

Media and Feedback Monitoring Using Machine Learning

Ms. Mandira Banik¹, Mr Sudeep Ghosh², Dhrubajyoti Adhikary³, Iman Das³, Arnab Banerjee³

¹⁻⁶ Guru Nanak Institute Of Technology, Kolkata

1. Introduction

In public relations and Comms, estimating a campaign's success (Steinberger et al., 2013), dealing with a crisis quickly, and listening out for negative publicities are imperative and critical. These tasks are achieved by effectual media monitoring. It uses ML strategies to crawl and index online platforms(Waardenburg et al., 2014). After indexing, it searches those media outlets for customers' reviews/suggestions, developments, trends, contemporary rivals, negative publicities, etc.

1.1 Background research

For a brand/product's triumph, user feedbacks are momentous(Alberghini et al., 2014). Retrieving/getting a piece of detailed information from even a single medium like Facebook, YouTube, Flipkart, Amazon, Etc., is quite a time-consuming task and also impossible to get it instantly by the manual system. So, Media monitoring exists to improve a company's performance and productivity(Zin T.). Large companies have separate and dedicated sectors/departments as a media praepostor to track the pertinent inventions and upgradation and to have eyes like a hawk on how recent news is representing them(Steinberger et al., 2013). In this era, customers share their views(Gasco et al., 2019) and feedback/suggestions about products and services on various digital platforms(Alorini et al., 2021). Negative publicities affect(downgrade) the reputation and economy of a company. So, making the customers aware of the truth about the product and caring is the way to shield the company's reputation. A company can improve its products and services, manage its reputation, and respond even in a crisis by monitoring customer reviews(Havas et al., 2021). The business market is full of competitors, so knowing about the contemporary rivals of a company is also a critical and crucial task for surviving in the market. Moreover, knowing about the latest and future trends and developments is also a crucial and significant business task. Therefore, it is essential to develop a media monitor.

1.2 Aim of the paper

The project aims to develop an ML model that tracks and arranges customers' online mentions/feedback according to their sentiments so that companies can improve and elevate their products and services. We aim to rank the comments by sentiment analysis so we do not get hung up on other details. In an advanced era where market competition has become enormous, for a company/business to survive and beat others, keeping track of rival companies' inventions and development status is essential. Keeping track of the latest and future trends, the scope of the progress of their products and services, taking care

of customers' needs, keeping track of negative publicities, and convincing customers is a critical task. So, a tool like a media monitor is required to surveil the media flow to fulfill the above tasks.

1.3 Research Questions

1. Can the model surveil the online mentions properly and contribute to the NLP technology?
2. Can the model rank the posted reviews and arrange them according to their sentiments?
3. Does the deep learning model I explored and finetuned give a better result than other deep learning models?
4. How does the training dataset affect the model?

The research project is heading in a python programming language with deep learning models. I have gone through quite a few research papers relevant to the topic to accomplish the task as it gives a complete idea and the right direction about an issue; it also gives the idea and familiarity with the latest developments and the loopholes in a particular field of technology.

2. Literature Review

Hu L. et al. (2019) proposed an integrated approach of some ML models and a hedonic model. The authors collated the information from several online housing rental websites. Models used by the author are random forest, extra trees, gradient boosting, support vector regression, MLP-NN, and k-NN. After training the models, the author calculated the relative importance of the determinants, and then with the help of partial dependence plots, the author visualized it. And then, the models get ready to be used in monitoring housing rental cost dynamics inside Shenzhen. The author concluded that random forest regression and extra-trees regression performed very well among the ML models he exploited and experimented upon. The following tables represent the acquired accuracy by exploiting the mentioned models:

Model performance in October 2017:

	RFR	ETR	GBR	SVR	MLP-NN	k-NN
RMSE	6.951	7.053	7.514	10.54	7.231	7.364
%RMSE	10.53%	10.68%	11.38%	15.96%	10.95%	11.15%
MAE	5.568	5.679	5.809	7.790	5.718	5.854
%MAE	8.92%	9.10%	9.06%	12.98%	8.96%	9.52%
P	0.911	0.909	0.909	0.870	0.910	0.905
R square	0.740	0.732	0.696	0.403	0.719	0.708

Model performance in February 2018:

	RFR	ETR	GBR	SVR	MLP-NN	k-NN
RMSE	8.506	8.447	9.321	11.62	9.194	9.007
%RMSE	12.65%	12.56%	13.86%	17.28%	13.67%	13.39%

MAE	7.055	6.945	7.540	9.321	7.688	7.424
%MAE	11.40%	11.20%	11.66%	15.33%	12.33%	11.95%
P	0.886	0.888	0.883	0.847	0.877	0.881
R square	0.711	0.715	0.653	0.460	0.662	0.676

Lian et al. (2022) explored many machine learning approaches like SVM, random forest, logistic regression, extra trees, and gradient boosting and got good accuracies in the field of media monitor. They first collected tweets related to the COVID-19 vaccine by exploiting the Twitter streaming API. Almost all the selected tweets were about an individual's personal experience with the COVID-19 vaccine adverse event. Among the 111,229 selected tweets, they randomly picked 5600 tweets for manual annotation. After annotation, these tweets are served as training, testing, and validation dataset as 7:2:1 for classification (i.e., removing the tweets from the annotated dataset that are not about the personal experience about the conducted VAE) and recognition of labeled entity such as type of the vaccine, dose, and symptoms/VAE. Then for entity normalization, the authors leveraged an NLP pipeline, Conditional Random Field, provided by CLAMP. Then they scrutinized the extracted data based on time, frequency, and location. Among all the machine learning models that the author exploited, the random forest gives the best F1 score, 0.926. Following are the outputs provided by the exploited machine learning algorithms:

Algorithm		SVM	Extra Trees	Random Forest	Logistic Regression	Gradient boosting
Precision	Included	0.896	0.904	0.906	0.927	0.921
	Excluded	0.898	0.852	0.910	0.878	0.850
Recall	Included	0.940	0.905	0.946	0.921	0.901
	Excluded	0.830	0.850	0.847	0.886	0.879
F1-Score	Included	0.918	0.905	0.926	0.924	0.911
	Excluded	0.863	0.850	0.877	0.882	0.864
Accuracy		0.897	0.884	0.908	0.908	0.892

The following table elaborates the performance of the CRF algorithm of CLAMP:

Entity	Vaccine	Dose	Adverse Event
Precision	0.856	0.851	0.784
Recall	0.846	0.862	0.755
F1-Score	0.851	0.857	0.770

Samuel, J. et al. (2020) used a naïve-based approach and logistic regression model to build a media monitor for tweets classification in their research project. For short tweets, the author got 91% accuracy with a naïve-based algorithm and 74% accuracy with logistic regression.

Waardenburg et al. (2014) developed a social media monitor prototype containing the latest and significant online activities on five famous online platforms (Facebook, Twitter, YouTube, Foursquare, and Flickr) of registered Dutch museums. They used APIs provided by the social media platform to collect the data. The authors used PHP language to extract data fields from the JSON file provided by the APIs. Then they stockpiled the data in a MySQL database. Then they represented meaningful information (from the collected data) like tables, graphs, tag clouds, Google charts, and other open-source tools.

2.1 Summary table of literature review

Author	Best Model	Performance
Hu L. et al	extra-trees regression	0.715 (R square)
Hu L. et al	Random forest regression	0.740 (R square)
Lian et al.	random forest	0.926 (F1 score)
Samuel, J., et al	Naïve-based	91% (accuracy)
	Logistic regression	74% (accuracy)

3. Progress to date

I have studied quite a few research papers and reviews relevant to media monitor and done the literature review. I have also explored some machine learning models and approaches useful for the research project. Later, I took datasets from the Kaggle website. Then I analyzed and exploited it.

Setup

I conducted the project in python language. I used Google Colab for programming because it provides all the facilities required for machine learning projects and a GPU facility. We do not need to download

and install anything on our system; everything just gets stored on the cloud.

Dataset Description

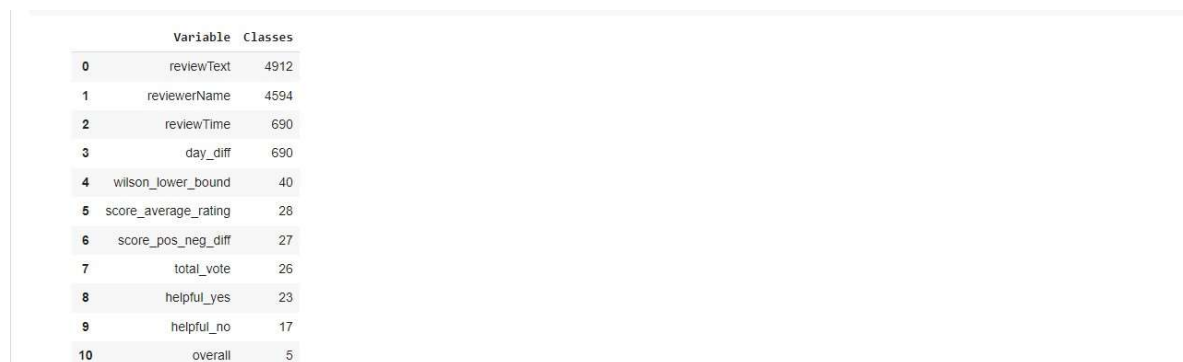
The dataset that I chose and am working on is picked from the Kaggle website; it is a .csv file and contains 4915 rows and 11 columns with reviews and metadata of the reviews given/posted by the customers on the Amazon website. For further finetuning of the model, I trained the model with other datasets taken from the data—world website.

Labels

The dataset contains 11 columns regarding reviews on various products given by customers. The columns are reviews and metadata of the reviews posted by customers.

Class imbalance

After going through the dataset, I found that the dataset has a class imbalance issue. The column 'overall' has only five distinct values, while the column 'reviewText' has 4912 distinct values. The following image shows the class imbalance of the dataset:



	Variable	Classes
0	reviewText	4912
1	reviewerName	4594
2	reviewTime	690
3	day_diff	690
4	wilson_lower_bound	40
5	score_average_rating	28
6	score_pos_neg_diff	27
7	total_vote	26
8	helpful_yes	23
9	helpful_no	17
10	overall	5

Dataset preparation

The .csv file/dataset contains reviews of customers on several products. The dataset was created by gathering customer reviews on the Amazon website. After collecting the dataset, cleaning and preprocessing are done before applying them to the models.

4. Consideration of ethical/legal/professional and social issues

Ethics

I conducted this research work with no human participation. It has never been published anywhere in this manner. It has no conflict of interest.

Legal

The dataset used in this research is taken from the Kaggle website, is publicly available, and can be used by everyone. The data collected is used only for this research strictly.

Professional

I have completed studying 31 research papers related to the topic, and now I can conduct this research. I do not intend to harm anyone through this research work. If anyone has any disagreement with this research work, please share.

Social

This research work does not intend to hurt anyone's emotions or beliefs. It does not harm society regarding inequalities, environment, culture, and gender. The research work does not intend to damage or disrupt society.

5. Planned Work

I strategize to organize the research project in the following manner:

1. Reading and understanding the dataset.
2. Knowing about the shape, types, duplicate values, and quantiles of the dataset
3. Checking for classes
4. Analysis of the categorical variables
5. Cleaning the dataset
6. Finding if we have an unbalanced data problem
7. Finding if there is an imbalance in the scoring?
8. Training and evaluating the model
9. Conclusion of the proposed approach
10. Final documentation of the conducted research work

The following Gantt chart elaborates on the planned work of the project:

							weeks							
Tas k ID	Task Description	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Planning													
2	Research on the topic													
3	Project proposal													
4	Data Preprocessing													
5	Interim Projec t Report													
6	Model implementation													
7	Evaluation													
8	Discussion o f the results													
9	Final projec t report													

6. Appendices

Data Sources

<https://data.world/promptcloud>
<https://www.kaggle.com/>

Installation of required libraries

```
[1] !pip install SentimentIntensityAnalyzer
!pip install chart_studio
!pip install TextBlob
!pip install plotly
!pip install WordCloud
!pip install cufflinks
```

Importation of required modules

```
[1] import pandas as pd
    from nltk.sentiment.vader import SentimentIntensityAnalyzer
    import nltk
    import re
    from textblob import TextBlob
    from wordcloud import WordCloud
    import numpy as np
    import seaborn as sns
    import matplotlib.pyplot as plt
    import cufflinks as cf
    %matplotlib inline
    from plotly.offline import init_notebook_mode, iplot
    init_notebook_mode(connected = True)
    cf.go_offline();
    import plotly.graph_objs as go
    from plotly.subplots import make_subplots
```

```
import warnings
warnings.filterwarnings("ignore")
warnings.warn("this will not show")
pd.set_option('display.max_columns', None)
```

Uploading and reading the dataset

```
[1] df_ = pd.read_csv("/content/drive/MyDrive/Content/amazon_reviews.csv")

[1] df = df_.copy()

[1] df = df.sort_values("wilson_lower_bound", ascending=False)
    df.drop('Unnamed: 0', inplace = True, axis = 1)
    df.head()
```

Output

	reviewerName	overall	reviewText	reviewTime	day_diff	helpful_yes	helpful_no	total_vote	score_pos_neg_diff	score_average_rating	wilson_lower_bound
2031	Hyoun Kim "Faluzure"	5.0	[[UPDATE - 6/19/2014]]So my lovely wife boug...	2013-01-05	702	1952	68	2020	1884	0.966337	0.957544
3449	NLee the Engineer	5.0	I have tested dozens of SDHC and micro-SDHC ca...	2012-09-26	803	1428	77	1505	1351	0.948837	0.936519
4212	SkincareCEO	1.0	NOTE: please read the last update (scroll to ...	2013-05-08	579	1568	126	1694	1442	0.925620	0.912139
317	Amazon Customer "Kelly"	1.0	If your card gets hot enough to be painful, it...	2012-02-09	1033	422	73	495	349	0.852525	0.818577
4672	Twister	5.0	Sandisk announcement of the first 128GB micro	2014-07-03	158	45	4	49	41	0.918367	0.808109

Knowing about the shape, types, duplicate values, and quantiles of the dataset

```
[1] def missing_values_analysis(df):
    na_columns_ = [col for col in df.columns if df[col].isnull().sum() > 0]
    n_miss = df[na_columns_].isnull().sum().sort_values(ascending=True)
    ratio_ = (df[na_columns_].isnull().sum() / df.shape[0] * 100).sort_values(ascending=True)
    missing_df = pd.concat([n_miss, np.round(ratio_, 2)], axis=1, keys=['Total Missing Values', 'Ratio'])
    missing_df = pd.DataFrame(missing_df)
    return missing_df

def check_dataframe(df, head=5, tail=5):
```

```
    print(" SHAPE ".center(82,'~'))
    print('Rows: {}'.format(df.shape[0]))
    print('Columns: {}'.format(df.shape[1]))
    print(" TYPES ".center(82,'~'))
    print(df.dtypes)
    print(''.center(82,'~'))
    print(missing_values_analysis(df))
    print(' DUPLICATED VALUES '.center(83,'~'))
    print(df.duplicated().sum())
    print(" QUANTILES ".center(82,'~'))
    print(df.quantile([0, 0.05, 0.50, 0.95, 0.99, 1]).T)

check_dataframe(df)
```

Output

```
~~~~~ SHAPE ~~~~~
Rows: 4915
Columns: 11
~~~~~ TYPES ~~~~~
reviewerName    object
overall         float64
reviewText      object
reviewTime      object
day_diff        int64
helpful_yes     int64
helpful_no      int64
total_vote      int64
score_pos_neg_diff  int64
score_average_rating  float64
wilson_lower_bound  float64
dtype: object
~~~~~ Total Missing Values Ratio ~~~~~
reviewerName    1  0.02
reviewText      1  0.02
~~~~~ DUPLICATED VALUES ~~~~~
0

~~~~~ QUANTILES ~~~~~
      0.00  0.05  0.50   0.95   0.99   1.00
overall    1.0  2.0  5.0  5.000000  5.00000  5.000000
day_diff   1.0 98.0 431.0 748.000000 943.00000 1064.000000
helpful_yes    0.0  0.0  0.0  1.000000  3.00000 1952.000000
helpful_no    0.0  0.0  0.0  0.000000  2.00000 183.000000
total_vote    0.0  0.0  0.0  1.000000  4.00000 2020.000000
score_pos_neg_diff -130.0  0.0  0.0  1.000000  2.00000 1884.000000
score_average_rating  0.0  0.0  0.0  1.000000  1.00000  1.000000
wilson_lower_bound  0.0  0.0  0.0  0.206549  0.34238  0.957544
```

Checking for classes

```
[1] def check_class(dataframe):
    unique_df = pd.DataFrame({'Variable': dataframe.columns,
                              'Classes': [dataframe[i].unique() \
                                           for i in dataframe.columns]})

    unique_df = unique_df.sort_values('Classes', ascending=False)
    unique_df = unique_df.reset_index(drop = True)
    return unique_df

check_class(df)
```

Output

	Variable	Classes
0	reviewText	4912
1	reviewerName	4594
2	reviewTime	690
3	day_diff	690
4	wilson_lower_bound	40
5	score_average_rating	28
6	score_pos_neg_diff	27
7	total_vote	26
8	helpful_yes	23
9	helpful_no	17
10	overall	5