Carrie McClanahan and David Wade

P2: Machine Learning Applications (Part 3) - Online News Popularity

Methodology

LOGISTIC REGRESSION / CLASSIFICATION STEPS

1. In order to classify based on low vs. high shares, create a new variable in the excel version of the data file called low_high_shares which separates the shares by low (unpopular) and high (popular). In excel, =IF(shares <= 1400,"low","high").
    a. If low is 1 - 10,000, distribution will be too skewed since there will be too many low values. When we separate the dataset, if it is too skewed then this could be a problem.

```
Value    Count   Percent
  low    37429    94.41%
 high     2215     5.59%
```

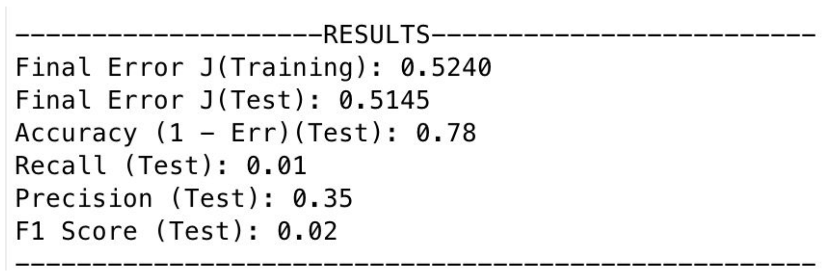
    b. For low being 1 - 3000 shares, distribution is less skewed (low_high_shares)

```
Value    Count   Percent
  low    30718    77.48%
 high     8926    22.52%
```

2. Use strcmp to make the y array, so that y = 1 if low_high_shares = ‘high’ and y = 0 if low_high_shares is not ‘high’ (= ‘low’)
3. Split data into 70% training 30% testing. Take random 70% of training samples and 30% testing
4. Do feature scaling and mean normalization on all data using the average and standard deviation from the training set

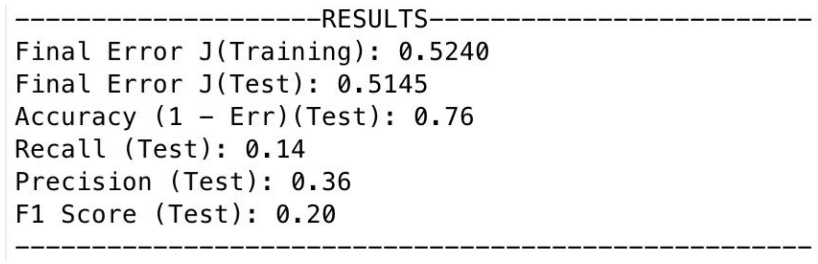
FEATURE SELECTION

5. FIRST ATTEMPT: 9 features with top correlation coefficients.
    a. y = 0 for 1-3000 shares, threshold = 0.5: Very low test set F1 score, underfitting.



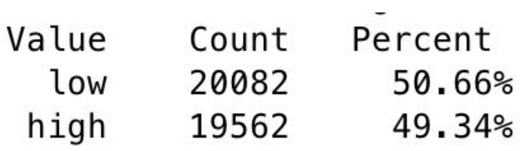
```
---------------------RESULTS-------------------------
Final Error J(Training): 0.5240
Final Error J(Test): 0.5145
Accuracy (1 - Err)(Test): 0.78
Recall (Test): 0.01
Precision (Test): 0.35
F1 Score (Test): 0.02
-----------------------------------------------------
```

b. y = 0 for 1-3000 shares, threshold = 0.3: Makes F1 score higher, still underfitting.



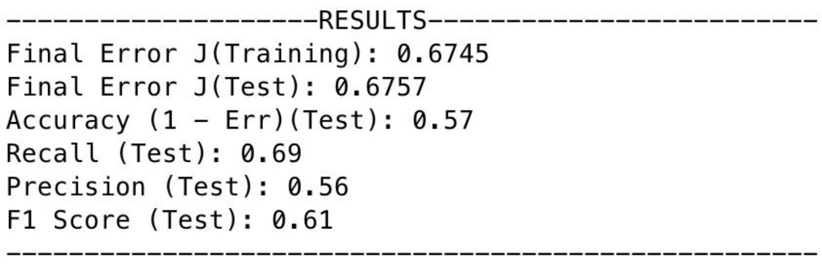
```
---------------------RESULTS-------------------------
Final Error J(Training): 0.5240
Final Error J(Test): 0.5145
Accuracy (1 - Err)(Test): 0.76
Recall (Test): 0.14
Precision (Test): 0.36
F1 Score (Test): 0.20
-----------------------------------------------------
```

c. After seeing that the 3000 shares dividing line caused too much skewness for logistic regression, we created a new variable low_high_shares2 which we will use as our main classification. For low 1-1400 shares (the median 1400 as the dividing point so that we can say the article is below average vs. above average), the classes are about 50/50.



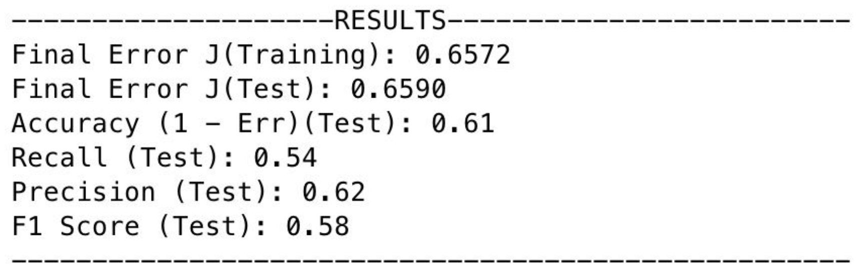
```
Value     Count   Percent
  low     20082    50.66%
 high     19562    49.34%
```

d. y = 0 for 1-1400 shares, threshold 0.5: Much better F1 score, still underfitting



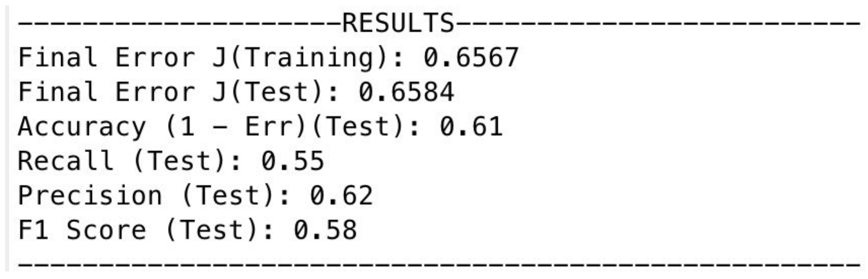
```
---------------------RESULTS-------------------------
Final Error J(Training): 0.6745
Final Error J(Test): 0.6757
Accuracy (1 - Err)(Test): 0.57
Recall (Test): 0.69
Precision (Test): 0.56
F1 Score (Test): 0.61
-----------------------------------------------------
```

6. SECOND ATTEMPT: Add 3 features of our choosing
    a. Low_high_shares2, threshold 0.5: Worse F1, better accuracy



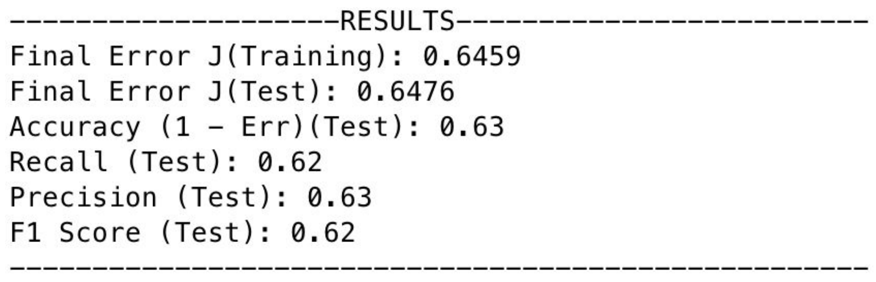
```
--------------------RESULTS-------------------------
Final Error J(Training): 0.6572
Final Error J(Test): 0.6590
Accuracy (1 - Err)(Test): 0.61
Recall (Test): 0.54
Precision (Test): 0.62
F1 Score (Test): 0.58
----------------------------------------------------
```

7. THIRD ATTEMPT: Add 3 features with next highest correlation coefficients
    a. low_high_shares2, threshold = 0.5: Unchanging F1



```
--------------------RESULTS-------------------------
Final Error J(Training): 0.6567
Final Error J(Test): 0.6584
Accuracy (1 - Err)(Test): 0.61
Recall (Test): 0.55
Precision (Test): 0.62
F1 Score (Test): 0.58
----------------------------------------------------
```

8. FOURTH ATTEMPT: Add next 3 features with next highest correlation coefficients (19 features including x0)
    a. low_high_shares2, threshold = 0.5: Better F1 score and accuracy



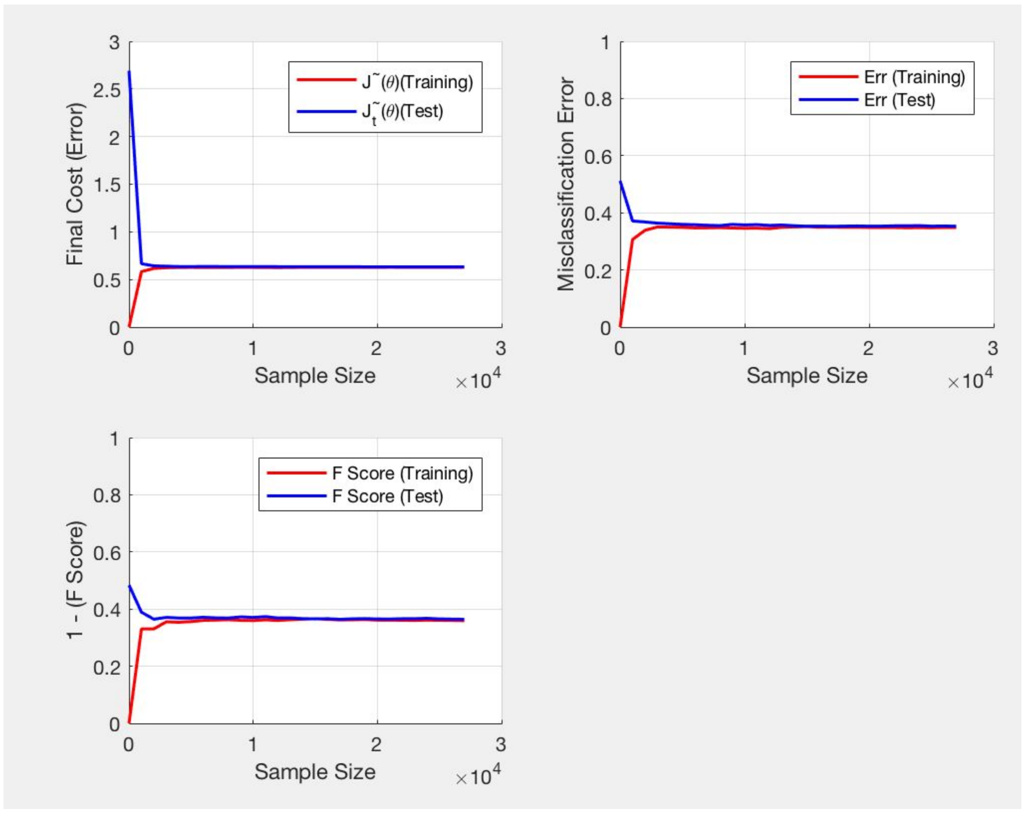
```
--------------------RESULTS-------------------------
Final Error J(Training): 0.6459
Final Error J(Test): 0.6476
Accuracy (1 - Err)(Test): 0.63
Recall (Test): 0.62
Precision (Test): 0.63
F1 Score (Test): 0.62
----------------------------------------------------
```

REGULARIZATION

9. Tried increasing lambda (regularization) - does not help performance since our problem is underfitting, not overfitting.

LEARNING CURVES:

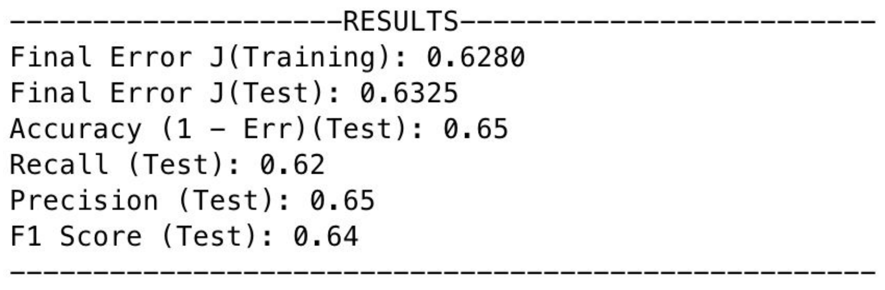
10. Made learning curve plots using fminunc, not our own classification with gradient descent. Increasing sample size does not help since we have an underfitting problem. The training and test errors end up being basically the same.



PCA + LOGISTIC REGRESSION

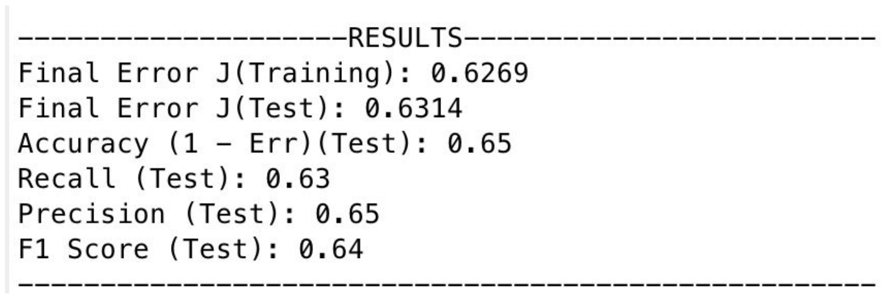
11. Applied PCA in order to choose features with highest variance and reduce the amount of features. Changed r iteratively to see which amount of features gave highest F1 score. Highest could get was 0.64 with a minimum of 42 features. Retained 100% of variance. 100% of variance within 8 features, 99.4% within 4 features. There seems to be too much redundancy in our features.

a. FINAL MODEL: With r = 42 (42 features), obtained highest F1 score possible with minimal features.



```
--------------------RESULTS-------------------------
Final Error J(Training): 0.6280
Final Error J(Test): 0.6325
Accuracy (1 - Err)(Test): 0.65
Recall (Test): 0.62
Precision (Test): 0.65
F1 Score (Test): 0.64
----------------------------------------------------
```

b. With r = 59 (All features)



```
--------------------RESULTS-------------------------
Final Error J(Training): 0.6269
Final Error J(Test): 0.6314
Accuracy (1 - Err)(Test): 0.65
Recall (Test): 0.63
Precision (Test): 0.65
F1 Score (Test): 0.64
----------------------------------------------------
```

POLYNOMIAL EXPANSION

12. Tried adding polynomial features up to degree 4 with 15 of the top features from PCA (15 features with greatest variance). Performance did not improve significantly if at all. F1 score about the same as using features as is.

NAIVE BAYES

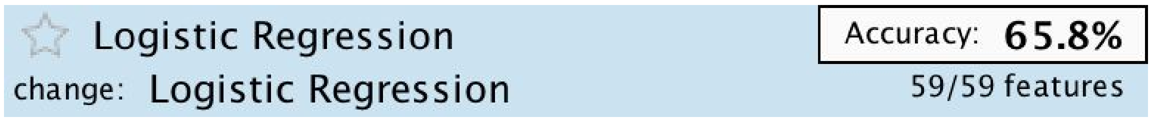
13. After implementing Naive Bayes on the training set with all the features, the predictions for the test set are not very accurate or consistent. With seed at rng(10), F1 score was only 0.19. Moving the seed to rng(11) gave a better F1, 0.50. This shows how inconsistent it is depending on what training and test set is used. The precision is much better than the recall. The Naive Bayes model predicts a lot of data points as being low but not many high. Logistic Regression seems better for this dataset.

```
--------------NAIVE BAYES RESULTS-------------------
Accuracy (1 - Err)(Test): 0.54
Recall (Test): 0.11
Precision (Test): 0.68
F1 Score (Test): 0.19
----------------------------------------------------
```

```
--------------NAIVE BAYES RESULTS-------------------
Accuracy (1 - Err)(Test): 0.60
Recall (Test): 0.40
Precision (Test): 0.65
F1 Score (Test): 0.50
----------------------------------------------------
```

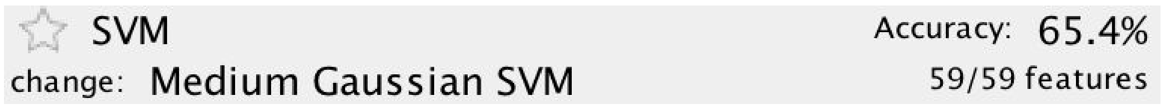
CLASSIFICATION LEARNER

14. Using Classification Learner with all features:
    a. Best accuracy result using logistic regression and all features, no PCA:



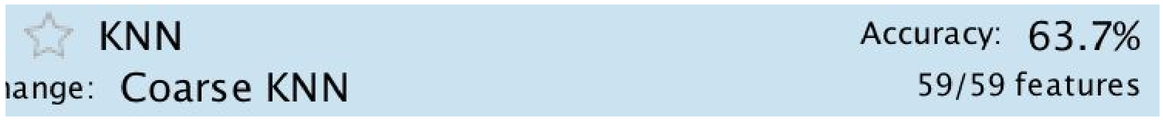
Logistic Regression
change: Logistic Regression
Accuracy: 65.8%
59/59 features

    b. Best SVM result with Medium Gaussian SVM (kernel scale 7.7, box constraint level 1, standardized data)



SVM
change: Medium Gaussian SVM
Accuracy: 65.4%
59/59 features

    c. Best KNN result with Coarse KNN (100 nearest neighbors, standardized data)



KNN
ıange: Coarse KNN
Accuracy: 63.7%
59/59 features