Classifying Emotions from Tweets

Chiratidzo Sanyika, Tigist Tefera, Niru Shanbhag, Isabel Heard, &
Carol Jiang

Purpose

- We want to classify six different emotions from tweets:
  1. Sadness
  2. Joy
  3. Love
  4. Anger
  5. Fear
  6. Surprise
- Find tweets where the classifier was confident or uncertain.
- Analyzing the diverse spectrum of emotions expressed in short-form text on social media.

Workflow

01 Data Set Description

02 Data Cleaning

03 Baseline Model

04 Training Different Models

05 Testing Accuracy

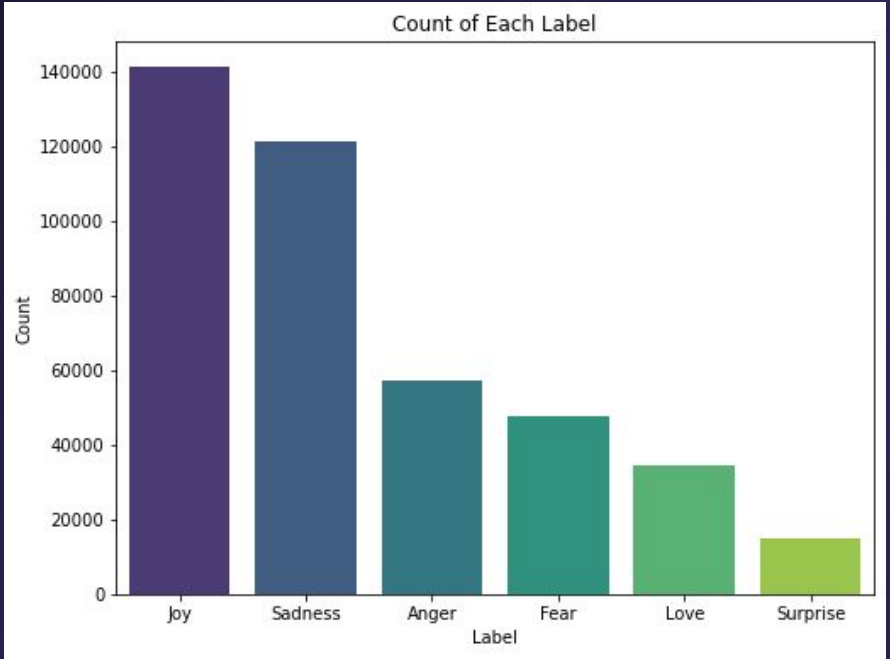
06 Examples & Conclusions

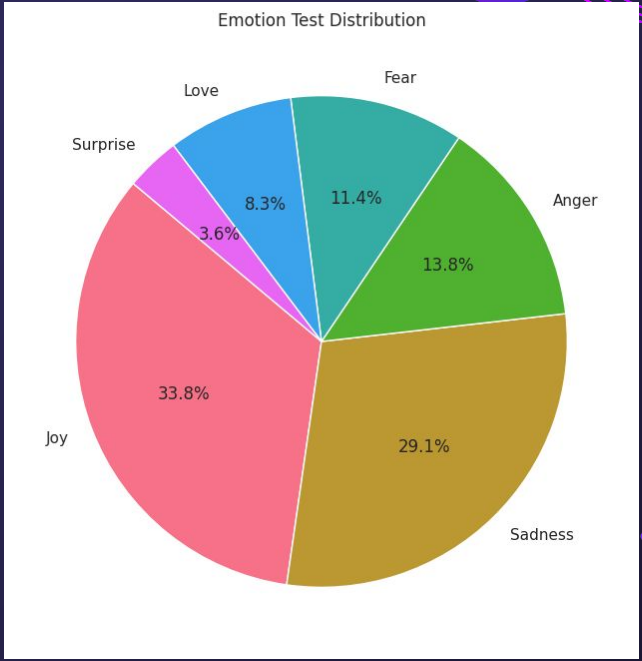
Data Set Description

- A collection of English Twitter messages
- 420K tweets in data set
- Unbalanced dataset
- No null values

Label	Text
Anger	feel like jerk library students claim love scrabble cant bothered participat...
Fear	feel really helpless heavy hearted
Joy	fear ever feel delicious excitement christmas eve least way remember
Love	would think whomever would lucky enough stay suite must feel like romantic p...
Sadness	ive enjoyed able slouch relax unwind frankly needed last weeks around end un...
Surprise	im forever taking time lie feel weird

Data Visualization

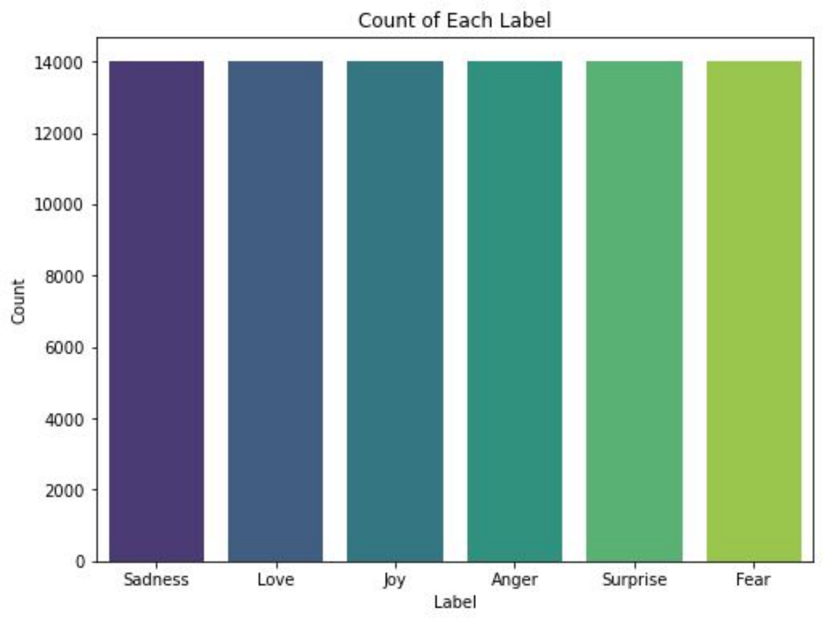


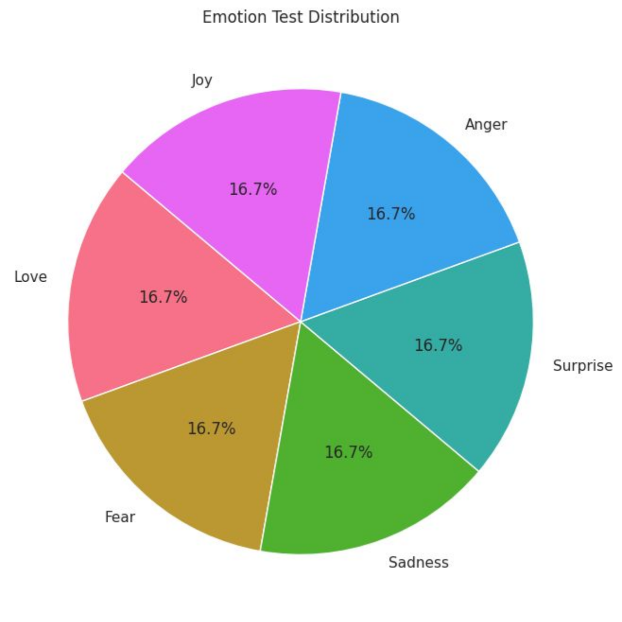


Data Cleaning

- Data Cleaning process:
  - Restructured the data so that we would replace ‘1’ with ‘Sadness’
  - Removed URLs
  - Removed extra white spaces
  - Removed special characters and punctuation
  - Made everything lowercase
  - Removed stop words
  - Removed Abbreviation words
  - Transfer emoji into regular text
- Get 14,000 rows from each emotion at random
- Split data into training (70%), development (15%), and test (15%).

Data Visualization – After Adjustment





Baseline Model

- Logistic Regression with no tuning:
  - Accuracy: 86%
- Naive Bayes with no tuning:
  - Accuracy: 85%

Logistic Regression Model A

- **Does not** include stop words, special characters, or punctuation.
- Best hyperparameters (tuned on development set):
  - C: 10,
  - Penalty: l2,
  - Solver: lbfgs,
  - Max_iter: 1000,
  - Class_weight: balanced
- Training accuracy: 89%
- Development accuracy: 87%
- Test accuracy: 86%
- Now trying with stop word, special characters, and punctuation to see if there is some improvement...

Logistic Regression Model B

- **Does include** stop words, special characters and punctuation
- Best hyperparameters (tuned on development set):
  - 'C': 1,
  - 'penalty': 'l1',
  - 'solver': 'liblinear',
  - 'max_iter': 1000,
  - 'class_weight': None
- Development accuracy: from 91.26% to 92.06%
- Test accuracy: 92.48%

Naive Bayes

- **Does not** include stop words, special characters, or punctuation.
- Best hyperparameters (tuned on development set):
  - alpha: 0.5,
  - fit_prior: False
- Development accuracy: 85%
- Test accuracy: 85%
- Does not change from baseline model.

RNN

- **Does not** include stopwords, punctuation, urls, whitespaces
- Tokenizes reviews to only include top 10000 most frequent words
- Sequences padded to max length of review
- RNN model
  - Embedding Layer
    - Vocabulary length = # unique words in reviews
    - Input length = max length of reviews
    - Embedding length = size of output layer
  - Simple RNN
    - 128 units
    - drop out = 0.2, recurrent_dropout = 0.2
  - Dense Layer
    - 6 nodes
    - Softmax (probability distribution of multiple outcomes)
- RNN model uses sparse categorical cross entropy, adam optimizer
- RNN model fit on training, batch_size = 128, epochs = 20
  - Validation on development set
  - Early Stopping

Training
→ Accuracy = 0.9475
→ Loss = 0.1382

Testing
→ Accuracy = 0.9094
→ Loss = 0.3023

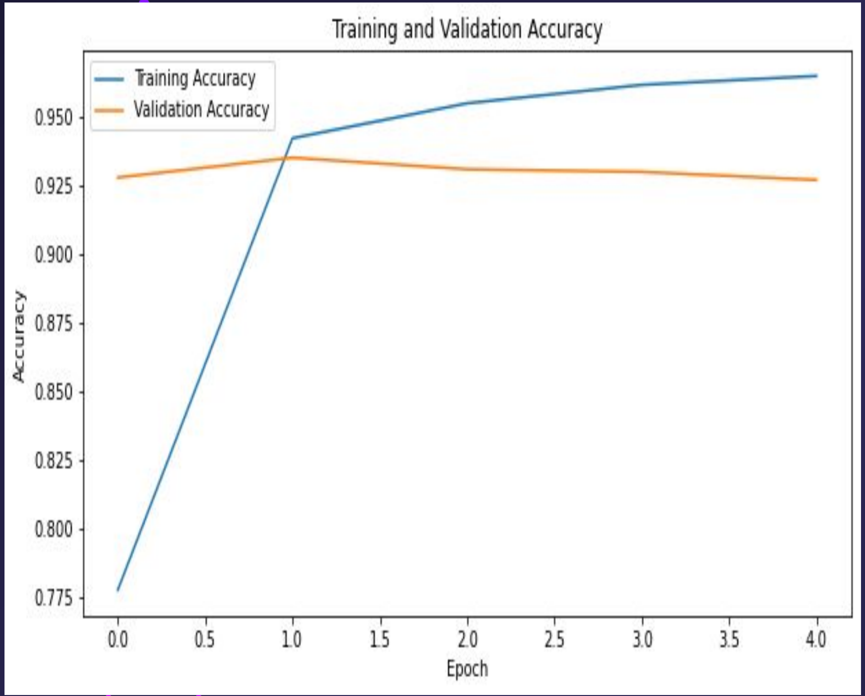
CNN

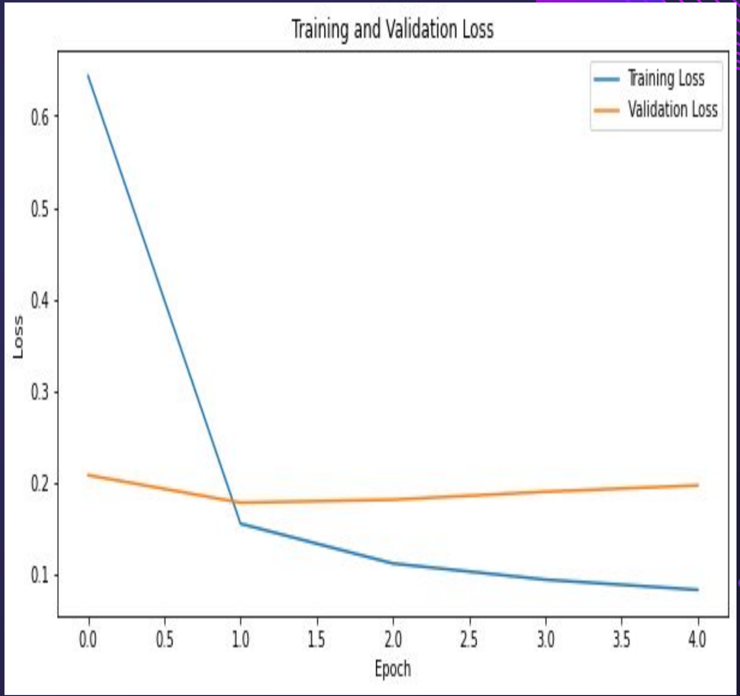
- **Does not** include stop words, special characters, or punctuation.
- includes embedding to convert text into dense vectors
- convolutional layers with 64 filters and a kernel size of 3 for feature extraction with ReLU activation
- A BatchNormalization layer is added after each convolutional layer to help in reducing internal covariate shift and speeds up training.
- GlobalMaxPooling1D layers are used
- batch_size = 256   epochs = 25
- Performance
  Training dataset: Loss: 0.11, Accuracy: 0.96, Precision: 0.96, Recall: 0.96
  Testing dataset: Loss: 0.34, Accuracy: 0.91, Precision: 0.91, Recall: 0.9

LSTM

- **Does not** include stop words, white space , special characters or punctuation. LOL & BRB replace with their expanded words
- Tokenized and pad sequences to ensure uniform length (max review)
- Embedding layer for word embedding (e dimension 100)
- LSTM layer with 128 units, for sequential data processing and dropout= 0.2, recurrent dropout = 0.2 to prevent overfitting
- Dense layer with softmax activation for multi class classification(6 class)
- Compile the model with categorical cross entropy loss and Adam optimizer
- Train the model with early stopping with validation loss
- Batch size 128, epoch 20
- **Performance**
  Training dataset = Accuracy : 0.94 - Loss: 0.1516
  Testing dataset = Accuracy: 0.9378- Loss: 0.1575

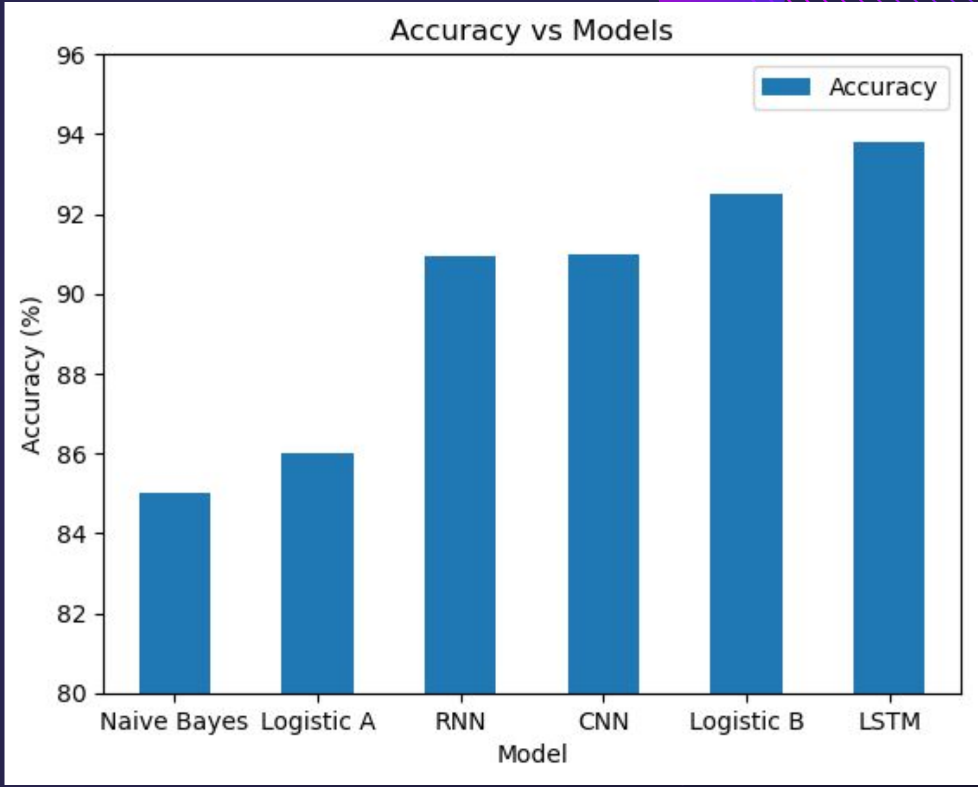
Accuracy and Loss Graph of LSTM





Comparison of Models

Model	Accuracy
Logistic A	86%
Logistic B	92.48%
Naive Bayes	85%
LSTM	93.78%
RNN	90.94%
CNN	91%



Examples with High Accuracy

```
Text: feel humiliated
Predicted Class: 5
Max Probability: 0.945466670372164
True Class: 5
```

```
Text: feeling threatened research paper
Predicted Class: 4
Max Probability: 0.9224259665925113
True Class: 4
```

```
Text: feel lucky friends already look forward seeing
Predicted Class: 2
Max Probability: 0.9251713290670955
True Class: 2
```

sadness (0), joy (1), love (2), anger (3), fear (4), and surprise (5).

Examples with Low Accuracy

```
Text: would left feeling needy
Predicted Class: 0
Max Probability: 0.21950173424534228
True Class: 3
```

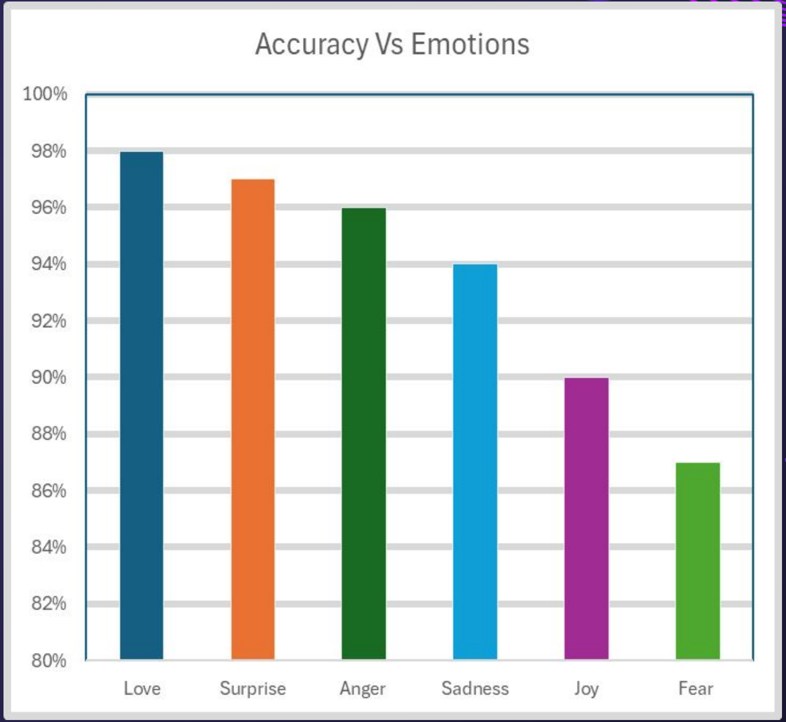
```
Text: feel excited
Predicted Class: 1
Max Probability: 0.36492171040112736
True Class: 5
```

```
Text: feel like stupid immature people high school could grow old together
Predicted Class: 0
Max Probability: 0.3311569496720529
True Class: 3
```

sadness (0), joy (1), love (2), anger (3), fear (4), and surprise (5).

Conclusion

- LSTM identified as an effective model in capturing text pattern and text dependency.
- Accuracy of 93.78%.
- Out of all 6 emotions, Love has highest accuracy.
- Distribution of sentiment expression and unique linguistic features use to distinguish each emotion categories.



Questions?

References

- Kaggle
- Github
- Emoji detection
- Common Text Abbreviations & Acronyms