Kaggle Competition 2: report

Text Classification Challenge

1. Our information:

Team name: FR

Team members and student numbers: Rachel Ebrahimi (20237647), Faezeh PouyaMehr (20241704)

Kaggle Usernames: rachelebrahimi98, faezehpouyamehr

2. Introduction

In this project, we had to do text semantic analysis which means we had to classify a dataset containing texts, and label them as positive, negative and neutral. We needed to create features based on the data we had and train algorithms using that. The dataset contained about 1 million text seen in figure 1, which were all labeled.

index	id	text
0	0	Anyway Im getting of for a while
1	1	My red, Apache isn't feelin too well this morning.. http://mypict.me/49n5
2	2	@danyelljoy you should be its great. friday will be great tooooooo))))
3	3	its 11:30pm and i dont wanna sleep; so i debated with myself, and in the end i decided what a perfect time to BAKE! no kidding.
4	4	Why does twitter eat my DM's? Not happy
5	5	@mranstey hey there. Drivin north. I guess we will miss u tonite? http://myloc.me/2NIn
6	6	is making cheese today in biology
7	7	cant sleep its already 2:00 am
8	8	What a rainy gloomy week...cant even get into our new pool
9	9	some bitch stole my blackberry the other night in Santa Monica. Still pissed, WHY SHE GOTTA TAKE MA BABY AWAY
10	10	YEEEE! STILL READY TO JUMP OUT OF MY OWN SKIN!!! I CANNOT WAITTTTTT!!!!
11	11	@BL4CKB4NN3R: why sadfaces?
12	12	is quite bummed that he didnt bump into Jonathon Ross yesterday at the Trocadero. I would of asked him to play Rambo with me...
13	13	I'm officially out of gas we are sitting on the side of the road can anyone come save us HCIBTHWDFM?
14	14	last night was crazy. gosh i love kiewit boys. my boy toy leaves today
15	15	@ilove2blogg i know...i was broke and had work in the mornin. how was it? Wat lifesavers were there?
16	16	@geisha2me boo i can never see your postings proper on here...anyway, is the hubby going away so much?
17	17	@FashionGuru nooooooooooooo! I wanted to see that movie I can't believe you just did that without a spoiler alert!
18	18	@jordanknight ..jordan the philosopher..didn't know you had it in you xoxo
19	19	Seat service please. I want a hot dog, a soda, some fries... I'm in seat 9 row 8 sec 418. Please hurry. I'm hungry.
20	20	@michellereneex haha, its alright. im dying of heat though. and wishing i was in dallas to see the jonas brothers!
21	21	I just wanna curl up on the couch with Stinky and Jar... instead... I'm at effing work... doing absolutely nothing
22	22	What the hell? There's like a congregation of indie adults at the coffee bean on beach and talbert ahahaha! its pretty intense.
23	23	@MadgeAsimo its allright dont worry dear its just boring anyway ^^ i like talking to you: MADONNA LOVERS DO IT BETTER! right? (L)

Figure 1- A sample part of the dataset used in this project

We used six algorithms for this classification. First, we used a **Naïve Bayes Classifier** for which we used bag of words as its features. The accuracy we got for this method was about 75.5%. Second, we used **SVM** with RBF kernel [1] for which we got the accuracy of 77.7%. We got the best accuracy on Tfidf features. Third, we used **Multiclass Logistic Regression** using Newton method and Tfidf features and we got the accuracy of 76.7%. For the Fourth algorithm, we tried **Decision Tree** with Tfidf features and got accuracy of 70.5%. Fifth, we used a **Random Forest** algorithm with entropy criterion and Tfidf features and got the accuracy of 76.4%. For the sixth algorithm, we used **Neural Network:** Bidirectional NN with LSTM [2] with an accuracy of 79.6%, which was the best performing method for us. We used Adam optimization with cross entropy loss for this Algorithm.

3. Feature Design

For this project we extracted the words in each text. For this, we first extracted the emoji in the text that were created with punctuation marks (figure 2) because they have semantic information and then we removed all the punctuation marks and numbers and extracted the words. Then we realized there are many words such as run, running, ran, etc. that are actually the same in meaning, but were identified separately. So, we used Stemming on them. The result was not perfect, and could still be improved. So, we finally used Lemmatization on the words which resulted a cleaner dictionary. Finally we created Bag of Words and Tfidf features out of the dictionary we and trained our models using them.

```
emoji = [':)', '(;', '(:', ';)', '|:', ':|', '@_@', '):', ':(', '),:', ':,(', ')\';', ':\'(', ':D', ':}', '{:', '}:',
        ':{', ':]', ':[', '[:', ']:', '\\:', '/:', ':/', ':\\', ':p', ';p', '*_*',
        ':*', '*:', '^^', '^_^', '^-^', ':O', 'O:', ':o', 'o:']
```

Figure 2- List of emoji we used to extract from the texts.

- Naïve Bayes
  We used Bag of Words as the features of this algorithm.

- SVM
  As number of all the words in the whole dataset was so huge and would make training time of SVM very high, we decided to remove the words which were not frequent in the whole dataset before extracting the Tfidf features. So we tested removing the words with less than “n“ repeating and tried different n for this. Finally we used n=100 to get the best result on SVM. Also, we used Tfidf as the features of this algorithm and used RBF as its kernel.

- Random Forest
  We did the same as SVM for feature selection and after testing different hyper parametes, we used n_estimator = 50 and criterion of entropy.

- Decision Tree
  We did the same as SVM for feature selection and we chose our hyper parametes using grid search. The criterion we used for this algorithm was entropy.

- Neural Networks
  We implemented two Neural Networks:

  Bidirectional LSTM: We used all the words we got in our preprocessing part as the features . We transformed each of our texts to a list with ids instead of the words. We used Adam optimization and cross entropy loss for this algorithm.

4. Algorithms

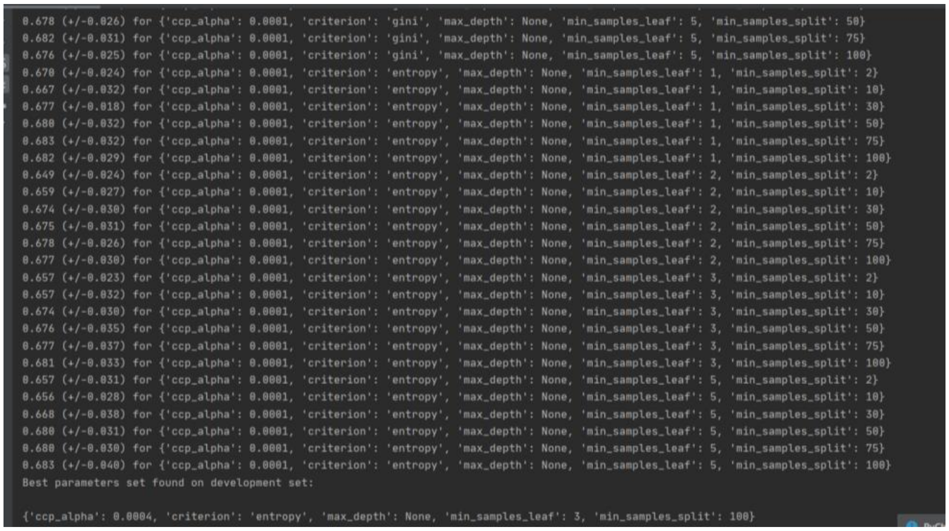
- Naïve Bayes
  Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object [3].
  We used Multinomial Naïve Bayes classifier suitable for using Bag of Words as it features. It was fast to train and we got accuracy of 75.5%.

- Logistic Regression
  Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables [4].
  As in this dataset we had 3 classes, we needed a multiclass logistic regression for this project. After testing different hyper parameters, we finally used 2000 iterations using Newton method for its optimization and L2 as its regularizer.

- SVM
  As the second choice, we used SVM. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N is the number of features) that distinctly classifies the data points [5].
  After testing many parameters, we finally used C = 50 and gamma = 0.0045 and RBF as its kernel.

- Random Forest
  A random forest is a Meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [6]. This algorithm was slow and the best accuracy it gave us was 76.4%.

- Decision Tree
  A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes [7]. We used grid search (figure 3) for it to search among different hyper parameters we could choose and the best performing hyper parameters we found were ccp_alpha= 0.000095, criterion= 'entropy', max_depth= None, min_samples_leaf= 7, min_samples_split= 500 which gave us the accuracy of 70.5%.



```
0.678 (+/-0.026) for {'ccp_alpha': 0.0001, 'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 50}
0.682 (+/-0.031) for {'ccp_alpha': 0.0001, 'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 75}
0.676 (+/-0.025) for {'ccp_alpha': 0.0001, 'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 100}
0.670 (+/-0.024) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
0.667 (+/-0.032) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10}
0.677 (+/-0.018) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 30}
0.680 (+/-0.032) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 50}
0.683 (+/-0.032) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 75}
0.682 (+/-0.029) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 100}
0.649 (+/-0.024) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2}
0.659 (+/-0.027) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 10}
0.674 (+/-0.030) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 30}
0.675 (+/-0.031) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 50}
0.678 (+/-0.026) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 75}
0.677 (+/-0.030) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 100}
0.657 (+/-0.023) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 3, 'min_samples_split': 2}
0.657 (+/-0.032) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 3, 'min_samples_split': 10}
0.674 (+/-0.030) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 3, 'min_samples_split': 30}
0.676 (+/-0.035) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 3, 'min_samples_split': 50}
0.677 (+/-0.037) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 3, 'min_samples_split': 75}
0.681 (+/-0.033) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 3, 'min_samples_split': 100}
0.657 (+/-0.031) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 2}
0.656 (+/-0.028) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 10}
0.668 (+/-0.038) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 30}
0.680 (+/-0.031) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 50}
0.680 (+/-0.030) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 75}
0.683 (+/-0.040) for {'ccp_alpha': 0.0001, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 5, 'min_samples_split': 100}
Best parameters set found on development set:

{'ccp_alpha': 0.0004, 'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 3, 'min_samples_split': 100}
```

Figure 3- A sample part of the output of Grid search.

- Neural Network

  Neural nets take inspiration from the learning process occurring in human brains. They consists of an artificial network of functions, called parameters, which allows the computer to learn, and to fine tune itself, by analyzing new data. Each parameter, sometimes also referred to as neurons, is a function which produces an output, after receiving one or multiple inputs. Those outputs are then passed to the next layer of neurons, which use them as inputs of their own function, and produce further outputs. Those outputs are then passed on to the next layer of neurons, and so it continues until every layer of neurons have been considered, and the terminal neurons have received their input. Those terminal neurons then output the final result for the model [8].

  For this part we used Bidirectional Neural Network with LSTM. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems [9]. Bidirectional long-short term memory (Bidirectional LSTM) is the process of making any neural network o have the sequence information in both directions backwards (future to past) or forward(past to future)[10].

5. Methodology

We decided to use the last 100000 data points as the validation set and the first 900000 points as the training set. They had almost the same distribution as seen in figure 4. So, I used this simple split for computation simplicity as sklearn split method was not able to split the sparse matrix we had.

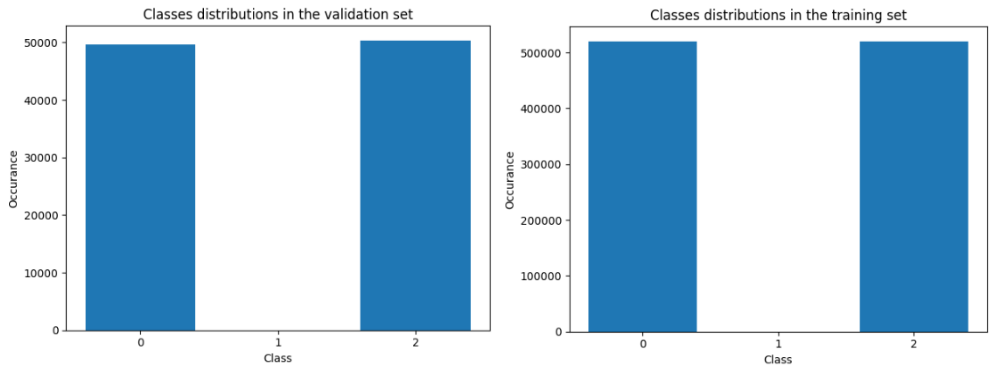


Figure 4 – Data distribution in different classes. Right plot is this distribution on the whole dataset and the left Image the distribution on the validation set.

We did not use any regularization on Naïve Bayes. For SVM we used squared L2 penalty is used as the regularization which is specified by the parameter C. The strength of the regularization is inversely proportional to C. We used L2 regularizer for Logistic Regression and L1 for Bidirectional LSTM Neural Network.

For optimization, Newton Method was used for Logistic regression and Adam was used for Bidirectional LSTM Neural Network.

After trying many different hyper parameters for all the algorithms used, I picked the ones that performed the best on my validation set. The final parameters I used for each of my algorithms is as follows:

- For Naïve Bayes we used alpha = 1.
- For Logistic Regression, we used 2000 iterations.
- For SVM, we used C = 50 and Gamma = 0.0045.
- For Random Forest we used n_estimators = 100 and criterion = entropy
- For Decision Tree we used ccp_alpha= 0.000095, criterion= 'entropy', max_depth= None, min_samples_leaf= 7, min_samples_split= 500
- For Bidirectional LSTM Neural Network we used 64 as dimentioanilty of output space, dropout = 0.1 and learning of Adam optimizer= 0.002.

6. Results

The following tables show a few of our results using different parameters at final stages of tuning.

- Logistic Regression

Max Iter	penalty	solver	Training samples	Training accuracy	Validation accuracy
2000	L2	Newton-cg	9000	83.6%	73.6%
20000	L2	liblinear	90000	78.89%	76.35%
2000	L2	Newton-cg	90000	79.1%	76.2%

- SVM:

C	Gamma	Threshold for removing less frequent words	Number of training samples	Validation accuracy
50	0.002	500	9000	72.95%
50	0.003	500	9000	72.97%
50	0.003	500	19000	74.07%
50	0.003	200	39000	75.14%
50	0.003	100	39000	75.26%
50	0.003	100	79000	75.97%
50	0.004	100	79000	75.98%
50	0.004	100	9000	73.37%
50	0.0045	100	9000	73.38%

- Random Forest

Criterion	n_estimators	Validation accuracy
Gini	50	76.39%
entropy	50	76.42%

- Decision Tree

Criterion	ccp_alpha	Min samples split	Min samples leaf	Validation accuracy
entropy	0.0008	50	5	60-70%
entropy	0.002	2	5	40-50%
entropy	0.0004	100	3	60-70%

- Bidirectional LSTM :

epoch	Training loss	Validation loss	Training accuracy	Validation accuracy
3	0.444	0.414	81.1%	79.6%
5	0.450	0.397	82.1%	79.4%
8	0.460	0.379	83%	79.2%
10	0.456	0.378	83.3%	79%

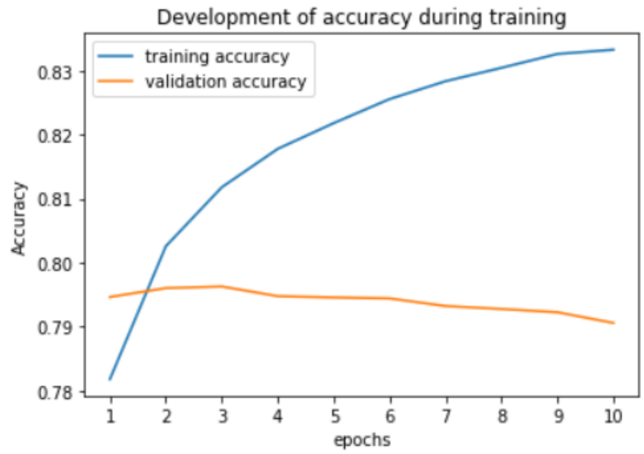


Figure 5- Development of accuracy during training in Bidirectional LSTM

Figure 6- Development of accuracy during training in Bidirectional LSTM

As seen in figure 5 and 6, the loss and accuracy of validation set decreased until the third epoch and then stated increasing after that. So, we decided to train our model only for 3 epochs.

Comparing the three algorithms, we see that Bidirectional LSTM was the best option with the best performance:

Algorithm	Training accuracy	Validation accuracy	Kaggle Test accuracy
Naïve Bayes	79%	75.5	-
Logistic Regression	77%	76.92%	76.78%
SVM	95.31%	80.5%	79.72%
Random Forest	-	76.42%	76.44%
Decision Tree	-	70.5%	70.7%
NN with LSTM	81.1%	79.6%	79.65%

Overall, the Neural Network method had better performance. SVM had a performance so close to NN, but the training time was so long on the huge dataset we had. Random Forest had a very long training time as well and Naïve Bayes had the shortest training time. Decision Tree had the least performance among all.

7. Discussions

In this project, the most important part was data cleaning and feature extraction. The approach we had was not computationally heavy and could perform well. However, for better performance we will need more complex semantic analysis methods. Also, using some pre-trained models such as BERT could improve our work, because only using the Tfidf or BoW features were not enough for semantic analysis in this problem.

We could also improve our current models by more hyper parameter tuning and more searching through them. But it was a computationally heavy task and could not be applied widely.

Another further improvement can be to cope with the imbalanced data we had. Class 0 and 2 were almost equally occurred, but there were only a few cases of class 1 in the whole dataset which would result in almost never predicting class 1. For instance, we could apply some data augmentation.

8. Explainability

Explainability was done using LimeTextExplainer from lime.lime_text import. We first train our model (Random Forest) on a small portion of the whole dataset and use this trained model and trained tfidf_vectorizer from our preprocess step to use the lime package and explain our whole text classification problem on some random samples of the validation dataset. Based on the figures 7 and 8 the model is recognizing positive and negative words with good probability. For Instance "hurt" is classified as negative, while "happy", "love", "yes" as positive which makes sense.

Figure 7- Output of Lime-1

Figure 8- Output of Lime-2

9. Statement of Contributions

We both contributed equally in this project.

- Faezeh did the research about the problem and the preprocessing method we should go on with- She also implemented some of the algorithms and also the Lime- Wrote explainability part
- Rachel did the preprocessing parts- She implemented some of the algorithms- She wrote the report

We hereby state that all the work presented in this report is that of the author.

10. References

[1]: https://github.com/sid-thiru/Text-Classification-with-TFIDF-and-sklearn/blob/master/sklearn_classifiers.py

[2]: https://github.com/changhuixu/LSTM-sentiment-analysis/tree/35ed3660cb11fb7a366230331be5d747d63bc492

[3]: https://www.javatpoint.com/machine-learning-naive-bayes-classifier

[4]: https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression

[5]: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[6]: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#:~:text=A%20random%20forest%20classifier.,accuracy%20and%20control%20over%2Dfitting.

[7]: https://www.ibm.com/topics/decision-trees

[8]: https://towardsdatascience.com/classification-using-neural-networks-b8e98f3a904f

[9]: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

[10]: https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/#:~:text=Bidirectional%20long%2Dshort%20term%20memory,forward(past%20to%20future).