Project Report: Email Classification Using Support Vector Machine (SVM)

This project aims to develop a machine learning-based email classification system using Support Vector Machine (SVM). The objective is to categorize emails into predefined classes by analyzing their subject lines and message bodies.

# 1. Project Objective

The primary goal of this project is to build a robust and scalable classification model that can automatically detect the category of an email based on its content. This type of classification is essential for spam filtering, routing messages to correct departments, or prioritizing communications in organizational settings.

# 2. Dataset Description

The dataset used for this project is a CSV file containing multiple emails with the following features:  
- Subject: The subject line of the email  
- Text: The main body of the email  
- Type: The category or label assigned to the email (target variable)

# 3. Data Preprocessing

To prepare the data for modeling, several preprocessing steps were taken:  
• Lowercased all text  
• Removed punctuation using Python's `string` module  
• Removed numbers using regular expressions  
• Cleaned both the subject and the body text  
• Combined cleaned subject and body text into a new feature called 'combined\_text'

# 4. Feature Extraction with TF-IDF

Text data was transformed into numerical vectors using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer. This method highlights important words by considering their frequency in a document versus their frequency across the corpus. The vectorizer was limited to the 5000 most relevant features for performance optimization.

# 5. Model Building and Training

A Support Vector Machine (SVM) classifier with a linear kernel was selected for its effectiveness in high-dimensional spaces like text data. The TF-IDF matrix was split into training and test sets with an 80-20 ratio. The model was then trained using the training data.

# 6. Model Evaluation

The trained model was evaluated on the test set using metrics such as accuracy and the classification report (precision, recall, F1-score). These metrics provide insights into how well the model performs for each class and overall.

# 7. Model Serialization

To preserve the trained model and the vectorizer for future use, both were saved using Python's `pickle` library:  
- `svm\_model.pkl`: Serialized SVM model  
- `tfidf\_vectorizer.pkl`: Serialized TF-IDF vectorizer

# 8. Why Use Support Vector Machine?

SVMs are well-suited for text classification tasks due to:  
- Their effectiveness in high-dimensional spaces  
- Strong performance in binary and multi-class classification  
- Ability to find an optimal separating hyperplane  
- Flexibility with different kernels (linear in this case for performance and interpretability)

# 9. Conclusion

This project successfully implemented an SVM-based email classifier using real-world text data. The results showed strong performance in classifying emails into multiple categories. The approach is scalable and ready for deployment in applications like spam detection, customer service triaging, or email automation.