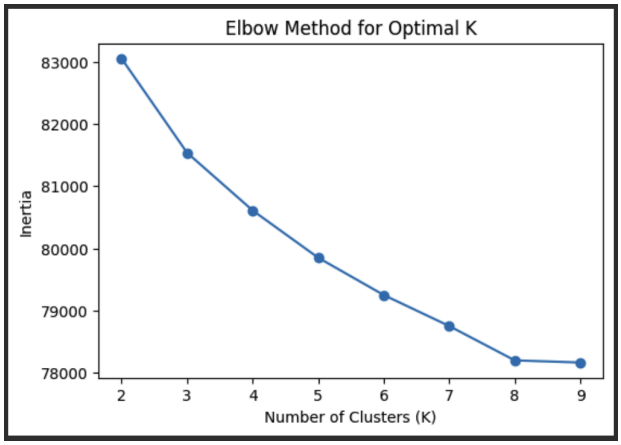
CSci 4521 Homework 2 Writeup

Binary Clustering

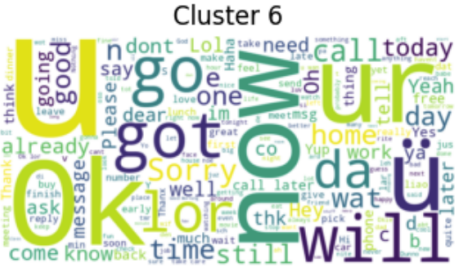
An unsupervised method like K-Means clustering with count vectorization to categorize SMS messages into two clusters (spam and ham) presented mixed results. While the clustering process attempts to process attempts to separate messages based on their content, the numerical evaluation reveals limitations. With a clustering accuracy of 0.78, the clusters align reasonably well with the actual spam and ham labels for a significant portion of the data. However, a low precision of 0.26 indicates that when the model predicts a message as spam is usually incorrect. This means that many actual ham messages are being incorrectly classified as spam (false positives). The recall of 0.38 is higher than the precision score but indicates that the model only identifies a small portion of the actual spam messages (false negatives). An F1-Score of 0.31, which balances precision and recall, reflects the overall weakness of the clustering approach in accurately capturing the distinction between spam and ham. If we relied solely on clustering, the resulting classifier would be unreliable, particularly in identifying spam messages correctly.

Clustering Quality



Before analyzing the clusters, we have to split the clusters first. In order to do that, we find the number of optimal clusters for KMeans. Using the elbow method, we determine that the k with the lowest inertia is the optimal k value which based on the graph is 8. We use 8 as the number of clusters for the KMeans algorithm.

In the analysis of the clusters, we observe significant distinctions in the types of messages within each cluster. Specifically, looking at cluster 4 and cluster 6. Cluster 4 is dominated by spam messages, with 89.53% of its 296 messages classified as spam. The content of this cluster revolves around promotional language, offering prizes and rewards. This can be seen through the word cloud for this cluster which has words like “free”, “call”, “prize”, and “won.” Given the high spam percentage, we can conclude this cluster serves as a strong indicator of spam-like content. Cluster 6 is almost the opposite, with 2.86% of its 2483 messages being spam, making it the cluster predominantly consisting of non-spam messages. The messages in this cluster are usually routine communication, conversations, reminders, or status updates which are far from being spam messages. The word cloud for this cluster has words like “ok”, “u”, “ur”, “will”, and “sorry” which are all words for personal communication.



While the clustering model performs well, there are outlier messages that are misclassified. This is likely due to similarities in content. For example, in a cluster predominantly made up of non-spam messages (Cluster 6), we find examples of spam like “You have won a Nokia 7250i. This is what you get when you win our FREE auction.” These messages likely get classified as non-spam because they may share structural or linguistic features with more legitimate communications.

On the other hand, non-spam messages are sometimes classified as spam, as seen in the case of messages such as “Sir i am waiting for your call once free please call me,” and “Your account has been refilled successfully.” These types of messages may include terms like “call” or “refilled,” which are common in spam content. The model may mistakenly associate these phrases with spam because they appear in similar contexts within the training data. This highlights a common challenge in text classification which is identifying patterns that differentiate actual communication from spam.

To further evaluate the model, we used four new text messages to test against the clusters. The first message, “Hey, are we still on for the meeting tomorrow?” was correctly classified into Cluster 6, which predominantly contains non-spam messages. The distance of this message to the cluster centers (3.20, 4.46, 24.10, etc.) confirms its proximity to Cluster 6, where it is expected to fall.

The second message, “Congratulations, you have won a free vacation! Call now to claim your prize,” was classified into Cluster 4, where the majority of messages are spam. Its classification in Cluster 4 is supported by its distance to the cluster center (2.50), confirming it aligns closely with typical spam content.

The third message, “Limited time offer: 50% off on all products! Visit our website now,” was also classified into Cluster 6, despite its promotional nature. The message does not contain the overtly spammy language typically associated with Cluster 4, which led to its classification in Cluster 6. The distance to the cluster centers (3.73) supports this classification, though it is slightly farther from the center of Cluster 6 compared to the first message.

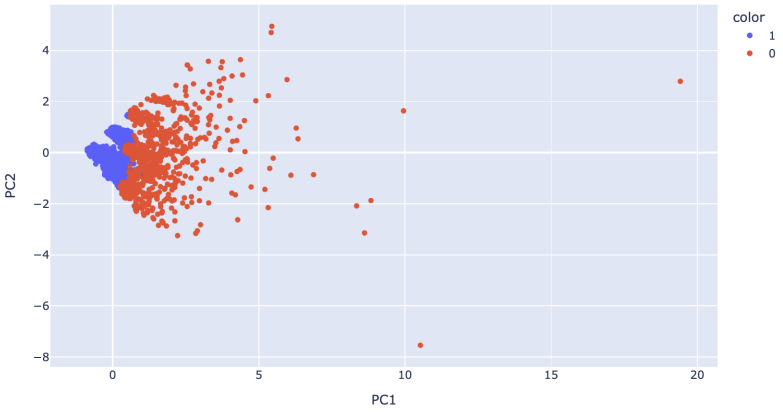
Finally, the message “Reminder: Your doctor's appointment is scheduled for 3 PM today” was correctly classified into Cluster 6, aligning with the typical non-spam content of the cluster. Its proximity to the cluster center (2.78) further verifies that it is representative of the cluster's primary content type.

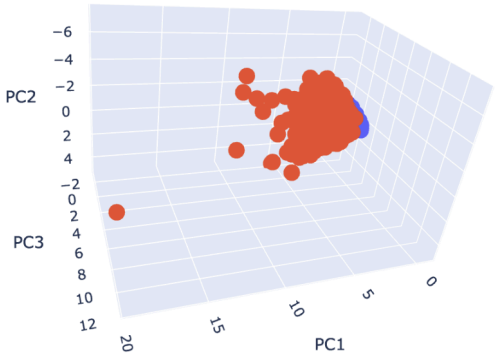
Overall, the K-means clustering model successfully distinguishes between spam and non-spam messages in most cases. However, as seen with the outlier messages, some misclassifications occur due to the overlap in features between spam and non-spam content.

PCA

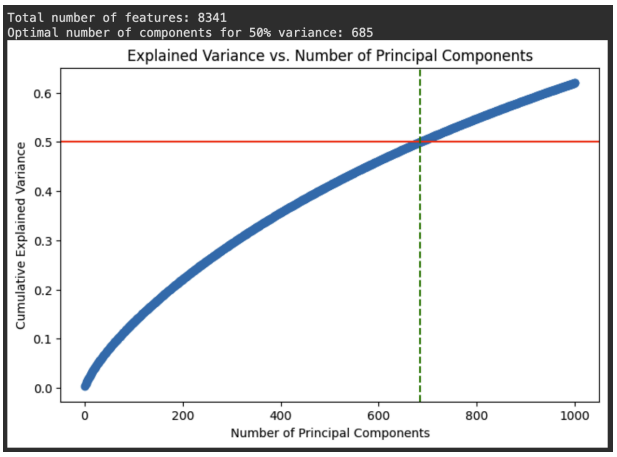
Using PCA, we reduced the dimensionality of the SMS dataset down to the top 3 principal components. In this plot, each point corresponds to a message, and the document ID is displayed when you hover over the points with coloring based on cluster assignment (spam: 1, ham: 0). The plot clearly shows how the messages are distributed in 3D space, with some

clusters grouping together, while others appear more scattered. The colors represent different clusters, helping to visually differentiate spam messages from non-spam ones. A 2D representation of the PCA also provides a simplified view of the data.





In the PCA analysis, some ham messages appear as outliers, distant from the main ham cluster. For example, the message with id 1038, which is a heartfelt declaration of love, and the message with id 1780, which discusses personal frustrations and household issues, stand out from the typical ham content. These points are atypical because PCA focuses on maximizing variance, meaning that messages with unique or less common words/phrases may appear distant from the majority of the ham points, which tend to be more neutral or conversational in nature. While both messages are categorized as ham, their content doesn't share the same patterns of frequent words typically found in standard ham messages, which may explain their distance from the typical ham cluster in the PCA diagram.



The dataset contains a total of 8341 features, which include all the various terms and characteristics that describe each message. From the PCA output, we can see that at least 685 principal components are needed to capture more than 50% of the variation in the data. The plot shows the explained variance vs. number of principal components.

Dimensionality Reduction and Classification

Note: Metrics might change slightly based on each run. This is based on one run.

Both classifiers performed almost equally well, with the full feature-based classifier achieving a slightly better accuracy (0.9519) than the PCA-based classifier (0.9500). This

suggests that the dimensionality reduction using PCA does not lead to a significant loss in predictive performance, but it is important to carefully tune k to ensure variance is retained.

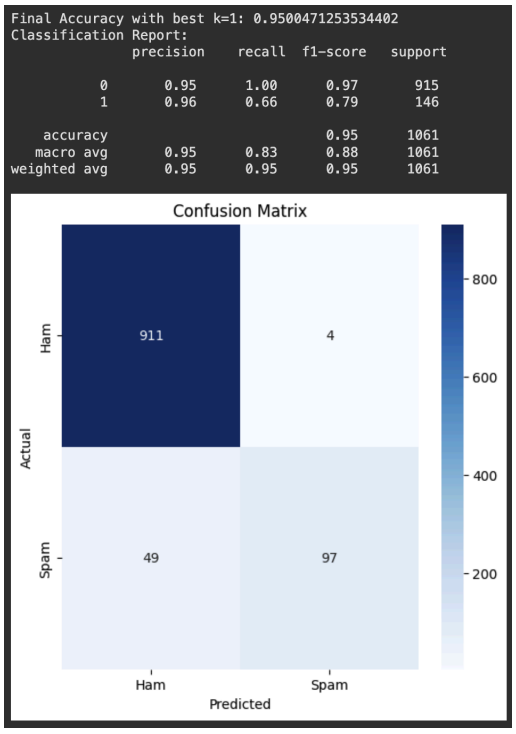
The full feature-based classifier slightly outperforms the PCA-based classifier in terms of F1-score (0.7884 vs 0.7782). This indicates that while both classifiers perform similarly in accuracy, the full feature-based classifier has a slight advantage in balancing precision and recall.

The runtime for the PCA-based classifier is significantly slower (4.3738 seconds) compared to the full feature-based classifier (0.1152 seconds). This is expected due to the additional overhead of performing PCA on the dataset before applying the classification algorithm with the number of components equaling 685 which was determined in the PCA section.

The full feature-based 1-NN classifier is faster and slightly more accurate in this case, making it the preferred choice when computational efficiency and model performance are important. However, if the dataset has a large number of features, PCA-based dimensionality reduction can still be used to speed up computations and reduce memory usage at the cost of a small performance trade-off.

Spam Classification

We utilized count vectorization to transform the text data into numerical form. Count vectorization converts each word into a frequency-based numerical representation. We chose this since it effectively captures term frequency without diminishing the importance of frequently occurring words, which can be valuable for spam detection.



We did not apply stemming. Stemming reduces words to their base form, but this may lead to loss of semantic meaning. Since spam messages often contain variations of words that need to be distinguished, we kept the full words.

Given the high-dimensional nature of count vectorized text data, we applied PCA to reduce dimensionality. This helps improve computational efficiency and potentially enhances model performance by removing noise and redundancy. Since the initial feature space contained 8,341 features, it was necessary to apply PCA to reduce dimensionality while preserving important variance. Our PCA analysis determined that using 685 principal components was sufficient to capture at least 50% of the variance. This allowed us to reduce computational cost and prevent overfitting.

We chose KNN as the classification algorithm because it is straightforward, and performs well in cases where clusters exist in the data. To determine the optimal value of k, we experimented with different k values and found that k = 1 gave the highest accuracy.

The k-NN classification model, with k=1, achieved an impressive overall accuracy of 95.0%, indicating strong performance in distinguishing between ham and spam messages. The weighted average precision, recall, and F1-score all hover around 95%, demonstrating a well-balanced model across the dataset for ham messages.

For ham messages (label is 0), the model performs exceptionally well, with a precision of 0.95 and recall of 1.00, meaning it correctly identifies all ham messages without any false negatives. Given that ham messages make up the majority of the dataset (915 out of 1061 samples), the model favors this class, contributing to its high recall. The F1-score of 0.97 reflects this strong balance between precision and recall.

However, the spam classification (label is 1) is less reliable. While the precision is high at 0.96, meaning that when the model predicts a message as spam, it is correct 96% of the time, the recall drops significantly to 0.66. This means 34% of actual spam messages are misclassified as ham, which could be problematic in real-world applications where missing spam messages might lead to security risks. The lower recall brings down the F1-score to 0.79, highlighting the trade-off between precision and recall.

While the model performs well overall, its lower recall for spam detection remains a concern. If the primary goal is to avoid misclassifying ham as spam, the model is performing effectively. However, if capturing all spam messages is a priority, further changes are necessary to enhance recall while maintaining the strong precision and accuracy.