Case Study 3

Daniel Chang

DS 7333 | Quantifying the World

2023

**Introduction**

**Background**

## In an era dominated by constant digital communication, the inundation of messages and emails has become an ever-present challenge. Within this deluge, a significant proportion is composed of unwelcome and potentially harmful spam messages, a pervasive issue that not only disrupts productivity but also poses significant security risks. To address this pressing concern, we embark on a comprehensive research endeavor centered on Data Science, aiming to construct a robust and highly effective spam classification system.

## Our primary objective is to harness the capabilities of advanced algorithms and machine learning techniques to develop an intelligent system capable of distinguishing between authentic messages and spam. By automating this process, we seek to offer users a streamlined and secure communication experience, devoid of the interference caused by unwanted messages. This research project represents a critical step towards achieving this vision.

## The project unfolds through several vital stages, encompassing data collection, meticulous preprocessing, intricate feature engineering, rigorous model selection, and meticulous evaluation. Our dataset, carefully curated to reflect a diverse spectrum of message characteristics, will serve as the foundation upon which our classification model is trained and refined. Throughout this journey, we will delve into the realm of natural language processing (NLP) and machine learning, employing cutting-edge techniques to not only enhance the performance of our model but also enable it to adapt to the ever-evolving landscape of digital communication.

## The successful culmination of this project carries immense significance, transcending individual convenience to influence broader domains such as email filtering, cybersecurity, and information management. By mitigating the disruptive impact of spam, our research endeavors to foster a safer, more efficient, and productive digital communication environment, ultimately contributing to the greater advancement of our digitally interconnected world.

## Objective and Scope

## The scope of this project is defined by the ambitious objective of developing a highly accurate and adaptive predictive model for detecting spam messages within the realm of digital communication. Leveraging the Gaussian Naive Bayes (GNB) model alongside clustering techniques, our project encompasses a multifaceted approach to tackle the pervasive spam issue. Our efforts extend from the collection and comprehensive preprocessing of diverse message data to the intricate process of feature engineering and model selection. This multifaceted approach not only aims to enhance the accuracy of spam detection but also to ensure the model's adaptability to the ever-evolving landscape of digital communication.

## The core objective of this project is twofold. First and foremost, we seek to harness the power of machine learning, data analysis, and the Gaussian Naive Bayes model to create a predictive system that intelligently distinguishes between spam and legitimate messages. By doing so, we aim to provide users with a refined and secure communication experience, free from the intrusion of spam messages. Secondly, by incorporating clustering techniques, we endeavor to enhance the adaptability of our model, enabling it to identify emerging patterns and variations in spam messages effectively. Thus, our overarching goal is not only to deliver a highly accurate and efficient spam classification system but also to contribute to the broader domains of email filtering, cybersecurity, and information management, fostering a safer and more productive digital communication environment for all users.

## Data Source

This case study will utilize five folders of data to create this model. three folders contain messages that are not spam, and two will contain spam messages. The data will be read into a pandas dataframe to be analyzed further.

## Data Inspection

Before creating any models or analysis with the data, our first step was to inspect our data to understand better data types (such as int, cat, object, etc.), distributions of values, text analysis, etc. This step is vital in understanding how we should approach any types of transformations or adjustments to the modeling and analysis process of our data.

From our inspection, we found that the messages variable is a string text while the target variable is categorical.

## Target Variable Inspection

The count plot vividly illustrates the stark contrast in the quantity of spam and not spam messages within our dataset. With a staggering count of approximately 7,000 instances, the "Not Spam" category dominates the visual landscape, underscoring the prevalence of legitimate messages in our digital communication ecosystem. In sharp contrast, the count of "Spam" messages, standing at around 2,200, is notably dwarfed by its counterpart. This visual representation not only highlights the scale of the spam issue but also underscores the importance of developing an effective spam detection system, a pursuit that lies at the heart of our data science project.

***Figure 1:*** *Count Plot of Spam Messages*

A graph of a spam count plot

Description automatically generated

Description: Count plot of Spam (denoted as 1) and Not Spam(denoted by 0).

## Modeling

## Transforming Explanatory Data(Messages)

## In pursuit of an efficient spam detection system, the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique was employed on the messages column(in text format), serving as a pivotal element in the feature engineering process. TF-IDF, as a text preprocessing technique, adeptly transforms textual data into a numerical format comprehensible by machine learning algorithms. In other words, these will be numbers.

## However, not all the numbers are on equal footing. The resulting feature space may encompass values with varying magnitudes, where some feature values could be considerably more significant than others. These feature-scale disparities can adversely affect the performance of specific machine learning algorithms, leading to suboptimal results.

## To address this issue and ensure that our models are not unduly influenced by feature scales, we incorporated the StandardScaler() function into our data preprocessing pipeline. StandardScaler() plays a crucial role in normalizing the feature values by centering them around a mean of zero and scaling them to have a standard deviation of one. This standardization process harmonizes the numerical feature values, placing them on an equal footing and mitigating any undue bias that might arise from differing magnitudes.

## By applying StandardScaler(), we enhance the robustness and interpretability of our machine learning models, allowing them to make informed decisions based on the actual patterns within the data rather than being influenced by the scale of individual features.

## Gaussian Naïve Bayes

The goal was to utilize these transformed data representations to train and deploy a Gaussian Naive Bayes model, a potent tool in the domain of classification, for the purpose of predicting whether a given message should be categorized as spam or not.

TF-IDF generates numerical vectors that encapsulate the essence of each message. These vectors serve as the feature set for the Gaussian Naive Bayes model. Leveraging the Gaussian Naive Bayes algorithm, well-suited for text classification, these TF-IDF vectors facilitate informed decisions regarding the classification of messages as spam or not. In essence, this approach not only automates the spam detection process but also harnesses the information latent within messages to achieve a high level of classification accuracy, contributing to the creation of a more secure and efficient digital communication environment.

**Internal Cross-Validation Results:**

The results of our internal cross-validation, performed with a 5-fold validation scheme, provide valuable insights into the performance of our classification model. The accuracy scores achieved across the five folds reveal the model's ability to classify messages as spam or incorrect. These accuracy scores, denoting the proportion of accurately classified instances in each fold, are as follows: approximately 84.50%, 95.51%, 92.46%, 92.67%, and 91.28%, respectively. Crucially, the mean accuracy score of approximately 91.29% consolidates the performance of our model across all folds.

|  |
| --- |
| Mean Accuracy |
| 91.3% |

**External Cross-Validation Results:**

In the context of external cross-validation, our model exhibited outstanding performance, achieving an accuracy of approximately 98.67%. This accuracy score signifies the proportion of correctly classified instances among the total instances within the test set, demonstrating the model's exceptional proficiency in correctly classifying approximately 98.67% of the messages as either spam or not spam.

|  |
| --- |
| Mean Accuracy |
| 98.67% |

Moreover, a detailed examination through the confusion matrix provides deeper insights into the model's performance in binary classification, distinguishing between spam and not spam. Within this evaluation, the model accurately classified 1401 messages as spam (True Positives) and 1344 messages as not spam (True Negatives). Notably, the model demonstrated a remarkable level of precision, with only 27 messages falsely categorized as spam (False Positives) and merely 10 messages incorrectly labeled as not spam (False Negatives).

These metrics collectively underscore the model's exceptional performance, marked by an impressive accuracy score and an exceedingly low occurrence of false positives and false negatives. Such results reaffirm the model's effectiveness and reliability, positioning it as a robust solution for the specific context and objectives of the spam classification task. The combination of a high accuracy score and relatively low occurrences of false positives and false negatives suggests that the model performs effectively in its spam classification task. Nevertheless, it is essential to consider the specific context and objectives of the classification task to gauge the implications and overall effectiveness of the model fully.

***Figure 2(Next Page):*** *Confusion Matrix of Gaussian Naïve Bayes*

*A blue squares with white text

Description automatically generated*

***Description:*** *Confusion Matrix of Gaussian Naïve Bayes Model to detect spam. Our confusion matrix doesn’t seem to be that bad.*

## Clustering | KMeans

In the pursuit of establishing a robust framework for spam message classification, we have adopted the KMeans clustering technique as a foundational element in our model development process. While our primary focus remains on enhancing spam detection, the incorporation of KMeans clustering serves as a pivotal step in feature engineering and model selection.

Initially, we harnessed KMeans clustering to group similar messages together, a task that facilitates the subsequent modeling efforts. By employing this technique, we aimed to create distinct clusters of messages, each with its unique characteristics. These clustered groups of messages were then used as a basis for training various classification models, including Random Forest, XGBoost, Logistic Regression, and Gaussian Naive Bayes.

Within this context, the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique played a vital role in the feature engineering process. TF-IDF efficiently converted textual data into numerical representations that are interpretable by machine learning algorithms. These numerical vectors, encapsulating the essence of each message, served as the fundamental feature set for our diverse set of classification models.

The Gaussian Naive Bayes model leveraged these TF-IDF vectors to make informed decisions about whether a given message should be categorized as spam or not. This approach automated the spam detection process while harnessing valuable information embedded within messages, ultimately achieving a high level of classification accuracy.

Incorporating KMeans clustering as a precursor to model development reinforces our commitment to creating an effective spam classification system. By combining clustering techniques with a diverse set of classification models, we aim to contribute to a more secure and efficient digital communication environment where spam messages can be accurately identified and mitigated.

**Obtaining the Optimal Number of Clusters for KMeans**

The optimal number of clusters is an essential aspect of clustering tasks such as spam message classification. The Silhouette Score is a valuable metric for this purpose, providing a way to evaluate the quality of clustering solutions. A systematic approach is used to determine the optimal number of clusters using the Silhouette Score. First, a possible cluster number range is defined, ranging from a minimum to a maximum number of clusters deemed relevant for the dataset. The Silhouette Scores for each cluster number in this range are then computed. The Silhouette Score measures the cohesion of data points within the same cluster and the separation between clusters.

In our clustering analysis, we have identified the optimal number of clusters to be two, a pivotal insight that guides our model development. The line plot prominently showcases this optimal cluster count, confirming that partitioning into two distinct groups best captures the underlying structure in our data. At two clusters, the Silhouette Score is 0.92. This finding serves as a crucial foundation for our subsequent modeling efforts, ensuring that our spam classification system operates with precision and effectiveness.

***Figure 3:*** *Line graph for Optimal Number of Clusters  
  
A graph with blue lines

Description automatically generated*

***Description:*** *The line graph gives us the optimal number of clusters to achieve the best result possible.*

**Results**

The results have demonstrated promising outcomes, with each model achieving high accuracy scores:

1. Random Forest Accuracy: Our Random Forest model achieved an impressive accuracy of 96.21%. This ensemble learning method is known for its ability to handle complex data and perform well in classification tasks. In the context of spam detection, it excelled in accurately distinguishing between spam and non-spam messages.
2. Logistic Regression Accuracy: Logistic Regression, a fundamental classification algorithm, also exhibited strong performance with an accuracy of 93.32%. Despite its simplicity, it demonstrated its effectiveness in separating spam from legitimate messages.
3. XG Boost Accuracy: The XG Boost model outperformed other models with an accuracy of 96.95%. XG Boost is known for its robustness and capability to handle imbalanced datasets, making it a valuable asset in spam classification. It effectively identified spam messages while keeping false positives to a minimum.
4. Gaussian NB Accuracy: Gaussian Naive Bayes achieved an accuracy of 93.96%. Although this model assumes feature independence, it demonstrated competitive results in distinguishing spam from non-spam messages.

The consistently high accuracy scores across different models underline the importance of K-Means clustering as a preprocessing step. By leveraging K-Means clustering, we were able to transform the raw text data into structured features, reduce dimensionality, and enhance model performance. This approach enriched the feature space and empowered each model to capture subtle patterns and relationships within the data.

## Clustering | DBSCAN

In our pursuit of building a robust spam message classification system, we have shifted our clustering approach from KMeans to DBSCAN (Density-Based Spatial Clustering of Applications with Noise). While our primary goal remains enhancing spam detection, our adoption of DBSCAN clustering represents a significant evolution in our feature engineering and model selection strategies.

Initially, our use of DBSCAN clustering is aimed at effectively grouping similar messages together, laying the foundation for our subsequent modeling endeavors. Through this technique, we strive to create cohesive clusters of messages, each characterized by its distinct attributes. These message clusters will serve as the core data for training various classification models, including Random Forest, XGBoost, Logistic Regression, and Gaussian Naive Bayes. Similar to the previous clustering techniques, the TF-IDF vectorization tactics will be used.

Our transition to DBSCAN clustering underscores our unwavering commitment to crafting an exceptionally effective spam classification system. By seamlessly integrating DBSCAN clustering with diverse classification models, we aspire to contribute significantly to fostering a digital communication environment that is both secure and efficient, where spam messages can be reliably identified and effectively managed.

**Results:**

The results have unveiled impressive outcomes, showcasing the prowess of our models in accurately detecting spam using the DBSCAN clustering algorithm:

1. Random Forest Accuracy: The Random Forest model exhibited an impressive accuracy of 95.94%. This ensemble learning method is renowned for its ability to tackle complex data and excel in classification tasks. In the realm of spam detection, it distinguished itself by effectively discerning between spam and legitimate messages.
2. Logistic Regression Accuracy: Logistic Regression, a fundamental classification algorithm, delivered robust performance with an accuracy of 94.17%. Despite its simplicity, it proved its mettle in segregating spam from legitimate messages.
3. XG Boost Accuracy: The XG Boost model continued its dominance with an accuracy of 96.42%, outshining other models. XG Boost is celebrated for its robustness and adeptness in handling imbalanced datasets, making it an invaluable asset in the realm of spam classification. It adeptly identified spam messages while maintaining a low rate of false positives.
4. Gaussian NB Accuracy: Gaussian Naive Bayes achieved a commendable accuracy of 94.12%. Although this model assumes feature independence, it exhibited competitive performance in distinguishing spam from non-spam messages.

The consistently high accuracy scores across different models underscore the significance of DBSCAN clustering as a preprocessing step. By leveraging DBSCAN clustering, we transformed raw text data into structured features, reduced dimensionality, and amplified model performance. This approach enriched the feature space, empowering each model to capture intricate patterns and relationships within the data.

**Assumption Checks:**

***Figure 4:***

***Description:***

***Figure 11:*** *Feature Importance (Elastic-Net)*

**Conclusion**

The initial model employed Lasso Linear Regression on the raw data, resulting in an RMSE of 17.6157. During validation, the model demonstrated accurate predictions and a keen grasp of data patterns. However, it encountered challenges in generalizing to unseen data, evident by its higher RMSE of 23.5082 on the holdout set.

**Recommendations**

1. **Feature Engineering and Selection**: Using domain knowledge to engineer meaningful features can improve model performance significantly. Investigating interactions, polynomial terms, and domain-specific transformations may reveal hidden patterns in the data. Furthermore, feature selection techniques other than regularization, such as mutual information or recursive feature elimination, can help identify the most influential features. Experimenting with different combinations of features and selection methods iteratively can result in a more refined set of inputs for the models.
2. **Experimenting with Different Models:** Experimenting with models other than linear regression, such as decision trees, support vector machines, random forest or gradient boosting, can provide new insights and potentially improved performance. Each model has strengths and weaknesses, and experimenting with various approaches can reveal the best-fit approach for the specific dataset. Furthermore, neural networks and deep learning architectures can capture complex relationships that traditional models may overlook. Iteratively testing and comparing different models' performance can result in a more accurate and versatile predictive system.

**Appendix**