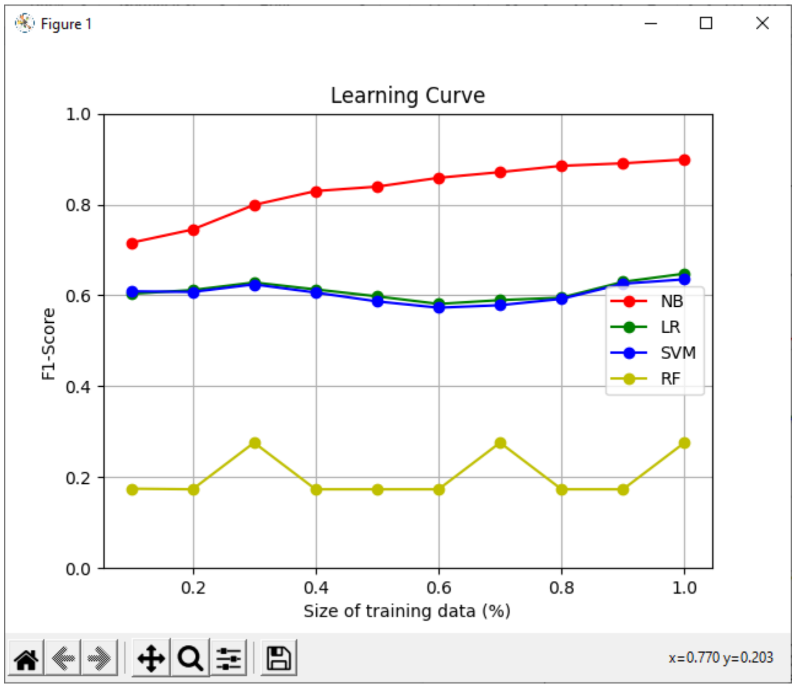
Gabriello Lima, 112803276

Part 1 Output:

```
NB,UB,0.9,0.9,0.9

NB,BB,0.86,0.86,0.86

LR,UB,0.65,0.65,0.65

LR,BB,0.76,0.76,0.76

SVM,UB,0.64,0.64,0.64

SVM,BB,0.75,0.75,0.75

RF,UB,0.28,0.28,0.28

RF,BB,0.27,0.27,0.27
```

IT IS IMPORTANT TO NOTE THAT THE LEARNING CURVE (as shown below)DOES NOT GO TO THE OUTPUT FILE, RATHER IT POPS UP AS A SEPERATE WINDOW IN PYTHON.



Part 1 Explanation:

- I believe my NB is extremely high because I'm using Multinomial Bayes, hence its able to easily identify the varying classes, especially with more data. My LR and SVM are awfully similar because I'm using linearSVC, so I'm fairly confident they're using similar regression formulas. My SVM is tuned to "ovr" or one vs rest, so it's kinda difficult to identify a class unless the given sample is really specific. My random forest is just a mess. I used Random Forest Classifier and tried fine-tuning the number of trees and features to be based on the number of classes we have, but I couldn't seem to find a decent configuration.

Part 2 Explanation:

- For this part, my Multinomial Naive Bayes clearly takes the cake as the winner. The other algorithms got close as I messed around with CountVectorizer, but NB was the clear winner. Aside from fine tuning the hyperparameters, mostly everything was kept the same as part 1, i.e. I skipped headers and used stemming to parse the input. I was able to improve the F score from part 1 by messing around with the alpha value (as shown by the list below). The lower the alpha value, the better (around a .2 -.3 range was the best). Why is this so? Essentially I'm modifying the smoothing parameter to preserve information in probability tables. I.e. a probability that would otherwise be zero would be increased by the alpha, preserving the probability information in the table. I also implemented english stopwords, accent stripping ascii, and a max_df to take out some of the language "clutter."

Part 2 Output:

NB,(Multinomial, Alpha=.25, Stopwords, Strip Accents Ascii, max_df=.8),0.93,0.93,0.93

LR,(Dual = True, solver = liblinear, classweight = 'balanced', Stopwords, Strip Accents Ascii,

max_df=.8),0.88,0.88,0.88

SVM,(C = .01, classweight = 'balanced', Stopwords, Strip Accents Ascii,

max_df=.8),0.87,0.87,0.87

RF,(class_weight='balanced', ccp_alpha=.75, Stopwords, Strip Accents Ascii,

max_df=.8),0.27,0.27,0.27

https://medium.com/syncedreview/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation-4f5271768ebf

https://link.springer.com/chapter/10.1007/978-3-642-23199-5_4

Alpha	Corresponding Score
0	.92
.1	.92
.2	.93
.3	.93
.4	.92
.5	.92

.6	.91
.7	.91
.8	.91
.9	.9
1	.90 (Part 1)