Comparing SVM and Naïve Bayes for Sentiment Analysis of Stack Overflow Comments

Tyler Sattler

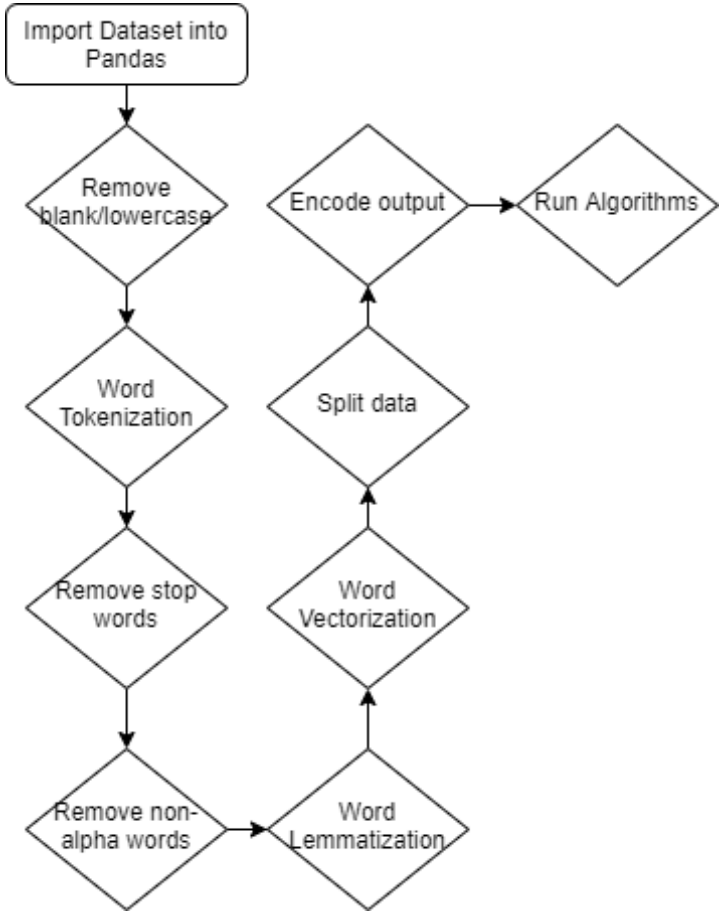
4/8/2020

Introduction

- Classifying sentiment of Stack Exchange Comments
  - i.e. Positive, Neutral, Negative polarity
- “Support-vector networks. Machine Learning”
  - Used the SVM algorithm for prediction
- “Multinomial Naive Bayes for Text Categorization Revisited”
  - Implemented a multinomial naïve bayes using TF-IDF weights

Experimental set-up

- Dataset:
  - 70% train/ 30% test
  - Text data from Stack Overflow comments
- Tokenized, cleaned, lemmatized text
- Vectorized using TF-IDF method
- Used scikit-learn library as baseline comparison



Results: Baseline

- Using scikit-learn library
- SVM outperformed Naïve Bayes by 7.54%
- Most misclassified occurred when classifying negative when it is neutral

Naïve Bayes Classifier

	Neutral	Negative	Positive
Neutral	163	33	3
Negative	176	395	83
Positive	21	80	372

Accuracy: 70.13%

SVM Classifier

	Neutral	Negative	Positive
Neutral	226	40	11
Negative	128	436	79
Positive	6	80	368

Accuracy: 77.67%

Results: My Implementation

- Wrote code from scratch
  - Naïve Bayes: numpy
  - SVM: Cvxpy for convex optimization
- Naïve Bayes outperformed baseline by 4.68%
- SVM underperformed baseline by 17.56%
  - Most likely due to improper implementation

Naïve Bayes Classifier

	Neutral	Negative	Positive
Neutral	267	91	15
Negative	76	329	45
Positive	17	88	398

Accuracy: 74.96%

SVM Classifier

	Neutral	Negative	Positive
Neutral	197	107	31
Negative	133	313	140
Positive	30	88	287

Accuracy: 60.11%

Conclusion

- Summary
  - Compared performance of SVM vs Naïve Bayes for text classification and sentiment analysis of Stack Overflow comments
  - Predicted polarity (negative, neutral, positive) with an accuracy of 74.96% (NB) and 60.11% (SVM)
- Key insights
  - SVM & Naïve Bayes can achieve decent accuracy for less computational cost
  - Incorporating TF-IDF score into Naïve Bayes can increase accuracy compared to standard libraries