Naïve Bayes Model-Analysis Report

NLP Project

GROUP 2

COMP 237

GROUP MEMBERS: Ali Algharibawi, Sophia Ojegba, Becky Rutherford, Shinumol Sebastian

December 5,2023

Table of Contents

[Naïve Bayes NLP Model 3](#_Toc152510769)

[Introduction 3](#_Toc152510770)

[Data Exploration 3](#_Toc152510771)

[Data Visualization 4](#_Toc152510772)

[Pre-Processing 4](#_Toc152510773)

[Model Building & Training 4](#_Toc152510774)

[Model Testing 5](#_Toc152510775)

[5](#_Toc152510776)

[Testing on Unseen data 5](#_Toc152510777)

[Conclusions 5](#_Toc152510778)

# 

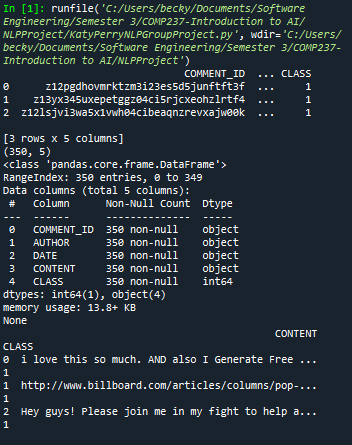
# Naïve Bayes NLP Model

## Introduction

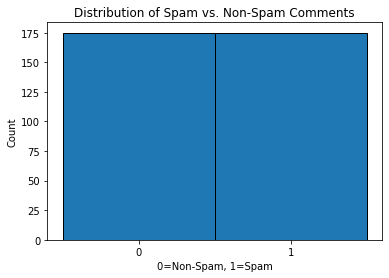
This analysis presents the results of a spam classification model for YouTube comments for a Katy Perry video. The analysis includes a description of the methods used for exploration of the dataset used to build the model, the pre-processing steps applied to the dataset to apply the Bag-of-Words model, and finally building and evaluating a Naïve Bayes classifier. We then test the model with new comments to check its ability to perform on unseen data. This analysis describes the dataset, the model’s performance and any areas for improvement.

## Data Exploration

The provided dataset is retrieved and stored in a Pandas data frame object. Initial exploration displays the first 3 rows from the dataset, the shape of the dataset, which contains 5 columns and 350 rows of data, and the general information about the dataset. The first three columns contain unique, irrelevant data; consequently, they are dropped.

 The first three rows of the newly defined data frame that contains the remaining two relevant columns are displayed.

## Data Visualization

The data is visualized using a matplotlib histogram to display the frequencies of spam vs non-spam comments in the testing dataset. The balanced representation of both spam vs non-spam will contribute to the creating of a more accurate model as bias towards one category will be reduced during training.

## Pre-Processing

Here we will transform the raw text data into a format that is appropriate for machine learning NLP models. We use a CountVectorizer object with the fit\_transform() method to process the ‘CONTENT’ column, removing stop words, tokenizing the words then creating token count vectors organized in a matrix. We chose to leave the punctuation as we felt it would be relevant in classifying spam vs non-spam, particularly in spam that contains URL links. The entry counts in these vectors represent how many times each word appears in the respective comment. The TfidfTransformer method is then applied to a TF-IDF which is a way of determining which words in the corpus carry significant meaning. The X\_train\_tfidf is the final data that will be used to train the NLP model.

## Model Building & Training

First the dataset is shuffled. The relevant features are stored in an X series and a Y series. The data is split using a 75/25 train/test split. The training data is vectorized and transformed then fit to the Naïve Bayes classifier created using the MultinomialNB().

Cross validation is then performed using 5 K-folds and then 9 k-folds to validate the model. The cross-validation with 5 k-folds results in a mean accuracy score of 0.8743. The cross-validation with 9 k-folds results in a mean accuracy score of 0.9047.

## Model Testing

The model is now tested using the 25% test datapoints. The resulting confusion matrix and Accuracy score of the model test demonstrates a robust model that performs very well on test data. There was an almost equal number of false positives vs false negatives, so the model is fairly well balanced in minimizing both types of errors. There is not much bias towards false negatives or false positives; this is great for a model that favours both specificity and sensitivity.

## 

## Testing on Unseen data

A screenshot of a computer screen

Description automatically generated Five new comments are formulated, transformed, and passed through the NLP model. The results in this test were 100%; the model accurately predicted whether each comment was spam or non-spam.

## Conclusions

Most of the spam comments contained in the original dataset used to create the model were quite obviously spam comments to the naked eye. They contained links, more punctuation than non-spam comments and frequent usage of specific words like ‘click’, ‘free’, ‘vote’ etc. The model was successful in predicting these patterns. Spammers could improve their strategies by avoiding the typical words and punctuation that are detected by spam filters.

The Naive Bayes spam model demonstrated very good performance in classifying comments as spam vs non-spam for the comments in the Katy Perry YouTube dataset. It achieved 87% accuracy when cross validated using 5 k-folds, 90% accuracy when cross validated using 9 k-folds, 94% accuracy when tested on the labelled testing dataset, and 100% accuracy on the unlabelled dataset.