Leticia Lima Huang CIS9660- Data Mining for Business Analytics Technical report

My classification project involved creating an app that sorts financial documents into categories, providing a useful tool for procurement professionals managing large volumes of paperwork. I selected this topic based on my experience conducting procurement through CUNY Buy, which frequently involves handling numerous financial documents. Such an application could streamline the process of reviewing and accurately sorting these documents by department. The dataset used comes from a Kaggle dataset (link available in my Jupiter notebook), but due to its size (about 9GB), I sampled the data to extract image features efficiently.

I started the project by loading my original dataset and revising its structure, which initial contained 3,482 images. Various dataset structures were tested to determine which would be most effective given system constraints and to achieve a reliable cross-validation score. A reduced dataset was created with 120 images per class to ensure fair distribution, as some classes had approximately 600 images while others, such as the resume category, only had 120. This adjustment aimed to prevent bias toward overrepresented classes. After creating my data frame, I reviewed sample documents, cleaned the data, and extracted image features for classification. One of the main considerations was whether to use TensorFlow, which offered faster and more accurate results, or Scikit-learn, a lightweight library compatible with Streamlit cloud deployment but slower in performance. Ultimately, TensorFlow was selected, and compatibility with Streamlit was achieved by using Python version 3.10, which supports the TensorFlow library. Just keep in mind that when working with the same repository on GitHub, it is advisable to create a file specifying the Python library used by each application. Some applications experienced failures due to dependencies on Python version 3.13. The use of config.toml to define the required libraries for each application successfully resolved these issues.
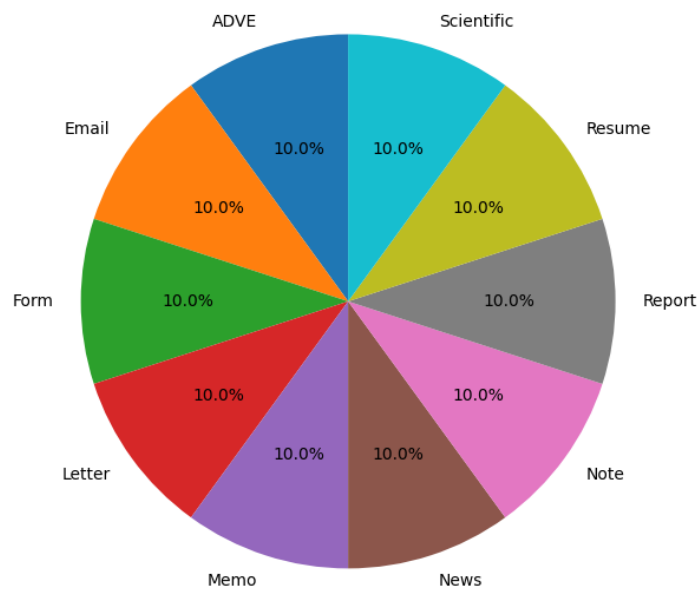
During preprocessing and model training, I first used K-Nearest Neighbors (KNN), finding an optimal K of 7 and tested various distance metrics, with both Euclidean and Minkowski yielding comparable results. A total of eleven models were assessed and evaluated using cross-validation; Logistic Regression achieved the highest accuracy at 63.61%. Unsupervised K-Means performed best with k=7 (score: 798116.94) and cluster accuracy of 3.89%, less accurate than supervised methods, that outperformed k-means by 59.72%. Overall, supervised models consistently achieved superior performance compared to unsupervised approaches. The confusion matrix revealed a satisfactory rate of correct predictions, with minor confusion observed between classifying 'letter' and 'report' due to similarities in image characteristics.

This model presents a valuable solution for environments that require daily sorting, offering the advantage of automating document classification and thereby reducing time spent on manual sorting. It enhances document organization by ensuring materials are correctly categorized and improves workflow efficiency when integrated with databases that route documents to the appropriate departments. By minimizing the need for manual intervention, it also contributes to cost savings through reduced staffing requirements.

Appendix

Distribution of Different Image Categories
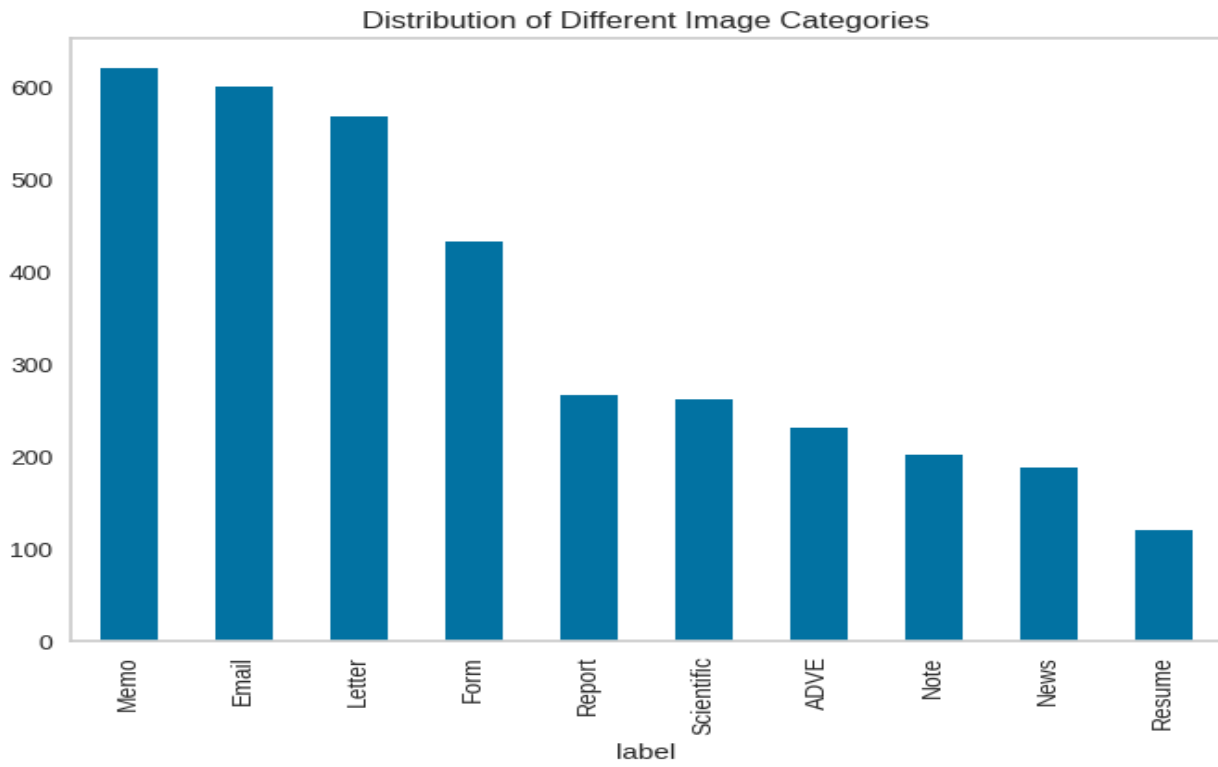
Distribution of Sampled Financial Documents Classes

Sample of Images included in the data frame

ADVE

ADVE

Email

Email

Form

Form

Letter

Letter

Memo

Memo

News

News

Note

Note

Report

Report

Resume

Resume

Scientific

Scientific

5-Fold Cross Validation Results

KNN Validation Curve: Finding Optimal k

Model vs Weighted Avg Of Various Metrics

Elbow Method for Optimal K

Elbow at k=7
Score=798116.94

Confusion Matrix - Logistic Regression