Approach

April 28, 2020

Problem statement

A new pharmaceutical startup is recently acquired by one of the world's largest MNCs. For the acquisition process, the startup is required to tabulate all drugs that they have sold and account for each drug's effectiveness. A dedicated team has been assigned the task to analyze all the data. This data has been collected over the years and it contains data points such as the drug's name, reviews by customers, popularity and use cases of the drug, and so on. Members of this team are by the noise present in the data.

Your task is to make a sophisticated NLP-based Machine Learning model that has the mentioned features as the input. Also, use the input to predict the base score of a certain drug in a provided case.

Data

The dataset has the following columns:
patient_id: ID of patients
name_of_drug: Name of the drug prescribed
use_case_for_drug:Purpose of the drug
review_by_patient:Review by patient
drug_approved_by_UIC:Date of approval of the drug by UIC
number_of_times_prescribed:Number of times the drug is prescribed
effectiveness_rating:Effectiveness of drug
base_score:Generated score (Target Variable)

Data description

The data folder consists of the following two .csv files:
train.csv - (32165x 7)
test.csv - (10760x6)

Approach

This is a regression problem. Approach to solve this problem is as follows:

1. Data Engineering: After observing train.csv , find out corroupted usecases like 'users found this comment helpful'. I imputed those values with maximum value of 'use_case_for_drug' for that drug name and results are stored in train1.csv

2. Descriptive analysis is done:

- Find mean, median and standard deviation.
- Checking for missing or null values
- Top 10 most reviewed drug names

- Top 10 most suffered condition by reviewers
- Top 10 drugs with best and worst rating
- Top 10 'significant' drugs with best and worst rating"
- Encoding the categorical attributes: -#'name_of_drug','use_case_for_drug'. -# date is converted to day, month and year.
- Sentiment score is computed for each review.

(Please refer 'descriptive-analysis.ipynb')

3. Feature Engineering:

    - Correlation with target attribute base_score is computed.
    - After model building try to understand feature importance.

4. I tried to solve this regression problem with following tree based algorithms:

    - Random Forest
    - XGBoost
    - Lightgbm

    (Plaese refer 'Xgboost.py' and 'Xgboost.ipynb', 'lgb.py' and 'RF.py')
    Observation is XGboost performed well for this problem. Analysis is done with tuning different hyper-parameters and cross-validation.