Amazon product review Helpfulness - Kaggle Competition

(Dataset - Amazon Product Reviews)

1. **Purpose**
   - Create a Predictive model for the "**Usefulness of an Amazon product review"** (the problem statement can be predicting the ratio of "number of Helpful reviews" over "total number of reviews" for a given product.

2. **Data Source**
   - Training Data, provided in the form of json file (train.json.gz). It contains information on:
     1. total number of reviews received (*"out of"*)
     2. how many of them were helpful (*"nHelpful"*)
   - Testing data provided (test_Helpful.json.gz). Only contains information on:
     1. total number of reviews received (*"out of"*)

3. **Task**
   - Identify and create informative features (independent variables) from given Amazon
   - On Testing data predict
     - how many of the reviews will be helpful (Target Variable)
   - Scoring metric is Mean Absolute Error (MAE)

4. **Data Loading and preparation (Feature Selection)**

   a) For Data loading into Pandas Data Frame and parsing Baseline code was provided as a reference.
   b) Feature Selection process – Used **Reverse Ablation technique** to create minimal set of features (metrics used was MAE). Following table is the final features used:

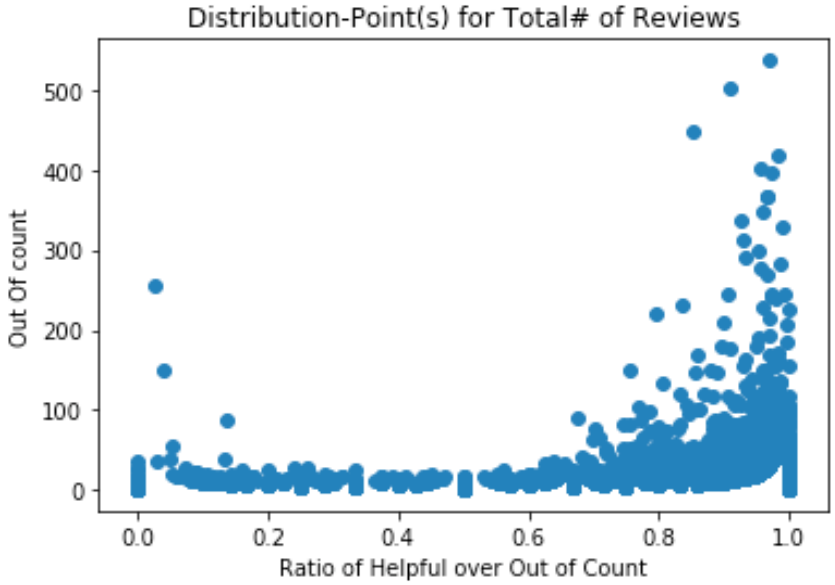
No.	Name	Description
1.	**rating**	User rating of the product (range 1 to 5 star)
2.	**len_reviewText**	Length of Review text
3.	**ratio_reviewOverSummary**	Ratio of length of Review over Length of Summary
4.	**outOf**	Total number of reviews received
5.	**nSentence_Clause**	Total number of sentences, clauses and comma separated words (i.e. cumulative count)

c) The above list has been filtered out using various experiments like best feature importance from Regression, Ablation technique, plotting graphs, correlation matrix etc.

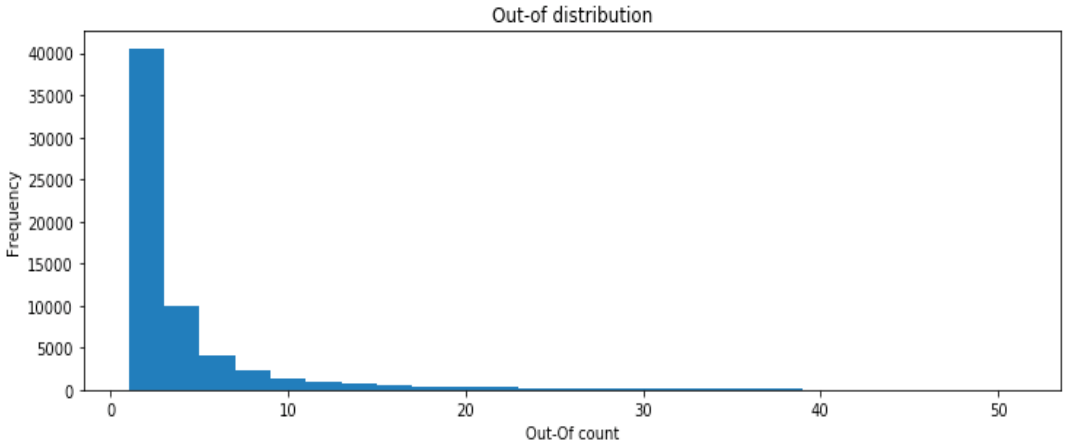
d) Target Variable ("ratio_nHelpful"):
   - **I.** It's the ratio of total number of reviews received (*"outOf"*), how many of them were helpful (*"nHelpful"*).
   - **II.** From the training set calculated the Target variable values (as *"outOf" and nHelpful"* both values are available.

5. EDA

a. Outlier Cases and Data Cleansing
   - **i.** Cases where "outOf" equals zero were dropped (as number of helpful in such cases will also be zero)
   - **ii.** Cases where "outOf" is greater 300 are few and are outliers
   - **iii.** Referenced the following scatter matrix plot to identify such cases for outlier points in Training data set.



b. Histogram of "out of" count, as can be noticed that the most (highest frequency) of the reviews have count equal to one.



Also the distribution has a long tail-end i.e. indicating that the behavior and model will be different at the section after and before the long tail. This also alludes to use more than one ML algorithm (model), something like an ensemble approach.

6. Modelling & solution approach

a. Split the test data into test and validation set using "Train_Test_Split" mechanism).

b. Used Ensemble method ***"GradientBoostingRegressor"*** and performed GridSearchCV to find the best parameters for fit and validation data set.

c. Measurement metrics used is MAE with final value = 0.1563

d. For this given set following are the best parameters (clf.best_params_):

{'learning_rate': 0.05,

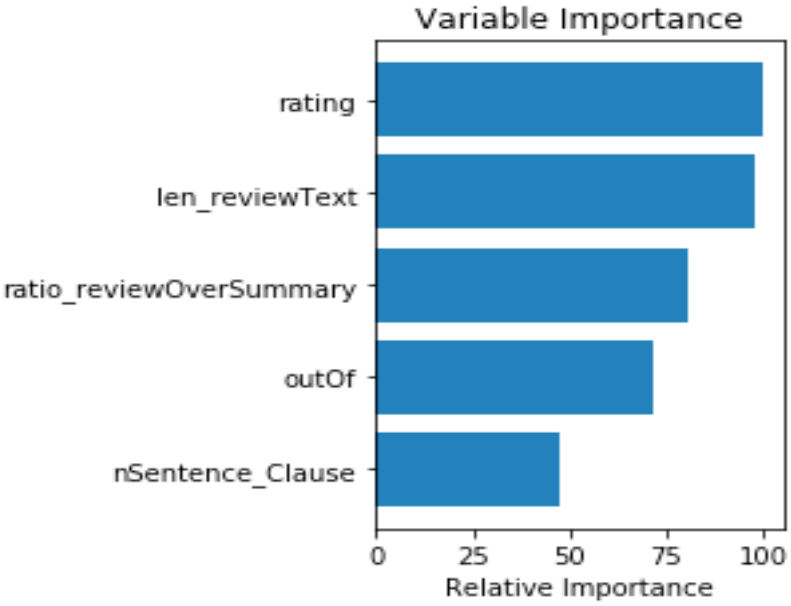
'loss': 'ls',

'max_depth': 4,

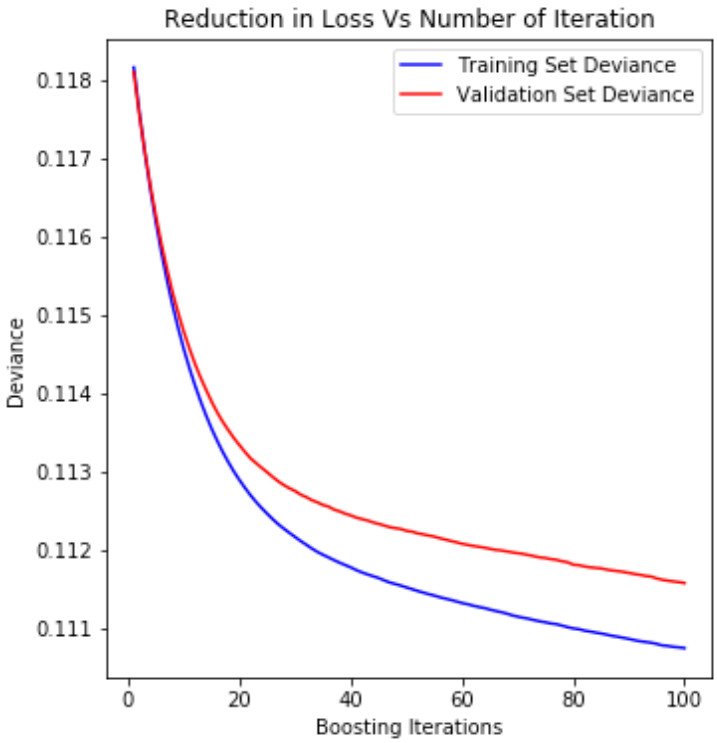
'n_estimators': 100}

e. Predictions were made and rounded before writing to file (predictions_Helpful.txt)

f. Feature importance depicted and identified as:



g. To strike a balance between boosting iterations & learning iterations analyzed the loss function for validation data (LSE). Figure below illustrates after 100 iterations deviation from actual levels-off:



7. Limitations & Recommendations future versions:

a) Case of Overfitting – noticed after final result that this approach was overfitting so using "`StratifiedKFold`" might alleviate this issue
b) As noticed in long tail graph in data-distribution, usage of different models on different parts of data will improve the prediction results.
c) Create more good features like "Sentiments Polarity", readability of text so on and so forth.
d) Use Logistic-Regression i.e. classification for cases when 'outOf' ==1 and Regression model for reviews for cases when 'out of' > 1.
e) All the selected features can transformed using Polynomial Features (**sklearn.preprocessing.PolynomialFeatures**) degree 2 and thereafter "Ablation experiment" can be performed to select the best features.
f) Use XGBoost instead of GBM (for both classification and regression).