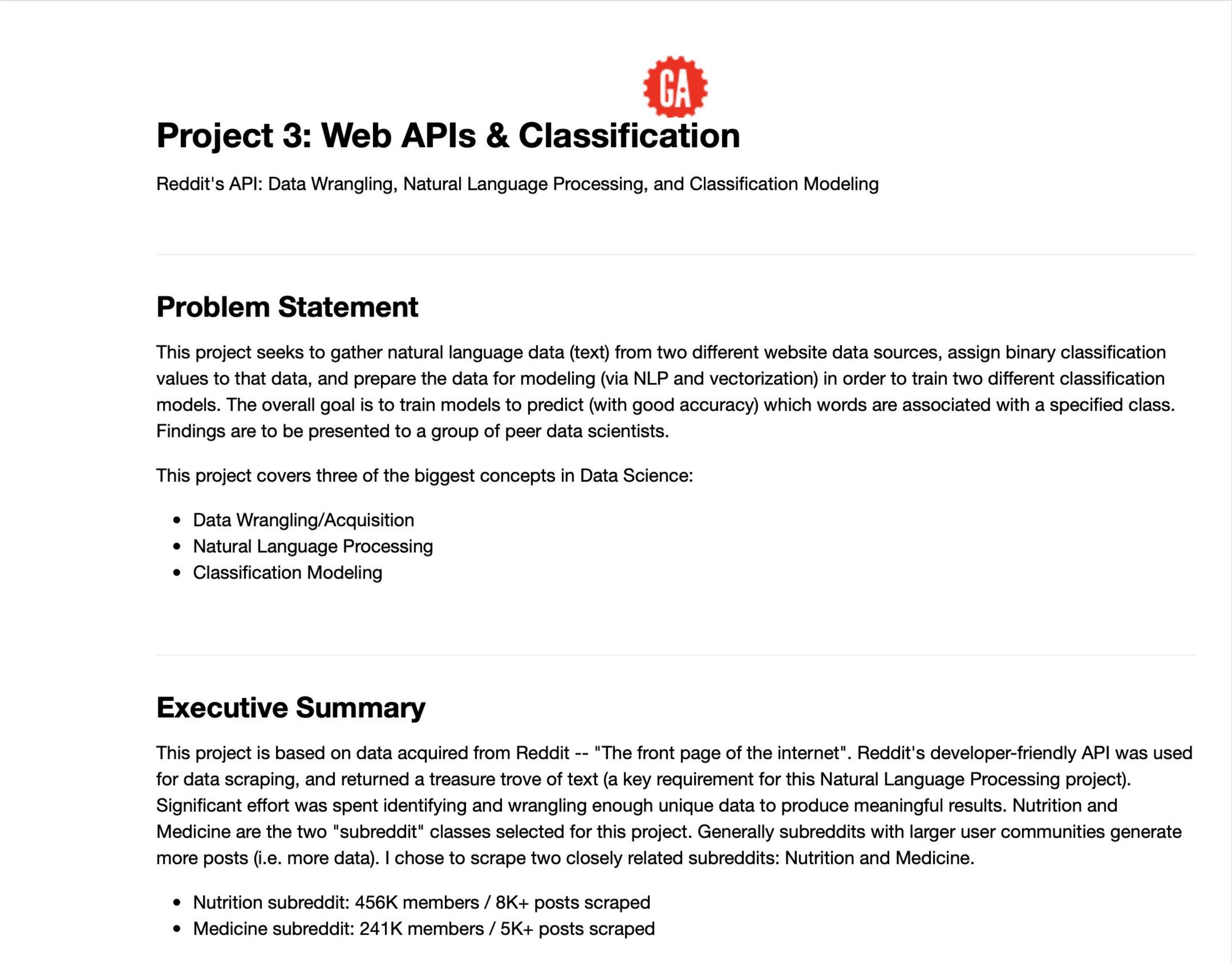
Project 3: Web APIs & Classification

General Assembly
DSI8



Project 3: Web APIs & Classification

Reddit's API: Data Wrangling, Natural Language Processing, and Classification Modeling

Problem Statement

This project seeks to gather natural language data (text) from two different website data sources, assign binary classification values to that data, and prepare the data for modeling (via NLP and vectorization) in order to train two different classification models. The overall goal is to train models to predict (with good accuracy) which words are associated with a specified class. Findings are to be presented to a group of peer data scientists.

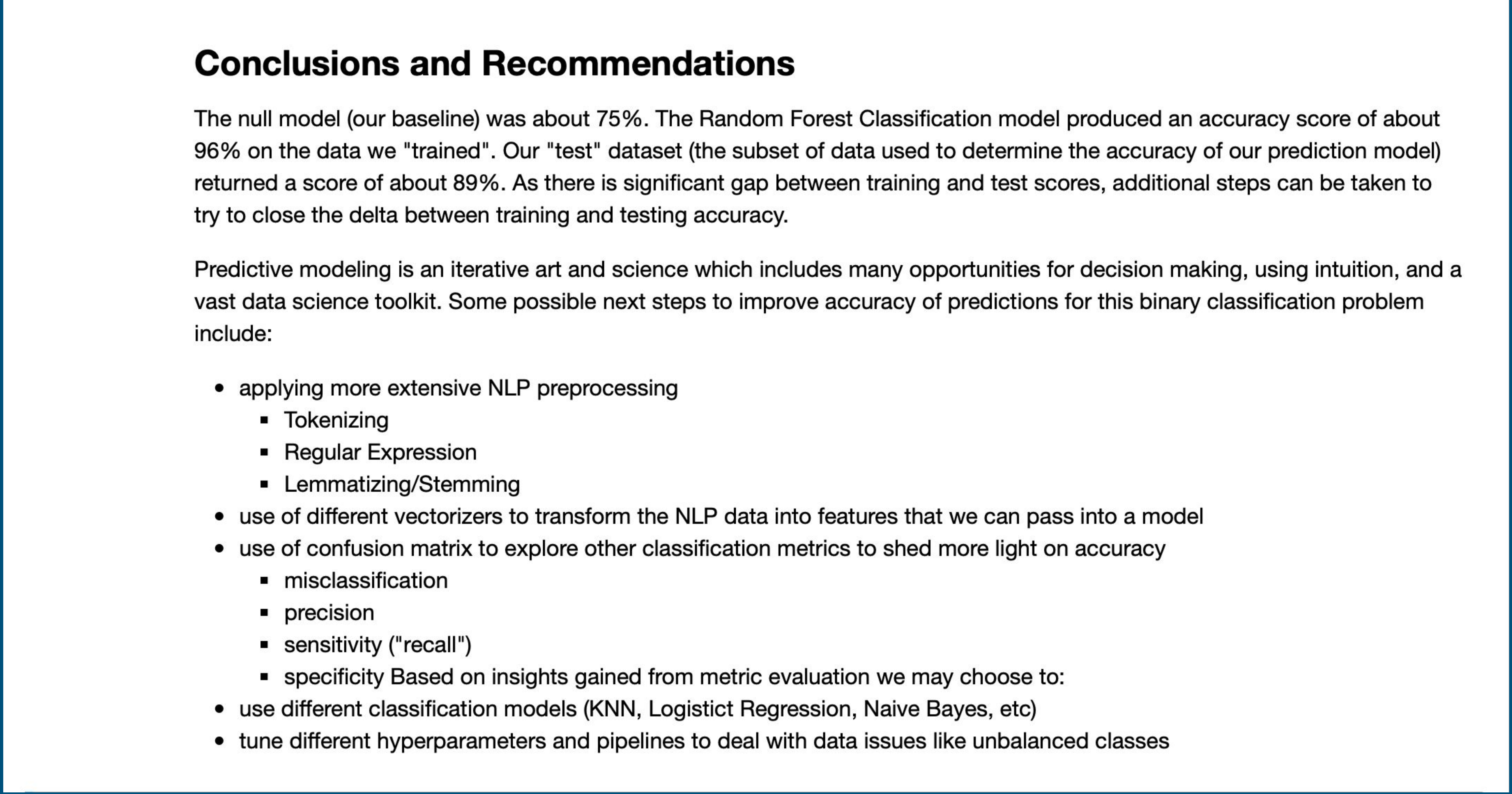
This project covers three of the biggest concepts in Data Science:

- Data Wrangling/Acquisition
- Natural Language Processing
- Classification Modeling

Executive Summary

This project is based on data acquired from Reddit -- "The front page of the internet". Reddit's developer-friendly API was used for data scraping, and returned a treasure trove of text (a key requirement for this Natural Language Processing project). Significant effort was spent identifying and wrangling enough unique data to produce meaningful results. Nutrition and Medicine are the two "subreddit" classes selected for this project. Generally subreddits with larger user communities generate more posts (i.e. more data). I chose to scrape two closely related subreddits: Nutrition and Medicine.

- Nutrition subreddit: 456K members / 8K+ posts scraped
- Medicine subreddit: 241K members / 5K+ posts scraped

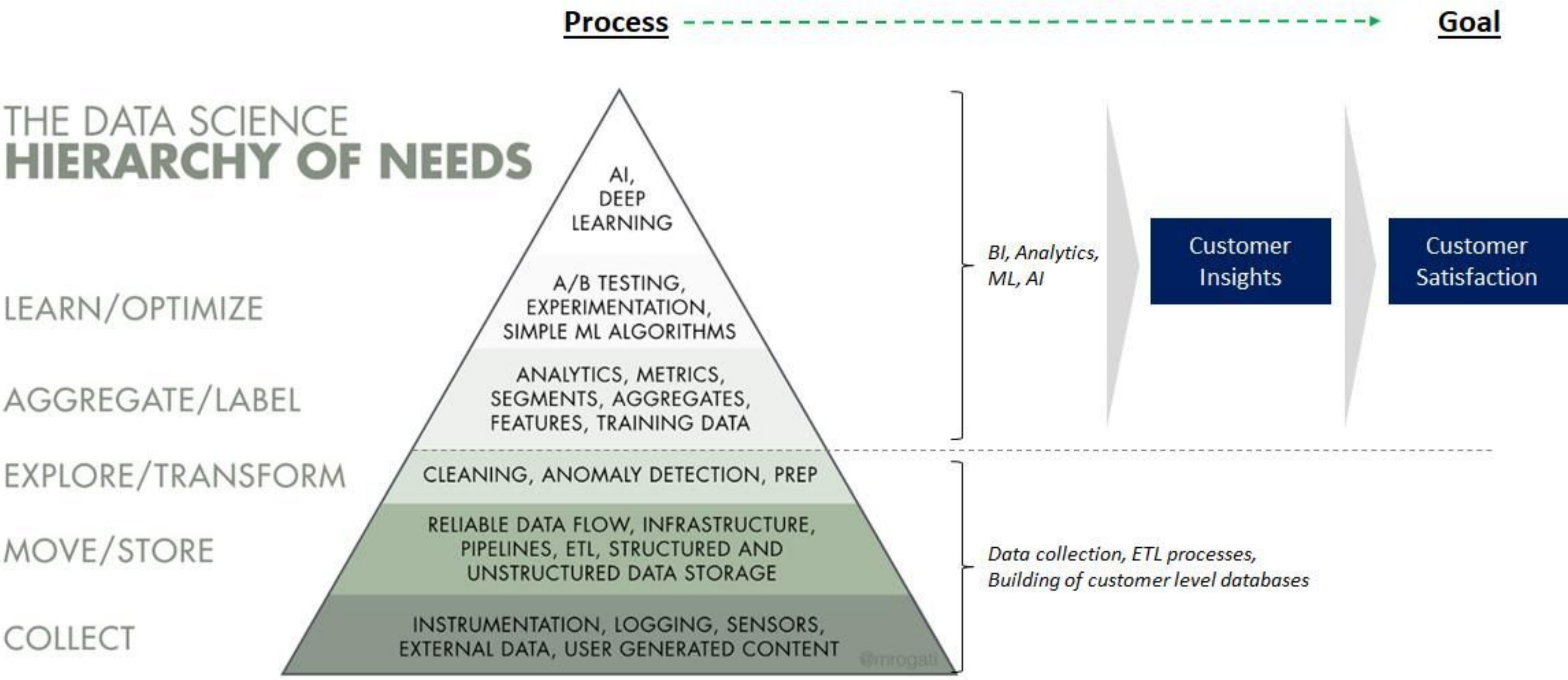


Conclusions and Recommendations

The null model (our baseline) was about 75%. The Random Forest Classification model produced an accuracy score of about 96% on the data we "trained". Our "test" dataset (the subset of data used to determine the accuracy of our prediction model) returned a score of about 89%. As there is significant gap between training and test scores, additional steps can be taken to try to close the delta between training and testing accuracy.

Predictive modeling is an iterative art and science which includes many opportunities for decision making, using intuition, and a vast data science toolkit. Some possible next steps to improve accuracy of predictions for this binary classification problem include:

- applying more extensive NLP preprocessing
  - Tokenizing
  - Regular Expression
  - Lemmatizing/Stemming
- use of different vectorizers to transform the NLP data into features that we can pass into a model
- use of confusion matrix to explore other classification metrics to shed more light on accuracy
  - misclassification
  - precision
  - sensitivity ("recall")
  - specificity Based on insights gained from metric evaluation we may choose to:
- use different classification models (KNN, Logistict Regression, Naive Bayes, etc)
- tune different hyperparameters and pipelines to deal with data issues like unbalanced classes



Process → **Goal**

THE DATA SCIENCE
HIERARCHY OF NEEDS

LEARN/OPTIMIZE

AGGREGATE/LABEL

EXPLORE/TRANSFORM

MOVE/STORE

COLLECT

AI, DEEP LEARNING

A/B TESTING, EXPERIMENTATION, SIMPLE ML ALGORITHMS

ANALYTICS, METRICS, SEGMENTS, AGGREGATES, FEATURES, TRAINING DATA

CLEANING, ANOMALY DETECTION, PREP

RELIABLE DATA FLOW, INFRASTRUCTURE, PIPELINES, ETL, STRUCTURED AND UNSTRUCTURED DATA STORAGE

INSTRUMENTATION, LOGGING, SENSORS, EXTERNAL DATA, USER GENERATED CONTENT

@mrogati

BI, Analytics, ML, AI

Customer Insights

Customer Satisfaction

Data collection, ETL processes, Building of customer level databases

Higher Hierarchy Tasks

- Data Vizualizations
- Descriptive & Inferential Statistics
- Outside Research*
- Model Tuning
- Conclusions & Recommendations