Abstract

StackOverflow is a public platform that aims to support the coding community by providing an online space for people to share their questions and for others to answer. Unfortunately, online communication can oftentimes be frustrating, leading to some users to become condescending in the comments sections of posts. This makes for an unproductive learning space. From topic modeling, we were able to divide comments up into three main categories — those that are mostly technical (as in, the answers), those that are looking for further clarification (as in, questions extending upon original post), and those that review other comments (as in, comments that commend or criticize other comments). These three categories were used in further classification modeling to try and predict whether or not comments were condescending. Overall, the community can benefit in productivity if the model is able to catch condescension before the comment is posted, prompting the user to rethink their words.

Design

Unsupervised learning algorithms used for this project were dimensionality reduction/topic modeling. Classification models used for this analysis were kNN, logistic regression, and random forests. The first two were used for their simplicity in explanation to business clientele, and the third was to try an ensemble method.

Data

These models were built using a dataset from Kaggle (https://www.kaggle.com/datasets/stackoverflow/stackoverflow?select=comments). The dataset has 75.4 million observations, with each row representing an individual comment. However, for this project, 3887 of the total comments were used, making a little over 100,000 aggregated terms. For the dataframe made for classification modeling, columns included the comments, a negative sentiment score by Vader, and scores of how much the comments were in each of the three topics.

Algorithms

Data Cleaning/EDA:

Data was from Kaggle. No null values were found, there was some scaling of features for purposes of the kNN and logistic regression models, as well as upsampling to account for highly imbalanced data.

Model: Topic Modeling/Dimensionality Reduction, kNN, Logistic Regression, Random Forests

Model Evaluation: The dataset was split into a 80/20 training set and testing set. Used accuracy, precision, recall, f1, confusion matrices. Decision threshold was lowered in an attempt to increase recall.

Tools

Numpy, Pandas, Matplotlib, Seaborn, Sklearn, Vader, nltk, scipy, gensim

Communication

Topic 1: asking for clarification

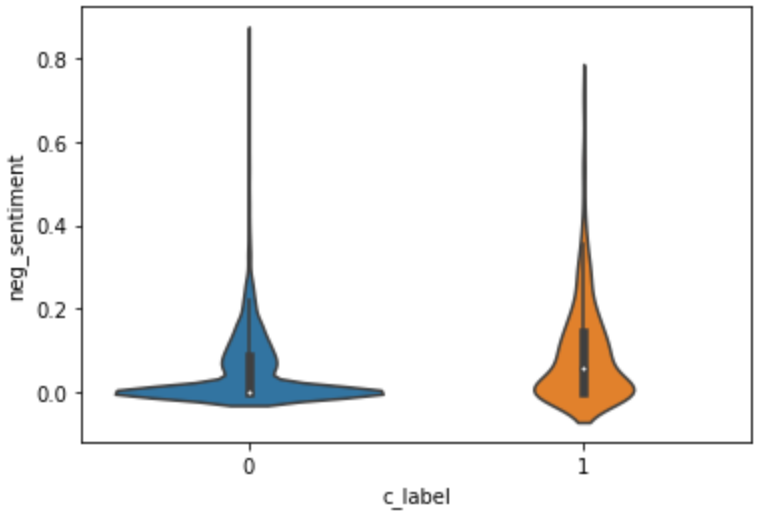
``, n't, 's, use, code, would, ..., one, like, answer, also, question, c, object, ., get, way, think, using, need

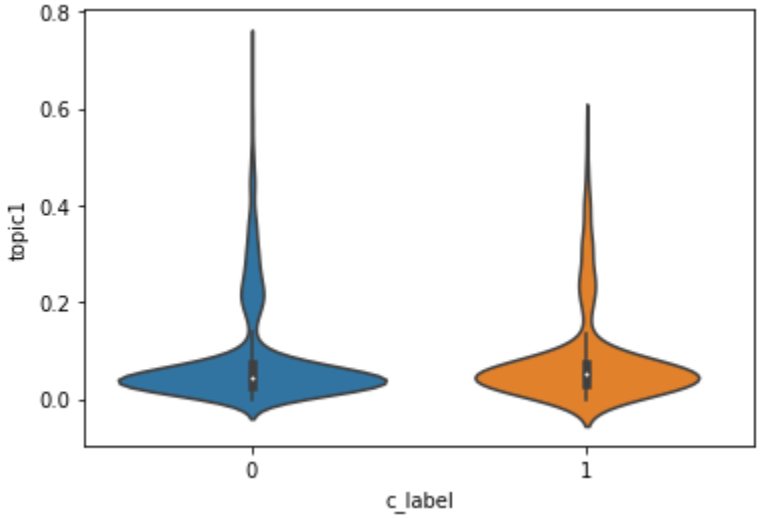
Topic 2: technical

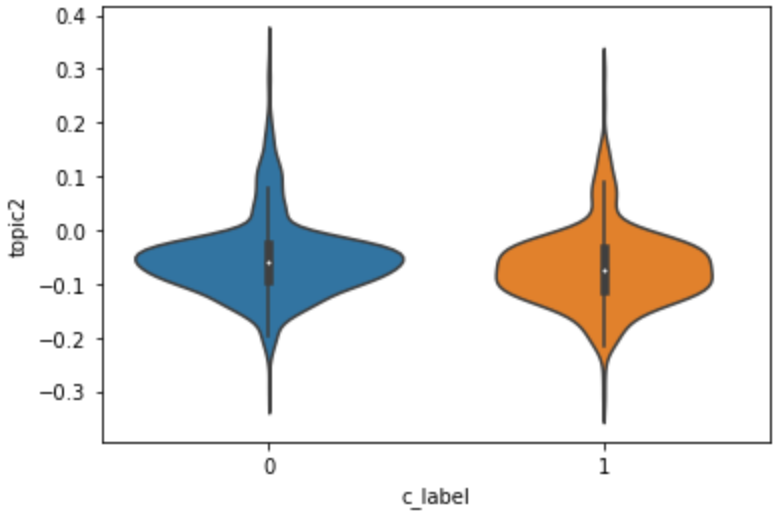
``, cast, ~, derived, straight, postgresql, exact, var, printf, typeof, derives, student, .constructor, computer, insane, controlchars.quote, repeating, intend, -name, namespacelist

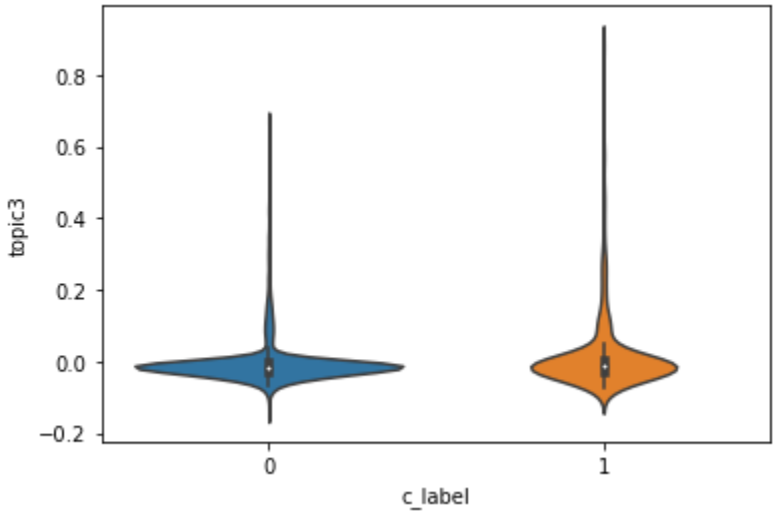
Topic 3: reviews on other comments

's, code, way, think, like, answer, function, much, data, language, also, right, problem, http, around, wrong, goto, since, pretty, lot


0.8
0.6
0.4
0.2
0.0
neg_sentiment
0
1
c_label


0.8
0.6
0.4
0.2
0.0
topic1
0
1
c_label



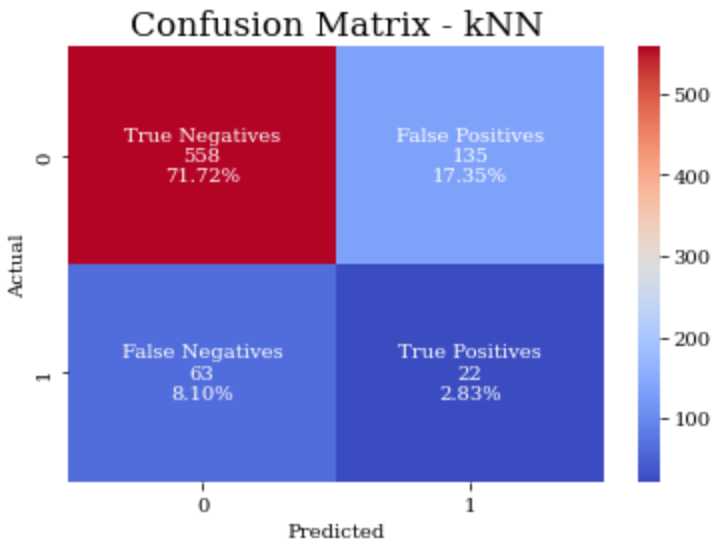


Model: Baseline Model

accuracy score: 89.07%
precision score: 0.00%
recall score: 0.00%
f1 score: 0.00%

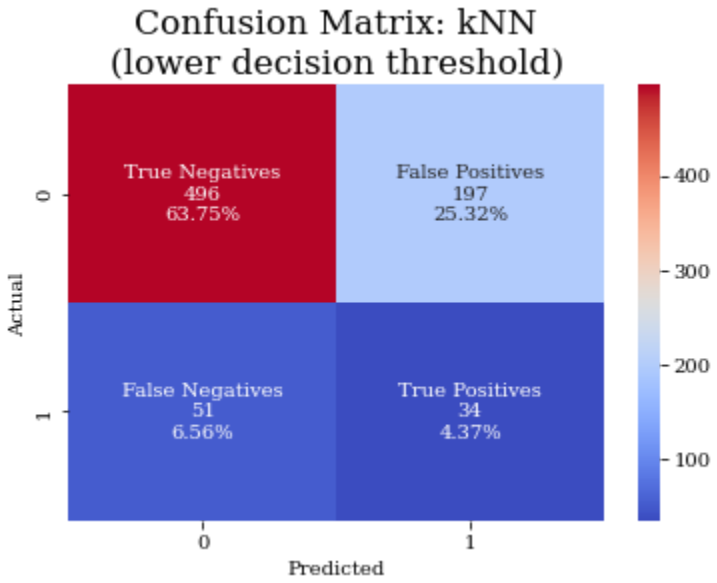
Model: kNN

accuracy score: 74.55%
precision score: 14.01%
recall score: 25.88%
f1 score: 18.18%



Model: kNN with lower decision threshold

accuracy score: 68.12%
precision score: 14.72%
recall score: 40.00%
f1 score: 21.52%



^final model