NLP SENTIMENT ANALYSIS

IMDB MOVIES REVIEW

PROBLEM & DATA

1. Problem Definition

The objective is to determine the polarity of of the reviews classifying whether the review is positive or negative.

2. Dataset

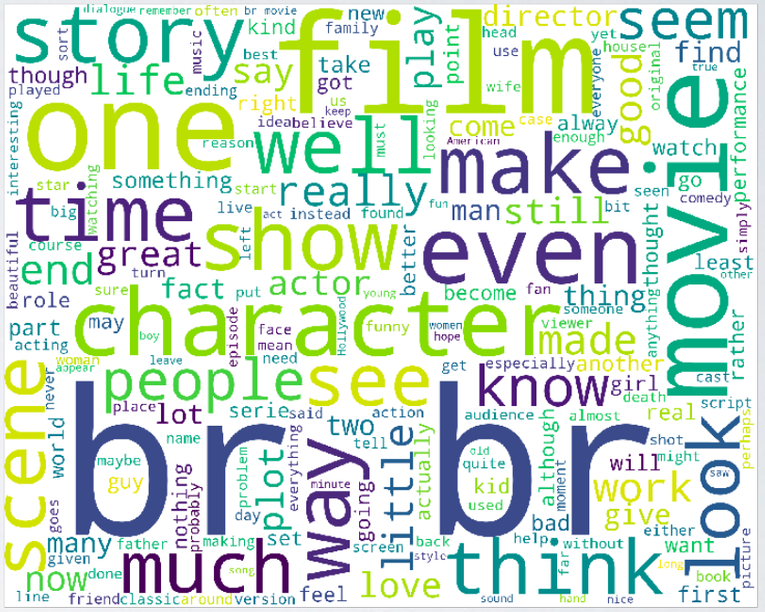
The IMDB dataset is a binary sentiment classification provided by Stanford University with a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. There is additional unlabeled data for use as well. Raw text and already processed bag of words formats are provided.

The labeled data set consists of 50,000 IMDB movie reviews, specially selected for sentiment analysis. The IMDB rating < 5 results in a sentiment score of 0, and rating ≥7 have a sentiment score of 1. No individual movie has more than 30 reviews. The 25,000 review labeled training set does not include any of the same movies as the 25,000 review test set. In addition, there are another 50,000 IMDB reviews provided without any rating labels.

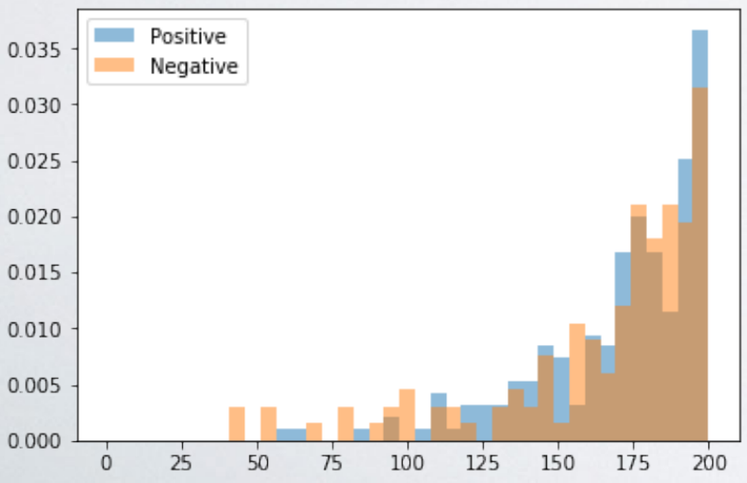
Location: http://ai.stanford.edu/~amaas/data/sentiment/

DATA CLEANING, FEATURE ENGINEERING

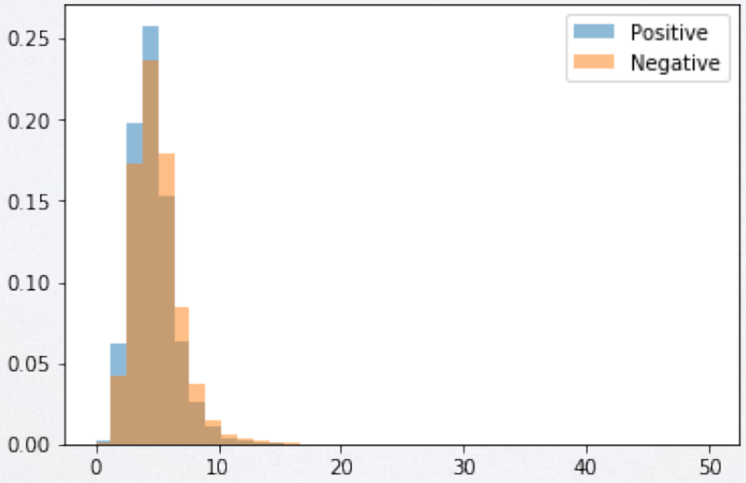
- Remove HTML characters
- Tokenizing and removing Nonstop words
- Feature Engineering:
- Creation of features: Length, punctuation, word counts
- Word embedding TF IDF
- Flair Embeddings



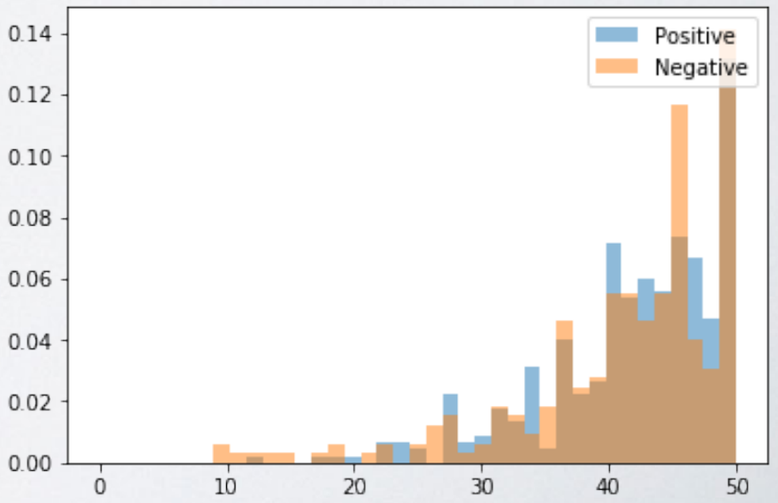
Text Length



Punctuation in the Text



Count of words



TRAINING MODELS

- Selected three models for text classification: Model Random Forest, Gradient Boosting and Flair NLP State of the Art

Random Forest

Feature Importance

```
[(0.0032239412208359213, 'len'),
 (0.002586011500927889, 'punct%'),
 (0.002574772837474503, 'text_nostop'),
 (0.0025358750164764608, 'word count')]
```

```
1 print(confusion_matrix(y_test,y_pred_rf))
2 print(classification_report(y_test,y_pred_rf))
3 print(accuracy_score(y_test, y_pred_rf))
```

```
[[4394  661]
 [ 694 4251]]
              precision    recall  f1-score   support

           0       0.86      0.87      0.87      5055
           1       0.87      0.86      0.86      4945

   micro avg       0.86      0.86      0.86     10000
   macro avg       0.86      0.86      0.86     10000
weighted avg       0.86      0.86      0.86     10000

0.8645
```

Gradient Boosting

Feature Importance

```
[(0.0002817854324905849, 'punct%'),
 (4.5316218563653224e-05, 'text_nostop'),
 (0.0, 'word count'),
 (0.0, 'len')]
```

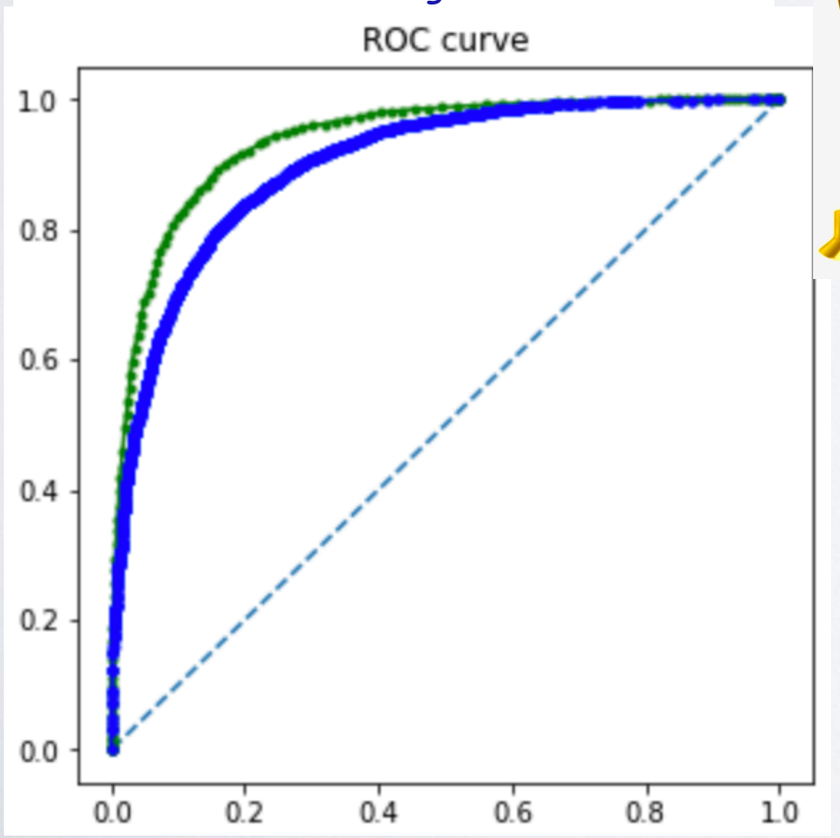
```
1 print(confusion_matrix(y_test,y_pred_gb))
2 print(classification_report(y_test,y_pred_gb))
3 print(accuracy_score(y_test, y_pred_gb))
```

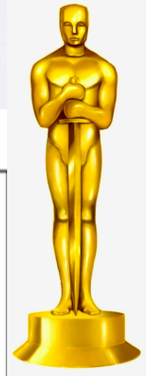
```
[[3836 1219]
 [ 649 4296]]
              precision    recall  f1-score   support

           0       0.86      0.76      0.80      5055
           1       0.78      0.87      0.82      4945

   micro avg       0.81      0.81      0.81     10000
   macro avg       0.82      0.81      0.81     10000
weighted avg       0.82      0.81      0.81     10000

0.8132
```

MODEL EVALUATION

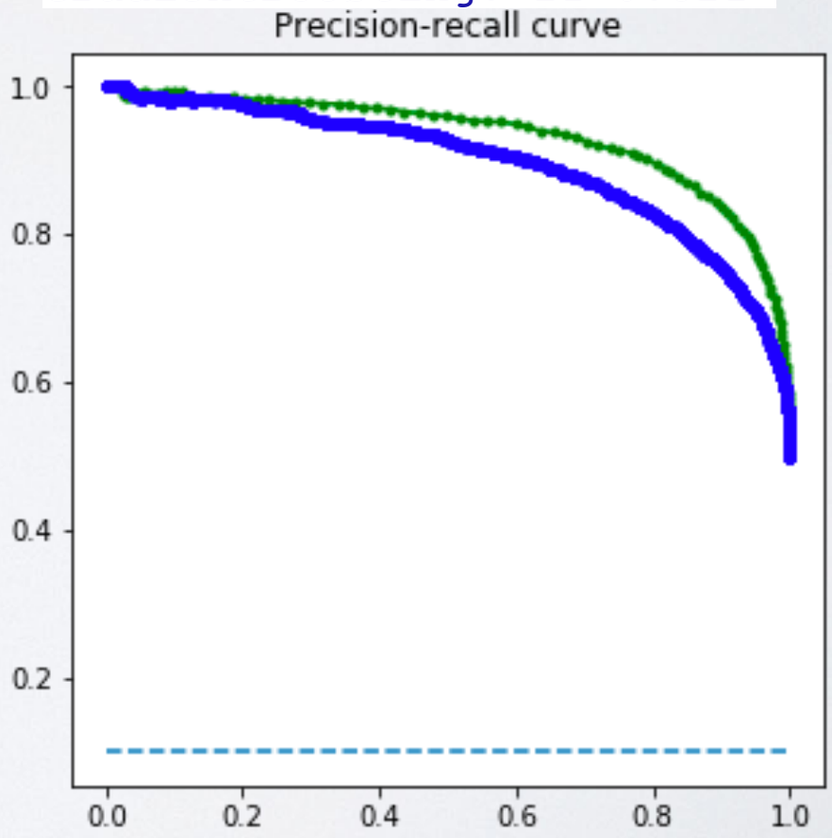
- Both models have great learning skill and a good precision curve which translated into great predicting power. However, based on performance, accuracy and precision we conclude Random Forest is better model

Randomforest AUC: 0.939
GradientBoosting AUC: 0.902

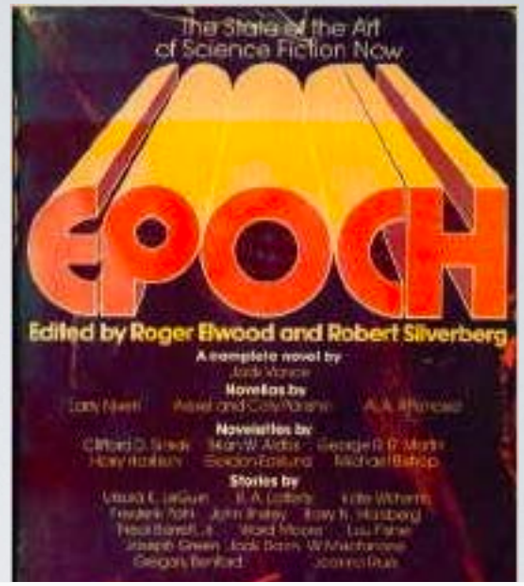




Randomforest: f1=0.863 auc=0.933
GradientBoosting: f1=0.821



FLAIR TEXT CLASSIFIER



- Flair Text Classifier
- We trained a Flair Text classifier Model 150 epochs during two days; Accuracy of 47% and F1 Score of 65%.
- Running a prediction we identify that Flair model accurately identify that the review is positive

```
-------
2019-04-22 19:02:38,666 EPOCH 150 done: loss 0.0001 - lr 0.0500 - bad epochs 1
2019-04-22 19:03:09,868 DEV  : loss 0.10816629 - f-score 0.6000 - acc 0.4286
2019-04-22 19:08:28,407 TEST : loss 0.10048141 - f-score 0.6532 - acc 0.4850
2019-04-22 19:08:47,224 ----------------------------------------------------------------------------------------------------
-------
2019-04-22 19:08:47,233 Testing using best model ...
2019-04-22 19:08:47,254 loading file resources/taggers/sentiment/best-model.pt
2019-04-22 19:12:52,708 MICRO_AVG: acc 0.4736 - f1-score 0.6428
2019-04-22 19:12:52,734 MACRO_AVG: acc 0.4736 - f1-score 0.64275
2019-04-22 19:12:52,736 neg        tp: 825 - fp: 468 - fn: 425 - tn: 782 - precision: 0.6381 - recall: 0.6600 - accur
acy: 0.4802 - f1-score: 0.6489
2019-04-22 19:12:52,737 pos        tp: 782 - fp: 425 - fn: 468 - tn: 825 - precision: 0.6479 - recall: 0.6256 - accur
acy: 0.4669 - f1-score: 0.6366
2019-04-22 19:12:52,739 ----------------------------------------------------------------------------------------------------
-------
```

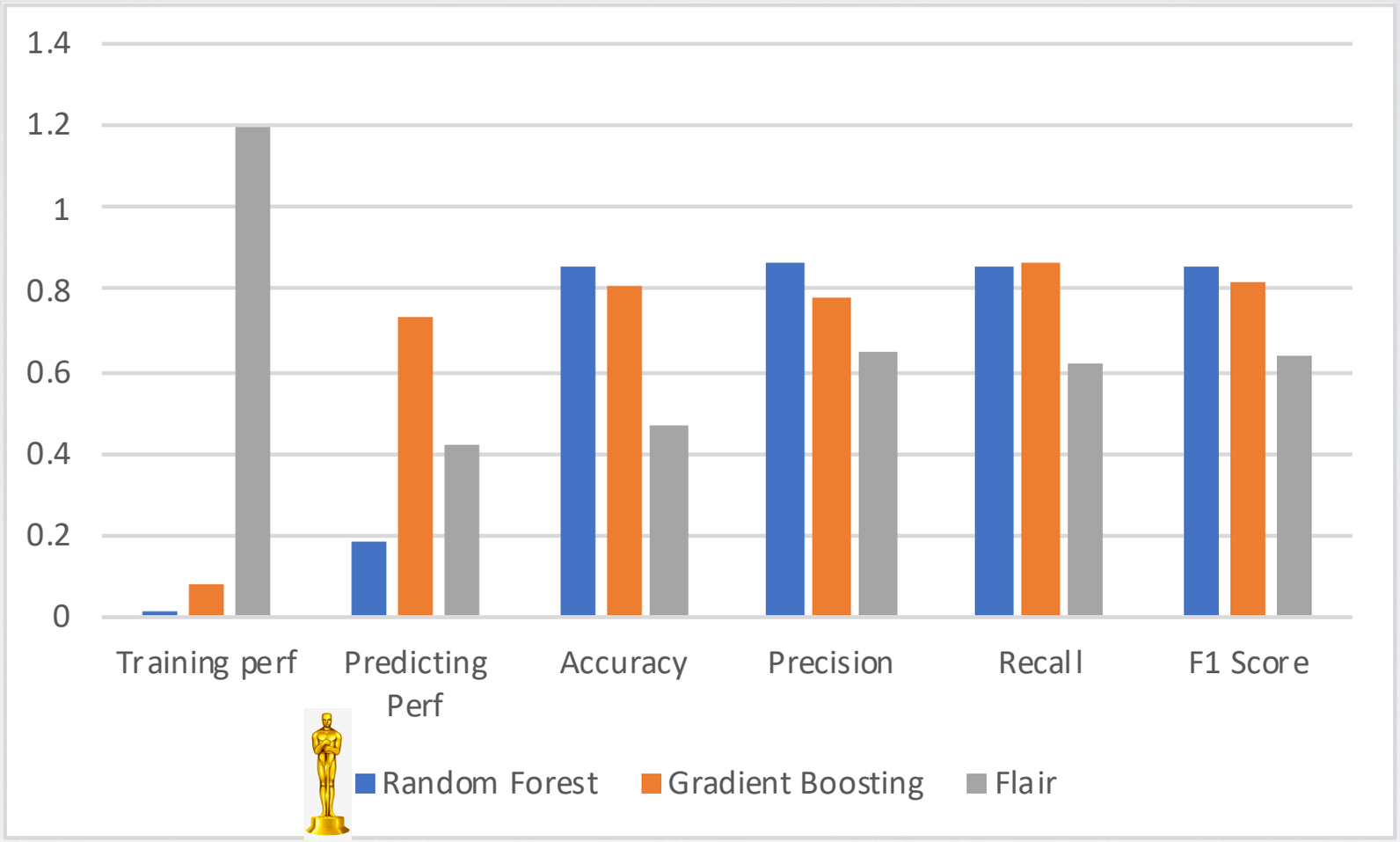
```
from flair.data import Sentence
# create example sentence
sentence = Sentence('I liked French Kiss, Kevin Kline and Meg Ryan delivered an amazing performan
ce; love the story and the connection to grapes and wine in France')

# predict tags and print
scores = classifier.predict(sentence)
print(sentence.labels)
print(scores)
```

```
[pos (1.0)]
[Sentence: "I liked French Kiss, Kevin Kline and Meg Ryan delivered an amazing performance; love t
he story and the connection to grapes and wine in France" - 25 Tokens]
```

```
# create example sentence
sentence = Sentence('I was very disapointed with the episode 1, 2 and 3 of star wars.  Bad script
and very regular acting.  I do not recommend it')

# predict tags and print
scores = classifier.predict(sentence)
print(sentence.labels)
print(scores)
```

```
[neg (1.0)]
[Sentence: "I was very disapointed with the episode 1, 2 and 3 of star wars. Bad script and very r
egular acting. I do not recommend it" - 25 Tokens]
```

CONCLUSION



Model	Training perf	Predicting Perf	Accuracy	Precision	Recall	F1 Score
Random Forest	6mins	18 secs	86%	87%	86%	86%
Gradient Boosting	5 hrs	1.4 mins	81%	78%	87%	82%
Flair	+2Days	~5 secs	47%	65%	62%	64%

- Find the code in Jairo's GitHub:

https://github.com/JAIR0MEL0/ML1010FinalProject/blob/master/ML1010MovieReview_JairoMelo.ipynb