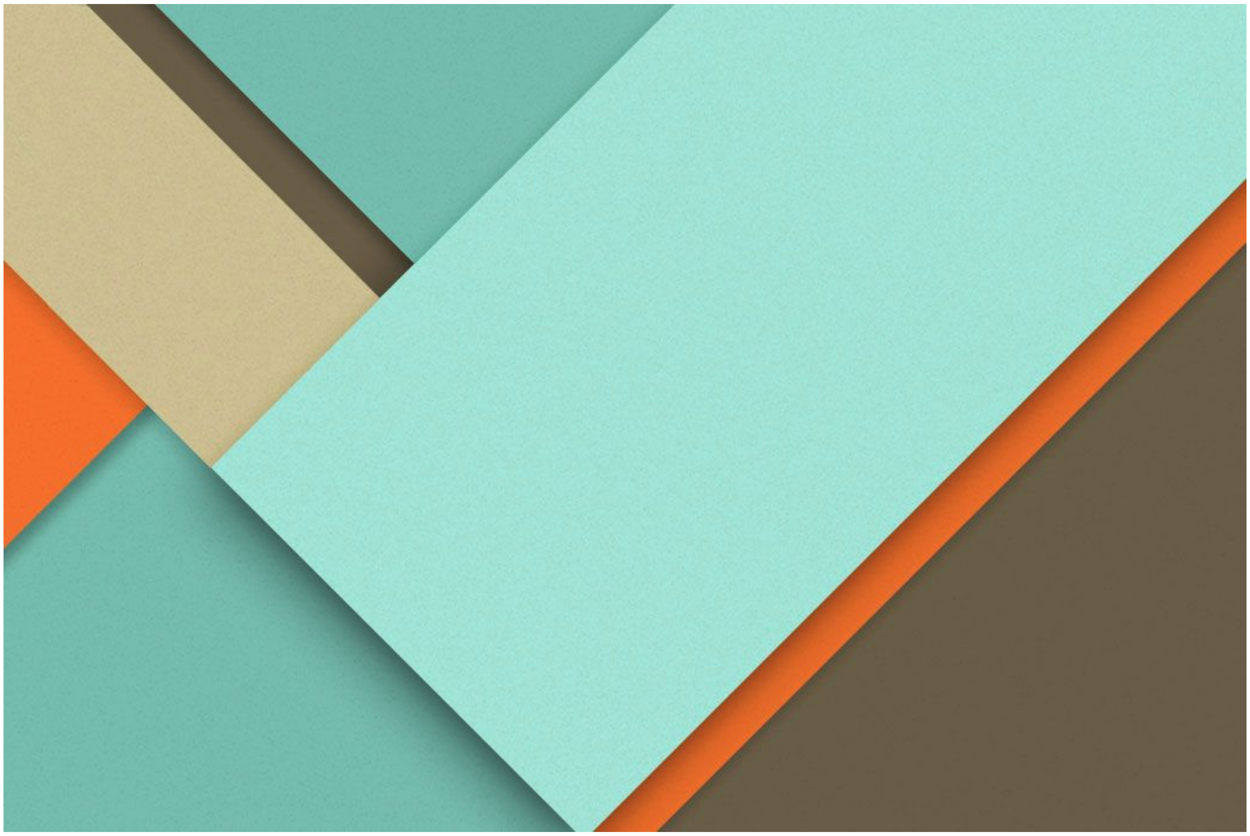


Fake News detection

Naman Agarwal

Overview

To use various natural language processing techniques and machine learning algorithms to classify fake news articles using sci-kit libraries from python.

Goals

1. Binary classification task (true, false)
2. Six-way classification task (pants on fire, false, mostly false, half-true, mostly true, true)

Prerequisites

1. Python 3.6
2. You will also need to download and install below packages after you install either python or anaconda from the steps above
    - Sklearn (scikit-learn)
    - numpy
    - Scipy
    - NLTK
    - Seaborn
    - Gensim
    - Matplotlib
    - Pandas

File Description

DataPrep.py

This file contains all the pre processing functions needed to process all input documents and texts. First we read the train, test and validation data files then performed some preprocessing like tokenizing, stemming etc. There are some exploratory data analysis is performed like response variable distribution and data quality checks like null or missing values etc.

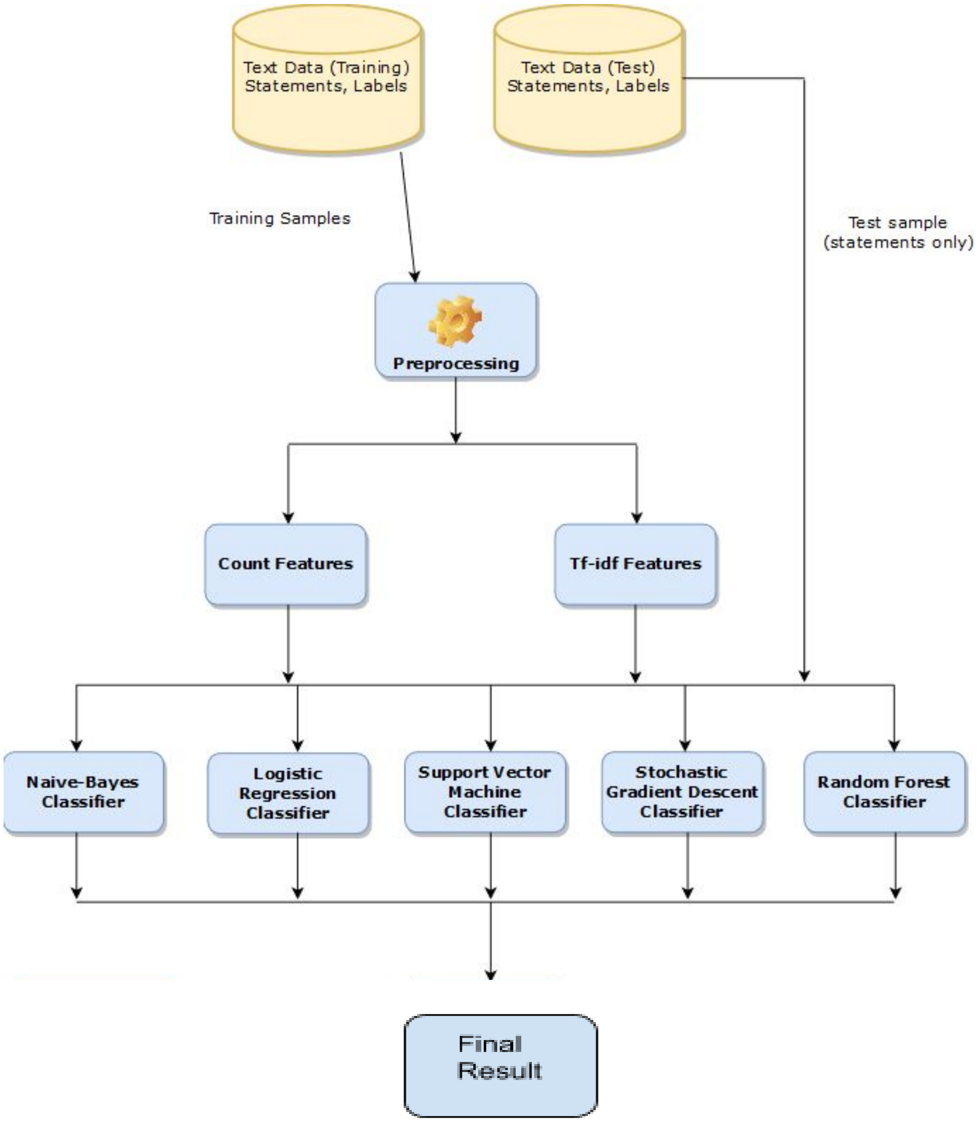
FeatureSelection.py

This file is used to perform feature extraction and selection methods from sci-kit learn python libraries. For feature selection, methods like simple bag-of-words and n-grams and then term frequency like tf-tdf weighting are used.

classifier.py

Here I have built all the classifiers for predicting the fake news detection. The extracted features are fed into different classifiers. I have used Naive-bayes, Logistic Regression, Linear SVM, Stochastic gradient descent and Random Forest classifiers from sklearn. Each of the extracted features were used in all of the classifiers. Once fitting the model, we compared the f1 score and checked the confusion matrix. After fitting all the classifiers, 2 best performing models were selected as candidate models for fake news classification. We have performed parameter tuning by implementing GridSearchCV method on these candidate models and chosen best performing parameters for these classifier. Finally selected model was used for fake news detection with the probability of truth. In Addition to this, We have also extracted the top 50 features from our term-frequency tfidf vectorizer to see what words are most important in each of the classes. We have also used Precision-Recall and learning curves to see how training and test set performs when we increase the amount of data in our classifiers.

Process Flow Chart



Performance

Below are the confusion matrices and F1 score for binary classifier using:

1. n-grams & tfidf

Naive bayes

[841 3647]

[427 5325]

f1-Score: 0.723262051071

Logistic regression

[1617 2871]

[1097 4655]

f1-Score: 0.70113000531

SVM

[2016 2472]

[1524 4228]

f1-Score: 0.67909201429

Sgdclassifier

[10 4478]

[13 5739]

f1-Score: 0.718731637053

random forest

[1979 2509]

[1630 4122]

f1-Score: 0.665720333284

2. Bag of words

Naive bayes

[2118 2370]

[1664 4088]

f1-Score: 0.669611539651

Logistic regression

[2252 2236]

[1933 3819]

f1-Score: 0.646909097798

SVM

[2260 2228]

[2246 3506]

f1-score: 0.610468748792

Sgdclassifier

[2414 2074]

[2042 3710]

f1-Score: 0.640874558778

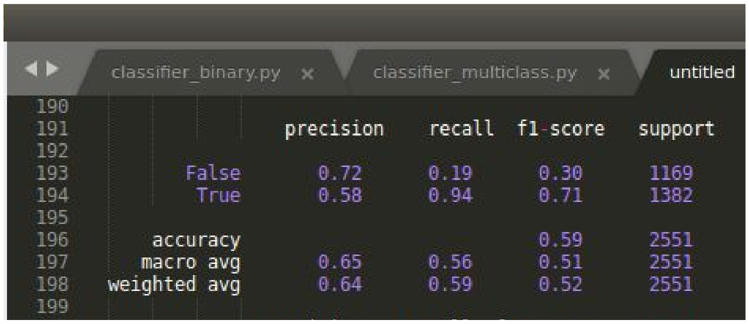
random forest classifier

[1821 2667]

[1192 4560]

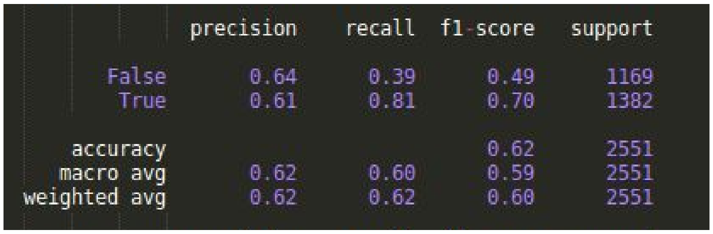
f1-Score: 0.702651511011

Below is the classification report for binary classifier:



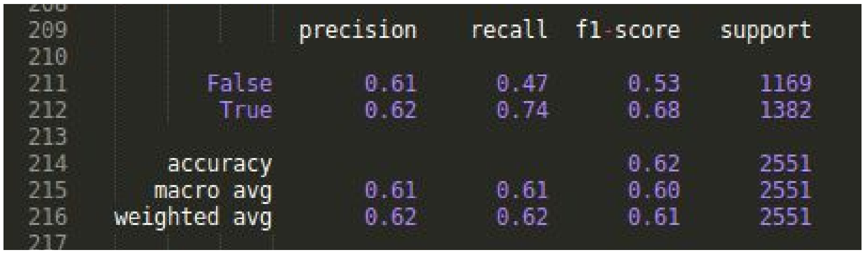
```
              precision    recall  f1-score   support

       False       0.72      0.19      0.30      1169
        True       0.58      0.94      0.71      1382

    accuracy                           0.59      2551
   macro avg       0.65      0.56      0.51      2551
weighted avg       0.64      0.59      0.52      2551
```

Naive Bayes



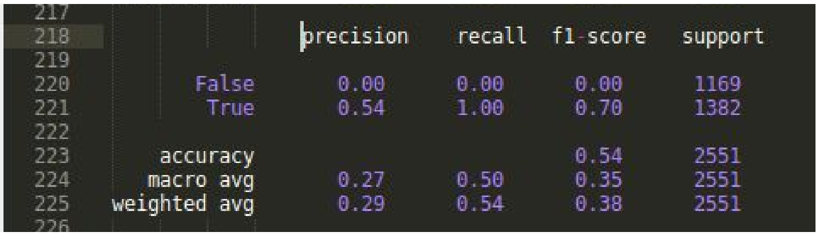
```
              precision    recall  f1-score   support

       False       0.64      0.39      0.49      1169
        True       0.61      0.81      0.70      1382

    accuracy                           0.62      2551
   macro avg       0.62      0.60      0.59      2551
weighted avg       0.62      0.62      0.60      2551
```

Logistic Regression



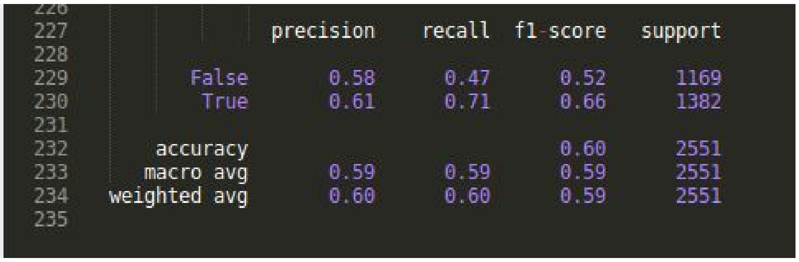
```
              precision    recall  f1-score   support

       False       0.61      0.47      0.53      1169
        True       0.62      0.74      0.68      1382

    accuracy                           0.62      2551
   macro avg       0.61      0.61      0.60      2551
weighted avg       0.62      0.62      0.61      2551
```

SVM Classifier



```
              precision    recall  f1-score   support

       False       0.00      0.00      0.00      1169
        True       0.54      1.00      0.70      1382

    accuracy                           0.54      2551
   macro avg       0.27      0.50      0.35      2551
weighted avg       0.29      0.54      0.38      2551
```

SGD Classifier



```
              precision    recall  f1-score   support

       False       0.58      0.47      0.52      1169
        True       0.61      0.71      0.66      1382

    accuracy                           0.60      2551
   macro avg       0.59      0.59      0.59      2551
weighted avg       0.60      0.60      0.59      2551
```

Random Forest

Below are the confusion matrices and F1 score for multi class classifier using:

1. Bag of words

Naive bayes

```
[[255 377 538 351  13 120]
 [275 519 582 414  15 190]
 [251 366 713 561  10 213]
 [171 294 615 593   9 280]
 [144 246 218 137  35  59]
 [115 281 456 550  18 256]]
```

f1-Score: [0.17810937 0.25459002 0.27235932 0.25946755 0.07458623 0.18311458]

Logistic regression

```
[[314 375 380 325  70 190]
 [295 586 403 332 108 271]
 [318 406 562 459  69 300]
 [251 349 449 483  52 378]
 [145 228 148 113 114  91]
 [169 324 344 443  53 343]]
```

```
f1-Score: [0.19963917 0.27494763 0.25496941 0.2346005  0.17429044
0.21102356]
```

SVM

```
[[316 356 350 295 106 231]
 [307 501 382 354 146 305]
 [340 388 506 439 103 338]
 [281 365 429 440  81 366]
 [143 193 141 102 144 116]
 [207 302 323 403  87 354]]
```

```
f1-Score: [0.19441198 0.24408367 0.23795384 0.22010213 0.1911165
0.20903274]
```

Sgdclassifier

```
[[251 379 413 299 128 184]
 [251 593 441 300 155 255]
 [245 416 627 450 118 258]
 [223 332 551 439  92 325]
 [117 212 168 104 154  84]
 [172 339 391 380  83 311]]
```

```
f1-Score: [0.1685202  0.27561171 0.26085534 0.21971733 0.1951177
0.19916655]
```

Random Forest

```
[[182 635 378 331   5 123]
 [150 913 431 330  20 151]
 [153 651 627 486   7 190]
 [102 549 446 611   7 247]
 [ 68 401 159 116  56  39]
 [ 80 490 356 497   3 250]]
```

f1-Score: [0.15225759 0.32428993 0.27751728 0.28154819 0.11931054 0.1866272]

2. N-grams and tfdif

Naive bayes

```
[[  60  398  884  292    0   20]
 [  48  577 1022  315    0   33]
 [  52  400 1169  456    0   37]
 [  17  330 1056  514    0   45]
 [  21  281  406  122    2    7]
 [  19  315  835  454    0   53]]
```

f1-Score: [0.06417057 0.2684663 0.31294388 0.24858862 0.00440801 0.05670925]

Logistic regression

```
[[201 438 557 375   0  83]
 [190 653 646 385   2 119]
 [177 437 771 574   2 153]
 [116 364 672 618   2 190]
 [107 297 242 145  14  34]
 [ 93 327 499 545   1 211]]
```

f1-Score: [0.15806962 0.28980254 0.28017956 0.26830834 0.03294609 0.17093725]

SVM

```
[[311 334 424 351  54 180]
 [310 476 497 387  83 242]
 [276 340 656 532  45 265]
 [210 295 501 565  34 357]
 [155 190 173 131 119  71]
 [177 280 365 483  35 336]]
```

```
f1-Score: [0.20104128 0.24343244 0.27733517 0.25611377 0.19658431
0.21453804]
```

Sgdclassifier

```
[[326 319 439 335  51 184]
 [324 458 501 371  85 256]
 [289 332 654 515  45 279]
 [217 285 515 550  38 357]
 [164 194 162 126 116  77]
 [193 267 375 460  34 347]]
```

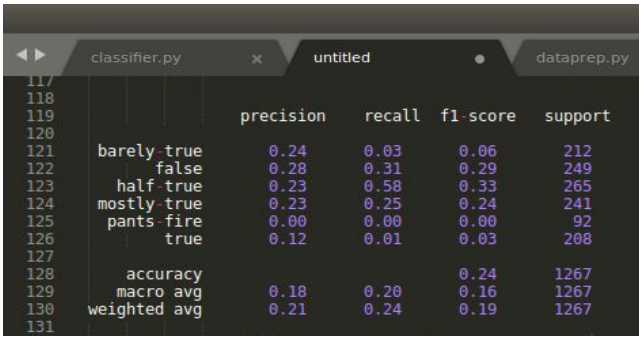
```
f1-Score: [0.20573008 0.23811802 0.27477963 0.25458604 0.19148554
0.21831309]
```

Random Forest

```
[[146 666 282 366  41 153]
 [152 902 309 415  53 164]
 [156 732 355 639  33 199]
 [115 602 310 664  25 246]
 [ 80 387 115 129  69  59]
 [101 560 205 521  23 266]]
```

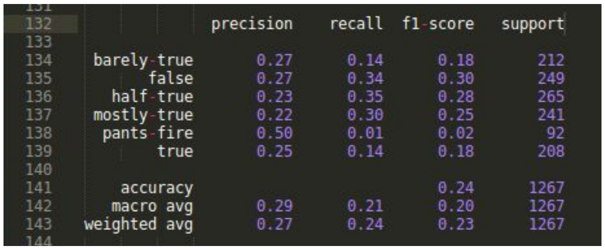
```
f1-Score: [0.12007153 0.30902868 0.19245476 0.28280816 0.1261867
0.19239621]
```

Below is the classification report for multiclass classifier:



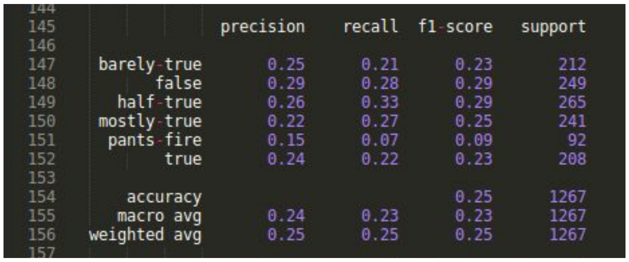
```
              precision    recall  f1-score   support

 barely-true       0.24      0.03      0.06       212
       false       0.28      0.31      0.29       249
   half-true       0.23      0.58      0.33       265
 mostly-true       0.23      0.25      0.24       241
  pants-fire       0.00      0.00      0.00        92
        true       0.12      0.01      0.03       208

    accuracy                           0.24      1267
   macro avg       0.18      0.20      0.16      1267
weighted avg       0.21      0.24      0.19      1267
```

Naive Bayes



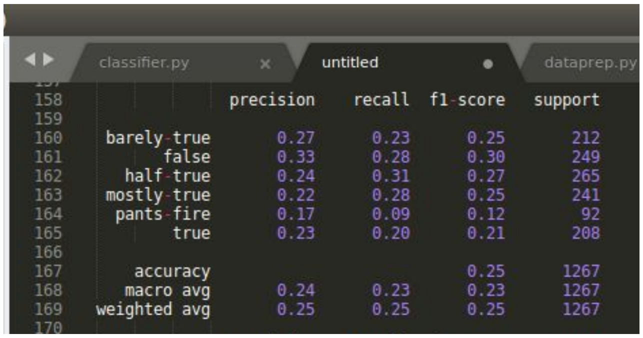
```
              precision    recall  f1-score   support

 barely-true       0.27      0.14      0.18       212
       false       0.27      0.34      0.30       249
   half-true       0.23      0.35      0.28       265
 mostly-true       0.22      0.30      0.25       241
  pants-fire       0.50      0.01      0.02        92
        true       0.25      0.14      0.18       208

    accuracy                           0.24      1267
   macro avg       0.29      0.21      0.20      1267
weighted avg       0.27      0.24      0.23      1267
```

Logistic Regression



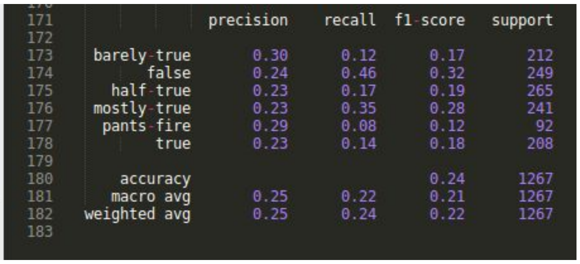
```
              precision    recall  f1-score   support

 barely-true       0.25      0.21      0.23       212
       false       0.29      0.28      0.29       249
   half-true       0.26      0.33      0.29       265
 mostly-true       0.22      0.27      0.25       241
  pants-fire       0.15      0.07      0.09        92
        true       0.24      0.22      0.23       208

    accuracy                           0.25      1267
   macro avg       0.24      0.23      0.23      1267
weighted avg       0.25      0.25      0.25      1267
```

SVM Classifier



```
              precision    recall  f1-score   support

 barely-true       0.27      0.23      0.25       212
       false       0.33      0.28      0.30       249
  half-true        0.24      0.31      0.27       265
 mostly-true       0.22      0.28      0.25       241
  pants-fire       0.17      0.09      0.12        92
        true       0.23      0.20      0.21       208

    accuracy                           0.25      1267
   macro avg       0.24      0.23      0.23      1267
weighted avg       0.25      0.25      0.25      1267
```

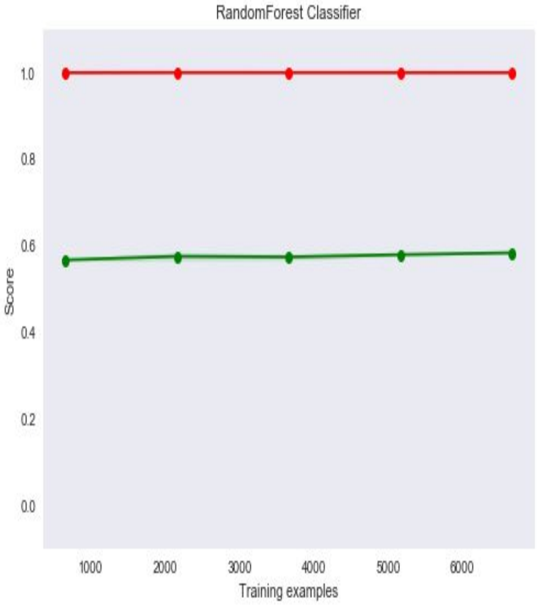
SGD Classifier

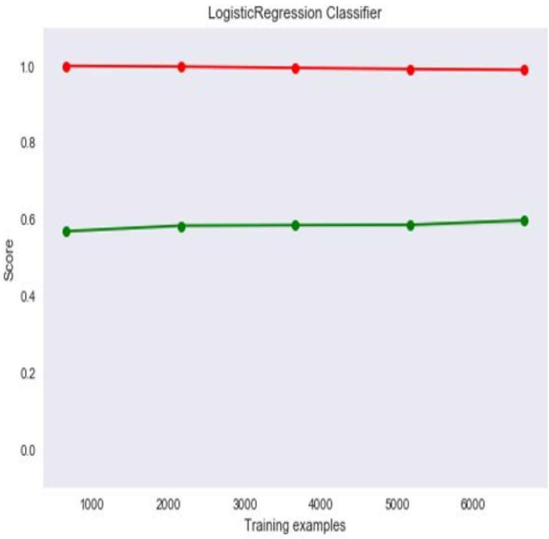


```
              precision    recall  f1-score   support

 barely-true       0.30      0.12      0.17       212
       false       0.24      0.46      0.32       249
  half-true        0.23      0.17      0.19       265
 mostly-true       0.23      0.35      0.28       241
  pants-fire       0.29      0.08      0.12        92
        true       0.23      0.14      0.18       208

    accuracy                           0.24      1267
   macro avg       0.25      0.22      0.21      1267
weighted avg       0.25      0.24      0.22      1267
```

Random Forest

The best classifiers were Logistic Regression and Random Forest and the learning curves for these classifiers are as follows:





References

1. Jones E, Oliphant T, Peterson P, others. SciPy: Open source scientific tools for Python [Internet]. 2001. Available from: "http://www.scipy.org/"
2. Oliphant TE. A guide to NumPy. Vol. 1. Trelgol Publishing USA; 2006.
3. McKinney W, others. Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference. 2010. p. 51–6.
4. Van Rossum G, Drake Jr FL. Python tutorial. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands; 1995.
5. Pedregosa F, Varoquaux, Ga"el, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of machine learning research. 2011;12(Oct):2825–30.
6. Bird, Steven, Edward Loper and Ewan Klein (2009).Natural Language Processing with Python. O'Reilly Media Inc.
7. Hunter JD. Matplotlib: A 2D graphics environment. Computing in science &amp; engineering. 2007;9(3):90–5.