AskReddit

Detecting troll questions on a subreddit

Nandakishore S Menon (IMT2019057)
Rachna S Kedigehalli (IMT2019069)

Preprocessing

Preprocessing

- For the sake of experimenting, we removed all preprocessing and tried training a model.
- Used Count Vectorizer and experimented with its parameter.
- Using Count Vectorizer with ngrams (1, 3): unigram, bigram and 3-gram gave us our highest score with Logistic regression.
- Tried using Tf-Idf instead of Count Vectorizer, but it gave a lower score.

Preprocessing

- We tried adding each preprocessing we had previously one-by-one, but they all gave a lower score.
- We also tried to use Word2Vec. We kept it running for 12 hrs and it still didn’t finish, so we abandoned it.

Training

Models

- Logistic Regression
- GridsearchCV was overfitting the training data, so we tried changing parameters manually.
- Used “lbfgs” solver with “l2” penalty over 1500 iterations to get our best score.

- Naive Bayes
- Tried Multinomial Naive Bayes, with different alpha values.
- Gave a decent score, but less than that of logistic regression.

Models Tried

- Tried multilayer perceptron (took 4 hrs to train).
  - Gave a relatively low score (0.49708 public score).
- Tried XGBoost and got a low score (0.16566 public score).
- Tried training AdaBoost and GradientBoostingClassifier, but they were taking too long to train.

- For the final submission, we used LogisticRegression(solver="lbfgs", penalty="l2", max_iter=1500)

Final Result

- For the final submission, we used the following:
- CountVectorizer(ngram=(1,3))
- LogisticRegression(solver="lbfgs", penalty="l2", max_iter=1500)

submission.csv 0.63808 0.62741

3 days ago by Nandakisho submission.csv

count_vectorizer = CountVectorizer(ngram_range = (1,3)) y_pred = (pred_prob > 0.2).astype(np.int) classifier = LogisticRegression(max_iter=1500, solver='lbfgs', penalty = 'l2') 0.9726866932078658 all data