage of correct predictions out of all samples. We aim for around 85% accuracy with the Decision Tree and SVM models.
- **F1-Score:** A harmonic mean of precision and recall, which is useful for handling class imbalance. We expect an F1-Score near 80%.
- **Silhouette Score:** Assesses the cohesion and separation of clusters in KMeans. A score around 0.6 is expected, indicating well-defined clusters.

7. Project Goals and Expected Results

The project's main goals include improving mental well-being by identifying and moderating harmful content. Our supervised model is expected to reach an accuracy of 85% and an F1-Score around 80%, while the KMeans model is expected to achieve a silhouette score of 0.6. These results will demonstrate the feasibility of using machine learning for meme sentiment analysis and contribute to safer social media spaces for youth.

8. Proposal Contributions and link to Gantt Chart

Name	Proposal Contributions
Aravinth	Scheduling
Ethan	Presentation, Literature Review, Dataset Research
Jay	Presentation, Report, Contribution Table, GitHub Repository
Shrey	Presentation, Model, Dataset Research
Siddhant	Presentation, Goal Setting, Dataset and Metric Research

Link to Gnatt Chart

9. References

1. Lai, Songning, et al. “Multimodal Sentiment Analysis: A Survey.” arXiv.org, 12 May 2023, arxiv.org/abs/2305.07611.
2. E. Ioanes, “Instagram’s Teen Accounts aren’t really for teens,” Vox, Sep. 18, 2024.
3. Instagram. “New Instagram Teen Accounts: Default Settings for Teen Safety | about Instagram.” Instagram.com, 2022
4. Hammad J. (2023). 6992 Meme Images Dataset. Available on Kaggle: https://www.kaggle.com/datasets/hammadjavaid/6992-labeled-meme-images-dataset
5. PyTorch. ResNet50 Documentation. Available at: https://pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html
6. Hugging Face. BERT Base Uncased. Available at: https://huggingface.co/google-bert/bert-base-uncased
7. Scikit-learn. SVM Documentation. Available at: https://scikit-learn.org/stable/modules/svm.html