Natural Language Processing

PLN Project 1 - FEUP | MEIC 1º

João Alves - up202007614

Marco André - up202004891

Rúben Monteiro - up202006478

Introduction

Hugging Face

This work was done for our *Natural Language Processing* course at FEUP. The main objectives of this project are the exploration of NLP techniques, including pre-processing, feature extraction, sparse vs dense feature representations, text classification and machine learning classifiers.

Emotion is a dataset of English Twitter messages with six basic emotions: anger, fear, joy, love, sadness, and surprise.

The data fields are:

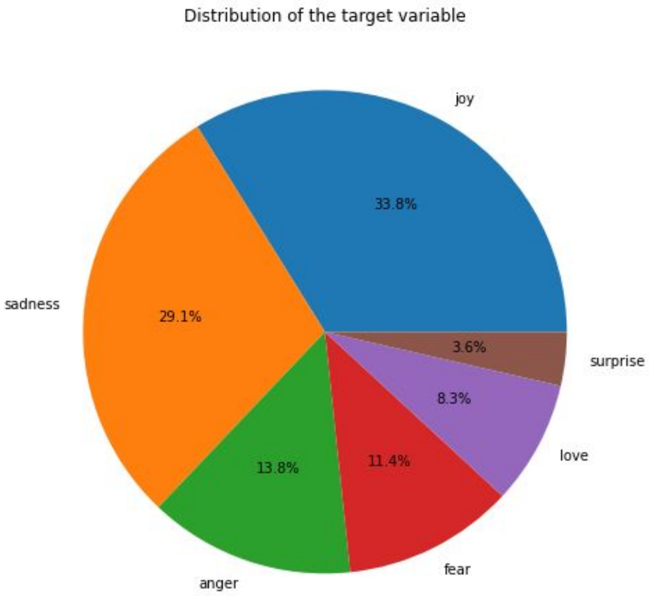
- **text:** a string feature
- **label:** a classification label, with possible values including sadness (0), joy (1), love (2), anger (3), fear (4), surprise (5)

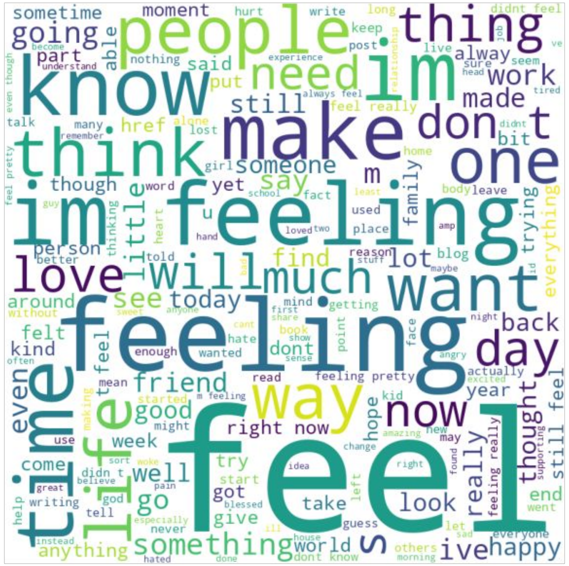
Number of rows: 436,809

Exploratory Data Analysis

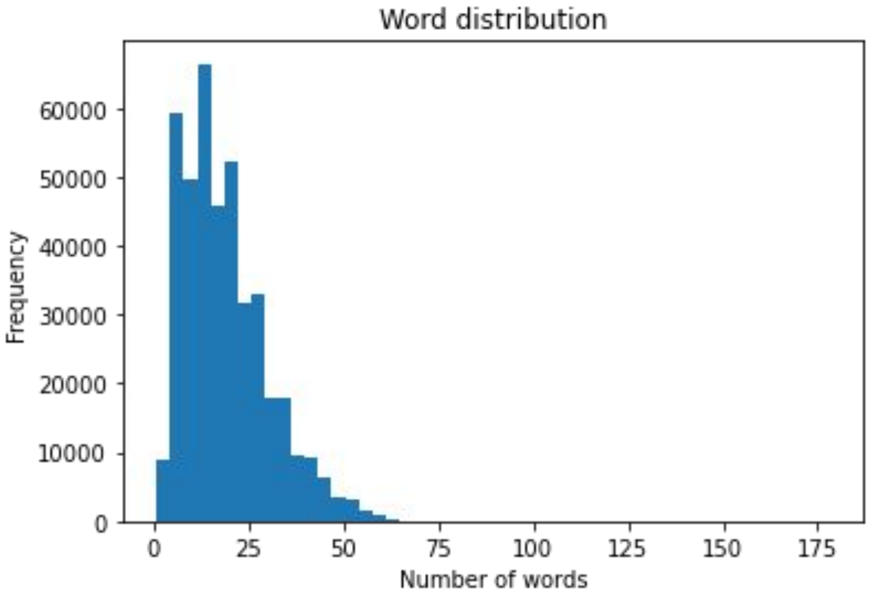
In order to perform the exploration on our dataset, we started by inspecting some of the following information:

- Distribution of the target Variable
- Wordclouds

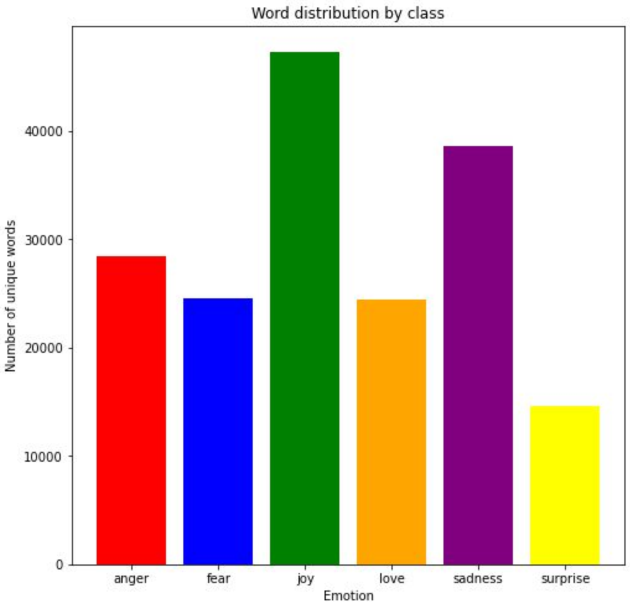




Distribution of the number of words per row



Distribution of the number of words per class



Further information:

- Most common/relevant words
- Use of n-grams to explore negation
- VADER analysis

Pre-processing

The following manual pre-processing steps were applied:

- Cleaning - words such as “*href*, *http*, *www*,...” were removed.
- Removing stop words (needed to consider versions with/without apostrophe due to badly written words)
- Replace all negative words with a prefix “not_” on the following word
- Remove Single character words because they were just noise
- Dealing with repeated letters (like "soooooo" or "amaaaaaazingggggg") by removing the duplicates
- Applied Stemming

Feature extraction / Representations

The following representation techniques were used during the development of our project:

- Bag-of-words (1500 features)
- One-hot (1500 features)
- Tf-idf (1500 features)
- Custom (Word2Vec) Embeddings - trained with our current dataset (150, 300 and 1500 features)
- Twitter (Word2Vec) Embeddings - pre-trained embeddings from tweets (100 features)
- Fasttext Embeddings - pre-trained on english webcrawl and Wikipedia (300 features)

Multiple representations techniques were explored in order to find the most suited one to our problem.

Models & Evaluation Metrics

Different kinds of models were tested with all previously presented representations:

Time	Model
Medium	MultinomialNB
Medium	DecisionTreeClassifier
Fast	GaussianNB
Medium	KNeighborsClassifier
Very Slow	GradientBoostingClassifier
Slow	RandomForestClassifier
Slow	MLPClassifier
Fast	LogisticRegression
Medium	LinearSVC
Fast	SGDClassifier

The following evaluation metrics were collected in order to evaluate the model's performance:

- Accuracy
- Precision
- Recall
- F1

Confusion Matrices were also used in order to visualize the results.

Model Results

We experimented different combinations of representations and models and recorded their results:

Accuracy									
Modelo	bow	one_hot	tf_idf	emb_original_150	emb_original_300	emb_original_1500	emb_twitter no_proc	emb_twitter	emb_fasttext
MultinomialNB	0,85	0,86	0,82	-	-	-	-	-	-
DecisionTreeClassifier	0,36	0,35	0,36	0,46	0,47	0,53	0,43	0,41	0,42
GaussianNB	0,64	0,64	0,68	0,59	0,60	0,58	0,42	0,39	0,25
KNeighborsClassifier	0,75	0,78	0,59	0,71	0,71	0,70	0,55	0,54	0,51
GradientBoostingClassifier	0,44	-	0,44	-	-	-	-	0,43	0,44
RandomForestClassifier	0,82	0,82	0,82	-	0,64	0,65	0,52	0,50	0,50
MLPClassifier	0,82	0,82	0,81	0,83	0,84	0,86	0,71	0,64	0,67
LogisticRegression	0,87	0,87	0,87	0,79	0,82	-	0,60	0,52	0,58
LinearSVC	0,87	0,87	0,87	-	0,82	-	0,59	0,51	0,59
SGDClassifier	-	-	-	0,69	0,71	-	-	0,50	0,47

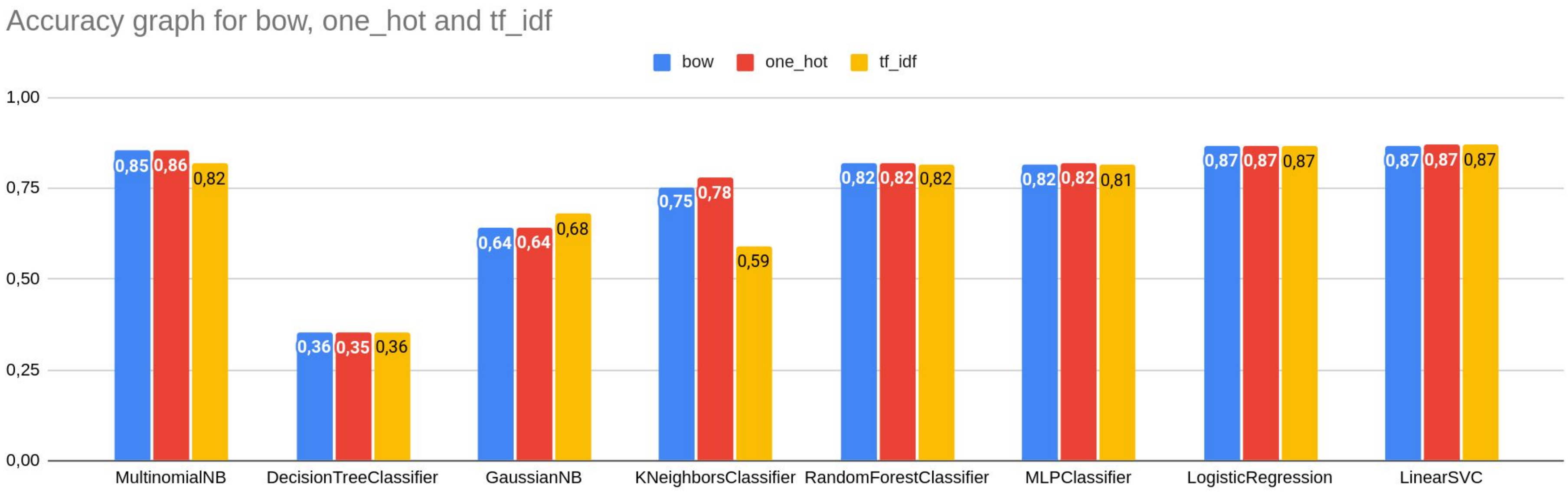
Besides accuracy, other measures such as precision, recall and f1 were also recorded and analysed. The full results can be found here.

Model Results

Some interesting things to note:

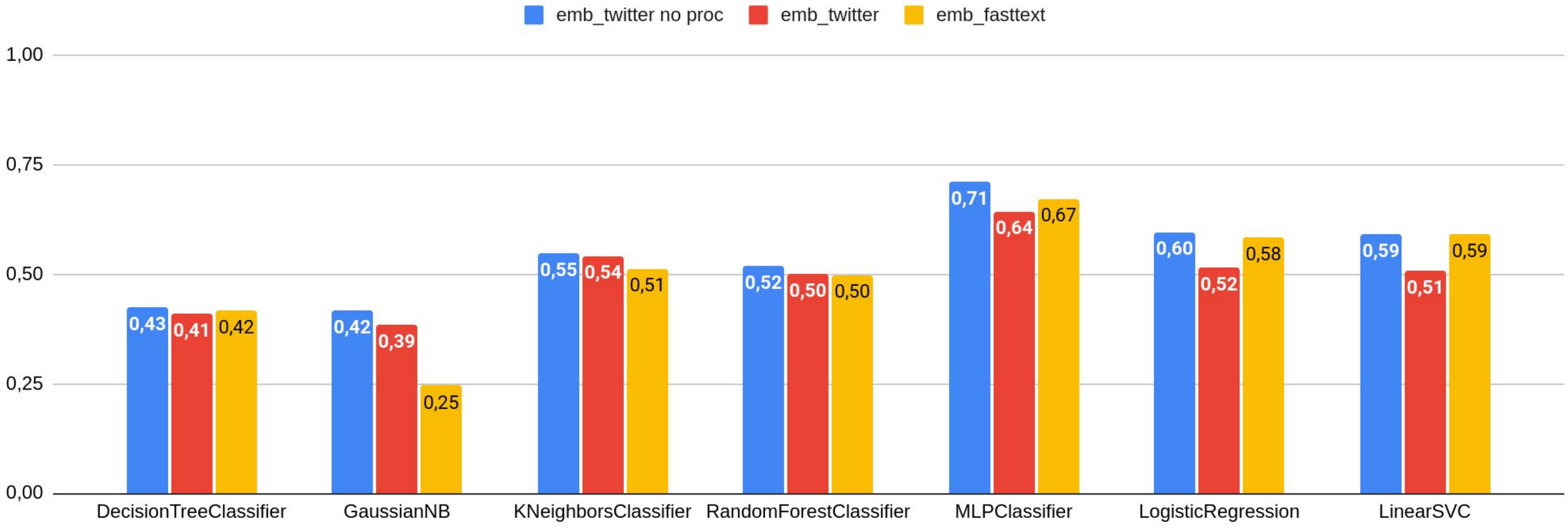
- The **worst** model by far, was the **Decision Tree** (even when tried with different max depth values)
- The **neural network** performed **better** for **embeddings** than it did for the rest of the models
- The **twitter** embeddings were **worst** of when the **pre-processing** step was applied
- The **simpler models**, while taking a lot more space (feature space of **1500**), performed **better** by far
- There was **no substantial difference** among **models**, for the **3 first representations**
- The **custom embedding** had **slightly better** performance when jumping from **150 to 300 and to 1500** features but at **cost of space** and training **time** (which we feel isn't worth it)
- **Gradient Boosting was not a very good model** at all and was axed midway through the results collection due to the very long training time (up to 8h) because of the algorithm itself

Model Results

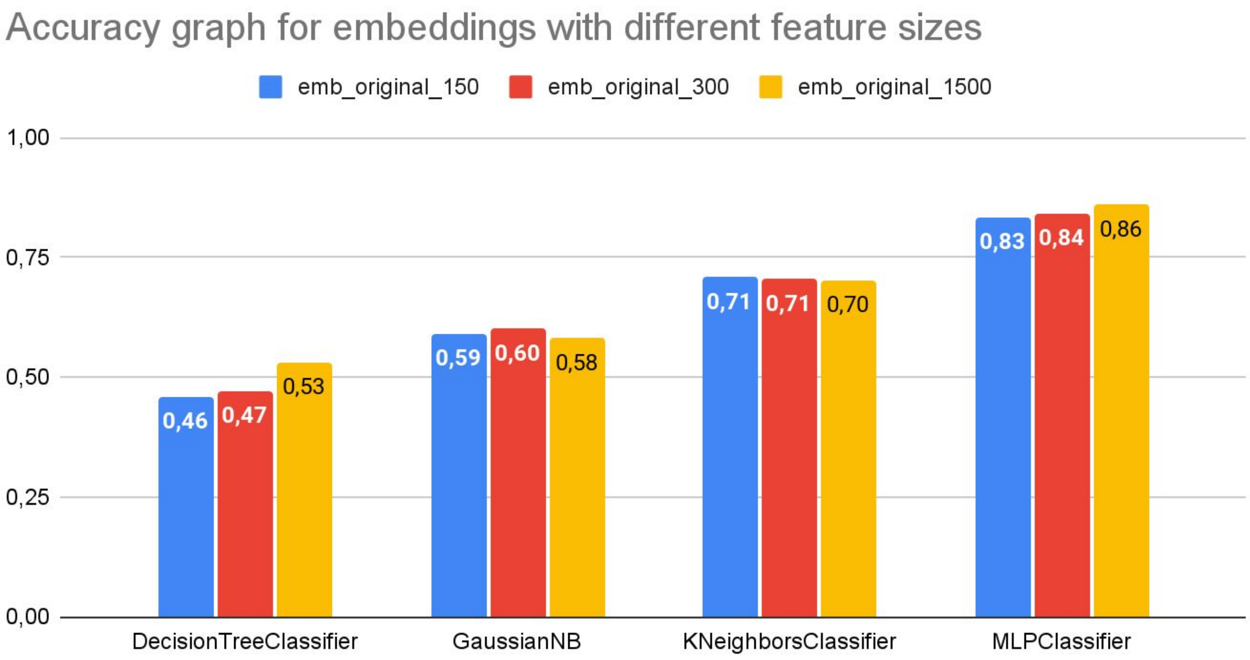


Model Results

Accuracy graph for twitter, twitter without processing and fasttext embeddings

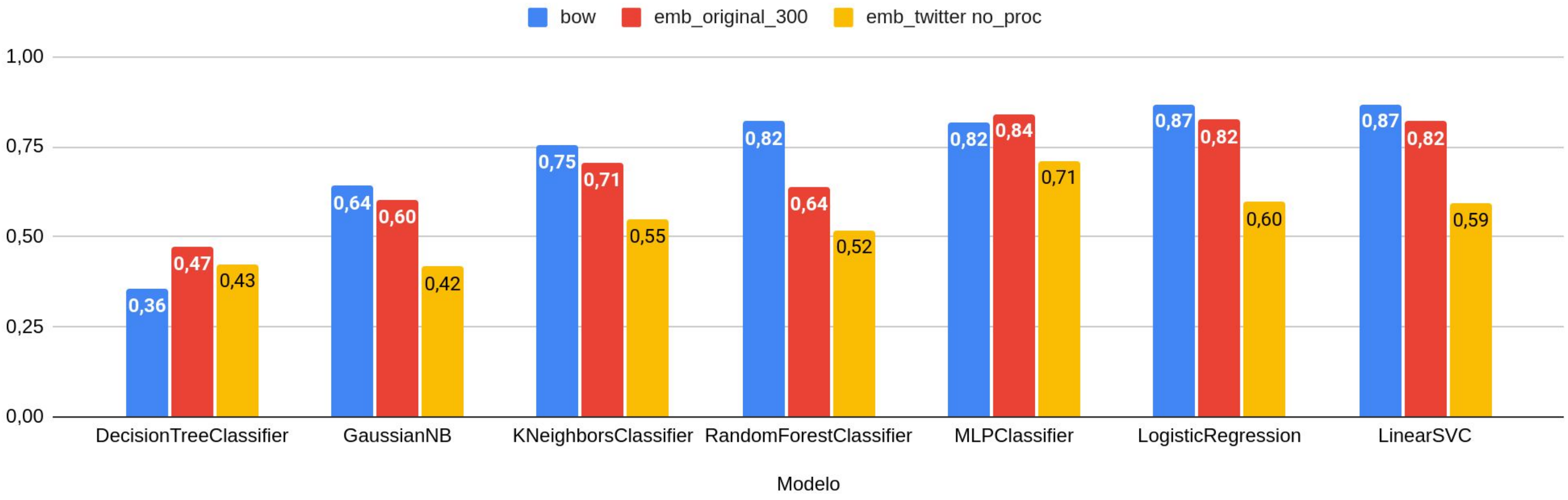


Model Results

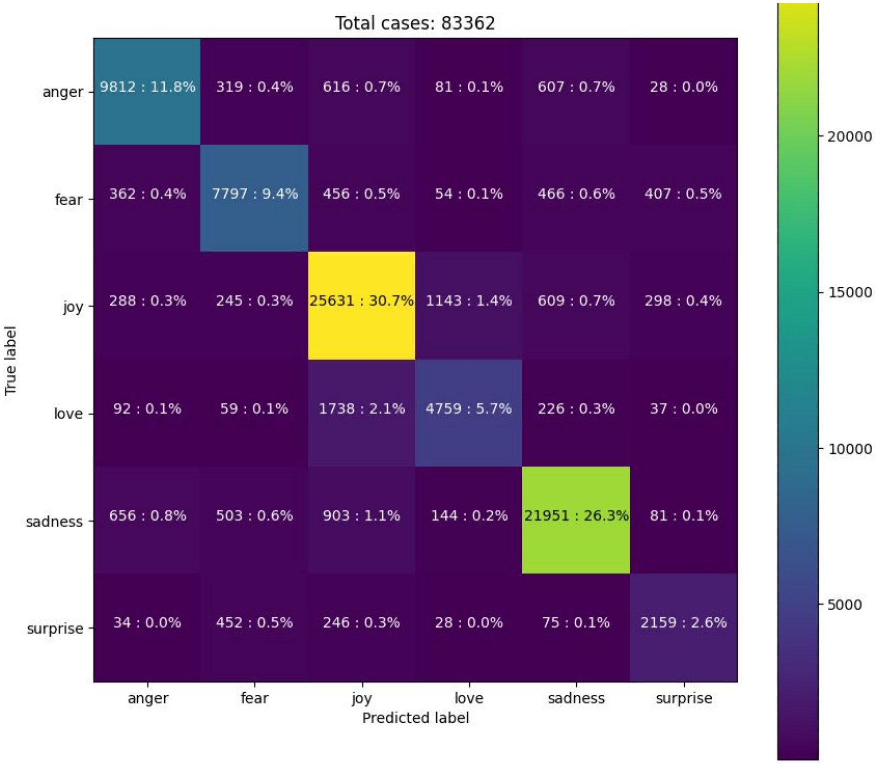


Model Results

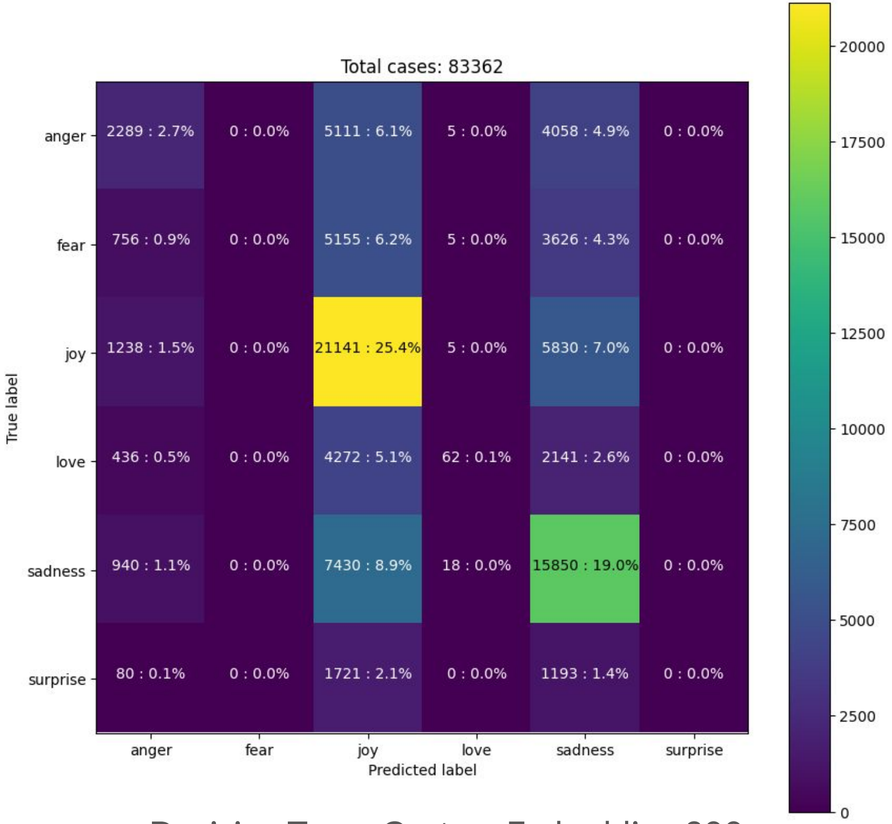
Comparing the best models of each group



Model Results



Logistical Regression - BoW



Decision Tree - Custom Embedding 300

Error Analysis

		Predicted Label						
		Anger	Fear	Joy	Love	Sadness	Surprise	Recall
True Label	Anger	9812	319	616	8	607	28	0,86
	Fear	362	7797	456	54	466	407	0,82
	Joy	288	245	25631	1143	609	298	0,91
	Love	92	59	1738	4759	226	37	0,69
	Sadness	656	503	903	144	2951	81	0,56
	Surprise	34	452	246	28	75	2159	0,72
Precision		0,87	0,83	0,87	0,78	0,60	0,72	**0,78**

Error Analysis

The team undertook an error analysis task for the best overall representation/model combination (BoW with Logistic Regression) and reached some conclusions:

- The dataset contains misclassified information (e.g. “I’m not feeling particularly creative at the moment” is classified as “joy”).
- Some phrases have more than one emotion attached (e.g. “I feel angry, ashamed and sad” is marked only as “angry”)
- The surprise class gets lower results due to being the less common one for the model to learn on.
- The model poorly classifies the sadness class, due to the weight the model attributes to some words. This can also be explained by the earlier table, where the model guessed “Anger” or “Joy”. The first is due to the common overlap in this sentiments, while the second can be explained by negative words not being correctly interpreted.

Related Works

Study 1

They found that more sentiment classes lead to a greater inaccuracy. Still, their overall results for 6 classes were 60.2%, while ours were 87%.

Study 2

Our results were better (87%) compared to theirs (61%). This is exacerbated given the fact that we are testing for 6 classes and they were only testing for 3.

Conclusions:

The field of Twitter Sentiment Analysis is still in its infancy and requires further research. If a good method for constant analysis is developed, Twitter may be used to feed specific opinions to specific people to analyze and even manipulate general opinions on certain topics.

End