CS 57300 Data Mining Assignment 3

Shuang Wu (wu1716@purdue.edu)

Environment

See `README.md`. Generally, the Python version we use with Poetry is `~3.9`.

In [ ]:

```
TEST_SET = 'testSet.csv'
TRAINING_SET = 'trainingSet.csv'
MODE_LR = 1
MODE_SVM = 2
```

Preprocessing

In [ ]:

```
import pandas as pd
import warnings

warnings.simplefilter(action='ignore', category=pd.errors.PerformanceWarning)

%run -i preprocess-assg3.py
```

```
Mapped vector for female in column gender: [1]
Mapped vector for Black/African American in column race: [0, 1, 0, 0]
Mapped vector for Other in column race_o: [0, 0, 0, 0]
Mapped vector for economics in column field: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Implement Logistic Regression and Linear SVM

In [ ]:

```
%run -i lr_svm.py {TRAINING_SET} {TEST_SET} {MODE_LR}
```

```
Training Accuracy LR: 0.66
Testing Accuracy LR: 0.66
```

If we set the learning rate (step size) to 0.001 and iterations to 3500. The accuracy for both sets can reach 0.78.

In [ ]:

```
%run -i lr_svm.py {TRAINING_SET} {TEST_SET} {MODE_SVM}
```

```
Training Accuracy SVM: 0.56
Testing Accuracy SVM: 0.55
```

If we set the learning rate (step size) to 0.001 and iterations to 2000. The accuracy for both sets can also reach 0.78.

Learning Curves and Performance Comparison

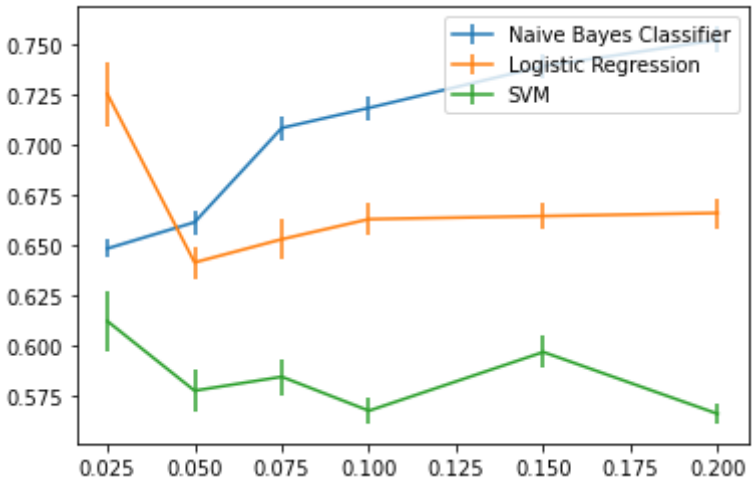
Formulate Hypothesis

According to the trait of the preprocessed data, it is likely that the data is not linear-separable. Thus, we believe the performance of NBC will be better than the performance of LR and linear-SVM.

In [ ]:
```
%run -i cv.py
```

```
t_frac: 0.025
[Naive Bayesian Classifier] Test Accuracy: 0.6484615384615384
[Naive Bayesian Classifier] CV Average Accuracy: 0.6494230769230769
[Naive Bayesian Classifier] CV Standard Error: 0.004734441976453226
[Logistic Regression] Test Accuracy: 0.7253846153846154
[Logistic Regression] CV Average Accuracy: 0.6688461538461539
[Logistic Regression] CV Standard Error: 0.015801561603601907
[SVM] Test Accuracy: 0.6123076923076923
[SVM] CV Average Accuracy: 0.5569230769230769
[SVM] CV Standard Error: 0.014819425920574135
t_frac: 0.05
[Naive Bayesian Classifier] Test Accuracy: 0.6615384615384615
[Naive Bayesian Classifier] CV Average Accuracy: 0.6696153846153846
[Naive Bayesian Classifier] CV Standard Error: 0.005957194447092801
[Logistic Regression] Test Accuracy: 0.6415384615384615
[Logistic Regression] CV Average Accuracy: 0.6590384615384615
[Logistic Regression] CV Standard Error: 0.007896566926884695
[SVM] Test Accuracy: 0.5776923076923077
[SVM] CV Average Accuracy: 0.5538461538461539
[SVM] CV Standard Error: 0.010444981241538667
t_frac: 0.075
[Naive Bayesian Classifier] Test Accuracy: 0.7084615384615385
[Naive Bayesian Classifier] CV Average Accuracy: 0.6975
[Naive Bayesian Classifier] CV Standard Error: 0.005839634522891537
[Logistic Regression] Test Accuracy: 0.6530769230769231
[Logistic Regression] CV Average Accuracy: 0.6688461538461539
[Logistic Regression] CV Standard Error: 0.009702680651980647
[SVM] Test Accuracy: 0.5846153846153846
[SVM] CV Average Accuracy: 0.5573076923076923
[SVM] CV Standard Error: 0.009100848520164599
t_frac: 0.1
[Naive Bayesian Classifier] Test Accuracy: 0.7184615384615385
[Naive Bayesian Classifier] CV Average Accuracy: 0.7092307692307692
[Naive Bayesian Classifier] CV Standard Error: 0.005780757715327088
[Logistic Regression] Test Accuracy: 0.6630769230769231
[Logistic Regression] CV Average Accuracy: 0.6755769230769231
[Logistic Regression] CV Standard Error: 0.008235844948263312
[SVM] Test Accuracy: 0.5676923076923077
[SVM] CV Average Accuracy: 0.5613461538461537
[SVM] CV Standard Error: 0.00636332176660315
t_frac: 0.15
[Naive Bayesian Classifier] Test Accuracy: 0.7392307692307692
[Naive Bayesian Classifier] CV Average Accuracy: 0.7249999999999999
[Naive Bayesian Classifier] CV Standard Error: 0.006153846153846151
[Logistic Regression] Test Accuracy: 0.6646153846153846
[Logistic Regression] CV Average Accuracy: 0.6782692307692308
[Logistic Regression] CV Standard Error: 0.006595060461072898
[SVM] Test Accuracy: 0.5969230769230769
[SVM] CV Average Accuracy: 0.5786538461538461
[SVM] CV Standard Error: 0.008179070200769306
t_frac: 0.2
[Naive Bayesian Classifier] Test Accuracy: 0.7523076923076923
```

```
[Naive Bayesian Classifier] CV Average Accuracy: 0.7332692307692307
[Naive Bayesian Classifier] CV Standard Error: 0.0066230389528081565
[Logistic Regression] Test Accuracy: 0.6661538461538462
[Logistic Regression] CV Average Accuracy: 0.6661538461538462
[Logistic Regression] CV Standard Error: 0.00749852056414381
[SVM] Test Accuracy: 0.5661538461538461
[SVM] CV Average Accuracy: 0.5713461538461538
[SVM] CV Standard Error: 0.004729752861427062
```



Test Hypothesis

From the plot, we can see that NBC has the best performance and smallest standard errors with the given dataset and parameters. NBC stably increases its accuracy with more training data. LR increases slowly its accuracy with more training data. On the other hand, SVM does not perform stably with different sizes of training data. In sum, the performance of NBC is the best and the most stable among the three classifiers. Namely, it is possible that the data is not linear-separable, and thus the performance of LR and linear-SVM is worse than the one of NBC.

The observed data does support our hypothesis.