Hands on Natural Language Processing Project

Md. Naim Hassan Saykat
Aloïs Vincent
Marija Brkic

February 21, 2025

1 Abstract

One of the most exciting challenges in the today's artificial intelligence community is to improve AI's ability to understand and recognize human emotions. Many believe that this capability is a key step towards more natural human-AI interactions. In this project, we are developing a couple of models for this task trained on the CARER dataset and comparing them to state-of-the-art models.

2 CARER Dataset

The CARER dataset was proposed in the paper "CARER: Contextualized Affect Representations for Emotion Recognition"[1][2]. The authors created a set of hashtags to collect a dataset of English tweets from the Twitter API. They used eight emotions: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. The hashtags (339 total) serve as noisy labels. To ensure data quality, they are considering the hashtag appearing in the last position of a tweet as the ground truth.

Emotions	Amount	Hashtags
sadness	214,454	#depressed, #grief
joy	167,027	#fun, #joy
fear	102,460	#fear, #worried
anger	102,289	#mad, #pissed
surprise	46,101	#strange, #surprise
trust	19,222	#hope, #secure
disgust	8,934	#awful, #eww
anticipation	3,975	#pumped, #ready

Figure 1: Dataset statistics[1]

In the previous image, we can see some of the hashtags used for data extraction. However, the dair-ai/emotion dataset, which is provided on Hugging Face[1], is an updated CARER dataset and contains six emotions: sadness, joy, love, anger, fear, surprise, and is used in this project.

It is important to mention that tweets are a very informal speech. That means that some of them have words that are not grammatically correct. For example, if someone is happy they would write 'whaaaat' instead of 'what', or users might use 'tnx' instead of 'thanks'. This special cases could be very useful in emotion detection.[2]

3 Benchmark models preview

Emotion detection problem has been solved by numerous models and on numerous datasets. The next two images show different solutions and their metrics[2]. The paper compares their solution, which is a semi-supervised, graph-based algorithm, with already existing solutions, but none of them is trained on this dataset. Some solutions are implementing traditional models, and some of them are deep-learning models[6][7].

Models	Features	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	F1 Avg.
BoW	**word frequency**	0.53	0.08	0.17	0.53	0.71	0.60	0.36	0.33	0.57
BoW$_{TF-IDF}$	**TF-IDF**	0.55	0.09	0.18	0.57	0.73	0.62	0.39	0.35	0.60
n-gram	**word frequency**	0.56	0.09	0.17	0.57	0.73	0.64	0.42	0.39	0.61
n-gram$_{TF-IDF}$	**TF-IDF**	0.58	0.12	0.17	0.60	**0.75**	0.67	0.47	0.45	0.63
char_ngram	**character frequency**	0.49	0.06	0.12	0.46	0.67	0.55	0.30	0.28	0.52
char_ngram$_{TF-IDF}$	**TF-IDF**	0.53	0.07	0.15	0.53	0.71	0.59	0.35	0.31	0.57
LIWC	**affective words**	0.35	0.03	0.11	0.30	0.49	0.35	0.18	0.19	0.35
CNN$_{w2v}$	**word embeddings**	0.57	0.10	0.15	0.63	**0.75**	0.64	0.61	**0.70**	0.65
EmoNet	**word embeddings**	0.36	0.00	0.00	0.46	0.69	0.61	0.13	0.25	0.52
DeepMoji	**word embeddings**	0.60	0.00	0.03	0.49	**0.75**	0.67	0.20	0.27	0.59
CNN$_{BASIC}$	**basic patterns**	0.65	0.10	0.22	0.64	0.73	0.56	0.15	0.08	0.52
CARER$_{\beta}$	**enriched patterns**‡	0.61	0.31	0.34	0.67	**0.75**	0.68	0.60	0.55	**0.67**
CARER	**enriched patterns**	**0.74**	**0.41**	**0.43**	**0.79**	**0.83**	**0.82**	**0.76**	**0.75**	**0.79**

Figure 2: Comparison of CARER model against various emotion recognition systems[2]

Model	Input	Epochs	Accuracy
RNN$_{w2v}$	word2vec (Mikolov et al., 2013)	24	0.53
CNN$_{char}$	character embeddings (end-to-end)	50	0.63
CNN$_{w2v}$	word vectors (Deriu et al., 2017)	33	**0.69**
EmoNet	word embeddings (end-to-end)	23	0.58
DeepMoji	word embeddings (end-to-end)	100	0.63
BiGRNN	our enriched patterns‡	12	**0.68**
CARER$_{\beta}$	our enriched patterns‡	12	**0.72**
CARER$_{EK}$	our enriched patterns	12	**0.81**

Figure 3: Comparison of CARER method against deep learning models[2]

As for the models trained on the CARER dataset, most of them are transformer-based models, such as BERT and distilBERT. The next table shows the accuracy of the original work and fine-tuned BERT models, and we can notice that the highest accuracy is obtained by using fine-tuned distilBERT-base-uncased model, which 0.94.

Table 1: Benchmark models on CARER dataset

Paper	Model	Biggest Accuracy
Saravia et al.[2]	Semi-supervised, graph-based algorithm	0.81
Wang et al.[3]	Fine-tuned BERT(transformer based)	0.93
Fengkai[4]	Fine-tuned distilBERT-base-uncased(transformer based)	0.94

We will try to implement our own Traditional Models, a custom Convolutional Neural Network, a fine-tuned BERT model and an Ensamble model, because studies show that combining multiple can improve generalization. In all cases, the features will be TF-IDF vectors.

4 Data analyzing, preprocessing and feature extraction

Once again, the used dataset has six different labels(emotions): sadness, joy, love, anger, fear, surprise. Each data-point is a line of text(a tweet) and an emotion label. The next image shows instances of data-points.

	text	label
0	i feel awful about it too because it s my job ...	0
1	im alone i feel awful	0
2	ive probably mentioned this before but i reall...	1
3	i was feeling a little low few days back	0
4	i beleive that i am much more sensitive to oth...	2

Figure 4: Data-point examples

We can notice that most of them have a similar format, and they mostly start with 'i' and sometimes with 'i feel'. It might be expected for this words to be very frequent, and not very informational. Furthermore, there is no punctuation in this dataset. That might be a lucky thing for the hardship of preprocessing, but it also makes predictions harder, because some punctuation could be very informal for emotions, such as '!'.

If we look at the distribution of number of tweets over classes, we get the next histogram:

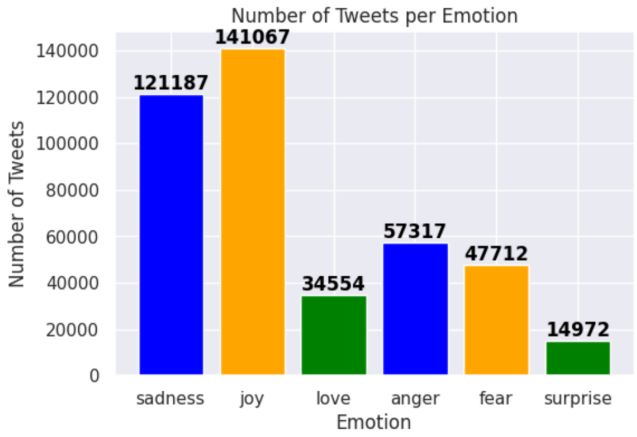

Figure 5: Distribution of number of tweets over classes

The dataset has 416809 instances. We can notice that the dataset is highly imbalanced. We have almost ten times more instances of joy than surprise. This might indicate that the largest classes will be dominant in classification and the smaller ones will be harder to detect.

Next thing we can observe the distribution of tweet lengths.

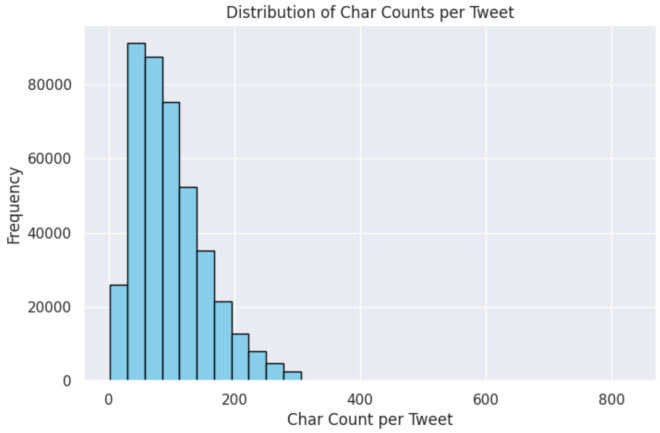

Figure 6: Distribution of tweet lengths

On the previous graph the mean is 97.03, standard deviation is 56.20, minimum value is 2.00 and maximum value is 830. The first percentile is 54, the median is 86 and the third percentile is 128. This distribution shows that most tweets have length less than 200, but it seems there are some outliers.
We can observe outliers a bit better if we look at separate distributions of tweet lengths over different emotions:

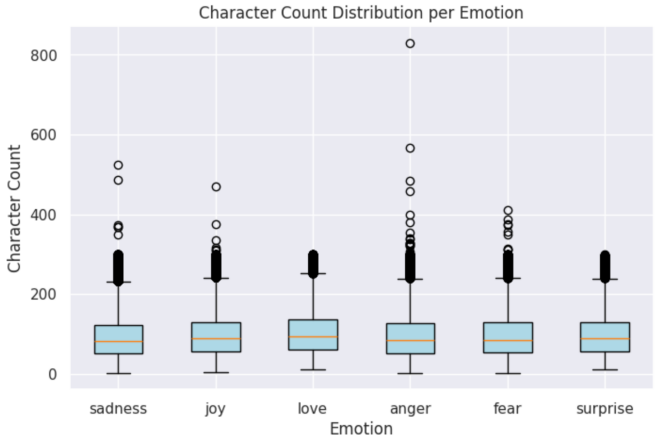

Figure 7: Distribution of tweet lengths for separate emotions

It is obvious that the class anger has one very prominent outlier, and there is only a few tweets with the length higher than 400 in the whole dataset. We will try to look at those tweets. Some of them are:

Tweet:
sadness
i have been thinking of changing my major for a few months my original major was chinese language and it blocks my way i have to face many problems at the sametime and i do not know what is real and virtuous i do not know if i am not interested in chinese or if there are other factors which make me sad i am not very clear on what i can do for the study of communication and journalism i am not clear about what i can do what are the limitations i hesitate but i feel more adjusted now

Tweet:
anger

i worked with several classmates on a project i was very anxious about the project while my partners showed no concern and when we had meetings on the project my classmates did not pay any attention some of them read books while the others argued on irrelevant questions the meeting would go on for two hours without the main theme being discussed we wasted time and could not reach a compromise my classmates avoided doing the work and the responsibilities

Tweet:

anger

a few days back i was waiting for the bus at the bus stop before getting into the bus i had prepared the exact amount of coins to pay for the bus fair and when i got into the bus i put these coins into the box meant to collect the bus fair i thought that i had paid and wanted to get inside however the bus driver called me and asked me in an impolite way if the coins were stuck at the opening of the box he had not seen me paying and there wasnt a stack of coins in the box i could not understand this and the driver kept questioning me he made me feel angry and at last i inserted a dollar coin in the box just to get away from him later i found that i had forgotten a few coins in my pocket and had not paid enough for the fair the first time after i had entered the bus i could still hear him scolding me and i felt disgusted

Since the length of a tweet might not be very influential in feeling prediction, but specific words(if we want to use N-Grams as features), long text lines could skew the visualization and the predictions. Furthermore, these tweets seem to be long because they are stories of certain situations. Therefore we decided that it would be reasonable to remove outliers. All texts that have length over 400 will be removed. The next graph shows separate distributions of tweet lengths over different emotions after removing outliers.

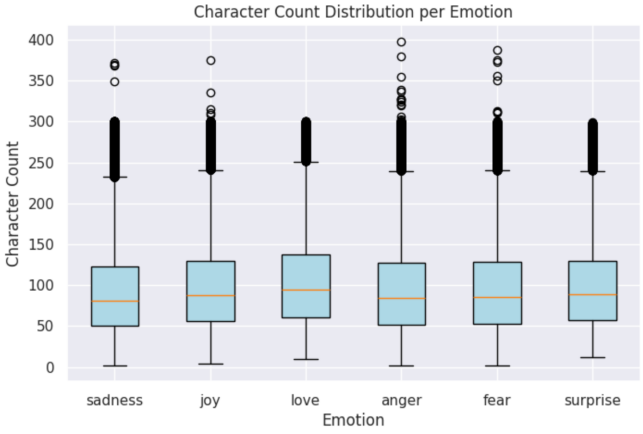

Figure 8: Distribution of tweet lengths for separate emotions

Next part we wanted to observe is tokenization. We used nltk.tokenize.word_tokenize[8] because it is separating punctuation as separate tokens, which would be very helpful in classification. However, it turns out that this dataset has no punctuation, which is why tokens are simply words. After that we observed vocabulary, and its lengths for different classes. Vocabulary length distribution could be shown on the next boxplot:

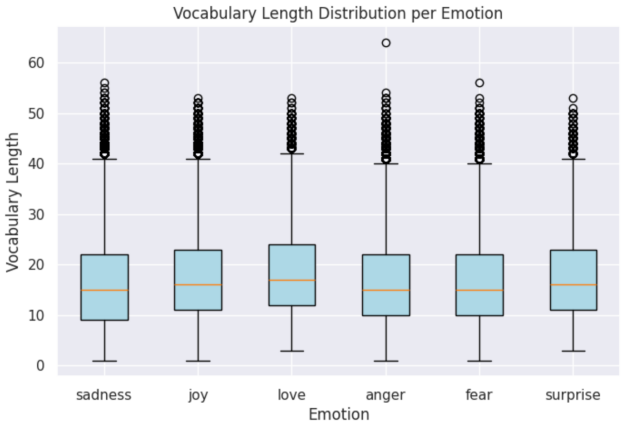

Figure 9: Vocabulary length distribution

We can notice that all classes mostly have the same length vocabulary. There is a slight outlier in the class anger. That means that there is one tweet with the largest number of unique tokens but the number is not significantly larger than the rest and we will not remove it from the dataset.
We can also observe the richness of vocabulary.

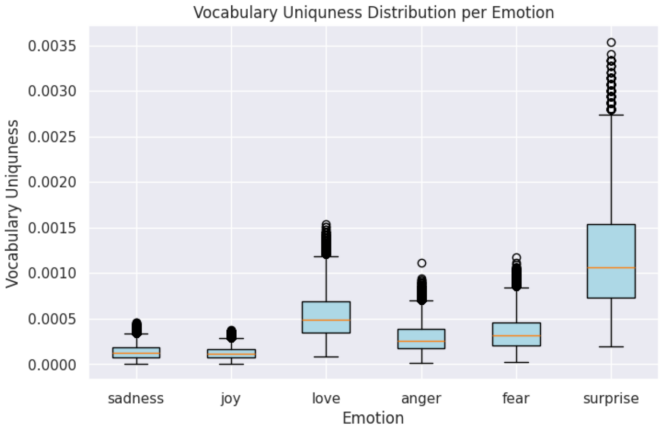

Figure 10: Richness of vocabulary

The previous boxplot shows the percentage of unique tokens in each tweet separated in classes. We can notice that 'surprise' has a significantly larger percentage of unique tokens, than the rest of the classes, which means it is by far the richest in the vocabulary, which absolutely makes sense because it is the smallest class.

Finally, we can observe frequency of tokens(words) in the whole dataset:

	token	frequency
0	i	676149
1	feel	289936
2	and	250251
3	to	233087
4	the	216591
...	...	...
75289	galleryimageborder	1
75290	danbo	1
75291	truc	1
75292	entrails	1
75293	usaully	1

Figure 11: Frequency of tokens

As expected, the most frequent words are 'i', 'feel', and stop words. This might indicate that we could remove stop words and use some vectorizing algorithm that will reduce the impact of very frequent words, since in this case they do not seem to be very informational about an emotion.

4.1 Vectorizing and visualization

We decided to use TF-IDF as a vectorizing algorithm[9]. TF-IDF vectorizer uses the next regular expression for tokenizing: '`(?u)\b\w\w+\b`'. This means that it is recognizing two or more word characters as tokens(A-Z, a-z, 0-9,_). Because of that, all one letter words, such as 'i' will not be considered tokens. After that, TF-IDF is counting appearances of tokens in text, but it is also dividing those numbers with the number of tweets the token appears in. Because of that, some very frequent words, such as 'feel', will lose importance, which is good because they are not very informative. On the other hand, some rare words will become more important features. This algorithm is balancing frequency of a token in the whole dataset and in the specific text line.

We used TF-IDF with and without removing stop words, which could be less informative, and with maximum number of 1000 features. Finally we applied Principal Component Analysis in order to visualize the data, and to hopefully extract some important features. PCA is a linear algorithm for dimensionality reduction and it is finding the components which maximize variance. However, this algorithm is unsupervised,

and therefore it is not separating classes, just looking for the most dispersed dimension, in order to find the most informative features.

The results we obtained are next:

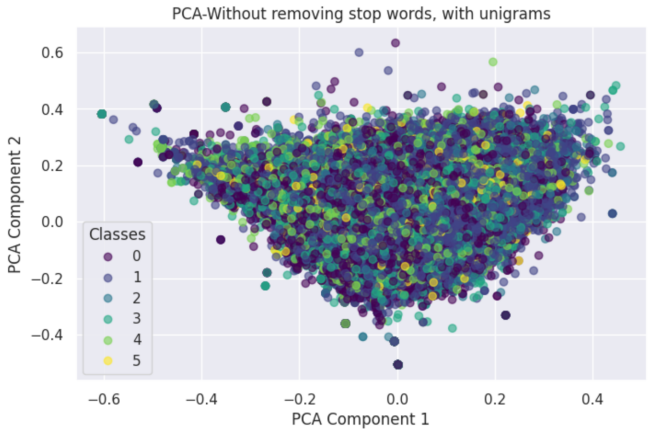

Figure 12: TF-IDF vectors with PCA without removing stop words

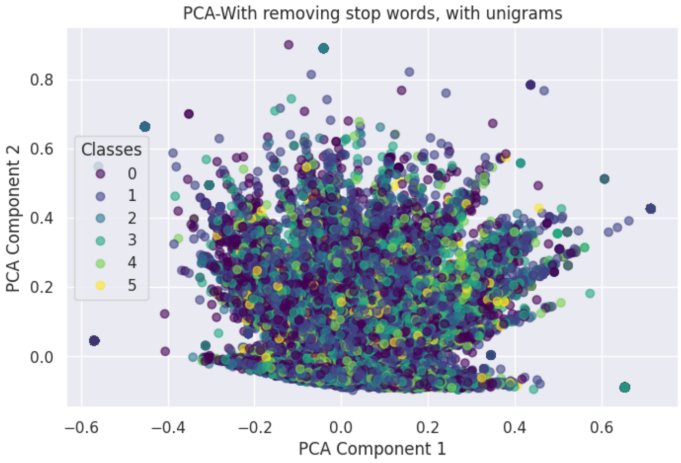

Figure 13: TF-IDF vectors with PCA without removing stop words

We can notice that in both cases classes are not separate at all. Perhaps after removing stop words we could see slight clustering of the class '0' on the left side and of classes '3' and '4' on the right, but this doesn't seem to be very promising for classification.

We tested some of the models with the data after applying TF-IDF with removing stop words and applied PCA with 100 components, but this gave us somewhat bad results. Therefore we finally decided on abandoning PCA altogether, and the final features are TF-IDF vectors with maximum length 5000, with removing stop words.

5 Models

We split the dataset into 80% training data, and 20% testing data, and then we split training data to validation and training set the same way. For some models, we also tried to balance classes by applying bigger class weights to smaller classes, and smaller class weights to bigger classes.

As evaluating metrics we observed accuracy, f1-score, precision and recall:

$$Recall = \frac{TP}{TP + FN}$$

$$Precission = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{N_{Correct}}{N_{Samples}}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Accuracy is not the best metric when we are working with unbalanced classes because it might give us false impression, therefore it is important to look at predictions for each class and their separate metrics.

5.1 Traditional approaches

For traditional approaches we opted for Logistic Regression(A simple linear classifier), Random Forest(An ensemble-based decision tree model)m and Support Vector Machine(A classifier using a linear kernel).

5.1.1 Support Vector Machine

Support vector machine classification report is the next one:

Classification Report for SVM:

	precision	recall	f1-score	support
sadness	0.93	0.92	0.93	581
joy	0.88	0.95	0.91	695
love	0.82	0.69	0.75	159
anger	0.89	0.88	0.88	275
fear	0.85	0.86	0.86	224
surprise	0.74	0.56	0.64	66
accuracy			0.89	2000
macro avg	0.85	0.81	0.83	2000
weighted avg	0.89	0.89	0.88	2000

Figure 14: Support Vector Machine classification report

We can notice that surprise has the smallest precission, recall and f1-score, because it is the smallest class, while sadness and joy have the biggest metrics, because they are the largest. Final accuracy was 89%
If we look at missclassified examples we can notice that predicted emotions are very similar to the true ones, such as sadness and fear:

```
Misclassified Examples (SVM Model):
                                                   text  label  \
103   i feel agitated with myself that i did not for...      4
1048  i wonder if the homeowners would feel weird if...      5
501   when we rearranged furniture in our flat and g...      3
1592        i have strong feelings about being faithful      2
575   i feel not having a generous spirit or a forgi...      2
625   i am feeling overwhelmed by trying to do it al...      5
1928  i feel inside cause life is like a game someti...      4
1296         i that it feels like she is being tortured      4
869   i feel like if people accepted that wed get al...      2
823   i dont remember how january was like last year...      3

      svm_predictions
103                 3
1048                4
501                 1
1592                1
575                 1
625                 4
1928                0
1296                3
869                 1
823                 0
```

Figure 15: Missclassified examples for SVM

Finally, we tested a few our own sentences on SVM model, and these are our predictions:

```
Custom Sentence Predictions (SVM):

Sentence: I am feeling very happy today!
SVM Prediction: joy

Sentence: This is the worst day of my life.
SVM Prediction: joy

Sentence: I can't stop smiling, this is the best surprise ever!
SVM Prediction: joy

Sentence: I am so scared to go outside alone.
SVM Prediction: fear

Sentence: I feel so loved and appreciated today.
SVM Prediction: love

Sentence: Why do you always make me so angry?
SVM Prediction: anger

Sentence: I feel like crying all day long.
SVM Prediction: joy
```

Figure 16: Our sentences

5.1.2 Random Forest Classifier

Random forest classification gave us 88% of accuracy
If we look at missclassified examples we can again notice that predicted emotions are very similar to the true ones, such as sadness and fear:

Misclassified Examples (Random Forest Model):

	text	label	rf_predictions
660	i was playing a sport in an advanced pe class ...	3	1
1530	i feel furious at love because i really though...	3	1
1146	i feel affirmed gracious sensuous and will hav...	2	1
119	i feel like i know who most of them are by now...	1	0
1402	i just keep on feeling blessed	2	1
565	i feel like an ugly monster where i cannot sho...	0	4
1791	i did a body scan and realized that everything...	5	1
941	i still feel confused and guilty about the who...	4	0
466	i feel his hand on me to stay faithful	2	1
468	i cant help feeling this way	0	1

Figure 17: Missclassified examples for Random Forest

Finally, we tested a few our own sentences on Random Forest model, and these are our predictions:

```
Custom Sentence Predictions (Random Forest):

Sentence: I am feeling very happy today!
Random Forest Prediction: joy

Sentence: This is the worst day of my life.
Random Forest Prediction: joy

Sentence: I can't stop smiling, this is the best surprise ever!
Random Forest Prediction: joy

Sentence: I am so scared to go outside alone.
Random Forest Prediction: fear

Sentence: I feel so loved and appreciated today.
Random Forest Prediction: love

Sentence: Why do you always make me so angry?
Random Forest Prediction: anger

Sentence: I feel like crying all day long.
Random Forest Prediction: joy
```

Figure 18: Our sentences

5.1.3 Logistic Regression

Logistic classification report is the next one:

```
Classification Report for Logistic Regression:
              precision    recall  f1-score   support

     sadness       0.90      0.93      0.91       581
         joy       0.85      0.96      0.90       695
        love       0.82      0.62      0.71       159
       anger       0.89      0.83      0.86       275
        fear       0.88      0.79      0.83       224
    surprise       0.80      0.53      0.64        66

    accuracy                           0.87      2000
   macro avg       0.86      0.78      0.81      2000
weighted avg       0.87      0.87      0.87      2000
```

Figure 19: Logistic Regression classification report

We can still notice that surprise has the smallest precission, recall and f1-score. Final accuracy was 87% If we look at missclassified examples we can still notice that predicted emotions are very similar to the true ones, such as sadness and fear:

```
Misclassified Examples (Logistic Regression Model):
                                                     text  label  \
433   i know that i have it nowhere near as worse as...      4
74    i were to go overseas or cross the border then...      2
1936  im polyamorous something im starting to feel t...      2
222   i think i wanted audiences to feel impressed i...      5
96    i love neglecting this blog but sometimes i fe...      2
715   i get to be creative if i feel like it or just...      2
242   i see you on the pitchers mound at our little ...      4
193   i really dont like quinn because i feel like s...      3
1387  im not sure but theres nothing that will get a...      2
121                       made a wonderfull new friend      1

      log_reg_predictions
433                     0
74                      1
1936                    1
222                     1
96                      1
715                     1
242                     0
193                     0
1387                    1
121                     0
```

Figure 20: Missclassified examples for Logistic Regression

Finally, we tested a few our own sentences on Logistic Regression model, and these are our predictions:

```
Custom Sentence Predictions (Logistic Regression):

Sentence: I am feeling very happy today!
Logistic Regression Prediction: joy

Sentence: This is the worst day of my life.
Logistic Regression Prediction: joy

Sentence: I can't stop smiling, this is the best surprise ever!
Logistic Regression Prediction: joy

Sentence: I am so scared to go outside alone.
Logistic Regression Prediction: fear

Sentence: I feel so loved and appreciated today.
Logistic Regression Prediction: love

Sentence: Why do you always make me so angry?
Logistic Regression Prediction: anger

Sentence: I feel like crying all day long.
Logistic Regression Prediction: sadness
```

Figure 21: Our sentences

5.2 Convolutional neural network

We decided on training a Convolutional Neural Network because we have a very high number of features, and a CNN could deal with that faster than the regular MLP(Multiple Layer Perceptron). Besides that, to our best knowledge, most of the models trained on this dataset for emotion recognition are more complicated deep-learning models such as transformer based models, or they have more complicated features. We wanted to test a bit simpler approach that could be promising for this type of problem.

We made a customed CNN with four convolutional layers, three max-pool layers and one fully connected layer. Activation functions were ReLU, except in the output where we used Softmax. As a form of regularization we used Dropout and EarlyStopping. For a loss function we used Sparse Categorical Crossentropy, and we used Adam optimizer. Finally, we trained the model for 20 epochs with batch size of 32.

Final classification report is the next one:

	precision	recall	f1-score	support
0	0.96	0.89	0.93	24400
1	0.96	0.86	0.91	28242
2	0.72	0.95	0.82	6872
3	0.88	0.91	0.89	11389
4	0.83	0.86	0.85	9451
5	0.64	0.96	0.77	3007
accuracy			0.89	83361
macro avg	0.83	0.91	0.86	83361
weighted avg	0.90	0.89	0.89	83361

Figure 22: Classification report of CNN

We can notice that the best predicted classes are joy('0') and sadness('1'), and the worst classified one is surprise ('5'). Even though the surprise was the richest class in tokens, it is still the smallest class, while joy and sadness are the biggest ones. We tried to balance classes, but this problem obviously still had an impact on classification.

Obtained confusion matrix is the next one:

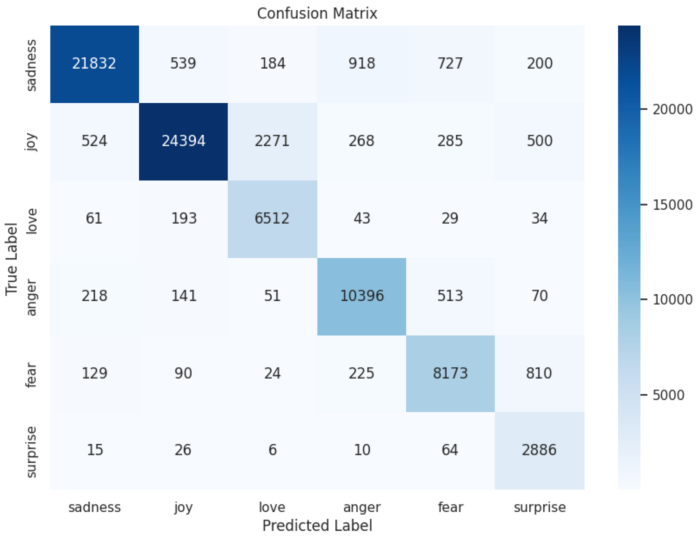

Figure 23: Confusion matrix for CNN

We can have the same conclusion as after observing classification report. It is also very interesting to mention that a lot of joy was classified as love, which could be understandable, because they are similar emotions.

We can also take a look at the missclassified cases:

Text: i was feeling a little low few days back
True label: fear
Predicted label: surprise

Text: i don t feel comfortable around you
True label: fear
Predicted label: surprise

Text: i am only having day a week where i am feeling depressed or seriously anxious
True label: sadness
Predicted label: fear

All of the presented wrongly classified instances are classified as very similar emotion, and in these instances even a human could make a mistake.

5.3 BERT fine-tuning

We also wanted to test and fine-tune a pretrained BERT(Bidirectional Encoder Representations from Transformers) model[5]. BERT implements WordPiece tokenization, and we also used class weights for classes balancing. We trained for 5 epochs, with the learning rate 10^{-5}. After that we improved it with adding EarlyStopping and Dropout for regularization. Final classification report is the next one:

	precision	recall	f1-score	support
sadness	0.96	0.96	0.96	581
joy	0.95	0.95	0.95	695
love	0.83	0.81	0.82	159
anger	0.93	0.90	0.91	275
fear	0.86	0.88	0.87	224
surprise	0.70	0.77	0.73	66
accuracy			0.92	2000
macro avg	0.87	0.88	0.87	2000
weighted avg	0.92	0.92	0.92	2000

Figure 24: BERT classification report

We can see a very high accuracy od 92%, but the imbalanced classes problem still makes the same impact as in previous cases.
If we look at missclassified examples we can still notice that predicted emotions are very similar to the true ones, such as sadness and fear:

	text	label	predictions
1479	i really feel and i know the devil hates that ...	1	3
457	i cant do strappy shoes at work i just feel we...	4	5
917	i feel the need to pimp this since raini my be...	1	2
1936	im polyamorous something im starting to feel t...	2	1
863	i feel betrayed and angry and sad at the same ...	3	0
1764	i don t know how else to describe it except to...	1	2
820	i found myself feeling a bit overwhelmed	5	4
625	i am feeling overwhelmed by trying to do it al...	5	4
1087	i feel for all of you who have been supporting...	1	2
688	i feel hated in cempaka	0	3

Figure 25: Missclassified examples for fine-tuned BERT model

Finally, we tested a few our own sentences on BERT model, and these are our predictions:

```
Text: I'm feeling really down today. → Predicted Emotion: sadness
Text: Wow! This is amazing! → Predicted Emotion: surprise
Text: I am so scared to go outside. → Predicted Emotion: fear
```

Figure 26: Our sentences

5.4 Ensemble(BERT + SVM + RF + LR)

Finally we decided to test a combination of models: BERT + Support Vector Machine + Random Forest + Logistic Regression. We combined the models using majority voting, a technique where the final prediction is the class that received the most votes from all models. If more than one class had the same number of votes, the first occurring one was chosen as the prediction. The classification report is the next one:

```
Ensemble Model Classification Report:
              precision    recall  f1-score   support

     sadness       0.93      0.96      0.94       581
         joy       0.88      0.97      0.92       695
        love       0.84      0.64      0.73       159
       anger       0.93      0.87      0.90       275
        fear       0.87      0.86      0.87       224
    surprise       0.87      0.50      0.63        66

    accuracy                           0.90      2000
   macro avg       0.89      0.80      0.83      2000
weighted avg       0.90      0.90      0.89      2000
```

Figure 27: Ensemble classification report

The accuracy is 90%, but the imbalanced classes problem still makes the same impact as in previous cases. If we look at missclassified examples we can still notice that predicted emotions are very similar to the true ones, such as sadness and fear:

```
Misclassified Examples (Ensemble Model - BERT + SVM + RF + Logistic Regression):
                                                   text  label  \
693   i can say is that as long as you enjoy the sto...      0
1533  i actually was in a meeting last week where so...      3
1714          i also do feel passionate about teaching      2
206   i wish to know whether i should feel sympathet...      2
476   i feel quite helpless in all of this so prayer...      0
254   i feel blessed beyond blessed to share my life...      2
1467  i seek out pain to feel tortured just to feel ...      4
433   i know that i have it nowhere near as worse as...      4
828      i feel unprotected even while travelling alone      4
861   i feel assaulted by this shit storm of confusi...      4

      ensemble_predictions
693                      3
1533                     0
1714                     1
206                      1
476                      4
254                      1
1467                     3
433                      0
828                      0
861                      0
```

Figure 28: Missclassified examples for Ensamble model

Finally, we tested a few our own sentences on Ensamble model, and these are our predictions:

```
Custom Sentence Predictions (Ensemble Model - BERT + SVM + RF + Logistic Regression):

Sentence: I am feeling very happy today!
Ensemble Model Prediction: joy

Sentence: This is the worst day of my life.
Ensemble Model Prediction: joy

Sentence: I can't stop smiling, this is the best surprise ever!
Ensemble Model Prediction: joy

Sentence: I am so scared to go outside alone.
Ensemble Model Prediction: fear

Sentence: I feel so loved and appreciated today.
Ensemble Model Prediction: love

Sentence: Why do you always make me so angry?
Ensemble Model Prediction: anger

Sentence: I feel like crying all day long.
Ensemble Model Prediction: sadness
```

Figure 29: Our sentences

5.5 Results comparison

We tested numerous models, some pretrained, some custom made. The next table presents our obtained accuracies with the highest benchmark accuracy:

Table 2: Model accuracies

Model	Accuracy
Logistic Regression	0.86
Random Forest	0.87
Support Vector Machine	0.89
Convolutional Neural Network	0.89
BERT	0.92
Ensemble (BERT + SVM + RF + LR)	0.90
Fine-tuned distilBERT-base-uncased(transformer based)[4]	0.94

Among our models, the highest accuracy is still the pretrained BERT model, but we managed to obtain very high accuracies with other models, such as Ensamble of models with 90% accuracy, and CNN and Support Vector Machine with 89% of accuracy. However, none of this could compare with the best benchmark model with the 94% accuracy.

6 Conclusion

In this project we implemented a few custom made objects that are fairly simple, but we managed to obtain very high accuracies. They are not comparable with the best benchmark model, but benchmark models are much more complicated, and we believe that we somewhat showed that similar impact could be made with much simpler models.

In our codes we used numerous toolboxes and they are: numpy[11], pandas[12], matplotlib[14], seaborn[15], nltk[8], collections[16], sckit-learn[10], tensorflow[13], datasets[17], transformers[18], torch[19], evaluate[20], and we had a help of ChatGPT[21].

Work distribution:
Marija Brkic: Data and Dataset analysis, State-of-the-art models research, Vectorization and Visualization, Convolutional Neural Network, Report
Aloïs Vincent: Data exploration and visualization, References, SVM, Logistic Regression, Random Forest, Notebooks Merging, Presentation
Md. Naim Hassan Saykat: SVM, Logistic Regression, Random Foresr, BERT, Ensamble, Included in Traditional models and BERT parts of the Report

References

[1] dair-ai/emotion dataset. Available at: https://huggingface.co/datasets/dair-ai/emotion

[2] Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. *CARER: Contextualized Affect Representations for Emotion Recognition.* Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 3687–3697, 2018. Available at: https://aclanthology.org/D18-1404.pdf.

[3] Wang, Y., and others. *Large Language Models on Fine-grained Emotion Detection Dataset with Data Augmentation and Transfer Learning.* arXiv preprint arXiv:2403.06108v1, 2024. Available at: https://arxiv.org/html/2403.06108v1.

[4] Fengkai Yu *Hugging Face Sollution* Available at: https://huggingface.co/Fengkai/distilbert-base-uncased-finetuned-emotion.

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* Proceedings of NAACL, 2019. Available at: https://arxiv.org/abs/1810.04805.

[6] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. *Distributed Representations of Words and Phrases and Their Compositionality.* Advances in Neural Information Processing Systems (NeurIPS), 2013. Available at: https://arxiv.org/abs/1310.4546.

[7] Hochreiter, S., & Schmidhuber, J. *Long short-term memory.* Neural Computation, 1997. Available at: https://doi.org/10.1162/neco.1997.9.8.1735.

[8] Natural Language Toolkit Available at: https://www.nltk.org/.

[9] sklearn.feature_extraction.text.TfidfVectorizer Available at: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.

[10] sckit-learn Available at: https://scikit-learn.org/stable/

[11] numpy Available at: https://numpy.org/

[12] pandas Available at: https://pandas.pydata.org/

[13] tensorflow Available at: https://www.tensorflow.org/

[14] matplotlib Available at: https://matplotlib.org/

[15] seaborn Available at: https://seaborn.pydata.org/

[16] collections Available at: https://docs.python.org/3/library/collections.html

[17] datasets Available at: https://pypi.org/project/datasets/

[18] transformers Available at: https://pypi.org/project/transformers/

[19] PyTorch Available at: https://pytorch.org/

[20] evaluate Available at: https://pypi.org/project/evaluate/

[21] ChatGPT Available at: https://chatgpt.com/