Spam Classification

with Word2Vec and Machine Learning Models

Stephen Ellingson

MS Analytics Student @ Georgia Tech

Contents

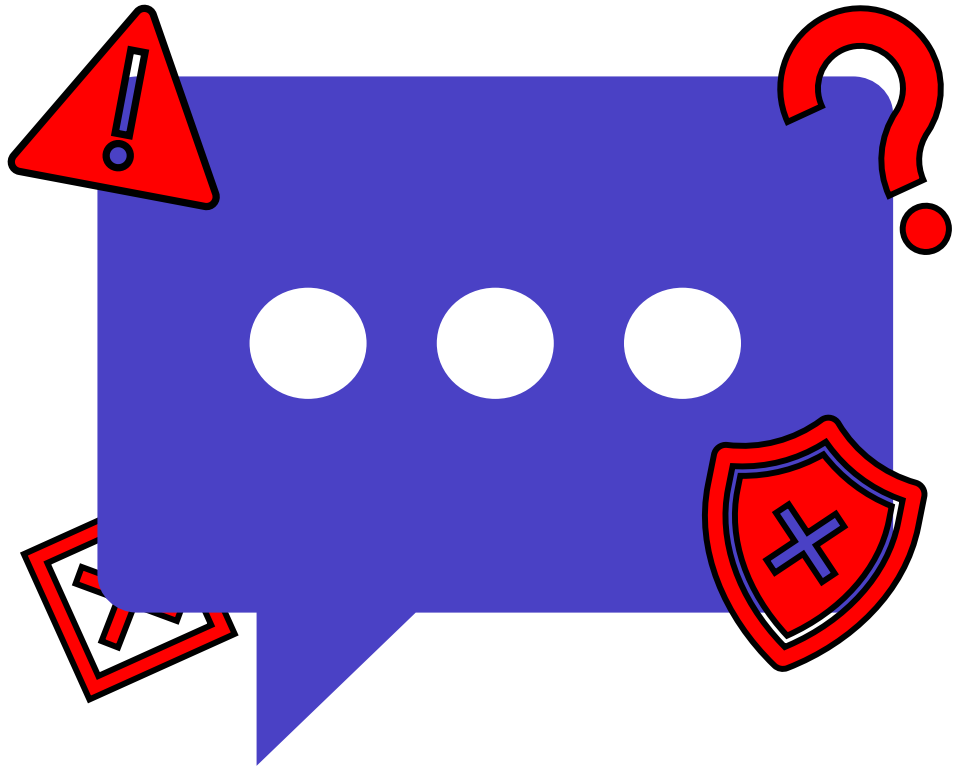
- Introduction: The Spam Problem
- Word Vectorization
- Classification Modeling
- The Complete Process
- Results
- Where Can We Improve?

Introduction: The Spam Problem

For better or worse, the COVID-19 pandemic brought along with it many technological and cultural changes, with one of them being an increase in remote lifestyles. As more people relied on messaging to communicate during this time period, so increased the amount of SMS spam messages - 2021 saw a 58% rise in spam texts sent from the previous year.

Unfortunately, this issue only seems to be getting worse - in fact, as of 2020, spam texts are now more popular than spam calls, and the average individual received more than 40 spam texts in April of 2022.

As this trend continues, I thought it would be prudent to showcase various data science methods and create some models of my own. Throughout this presentation, I'll define the techniques implemented as well as the metrics used to determine optimal performance. In doing so, I hope not only to highlight an increasingly important issue but also to educate others on the power behind the amazing world of data science!



Word Vectorization

What is Word Vectorization?

Word vectorization is the process of converting words into structured numeric representation. Depending on the purpose, those numbers can signify the existence of a word in a sentence or similarity of one word to other words.

For example, in a simple bag-of-words model we could obtain a sparse matrix with counts of words that appear in each phrase:

"The large fox"
"A red apple"
"This red fox"

	The	Large	Fox	A	Red	Apple	This
Phrase 1	1	1	1	0	0	0	0
Phrase 2	0	0	0	1	1	1	0
Phrase 3	0	0	1	0	1	0	1

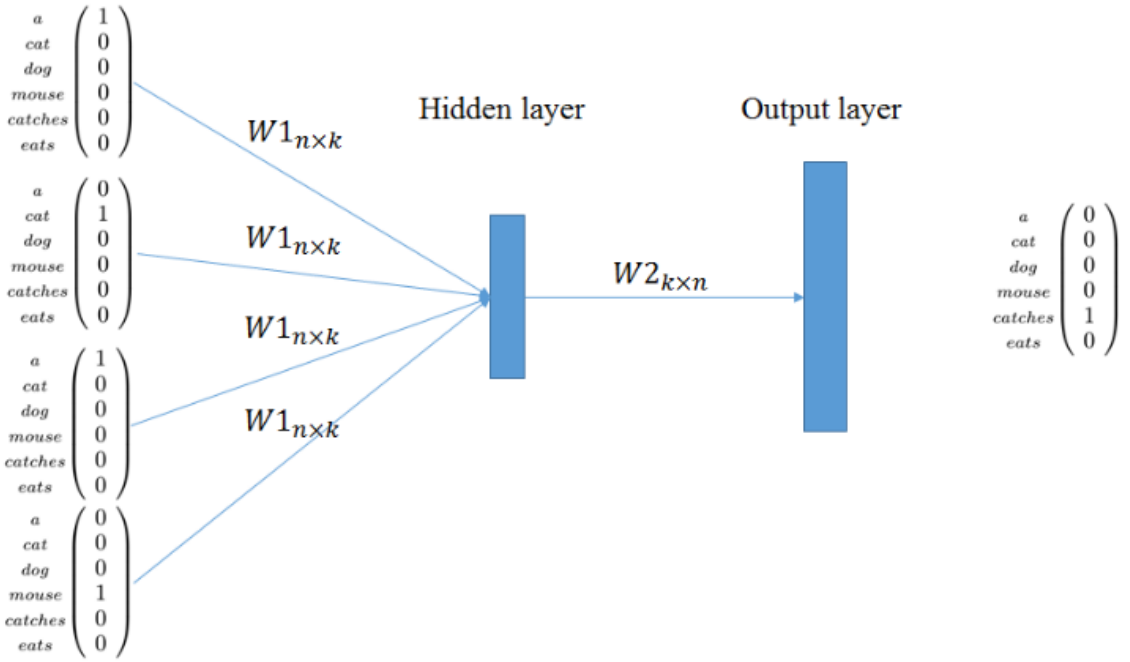
Word2Vec Explained

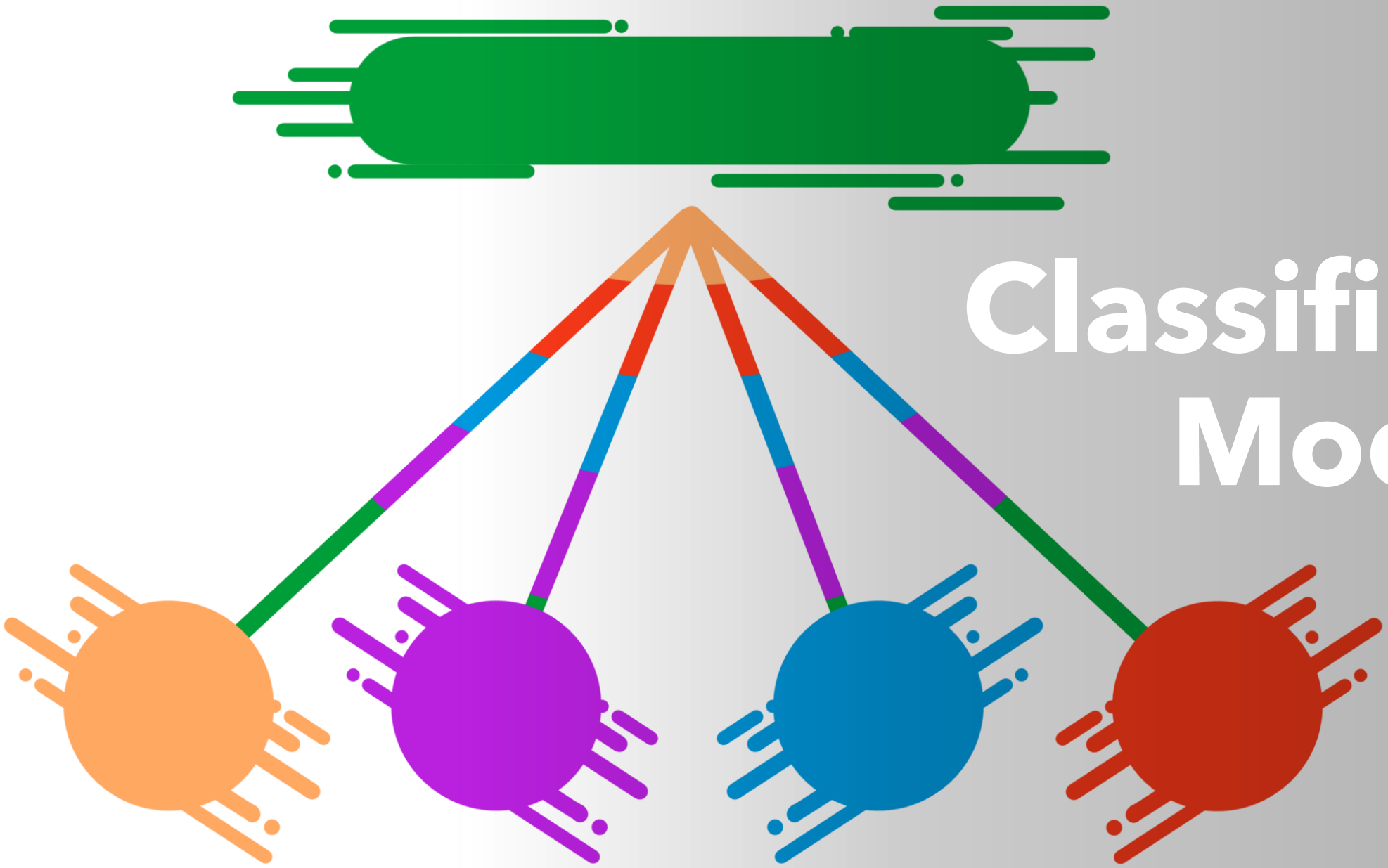
Word2Vec is a specific type of word vectorization technique that, instead of counting word occurrences like the previous example, finds associations between words through a shallow neural network model. Each distinct word is then represented as a list of numbers (or "vector") in a multidimensional space.

The implementation used for this project was Continuous Bag of Words (CBOW), which uses surrounding words in a phrase to predict the word in the middle. This is the method I chose to use, as it is a firmly established technique that provides greater accuracy than using more rudimentary methods.

CBOW ARCHITECTURE COMPONENTS

1. Input layer of one-hot encoded context words.
2. Hidden layer of *n* dimensions.
3. Output layer of the output word (in one-hot encoded format).





Classification Modeling

Classification Models

A classification model is a type of supervised machine learning algorithm that uses information on a subject to categorize it as a predefined class. There are many different types of classification models. In the case of spam, we have the following application problem:

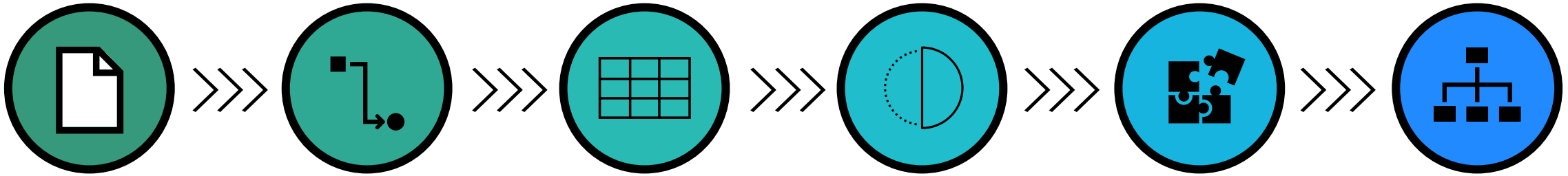
Given word vectorization information for a given SMS message, is that message most likely spam?

CLASSIFICATION MODELS USED

- **Logistic Regression:** statistical model that outputs probability of an event using Maximum Likelihood Estimation
- **Support Vector Machine (SVM):** non-probabilistic model that maps training points to a space and defines a hyperplane to best separate the two classes
- **K-Nearest Neighbors (KNN):** maps training points to a multi-dimensional space, randomly determines classes for each point, then iteratively updates class labels based on classes of neighboring data points
- **Naive Bayes:** probabilistic classifier based on independent assumptions
- **Random Forest:** ensemble learning method that involves creating multiple decision trees, with the class of a label determined by the one selected by the most trees

Project Pipeline

Project Pipeline



Data Gathering

- Test and train sets were found through a publicly available dataset from Kaggle which contained labeled SMS messages.

Data Preprocessing

- Conversion to lowercase
- Removal of stop words, punctuation, and whitespace
- Lemmatization
- Tokenization

Word Vectorization

- Used Word2Vec CBOW method to train model and find word embeddings
- Defined message vectors as average of word vectors

Data Preparation

- Due to small number of positive cases, SMOTE oversampling technique was used (synthetic examples created to reduce ratio of non-spam to spam)
- Split dataset into training and testing subsets

Parameter Optimization

- Used GridSearchCV to define optimal parameters for each classification model

Model Training/Testing

- Used training set to build each model with optimal parameters found in previous step
- Gather results from test set and compare performance

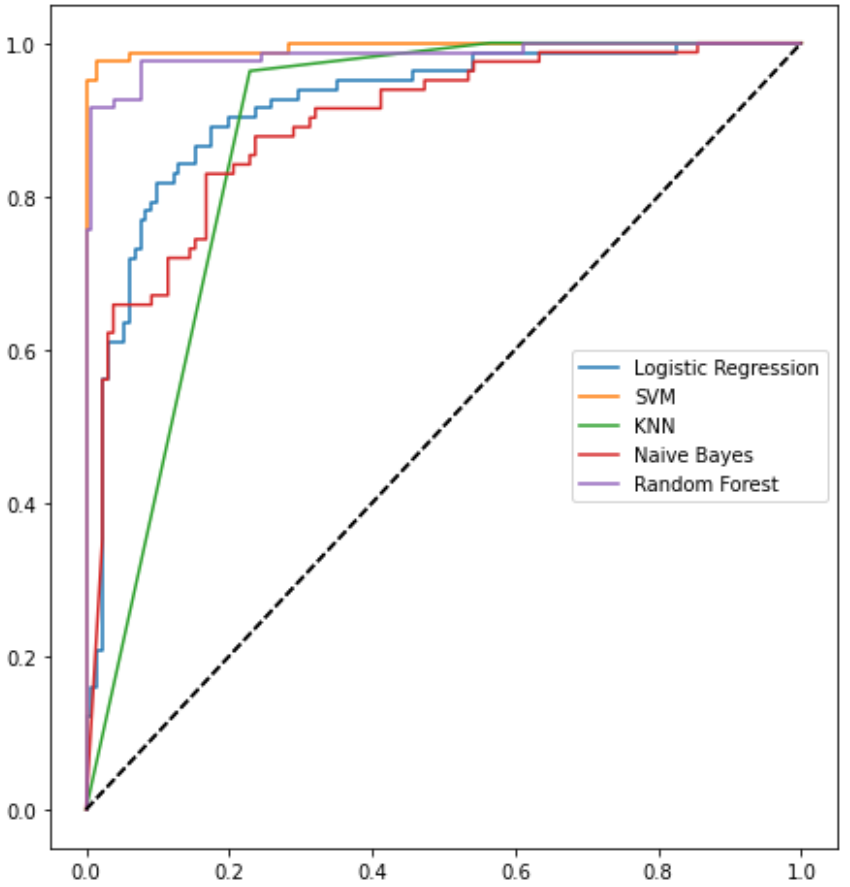
Results and Comparisons

ROC Curves

A receiver operating characteristic curve, or ROC, is a performative graph plotting False Positive Rate against True Positive Rate for all thresholds [0,1]. A desired ROC curve is one that comes closest to the upper left corner of the graph, which maximizes Area Under the Curve (AUC).

With some additional code, we can obtain ROC curves for all five of our models. As shown to the right, the **SVM model appears to outperform all others**, with Random Forest also doing quite well.

ROC Curve by Model



Precision, Recall, F1, Accuracy

In some classification problems, the metric we use to define optimal performance is simply the accuracy rate (the fraction of correctly labeled test points). However, in the case of spam we must put more value on correctly identifying spam texts rather than non-spam due to the greater impact on the person receiving those messages. This is where Precision, Recall, and F1 score come into play.

- **Precision:** A higher score means fewer regular messages incorrectly identified as spam.
  - True Positives / [True Positives + False Positives]
- **Recall:** A higher score means fewer spam messages were incorrectly identified as regular messages.
  - True Positives / [True Positives + False Negatives]
- **F1 Score:** This is the harmonic mean of precision and recall. We use harmonic mean instead of arithmetic mean because harmonic mean penalizes more for extreme scores. This is the primary metric for this situation.
  - [2 * precision * recall] / [precision + recall]

Coinciding with the findings from looking at ROC curves, it looks like **SVM offers the best performance**, with an F1 score of 0.97 and a general accuracy rate of 98%!

CLASSIFICATION MODEL RESULTS

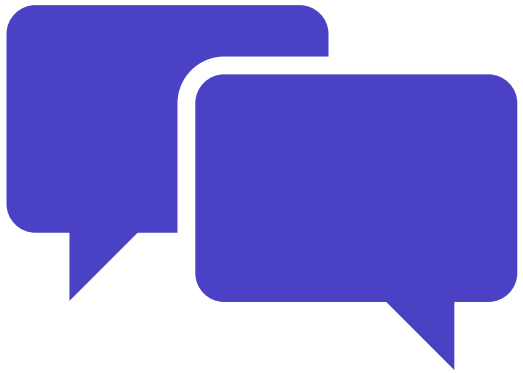
Model Type	Spam Precision	Spam Recall	Spam F1	Total Accuracy
Logistic Regression	0.79	0.84	0.82	0.85
SVM	0.98	0.96	0.97	0.98
KNN	0.72	0.96	0.83	0.85
Naive Bayes	0.66	0.88	0.75	0.78
Random Forest	0.97	0.91	0.94	0.96

Where Can We Improve?

While model performance exceeded my expectations, there are still many areas in the pipeline that could be changed in order to refine the results. Potential improvements include:

- **Other word vectorization methods:** more advanced techniques like BERT can be used to gain more contextual information from SMS messages.
- **Different classification models:** deep learning models can offer increased performance in certain problems when compared with classical algorithms.
- **Larger dataset:** More information can be gained from a dataset with more entries. With oversampling, one can run the risk of overfitting, so having more data from the beginning can eliminate this concern.

Summary



In this experiment, we analyzed text message data using word vectorization and derived a model that can predict spam messages extremely well.

What's wonderful about the field of data science is that it's always improving – new algorithms are being made and refined to increase performance and tackle previously unencountered problems. While spam texting is certainly a threat, implementations like these make great gains in keeping the communication methods we take for granted safe and secure.

Stephen Ellingson

Email: stephen.ellingson1@gmail.com
Portfolio: click here
GitHub: click here
LinkedIn: click here

To view the Jupyter notebook for this project, click here

Sources

- Spam Statistics: link
- Word Vectorization: link
- Word2Vec Explanation: link
- CBOW Diagram: link
- Classification Model Explanations: link
- Harmonic Mean: link
- Kaggle SMS Dataset: link