# Medical Text Analysis

Springboard Capstone 3

# Problem Statement

Use modern NLP methods to classify medical abstracts written by doctors to classify patient diagnosis

Diagnosis classes: digestive system disease, cardiovascular system disease, neoplasm, nervous system disease, general pathological disease

Compare results with modern AI chatbot

We want to answer:

- Can ML models classify complex medical text?
- What attributes are most important to model prediction?
- Can AI chatbots do as well as modern ML models trained with the data?

# The Data

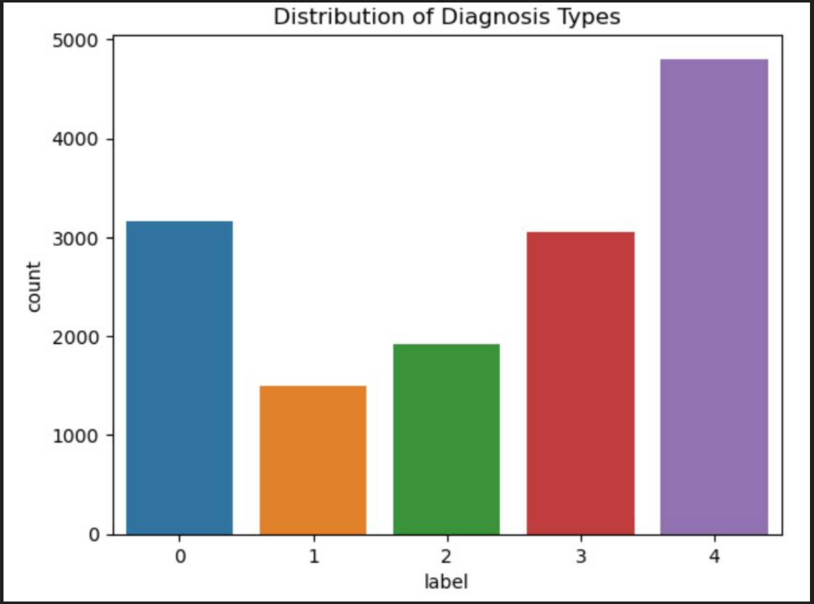
*Medical Text,* is publicly available on Kaggle

The train data contains text and diagnosis label for patients

The test data is not used in this project due to lack of labels

We train and test on 14438 records

# The Data



0 = neoplasms 1 = digestive system disease 2 = nervous system disease

3 = cardiovascular disease 4 = general pathological diseases

## Feature Engineering

I wanted to create descriptive features of the text to use during modeling

We modeled with and without the added features to test their effectiveness

- Character count
- Word count
- Capital word count
- Quotations count
- Sentence count
- Unique word count
- Stopword count
- Average word length
- Average sentence length
- Unique word ratio
- Stopword ratio
- Verb count
- Adverb count
- Adjective count
- Noun count

# Text Preprocessing

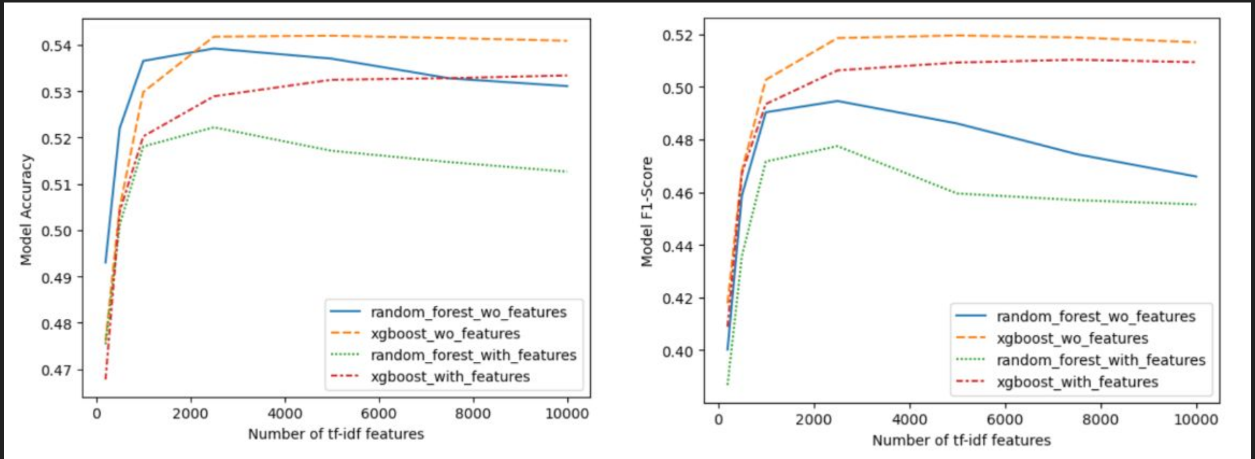
Preprocessing steps:

- Tokenize text
  - Break text up into words
- Lowercase all characters
- Remove stopwords
  - Remove common words (e.g. a, the, and, it, for, but, my, your)
- Lemmatize text
  - Group together different inflections of same word (e.g. improving, improvements, improved all stem to improve)
- TF-IDF vectorization
  - Weights the word counts by how often they appear in the text

# TF-IDF Vectorizer Parameter Tuning

By default, the vectorizer creates a feature for every unique word, but with over 14000 medical abstracts, this number is too large

I parameter tested the ‘max_features’ parameter with baseline Random Forest and XGBoost models and selected 2500 features

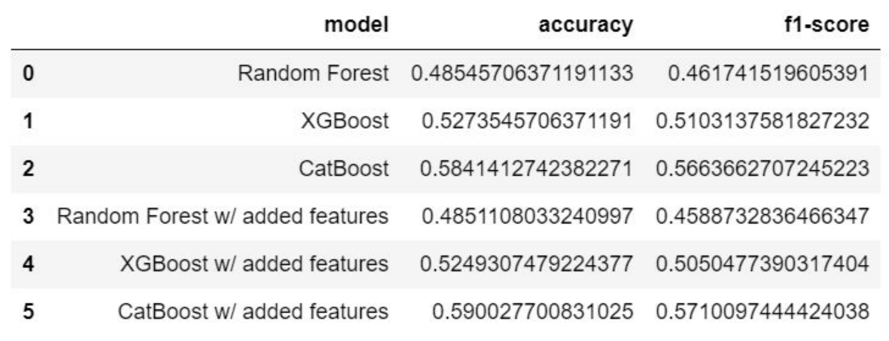


# Modeling

We trained and tested baseline Random Forest, XGBoost, and CatBoost classifiers with and without our added features from feature engineering

We use accuracy and F1-score as performance metrics because data is mostly balanced and these metrics will reflect misclassifications

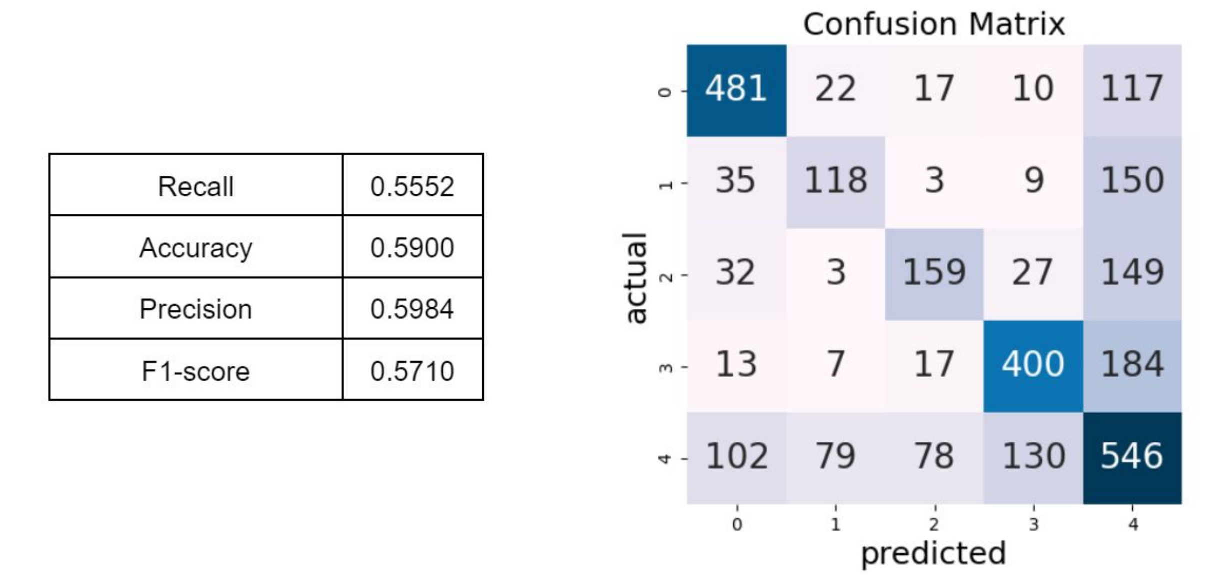
The best model is using CatBoost with the added features



|  | model | accuracy | f1-score |
|---|---|---|---|
| 0 | Random Forest | 0.48545706371191133 | 0.461741519605391 |
| 1 | XGBoost | 0.5273545706371191 | 0.5103137581827232 |
| 2 | CatBoost | 0.5841412742382271 | 0.5663662707245223 |
| 3 | Random Forest w/ added features | 0.4851108033240997 | 0.4588732836466347 |
| 4 | XGBoost w/ added features | 0.5249307479224377 | 0.5050477390317404 |
| 5 | CatBoost w/ added features | 0.590027700831025 | 0.5710097444424038 |

# Modeling

## Full CatBoost classification results:



| | |
|---|---|
| Recall | 0.5552 |
| Accuracy | 0.5900 |
| Precision | 0.5984 |
| F1-score | 0.5710 |

Confusion Matrix

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 481 | 22 | 17 | 10 | 117 |
| 1 | 35 | 118 | 3 | 9 | 150 |
| 2 | 32 | 3 | 159 | 27 | 149 |
| 3 | 13 | 7 | 17 | 400 | 184 |
| 4 | 102 | 79 | 78 | 130 | 546 |

# Modeling

Given the large number of features (2515) and our lack of computing power, parameter tuning the CatBoost model was not attainable

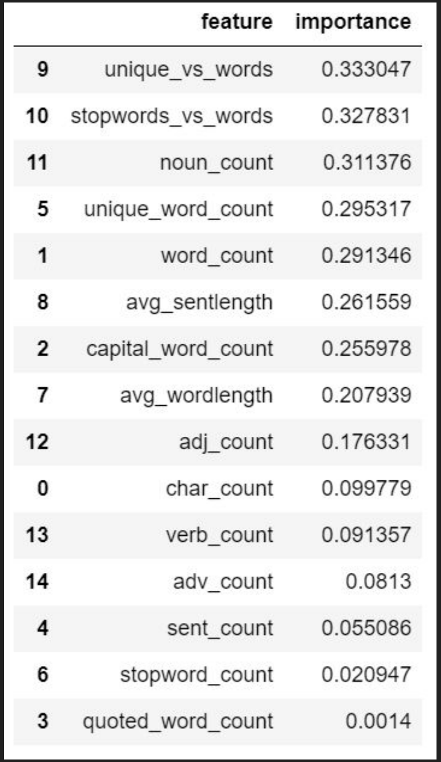
We parameter tuned the Random Forest classifier as a proof of concept

- We were able to gain an increase of ~4% in accuracy
- Parameters: max_depth=20, min_samples_split=20, n_estimators=200

Parameter tuning the CatBoost model would be beneficial in future work

# Modeling

The feature importance of the added features are:



|  | feature | importance |
|---|---|---|
| 9 | unique_vs_words | 0.333047 |
| 10 | stopwords_vs_words | 0.327831 |
| 11 | noun_count | 0.311376 |
| 5 | unique_word_count | 0.295317 |
| 1 | word_count | 0.291346 |
| 8 | avg_sentlength | 0.261559 |
| 2 | capital_word_count | 0.255978 |
| 7 | avg_wordlength | 0.207939 |
| 12 | adj_count | 0.176331 |
| 0 | char_count | 0.099779 |
| 13 | verb_count | 0.091357 |
| 14 | adv_count | 0.0813 |
| 4 | sent_count | 0.055086 |
| 6 | stopword_count | 0.020947 |
| 3 | quoted_word_count | 0.0014 |

It is beneficial to use the added features and including more would be useful in future work

# OpenAI ChatGPT Comparison

Now, we want to compare our ML model results with OpenAI

We will test using our preprocessed data (stopwords removed, etc.), raw data, and misclassified data from the CatBoost model

We will try both curie engines and davinci engines from GPT-3 and GPT-3.5 models, respectively

The curie models are faster and about 10x cheaper to use

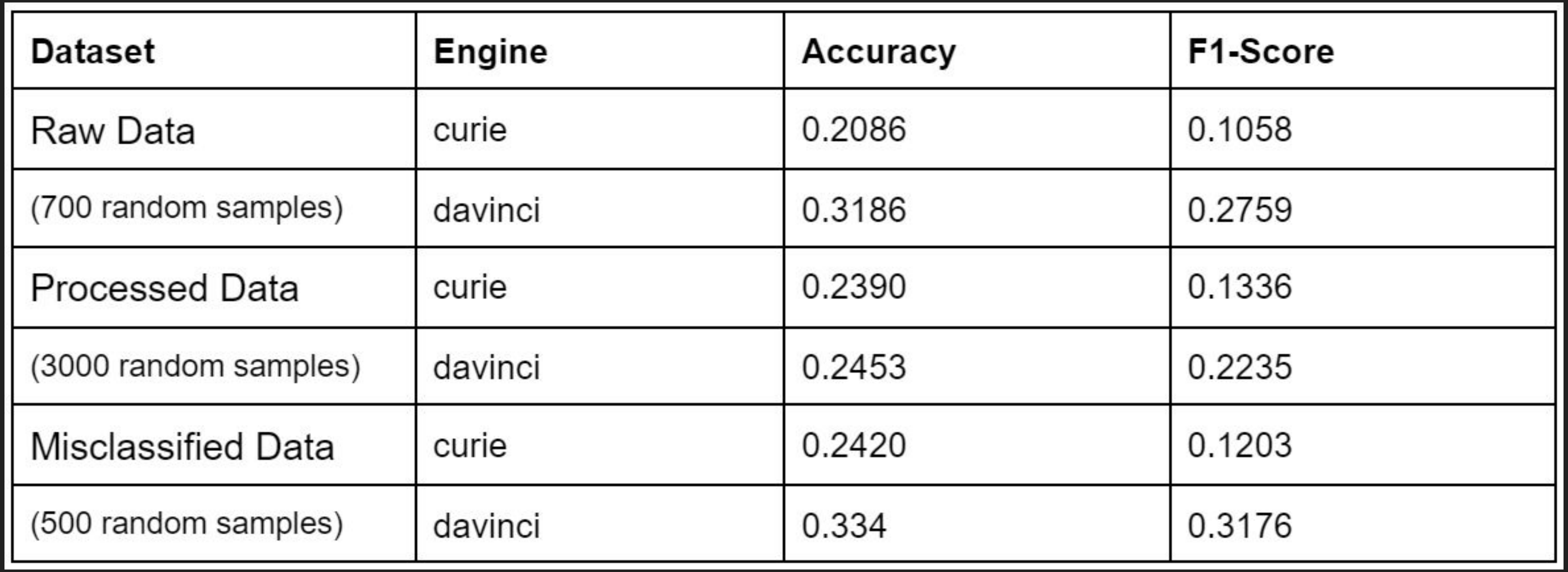
# OpenAI ChatGPT Comparison

An example of a (cardiovascular disease) prompt I use for the API request is:

*Is the diagnosis a digestive system disease, cardiovascular disease, neoplasms, nervous system disease, or general pathological condition if the patient has the following conditions: catheterization laboratory event hospital outcome direct angioplasty acute myocardial infarction safety direct infarct angioplasty without antecedent thrombolytic therapy catheterization laboratory hospital event assess consecutively treated patient infarction involve leave anterior descend patient right circumflex coronary artery group patient similar age leave anterior descend coronary artery year right coronary artery year circumflex coronary artery year patient multivessel disease leave anterior descend coronary artery right coronary artery circumflex coronary artery patient initial grade antegrade flow leave anterior descend coronary artery right coronary artery circumflex coronary artery cardiogenic shock present eight patient infarction leave anterior descend coronary artery four infarction right coronary artery four infarction circumflex coronary artery major catheterization laboratory event cardioversion cardiopulmonary resuscitation dopamine intra aortic balloon pump support hypotension urgent surgery occur patient infarction leave anterior descend coronary artery eight infarction right coronary artery four infarction circumflex coronary artery shock six nonshock patient less one laboratory death shock patient infarction leave anterior descend coronary artery*

# OpenAI ChatGPT Comparison

Complete results:



| Dataset | Engine | Accuracy | F1-Score |
|---|---|---|---|
| Raw Data | curie | 0.2086 | 0.1058 |
| (700 random samples) | davinci | 0.3186 | 0.2759 |
| Processed Data | curie | 0.2390 | 0.1336 |
| (3000 random samples) | davinci | 0.2453 | 0.2235 |
| Misclassified Data | curie | 0.2420 | 0.1203 |
| (500 random samples) | davinci | 0.334 | 0.3176 |

# OpenAI ChatGPT Comparison

We determined that davinci performs much better than curie engine

However, none of the results are as good as any of our ML model’s results

The chatbots perform better with the raw data which includes stopwords, punctuation, etc.

The davinci engine performed its best using a sample of the misclassified data from the CatBoost model

# Conclusion

We performed feature engineering, text preprocessing, and modeling with 5-class classification for complex medical text data

We compared our results with OpenAI chatbot using API requests

Found unique insights on medical diagnosis classifications and how it can fit in future medical world

## Future Work

Train and test with more labeled data

Include more feature engineering of the text data

Parameter tune the CatBoost model

Test more AI Chatbots including the state-of-the-art GPT-4 and Google PaLM’s medical api which will be released soon

Utilize transfer learning to train a better performing deep model with chatbot