Suheyla Iyimaya

Niranjan Balasubramanian

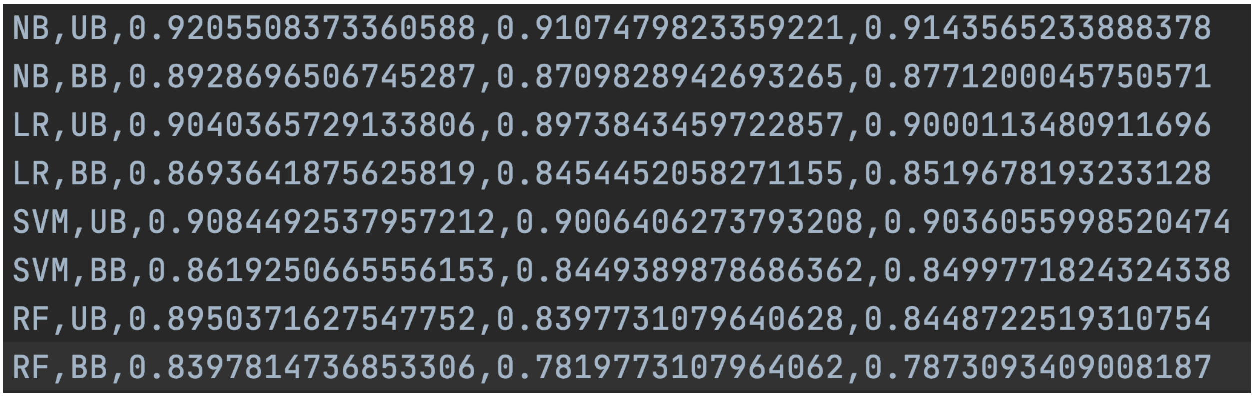
CSE 352

03/16/2021

Assignment 3: Text Classification

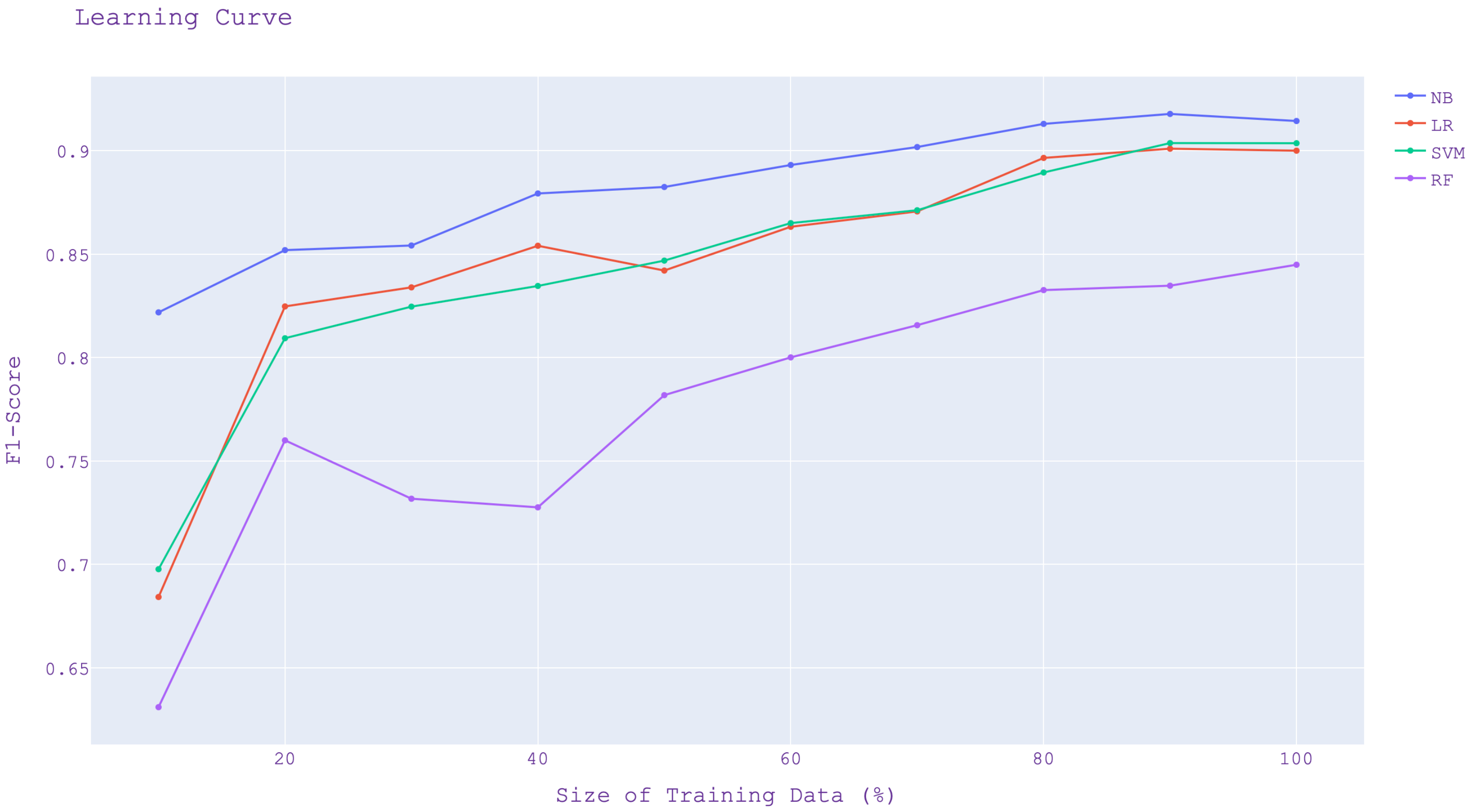
- **Basic Comparison with Baselines**

  - Output of all 4 methods with 2 configurations (UB_BB.py):



```
NB,UB,0.9205508373360588,0.9107479823359221,0.9143565233888378
NB,BB,0.8928696506745287,0.8709828942693265,0.8771200045750571
LR,UB,0.9040365729133806,0.8973843459722857,0.9000113480911696
LR,BB,0.8693641875625819,0.8454452058271155,0.8519678193233128
SVM,UB,0.9084492537957212,0.9006406273793208,0.9036055998520474
SVM,BB,0.8619250665556153,0.8449389878686362,0.8499771824324338
RF,UB,0.8950371627547752,0.8397731079640628,0.8448722519310754
RF,BB,0.8397814736853306,0.7819773107964062,0.7873093409008187
```

  - Learning curve:



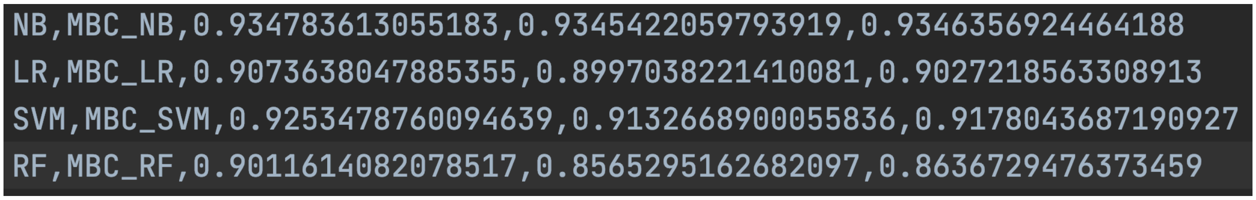
- Explanation:

    - For all four classification methods, we can clearly see that as more training data is provided to the classifier, the better their f1 score becomes. For Naive Bayes and Logistic Regression, after 80% of the training examples are given, not that huge of a change happens. Therefore, adding more training data will not be that much of a benefit for us. For SVM and Rain Forests, however, there is still slightly bigger improvement between 80% and 100% of given training data. Thus, adding more training data for those two classifiers may help us get a better f1 score over time.[1] Naive Bayes seems to have the best accuracy overall. All four classification methods use Count Vectorization. Naive Bayes seems to have the best f1 score with Count Vectorization. Random Forests on the other hand is not doing as well as the other classification methods.

- **My best configuration**

    - For each of four methods (NB, LR, SVM, and RF), pick at least two design choices and output evaluation results to a file(MBC_exploration.py)

        - Output:



```
NB,MBC_NB,0.934783613055183,0.9345422059793919,0.9346356924464188
LR,MBC_LR,0.9073638047885355,0.8997038221410081,0.9027218563308913
SVM,MBC_SVM,0.9253478760094639,0.9132668900055836,0.9178043687190927
RF,MBC_RF,0.9011614082078517,0.8565295162682097,0.8636729476373459
```

    - Explanation:

        - After observing the performances of the basic versions of the all four classification methods, I added some improvements for each classification method to increase their accuracy, precision and f1 scores. I noticed that Naive Bayes and Logistic Regression work best with Count Vectorizer while Random Forests and SVM have better evaluation scores with TFIDF. I also added feature selection using Select From Model method, which increased their performances,

[1] Validation Curves: Plotting Scores to Evaluate Models
Scikit, scikit-learn.org/stable/modules/learning_curve.html.

- For all four classification methods, I used stemming, ignoring stop words and lower casing words by default. In conclusion, my improved Naive Bayes gave me the highest evaluation scores, which is why I provided it as my final MBC.

Works Cited

Validation Curves: Plotting Scores to Evaluate Models
Scikit, scikit-learn.org/stable/modules/learning_curve.html.