Project Final Report

12-03-2024

Introduction

Summary:

We are analyzing emoji trends in social media using ML, based on a dataset of emoji usage by demographic and synthetic LLM-generated data. This proposal outlines literature, our problem and motivation, our approach to solving it, and discusses key project goals.

Literature Review:

The use of ML to analyze emoji sentiment trends is an emerging area aiming for accurate emotional interpretation of online communication, as emojis are an important source of nuance in communications. Several ML approaches are frequently used for emoji sentiment tasks. Supervised models like SVM and Naive Bayes can effectively classify emotions linked to emojis within their context (Yurtoz & Parlak, 2019). Neural nets are also gaining traction for emoji-based sentiment in more complicated linguistic environments (Pratibha et al., 2024). Literature also reviews using emojis to enhance the emotional expressiveness of artificial agents (Santamaria-Bonfil & Lopez, 2019). Challenges remain in improving detection of sarcasm or ambiguous emotions, as discussed by Shedthi and Shetty (2024), highlighting the need for further research into emojis in sentiment analysis.

Dataset Description:

Our dataset contains 4,000 uses of emojis by users across several platforms and demographic groups. There are five columns: the emoji itself, the platform (i.e. Snapchat or Instagram), user age, user gender, and the emotional context of the user. There are 25 emojis, each comprising approximately ~4% of the dataset.

The dataset is available on Kaggle.

Problem Definition

Motivation:

40% of Americans state their messages feel empty without emojis (Chaney, 2024), making emojis crucial to deciphering meaning. However, this meaning varies across different contexts.

For example, 💀 represents danger in plain conversation, whereas to younger generations it expresses something is funny or odd.

Problem:

Our problem is analyzing emoji usage in social media posts to predict the context associated with various emojis based on their user demographics, and predict what emoji a user in a demographic will use in a context.

Methods

Data Preprocessing:

Our proposed data preprocessing steps included 1) assigning emoji sentiment scores with an LLM 2) encoding categorical variables into numerical values and 3) split train and test samples for our supervised models. We executed upon these as steps as follows:

1. For each of 5000 samples in our dataset, we asked GPT to give us three scores ranging from 1 to 10: a positivity score, an engagement score, and a relevance score. Samples where GPT failed to produce a numerical output were rerun.
2. We attempted two different encodings for our categorical variables. First was get_dummies() in the pandas library, which converts each categorical feature into a set of binary features. The other was one-hot encoding from the SciKit learn library. The performance was similar as discussed later.
3. We split the data into testing and training samples with the appropriate sklearn library, with the knowledge that we will use cross val score to automatically do splitting for validation purposes at a later time.

Our clustering step, which will be discussed in the next subsection, may also be interpreted as a form of preprocessing, as it is used to engineer a new feature for our random forest emoji prediction model.

Models:

After preprocessing, we:

1. Clustered data with K-prototypes to segment users into demographic groups with high similarity based on age, gender, and platform. This is a deviation from our

original plan to use k-means. We had overlooked that k-means is designed for continuous features whereas our features are discrete/categorical. This is our only unsupervised model.

2. Used random forests to predict emojis based on the user segment, context, and sentiment scores. We picked this approach because we believe it might model the decision process a human may follow in picking what emoji to pair with a text, looking through different factors and narrowing down possibilities until reaching their choice. This is our first supervised model.
3. Used gradient boosting to predict the context given the emoji, demographics, and sentiment. We picked this model due to consistently high performance in literature, excellence at classification tasks very broadly, and sophisticated library support for tuning and optimization of the model. With a combination of bagging decision trees in the previous approach with boosting decision trees with this model, we hoped to gain some more insight into the tradeoffs between the models. This is our second supervised model.

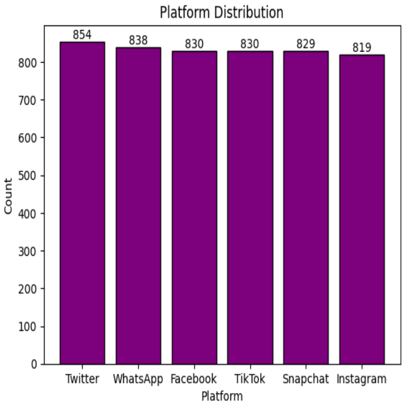
Code and Dataset:

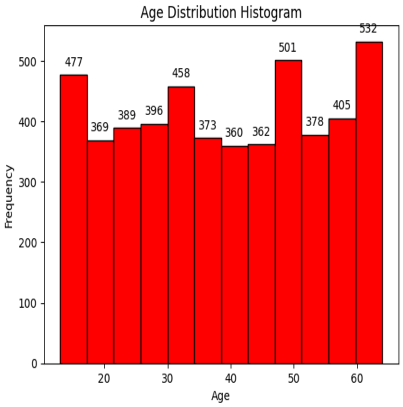
The code and preprocessed dataset can be found on the github repository: https://github.gatech.edu/nduggal3/emojipt.github.io/tree/main

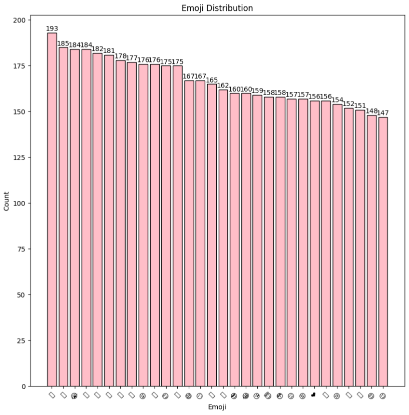
Results and Discussion

Basic EDA:

In order to understand our data, we prepared a few histograms regarding the distribution of our data. The gender, age, and platform distributions are relatively even. We do not see a need to do further processing to reduce imbalances in data, such as SMOTE.

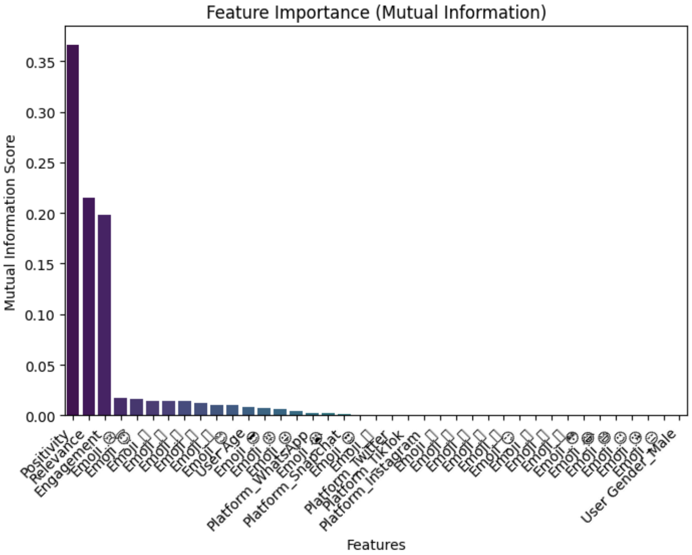


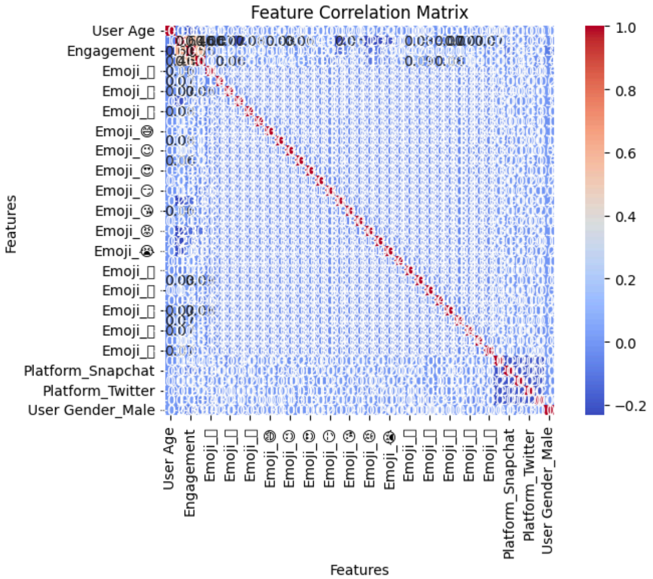




Preprocessing:

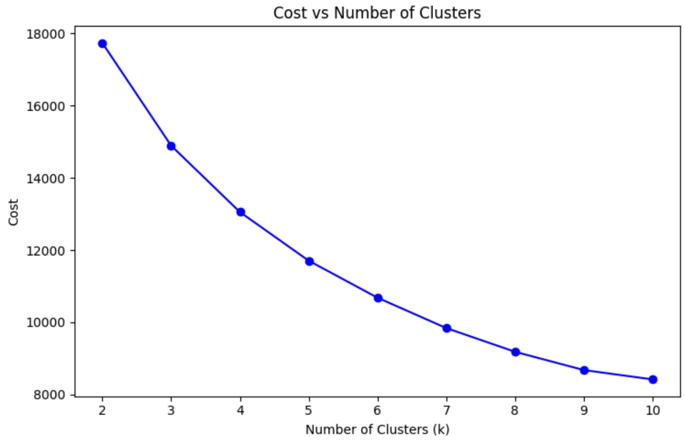
The interesting component of our preprocessing was using chatGPT to assign positivity, relevance, and engagement scores to the data. Each was an integer taking values between 1 and 10, and resulted in 15,000 total API calls. 13 values were not filled successfully by GPT, and those were handled with median setting, as they are low in volume. The scores generated from GPT exhibit high (mutual) information when ranked against all other features.

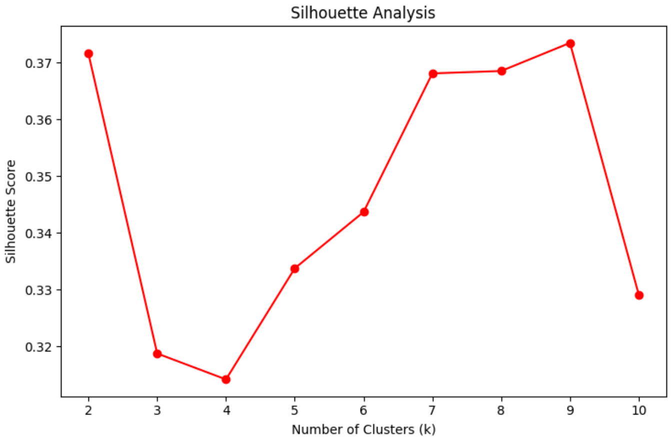




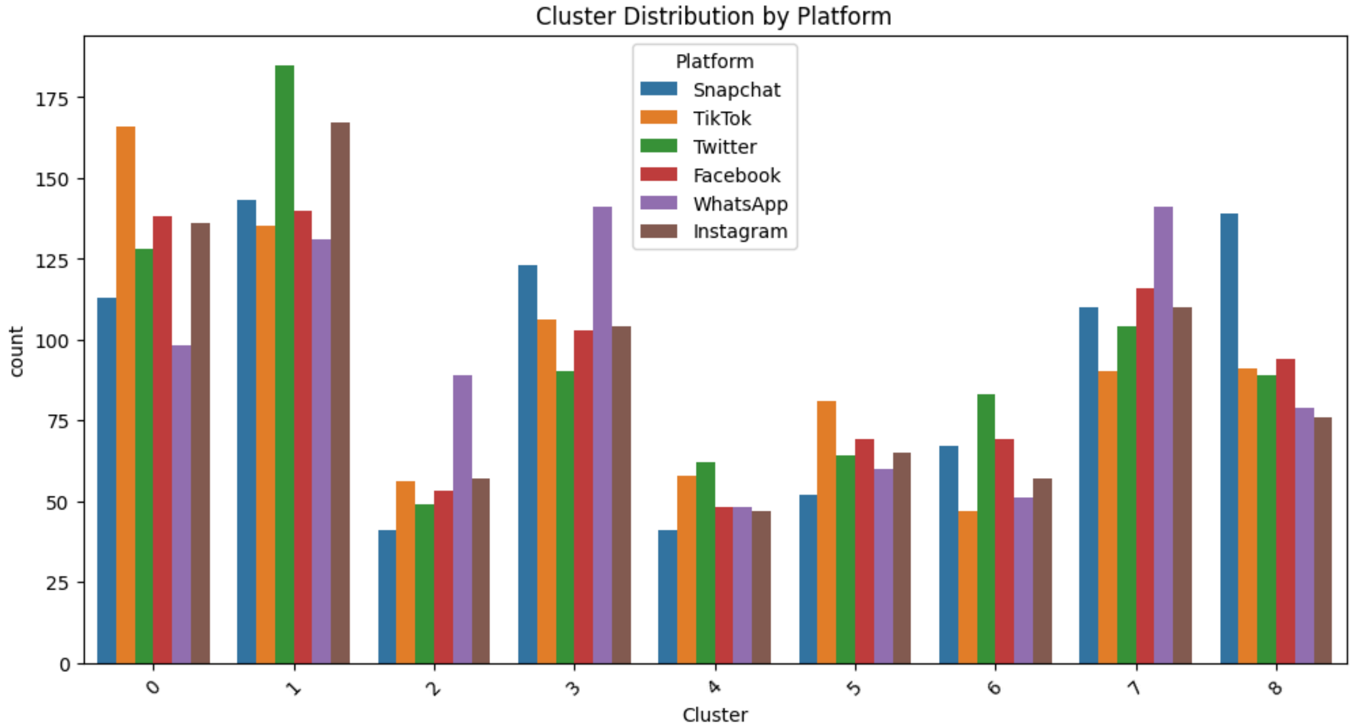
Models:

K-prototypes: we experimented with clustering our demographic features with various numbers of clusters. In order to determine whether our clustering was successful, we used the silhouette coefficient. Silhouette coefficient is a quantitative metric that measures the quality of clustering by evaluating how similar data points are to their own cluster compared to other clusters, with higher values indicating well-defined, cohesive clusters. Below, you can observe that silhouette coefficient is maximized with nine clusters, which is consistent with the plot to the right.

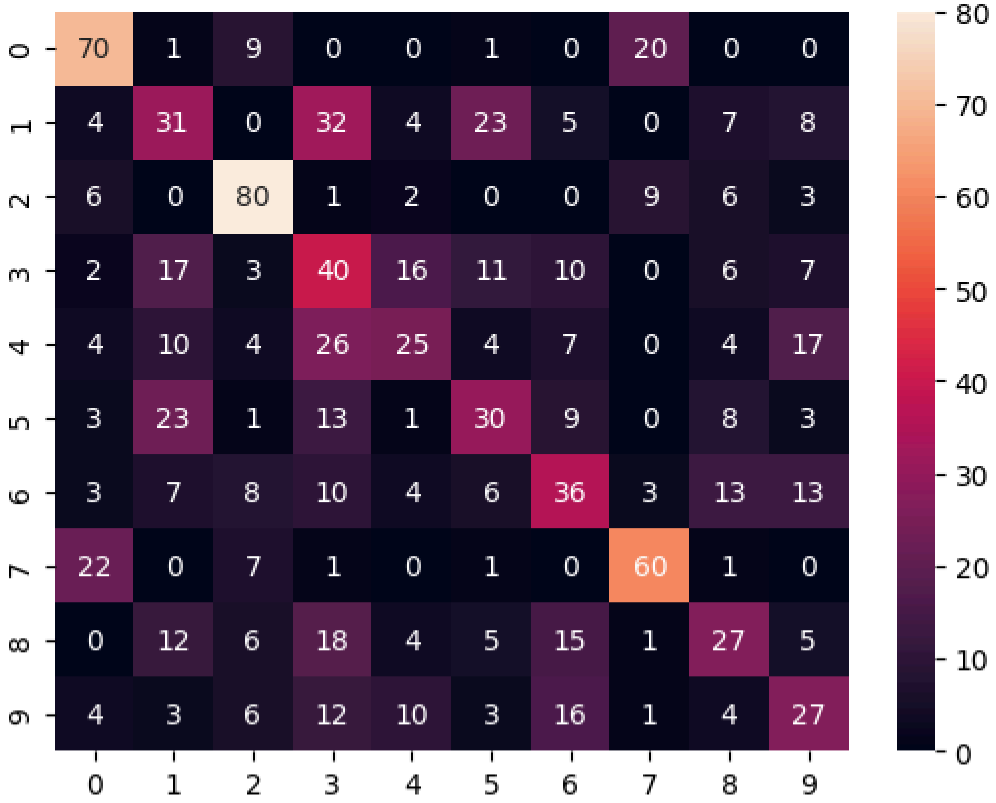




Interestingly, there was not a very explainable dimension for our clusters.They seemed to have a relative balance across platforms, perhaps indicating that platform is not as important of a feature as thought about previously. In our qualitative interpretation, this is because the variance in platform is only relevant to the extent that it captures other demographic information, such as age.



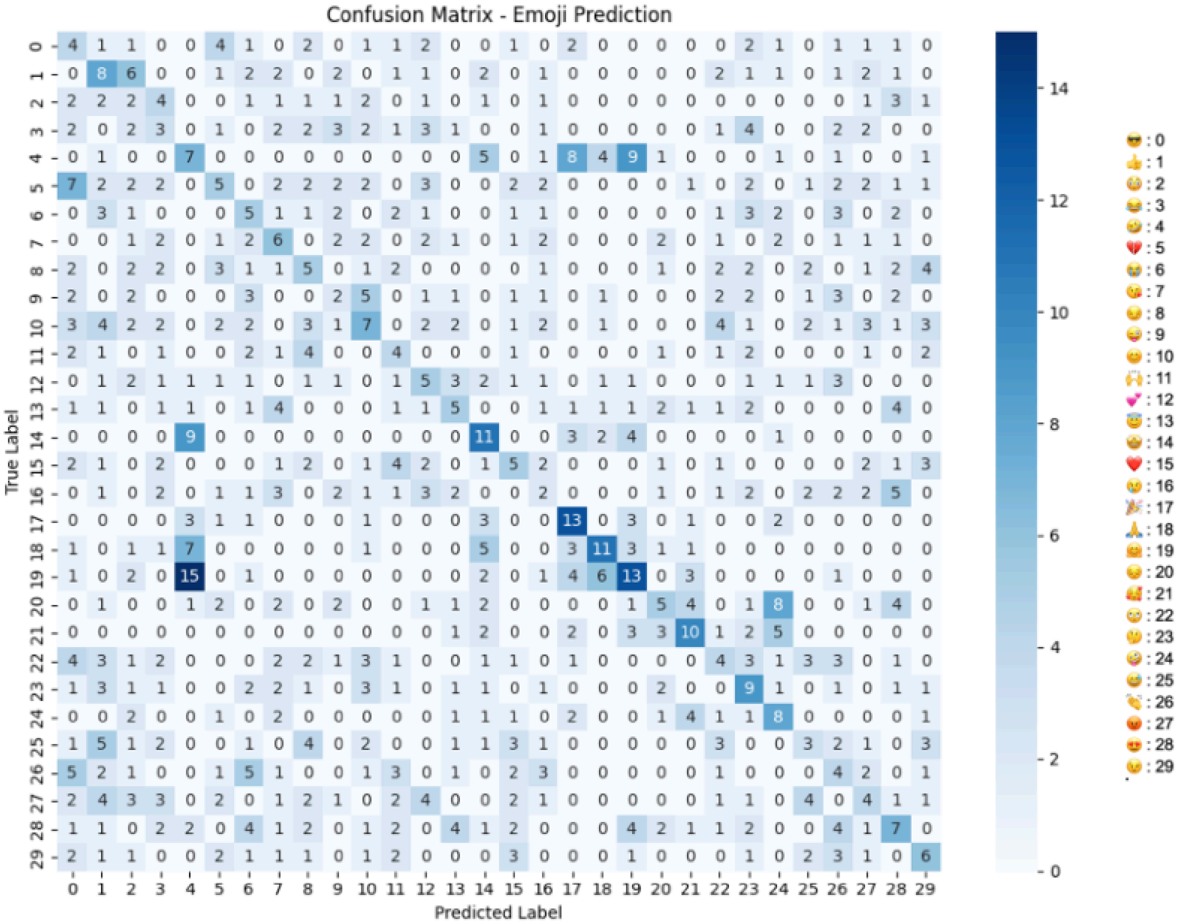
Gradient Boosting Classifier: This was trained to determine the context given the emoji and demographic related features, including the cluster features created above. We encoded our categorical features using binary 1/0 and then trained a classifier. With 5-fold cross validation, our classifier was predicting the correct context 48% of the time. With 12 classes, a classifier achieving 0.48 accuracy significantly outperforms random guessing (0.083) but indicates moderate predictive ability with room for improvement, especially if misclassifications or class imbalances persist.



	0	1	2	3	4	5	6	7	8	9
0	70	1	9	0	0	1	0	20	0	0
1	4	31	0	32	4	23	5	0	7	8
2	6	0	80	1	2	0	0	9	6	3
3	2	17	3	40	16	11	10	0	6	7
4	4	10	4	26	25	4	7	0	4	17
5	3	23	1	13	1	30	9	0	8	3
6	3	7	8	10	4	6	36	3	13	13
7	22	0	7	1	0	1	0	60	1	0
8	0	12	6	18	4	5	15	1	27	5
9	4	3	6	12	10	3	16	1	4	27

Interestingly, the most mispredicted features are 1 (corresponding to Anger) and 3 (corresponding to Happiness). We think that these are getting mispredicted because the model is not understanding satire or sarcasm in the emoji usage, for example blowing kisses after an angry message.

We originally planned to use a random forest model to predict emojis from a context. Upon attempting this, we determined this form of prediction not easily feasible, though it may be more applicable to generative models. Our random forest model achieved a maximum F1-score of .18 even after several optimizations.

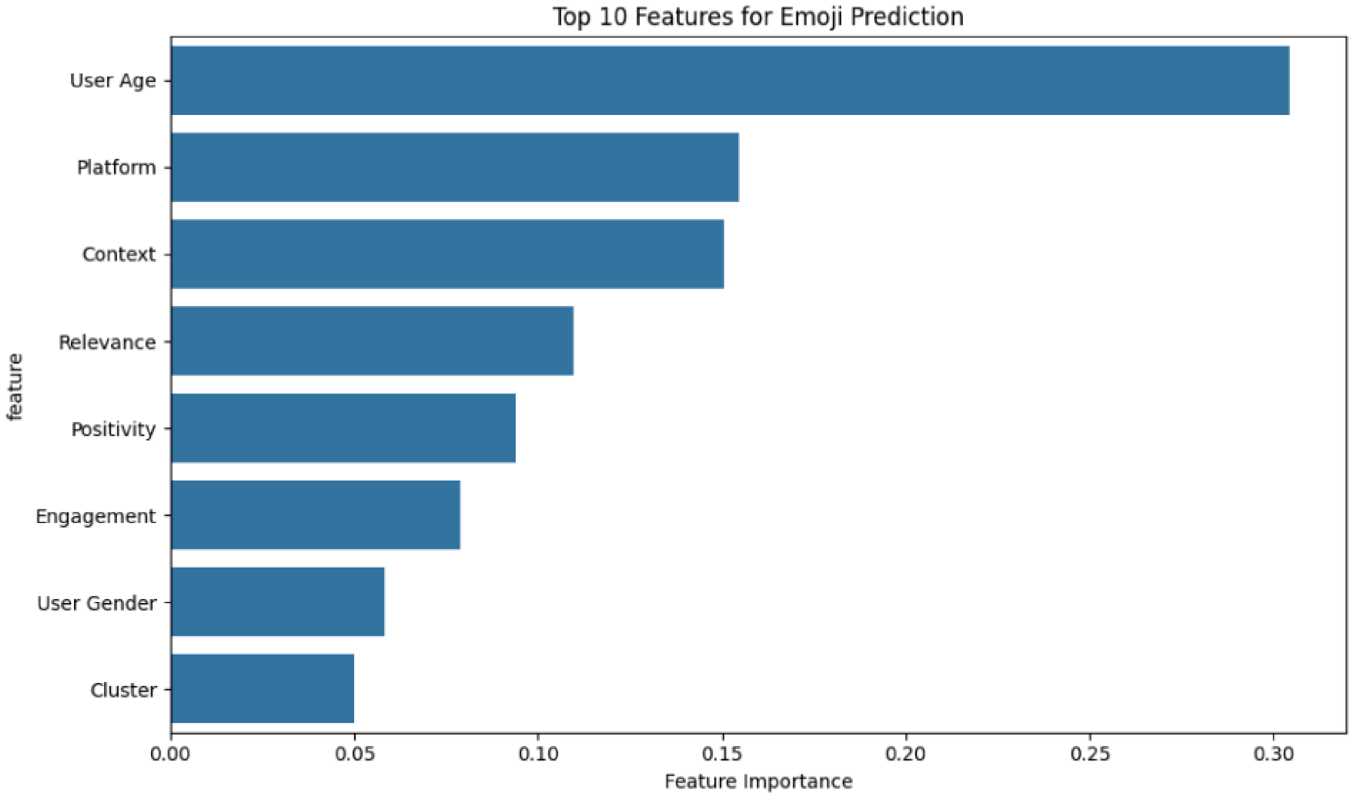


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	4	1	1	0	0	4	1	0	2	0	1	1	2	0	0	1	0	2	0	0	0	0	0	2	1	0	1	1	1	0
1	0	8	6	0	0	1	2	2	0	2	0	1	1	0	2	0	1	0	0	0	0	0	2	1	1	0	1	2	1	0
2	2	2	2	4	0	0	1	1	1	1	2	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	3	1
3	2	0	2	3	0	1	0	2	2	3	2	1	3	1	0	0	1	0	0	0	0	0	1	4	0	0	2	2	0	0
4	0	1	0	0	7	0	0	0	0	0	0	0	0	0	5	0	1	8	4	9	1	0	0	0	1	0	1	0	0	1
5	7	2	2	2	0	5	0	2	2	2	2	0	3	0	0	2	2	0	0	0	0	1	0	2	0	1	2	2	1	1
6	0	3	1	0	0	0	5	1	1	2	0	2	1	0	0	1	1	0	0	0	0	0	1	3	2	0	3	0	2	0
7	0	0	1	2	0	1	2	6	0	2	2	0	2	1	0	1	2	0	0	0	2	0	1	0	2	0	1	1	1	0
8	2	0	2	2	0	3	1	1	5	0	1	2	0	0	0	0	1	0	0	0	1	0	2	2	0	2	0	1	2	4
9	2	0	2	0	0	0	3	0	0	2	5	0	1	1	0	1	1	0	1	0	0	0	2	2	0	1	3	0	2	0
10	3	4	2	2	0	2	2	0	3	1	7	0	2	2	0	1	2	0	1	0	0	0	4	1	0	2	1	3	1	3
11	2	1	0	1	0	0	2	1	4	0	0	4	0	0	0	1	0	0	0	0	1	0	1	2	0	0	0	1	0	2
12	0	1	2	1	1	1	1	0	1	1	0	1	5	3	2	1	1	0	1	1	0	0	0	1	1	1	3	0	0	0
13	1	1	0	1	1	0	1	4	0	0	0	1	1	5	0	0	1	1	1	1	2	1	1	2	0	0	0	0	4	0
14	0	0	0	0	9	0	0	0	0	0	0	0	0	0	11	0	0	3	2	4	0	0	0	0	1	0	0	0	0	0
15	2	1	0	2	0	0	0	1	2	0	1	4	2	0	1	5	2	0	0	0	1	0	1	0	0	0	0	2	1	3
16	0	1	0	2	0	1	1	3	0	2	1	1	3	2	0	0	2	0	0	0	1	0	1	2	0	2	2	2	5	0
17	0	0	0	0	3	1	1	0	0	0	1	0	0	0	3	0	0	13	0	3	0	1	0	0	2	0	0	0	0	0
18	1	0	1	1	7	0	0	0	0	0	1	0	0	0	5	0	0	3	11	3	1	1	0	0	0	0	0	0	0	0
19	1	0	2	0	15	0	1	0	0	0	0	0	0	0	2	0	1	4	6	13	0	3	0	0	0	0	1	0	0	0
20	0	1	0	0	1	2	0	2	0	2	0	0	1	1	2	0	0	0	0	1	5	4	0	1	8	0	0	1	4	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	2	0	3	3	10	1	2	5	0	0	0	0	0
22	4	3	1	2	0	0	0	2	2	1	3	1	0	0	1	1	0	1	0	0	0	0	4	3	1	3	3	0	1	0
23	1	3	1	1	0	0	2	2	1	0	3	1	0	1	1	0	1	0	0	0	2	0	0	9	1	0	1	0	1	1
24	0	0	2	0	0	1	0	2	0	0	0	0	0	1	1	1	0	2	0	0	1	4	1	1	8	0	0	0	0	1
25	1	5	1	2	0	0	1	0	4	0	2	0	0	1	1	3	1	0	0	0	0	0	3	0	0	3	2	1	0	3
26	5	2	1	0	0	1	5	1	0	0	1	3	0	1	0	2	3	0	0	0	0	0	1	0	0	0	4	2	0	1
27	2	4	3	3	0	2	0	1	2	1	0	2	4	0	0	2	1	0	0	0	0	0	1	1	0	4	0	4	1	1
28	1	1	0	2	2	0	4	1	2	0	1	2	0	4	1	2	0	0	0	4	2	1	1	2	0	0	4	1	7	0
29	2	1	1	0	0	2	1	1	1	0	1	2	0	0	0	3	0	0	0	1	0	0	0	1	0	2	3	1	0	6

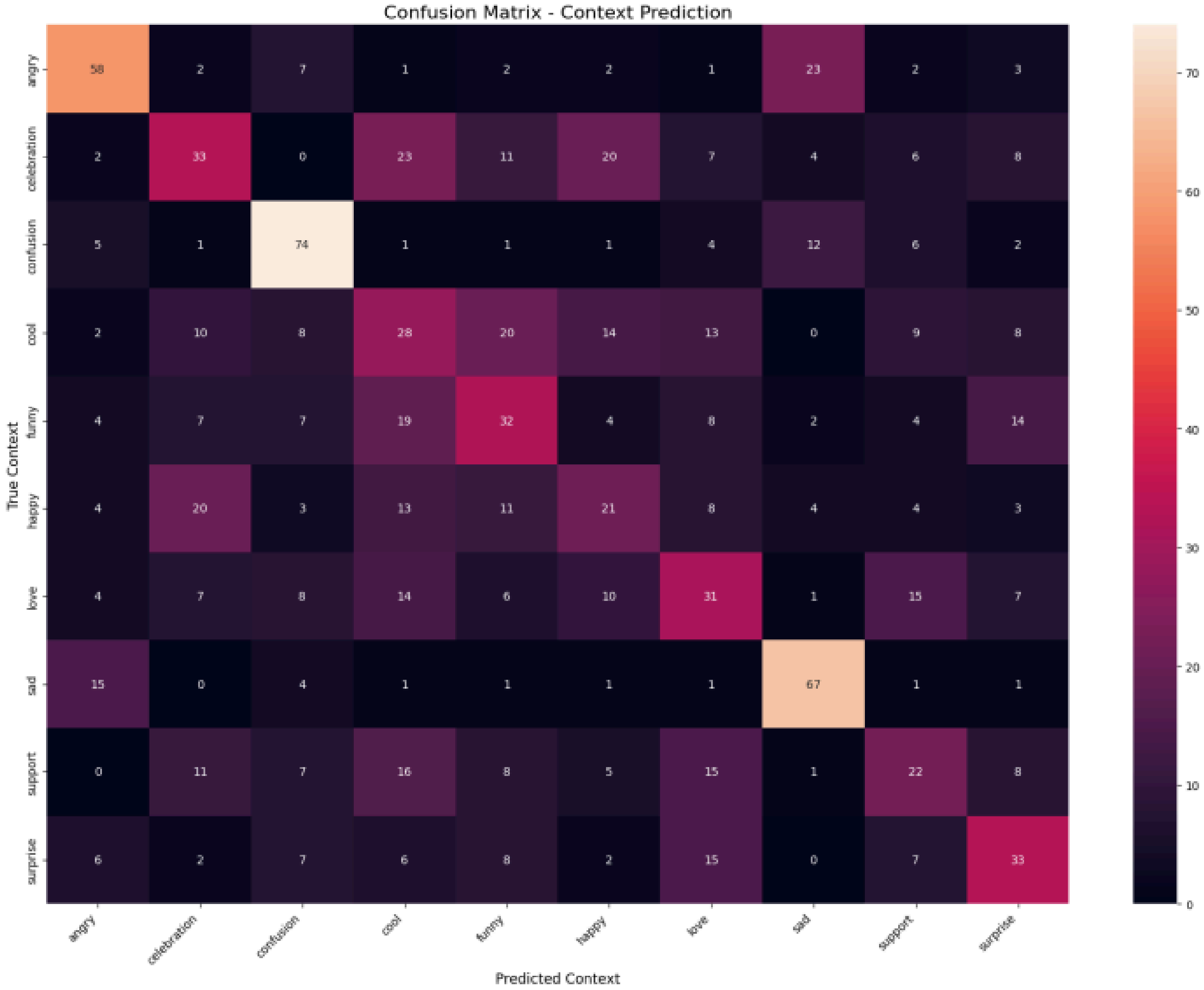
The difficulty in predicting emojis from context may stem from a poor selection of emoji-context pairs in the dataset, with several samples that completely overlapped. We also suffered from the curse of dimensionality - the approximately 150 samples per emoji that we had were not enough to provide coverage of the thousands of possible combinations of features. This may also come from noise in other uncontrolled factors like culture, and the more nuanced context than the categories in the data captured.

Because of this, and to provide a more concrete basis for model comparison, we instead chose to compare the accuracy of a random forest model and our original gradient boosting model on predicting the context from a given emoji.

Random Forest Classifier: determine the context given the emoji and demographic related features, including the clusters produced in our preprocessing step. We compared performance with using clusters as a feature, and without. We found that the model performed better without clustering as a pre-processing step, as measured by F1-score, the harmonic mean of precision and recall. Our analysis indicates that without clustering, our model had 11.7% higher precision and 5.4% higher F1 score. Indeed, clustering does not seem to be an important feature for our random forest model, as indicated by further feature-importance analysis.



Random forest achieved an accuracy of 0.40. We have similarly constructed a confusion matrix to identify misclassifications:



	angry	celebration	confusion	cool	funny	happy	love	sad	support	surprise
angry	58	2	7	1	2	2	1	23	2	3
celebration	2	33	0	23	11	20	7	4	6	8
confusion	5	1	74	1	1	1	4	12	6	2
cool	2	10	8	28	20	14	13	0	9	8
funny	4	7	7	19	32	4	8	2	4	14
happy	4	20	3	13	11	21	8	4	4	3
love	4	7	8	14	6	10	31	1	15	7
sad	15	0	4	1	1	1	1	67	1	1
support	0	11	7	16	8	5	15	1	22	8
surprise	6	2	7	6	8	2	15	0	7	33

Overall, the same context pairs were confused by both the gradient boosting classifier and the random forest classifier. However, there does exist slightly more noise across the contexts of the random forest, particularly between contexts of positive/negative connotation. For example, “cool”, “funny”, “happy”, “love”, and “celebration” were confused slightly moreso by the random forest than by the gradient boosting. This accounts for a small but significant drop in accuracy.

Comparison:

The three models that we trained demonstrate distinct trade-offs between interpretability, computational efficiency, and performance. K-prototypes, as an unsupervised method, offered exploratory insights but cannot directly predict outcomes. Its success is measured qualitatively and via metrics like the silhouette coefficient, though its clusters were not clearly interpretable, as explained prior. Computational demands grow with the number of clusters, but it proved valuable in preprocessing by capturing demographic relationships. Gradient boosting achieved 48% accuracy in predicting context with 5-fold cross-validation, but its computational cost is high due to iterative training. Misclassifications, particularly between anger and happiness, suggest difficulty capturing subtleties like sarcasm in emoji usage. Random forest, with and without augmented with K-prototype clusters, was outperformed by gradient boosting in classification (e.g., F1 score of 0.40 vs 0.48). While slightly less precise than gradient boosting, random forest achieved a better balance between precision and recall with manageable computational complexity. Overall, the choice of supervised model depends on the specific goals of efficiency, accuracy, and interpretability.

Next Steps:

We believe that model performance could be further improved with hyperparameter tuning and more sophisticated feature engineering. Furthermore, we could increase the scope of our project by benchmarking other models, like support vector machines, logistic regression, or deep learning approaches.

Significance:

Understanding emoji usage and sentiment provides insights for marketers, media platforms, and communication researchers. It also enables AI to more effectively communicate with youth. However, we should consider the risk of emotional persuasion of minors, which can be mitigated with legal and technological safe-guards.

References

[1] Pratibha, A. Kaur, and M. Khurana, “Multimodal Sentiments: Unraveling Text and Emoji Dynamics Through Deep Learning,” pp. 1–6, Mar. 2024, doi: https://doi.org/10.1109/icrito61523.2024.10522265.

[2] G. Santamaría-Bonfil and O. G. T. López, Emoji as a Proxy of Emotional Communication. IntechOpen, 2019. Available: https://www.intechopen.com/chapters/69271

[3] Cagatay Unal Yurtoz and Ismail Burak Parlak, “Measuring the effects of emojis on Turkish context in sentiment analysis,” IEEE Explore, no. Vol 17, Jun. 2019, doi: https://doi.org/10.1109/isdfs.2019.8757554.

[4] Shabari Shedthi B and V. Shetty, “Role of machine learning in sentiment analysis: trends, challenges, and future directions,” Elsevier eBooks, pp. 1–21, Jan. 2024, doi: https://doi.org/10.1016/b978-0-443-22009-8.00011-2.

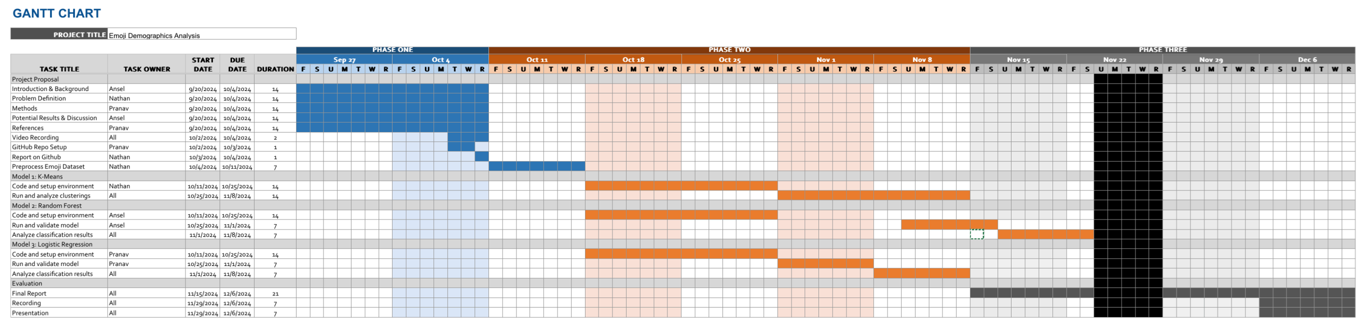
[5] K. Chaney, “Do Your Text Messages Really Need an Emoji??? - YR Media,” YR Media, Jul. 16, 2024. https://yr.media/tech/text-messages-emoji-gen-z/

Timeline and Gantt Chart

We will distribute work as illustrated in Figure 1, according to the timeline in Figure 2.

Task	Owner
Introduction	Ansel
Problem	Nathan
Methods	Pranav
Results and Discussion	Ansel
References	Pranav
Github	Pranav
Video	Nathan, Pranav, Ansel
Report on Github	Nathan

Figure 1: Work Distribution Table



GANTT CHART

PROJECT TITLE Emoji Demographics Analysis

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
Project Proposal				
Introduction & Background	Ansel	9/20/2024	10/4/2024	14
Problem Definition	Nathan	9/20/2024	10/4/2024	14
Methods	Pranav	9/20/2024	10/4/2024	14
Potential Results & Discussion	Ansel	9/20/2024	10/4/2024	14
References	Pranav	9/20/2024	10/4/2024	14
Video Recording	All	10/2/2024	10/4/2024	2
GitHub Repo Setup	Pranav	10/2/2024	10/3/2024	1
Report on Github	Nathan	10/3/2024	10/4/2024	1
Preprocess Emoji Dataset	Nathan	10/4/2024	10/11/2024	7
Model 1: K-Means				
Code and setup environment	Nathan	10/11/2024	10/25/2024	14
Run and analyze clusterings	All	10/25/2024	11/8/2024	14
Model 2: Random Forest				
Code and setup environment	Ansel	10/11/2024	10/25/2024	14
Run and validate model	Ansel	10/25/2024	11/1/2024	7
Analyze classification results	All	11/1/2024	11/8/2024	7
Model 3: Logistic Regression				
Code and setup environment	Pranav	10/11/2024	10/25/2024	14
Run and validate model	Pranav	10/25/2024	11/1/2024	7
Analyze classification results	All	11/1/2024	11/8/2024	7
Evaluation				
Final Report	All	11/15/2024	12/6/2024	21
Recording	All	11/29/2024	12/6/2024	7
Presentation	All	11/29/2024	12/6/2024	7

Figure 2: Gantt Chart

Nathan Duggal, Pranav Tadepalli, Ansel Erol

CS 7641 Group 65