DIAGNOSTIC ASSISTANCE USING CLINICAL TEXT ANALYSIS

Abhay Bedi

Problem statement

Clinical text classification aims to address the fundamental challenge in medical natural language processing of extracting meaningful information from unstructured clinical texts often expressed in layman language instead of computer understandable texts.

Additionally, the model aims to consider the patient's historical health records, particularly chronic conditions, to refine the diagnostic outcomes. This holistic approach necessitates the development of advanced deep learning frameworks capable of synthesizing multifaceted medical data into actionable diagnostic intelligence

Objectives

- **Developing an Accurate Classification Model**

  To create a model that can classify clinical texts with high precision and recall, facilitating better patient care and data management.

- **Translating Layman Descriptions to Medical Terminology:**

  To develop the capability to interpret descriptions provided by patients in non-technical language and accurately translate them into standardized medical terms for effective diagnosis and record-keeping.

- **Identifying Relevant Medical Specialties:**

  To enable the system to accurately predict the most relevant medical specialty for a patient's condition based on the analysis of clinical texts, thereby facilitating appropriate and timely medical referrals.

Dataset Explanation:

- The data was scraped from mtsamples.com.

Key columns include:

- description: A brief description of the case.
- medical_specialty: The medical specialty associated with the transcription.
- sample_name: Name or identifier of the sample.
- transcription: The actual text transcription.
- keywords: Relevant keywords associated with the transcription.

Example Entries:

- Entry 1 (Allergy / Immunology):
- Description: A 23-year-old white female presents with complaints of allergic rhinitis.
- Transcription: Subjective details about the patient’s condition.
- Keywords: allergy / immunology, allergic rhinitis, etc.

- Description: Consult for laparoscopic gastric bypass.
- Transcription: Details related to the patient’s medical history and procedure.
- Keywords: bariatrics, laparoscopic gastric bypass, etc.

Data preprocessing steps:

Word count and Sentence count extraction:

- The get_sentence_word_count function analyzes a list of texts to determine the total number of sentences and unique words across all texts. It counts sentences and tokenizes words, updating a dictionary to track word frequencies. The function returns the total sentence count and the number of unique words in the text list.

Removal of categories with low sample count

- Categories with too few samples may not provide enough data for the model to learn meaningful patterns or relationships. The model might overfit to the limited data, leading to poor generalization performance on unseen data. In this case, medical specialities with less than 50 samples in the database have been excluded. NULL values have also been excluded for model training.

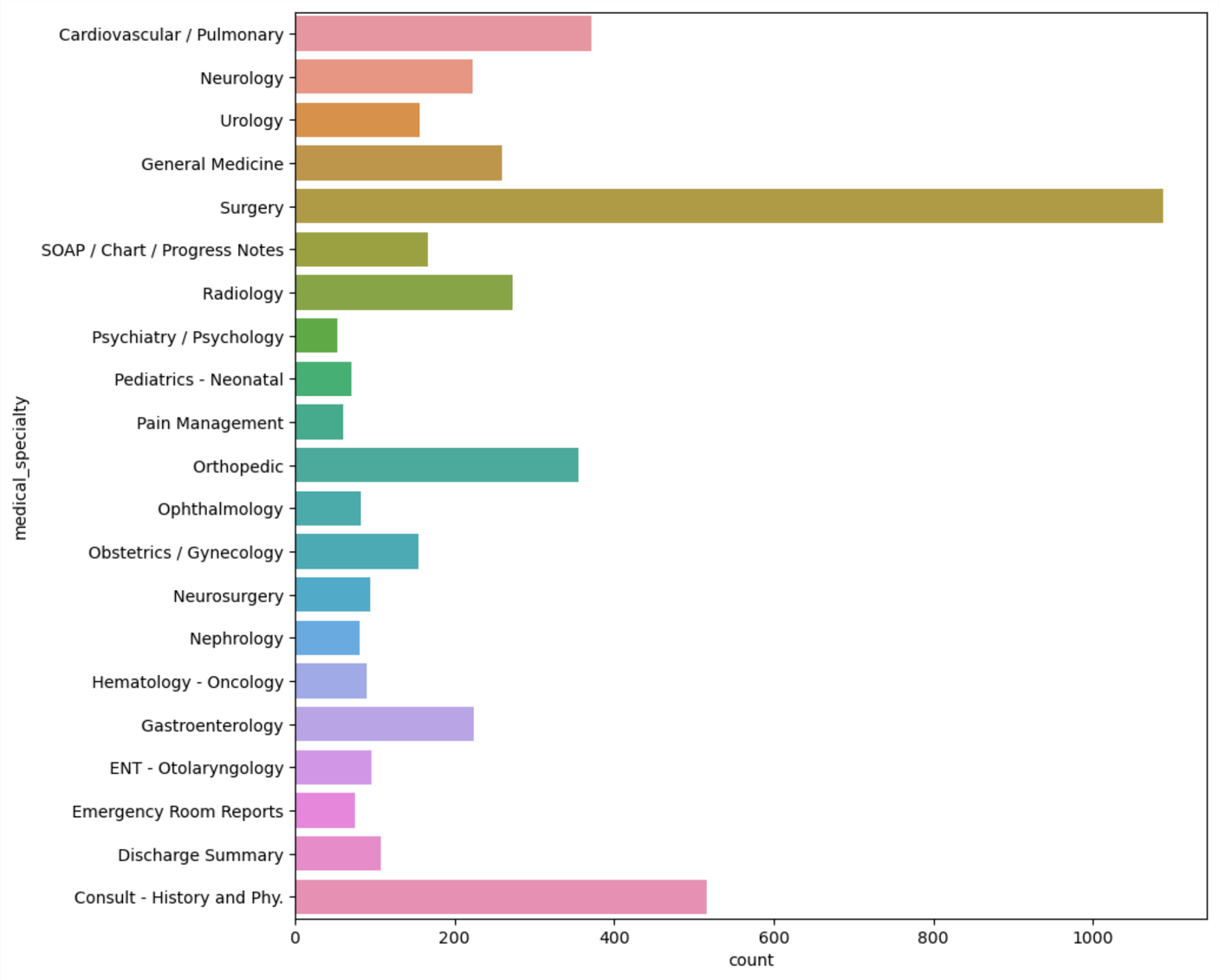
Data Cleaning

- This includes puntuation removal, removal of digits from the text using a list comprehension and converting them into a single string and replacing certain symbols such as '/', '()', '{}', '[]', '|', '@', ',', and ';') with spaces. Converting the final text to lower case and returning the same.

Lemmatization of text

- Begins by first tokenization of text into sentences and then tokenization of sentences into individual words
- Lemmatization reduces each word to its root form, also called a lemma. With this different forms of a word can be analyzed as a single item. (“going”, “went” and “go” all get converted to “go”)
- All words are then converted into a single string, with each element separated by a space.

Visual representation of med_specs and sample count



Term Frequency inverse Document Frequency

- In information retrieval, tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

- It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

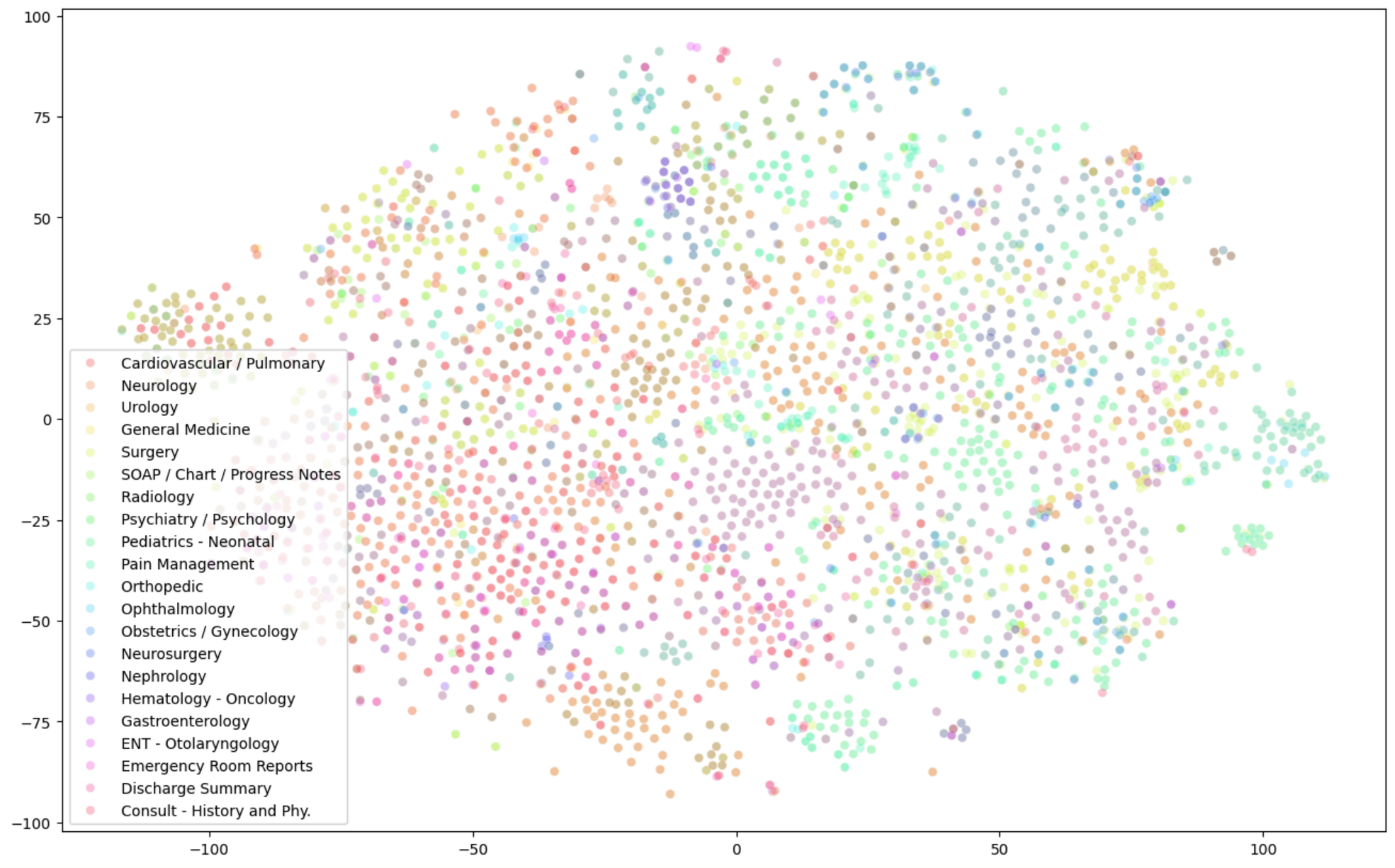
HyperParameters used in Model:

- ngram_range=(1,3) vectorizer should consider unigrams, bigrams, and trigrams as features. helps capture phrases instead of just words
- max_df=0.75 ignore words that are common in 75% of documents
- use_idf=True inverse document frequency (IDF) weighting should be applied, helps in giving more weight to terms that are rare across documents
- max_features=1000 (to include only top 1000)

Dimensionality reduction:

- t-SNE (t-Distributed Stochastic Neighbor Embedding) is a popular technique used for dimensionality reduction and visualization of high-dimensional data in a lower-dimensional space, typically 2D or 3D. It aims to preserve the local structure of the data points while reducing dimensionality. In simpler terms, t-SNE transforms complex and high-dimensional data into a simpler and more manageable form that can be visualized in a scatter plot. It is particularly useful for exploring and understanding the relationships between data points that may not be easily discernible in the original high-dimensional space.

Visual representation of data:



Dimensionality reduction:

- PCA (Principle Component Analysis) is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on

- In our case, pca object creation shall retain 95% of componenets
- The stratify=labels argument ensures that the splitting is done in a way that preserves the distribution of medical specialties in both the training and testing sets.
- only unique medical specialities are included

Logistic regression

- Logistic regression is a statistical method used for binary classification tasks, where the target variable (or dependent variable) has two possible outcomes, typically labeled as 0 and 1. It's called "logistic" because it models the probability of the outcome belonging to a particular class using the logistic function (also known as the sigmoid function).
- The sigmoid function maps any real-valued number into the range [0, 1], which is suitable for representing probabilities.
- The logistic regression model calculates a weighted sum of the input features and applies the logistic function to obtain the probability of belonging to the positive class (i.e., a specific medical specialty in this case).

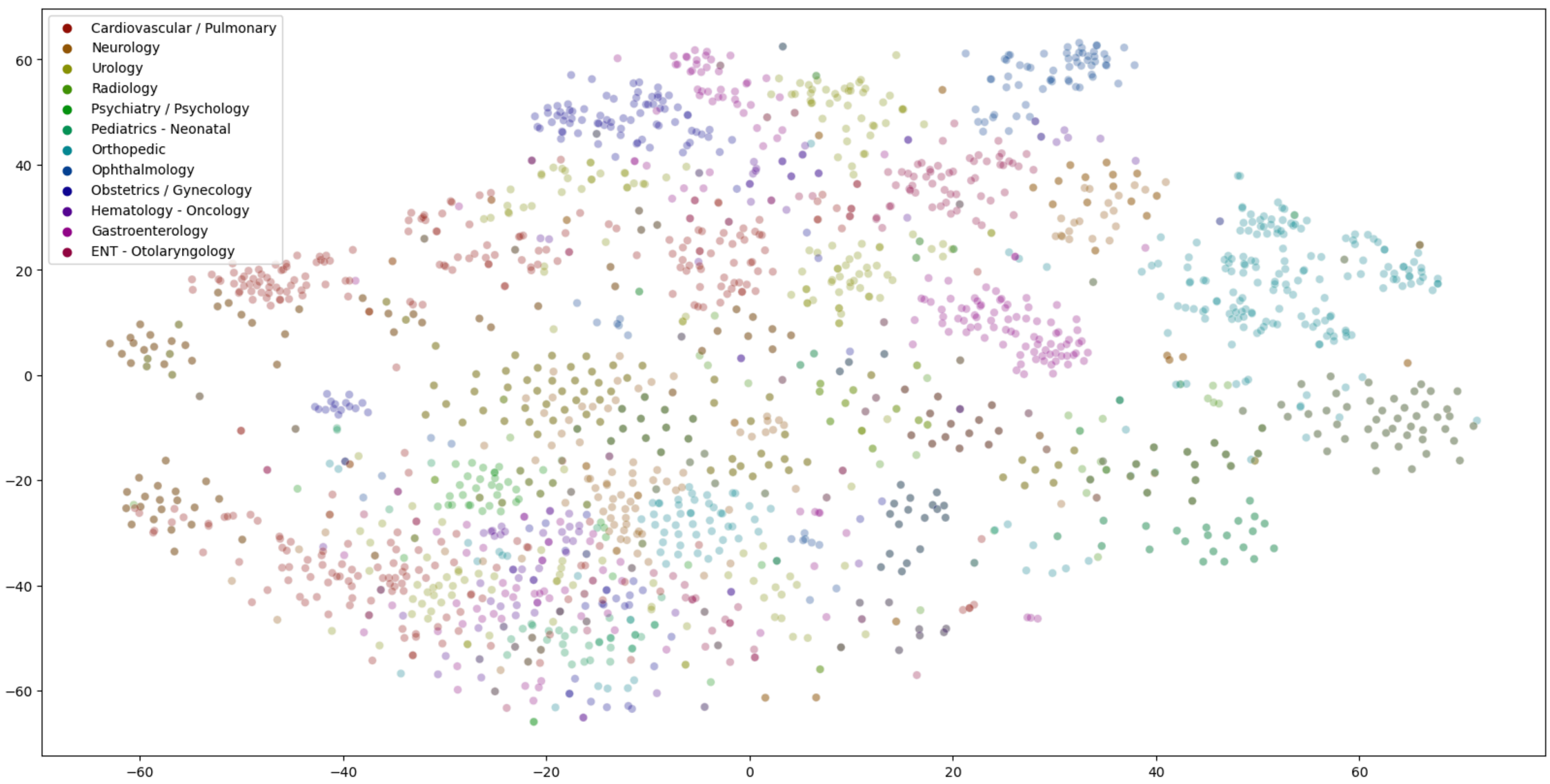
Post regression result analysis:

	precision	recall	f1-score	support
Cardiovascular / Pulmonary	0.33	0.29	0.31	93
Neurology	0.42	0.25	0.31	56
Urology	0.33	0.13	0.19	39
General Medicine	0.24	0.09	0.13	65
Surgery	0.44	0.79	0.57	272
SOAP / Chart / Progress Notes	0.38	0.36	0.37	42
Radiology	0.34	0.34	0.34	68
Psychiatry / Psychology	0.00	0.00	0.00	13
Pediatrics - Neonatal	0.00	0.00	0.00	17
Pain Management	1.00	0.20	0.33	15
Orthopedic	0.43	0.24	0.30	89
Ophthalmology	0.50	0.19	0.28	21
Obstetrics / Gynecology	0.11	0.03	0.04	39
Neurosurgery	0.00	0.00	0.00	24
Nephrology	1.00	0.05	0.10	20
Hematology - Oncology	0.00	0.00	0.00	22
Gastroenterology	0.29	0.07	0.11	56
ENT - Otolaryngology	0.00	0.00	0.00	24
Emergency Room Reports	0.00	0.00	0.00	19
Discharge Summary	0.50	0.56	0.53	27
Consult - History and Phy.	0.30	0.69	0.42	129
accuracy			0.38	1150
macro avg	0.32	0.20	0.21	1150
weighted avg	0.35	0.38	0.32	1150

Spacy Lib for NLP tasks

- SpaCy is a popular open-source library in Python used for natural language processing (NLP) tasks. It provides efficient and easy-to-use tools for processing and analyzing text data. Some key functionalities and tasks that SpaCy can perform include:
- The function used in the code ie nlp(text) processes the input text using the loaded SpaCy model (nlp) and generates a Doc object (doc) containing the analyzed information such as tokens, part-of-speech tags, and named entities.
- Remaining steps of data pre processing remain the same ie cleaning, lemmatization, tfidf vectorization etc

Visual representation of data post spacy implementation:



Model report (with spacy implementation)

	precision	recall	f1-score	support
Cardiovascular / Pulmonary	0.65	0.81	0.72	93
Neurology	0.55	0.65	0.59	79
Urology	0.79	0.85	0.82	59
Radiology	0.36	0.32	0.34	68
Psychiatry / Psychology	0.80	0.62	0.70	13
Pediatrics - Neonatal	0.67	0.44	0.53	18
Orthopedic	0.73	0.71	0.72	89
Ophthalmology	1.00	0.90	0.95	21
Obstetrics / Gynecology	0.80	0.72	0.76	39
Hematology - Oncology	0.50	0.36	0.42	22
Gastroenterology	0.77	0.71	0.74	56
ENT - Otolaryngology	0.89	0.71	0.79	24
accuracy			0.67	581
macro avg	0.71	0.65	0.67	581
weighted avg	0.67	0.67	0.67	581

SMOT

There is marked improvement in results. Since some classes are in minority we can use SMOTE(Synthetic Minority Over-sampling Technique) to generate more sample form minority class to solve the data imbalance problem

- SMOTE works by creating synthetic samples that are similar to existing samples in the minority class but are slightly modified to introduce variations.
- It does this by selecting a minority class sample and finding its k nearest neighbors in the feature space (using a distance metric like Euclidean distance).

Final Result:

	precision	recall	f1-score	support
Cardiovascular / Pulmonary	0.70	0.78	0.74	93
Neurology	0.53	0.54	0.54	79
Urology	0.74	0.83	0.78	59
Radiology	0.33	0.31	0.32	68
Psychiatry / Psychology	0.89	1.00	0.94	93
Pediatrics - Neonatal	0.62	0.47	0.53	17
Orthopedic	0.68	0.71	0.70	89
Ophthalmology	1.00	0.86	0.92	21
Obstetrics / Gynecology	0.81	0.74	0.77	39
Hematology - Oncology	0.53	0.35	0.42	23
Gastroenterology	0.77	0.66	0.71	56
ENT - Otolaryngology	0.90	0.75	0.82	24
accuracy			0.70	661
macro avg	0.71	0.67	0.68	661
weighted avg	0.69	0.70	0.69	661

THANK YOU