Michael Vanderlyn

9/3/19

CS 484

HW 2 Report

Name Registered on miner web-site:

ClassE

Rank & F1 score for your submission:

Rank: 18

F1: 0.74

Your Approach:

My approach was much different than the first project. This time I realized I needed to start much earlier, and do a lot of research into algorithms that I could use to do the feature selection, dimensionality reduction, and classification. So I made a google doc and just started reading up on several different features selection/classifiers. I didn't really look much into dimensionality reduction because I already knew about truncated singular value decomposition which is what is recommended to use for sparse data. For feature selection recursive feature elimination with and without cross validation, and chi^2 analysis with K-best selection all caught my eye. I then looked into classifiers and saw that Bernoulli Naive Bayes was designed for binary features so I wanted to try using this classifier. I then looked into decision trees vs. neural networks. I decided to go with decision trees because they seemed easier to implement at first, especially with the binary data for features. I also realized I needed to account for the imbalanced training data. I knew I needed to be careful and find some way to avoid oversampling or undersampling. I found an algorithm implemented in sklearn, SMOTE, which seemed to be pretty good at accounting for imbalances in data by oversampling the underrepresented class. So I wanted to try implementing SMOTE as well.

After this initial research I broke down the project into tasks and planned which days I wanted to work on which tasks, and in which order so that I would finish in time. I started with reading the data into sparse form, then feature engineering, then classifier fitting, then predictions/f1-scoring, and finally running the best classifier on the test data. I knocked these tasks out and was able to finish up earlier enough to tweak some of the parameters in my project until I got my max score.

Your methodology of choosing the approach and associated parameters:

After finishing all the basic structure of the app my approach was to try to fit different combinations of preprocessed data on both classifiers. I tried all the different feature selections, and started to try to combine them with each other as well. I also was making a copy of each preprocessed data and ran SMOTE on it to account for the imbalances on data, after generating all these different data sets I fit each classifier to them, and predicted the class of the original training data without the ranks. I then calculated the f1-score on all these different classifiers trained on

differently pre-processed data and was able to determine that chi^2 gave me the best accuracy, so for my final submission I made the naive bayes trained on the data set with chi^2 analysis and k-best feature selection predict the test data. I have below a table of the different combinations. For timing I run all the processes together, and pickling for the more time intensive steps, so with pickling, the whole program runs in 34 seconds. With nothing pickled, the TSVD, and RFE take around 2-3 minutes, and the RFECV after the RFE takes about 30-35 seconds.

Type/which dataset	F1 Score
Naive Bayes on original data no SMOTE	0.6644778436441713
Naive Bayes on original data w/ SMOTE	0.8079344166300688
Naive Bayes on TSVD, no SMOTE	0.8129277329439459
Naive Bayes on TSVD, SMOTE	0.8472222222222222
Naive Bayes on RFE(100 features), no SMOTE	0.8958333333333334
Naive Bayes on RFE, RFECV(92 features), SMOTE	0.8888888888888888
Naive Bayes on RFE, RFECV(92 features), no SMOTE	0.8934583422118048
Naive Bayes on K-best features of Chi^2 analysis	0.822430570353008
Decision Tree of RFECV with SMOTE	1
Decision Tree of RFECV without SMOTE	1
Decision Tree on original data	1
Decision Tree on original data with SMOTE	1
Decision Tree on TSVD data without SMOTE	1
Decision Tree on TSVD data with SMOTE	1

Final Thoughts:

Even after all my efforts to avoid oversampling, I still am pretty sure I had some issues with oversampling in the Decision trees. I also need to start submitting my classifications earlier, I waited to the last day to try submitting and realized that my f1 scored were very inaccurate compared to the ones on miner. I realized I had an overfitting issue and that is why I eventually decided to use the

chi^2 analysis with k-best feature selection. This probably performed so well because it didn’t allow for nearly as much overfitting as the other dimension reduction/feature selection methods did.