YASH RANDIVE




# Restaurant Rating Prediction Using Sentiment Analysis and Machine Learning

## Part I

# PROBLEM STATEMENT

YASH
RANDIVE

- To develop Machine Learning system(s) to predict the star rating of a review based on its meta features (e.g. reviewer ID, business ID etc.) and/or review text.
- To implement and compare different machine learning models and explore the effective features for this sentiment prediction task.

YASH
RANDIVE

# DATA ACKNOWLEDGEMENTS

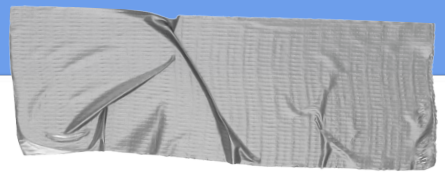
The data has kindly been provided to us, under the provision that any resulting work should cite these two resources:

- What Yelp fake review filter might be doing? A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance, ICWSM, 2013.
- Collective Opinion Spam Detection: Bridging Review Networks and Metadata. Shebuti Rayana, Leman Akoglu, ACM SIGKDD, Sydney, Australia, August 10-13, 2015

YASH
RANDIVE

# DATA DESCRIPTION

1. Meta features: date, unique review ID, reviewer ID, business ID, votes cast by Yelp users (whether they think the review is funny, cool or useful)
2. The original review text, provided as a single file with rows corresponding to the meta file
3. Text features: produced by various text encoding methods for review text
4. Class label: rating (3 possible levels, 1, 3 or 5)(Target)



YASH
RANDIVE

# <CODE/>

# Importing Relevant Libraries

YASH
RANDIVE

The libraries imported here are basic libraries that we would need to get started of with the Analysis and Modelling. All other relevant libraries will be imported as and when their functionality(ies) is/are required

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import preprocessing, feature_extraction, model_select
ion, linear_model
import seaborn as sns
sns.set()
```

# Loading the Data

YASH
RANDIVE

**The files loaded are:**

- #### review_meta_train.csv (rating label in this file)
- #### review_text_train.csv: original review text.
- #### review_meta_test.csv
- #### review_text_test.csv: original review text.

]:

```
review_text_train = pd.read_csv('../input/review_text_train.csv', i
ndex_col = False, delimiter = ',', header=0)
review_text_test = pd.read_csv('../input/review_text_test.csv', ind
ex_col = False, delimiter = ',', header=0)
review_meta_train = pd.read_csv('../input/review_meta_train.csv', i
ndex_col = False, delimiter = ',', header=0)
review_meta_test = pd.read_csv('../input/review_meta_test.csv', ind
ex_col = False, delimiter = ',', header=0)
```

# Examining the Shapes of the Data i.e Observing the number of Rows and Columns we will be dealing with

YASH
RANDIVE

*Here we examine the number of rows and columns that we'll be dealing with*

```
print('review_text_train :\t', str(review_text_train.shape))
print('review_text_test :\t', str(review_text_test.shape))
print('review_meta_train :\t', str(review_meta_train.shape))
print('review_meta_test :\t', str(review_meta_test.shape))
```

```
review_text_train :     (28068, 1)
review_text_test :      (7018, 1)
review_meta_train :     (28068, 8)
review_meta_test :      (7018, 7)
```

# Cleaning the Data

YASH
RANDIVE

## Checking for NaN Values

```
print('review_text_train :\n', str(review_text_train.isna().sum()),
"\n**********\n")
print('review_text_test :\n', str(review_text_test.isna().sum()),
"\n**********\n")
print('review_meta_train :\n', str(review_meta_train.isna().sum()),
"\n**********\n")
print('review_meta_test :\n', str(review_meta_test.isna().sum()),
"\n**********\n")
```

If there are NaN values in the data we can drop them using the,
df = df.drop(['<column1_name>','column2_name'], axis = 1)
Else,
The data has no missing values and we can continue with processing it further

# Creating the Train and Test DataFrames

YASH
RANDIVE

## df_train

Train DF

```
# Train DataFrame
df_train = review_text_train.copy()
```

```
df_train['vote_funny'] = review_meta_train.vote_funny
df_train['vote_cool'] = review_meta_train.vote_cool
df_train['vote_useful'] = review_meta_train.vote_useful
df_train['rating'] = review_meta_train.rating
df_train
```

# Creating the Train and Test DataFrames

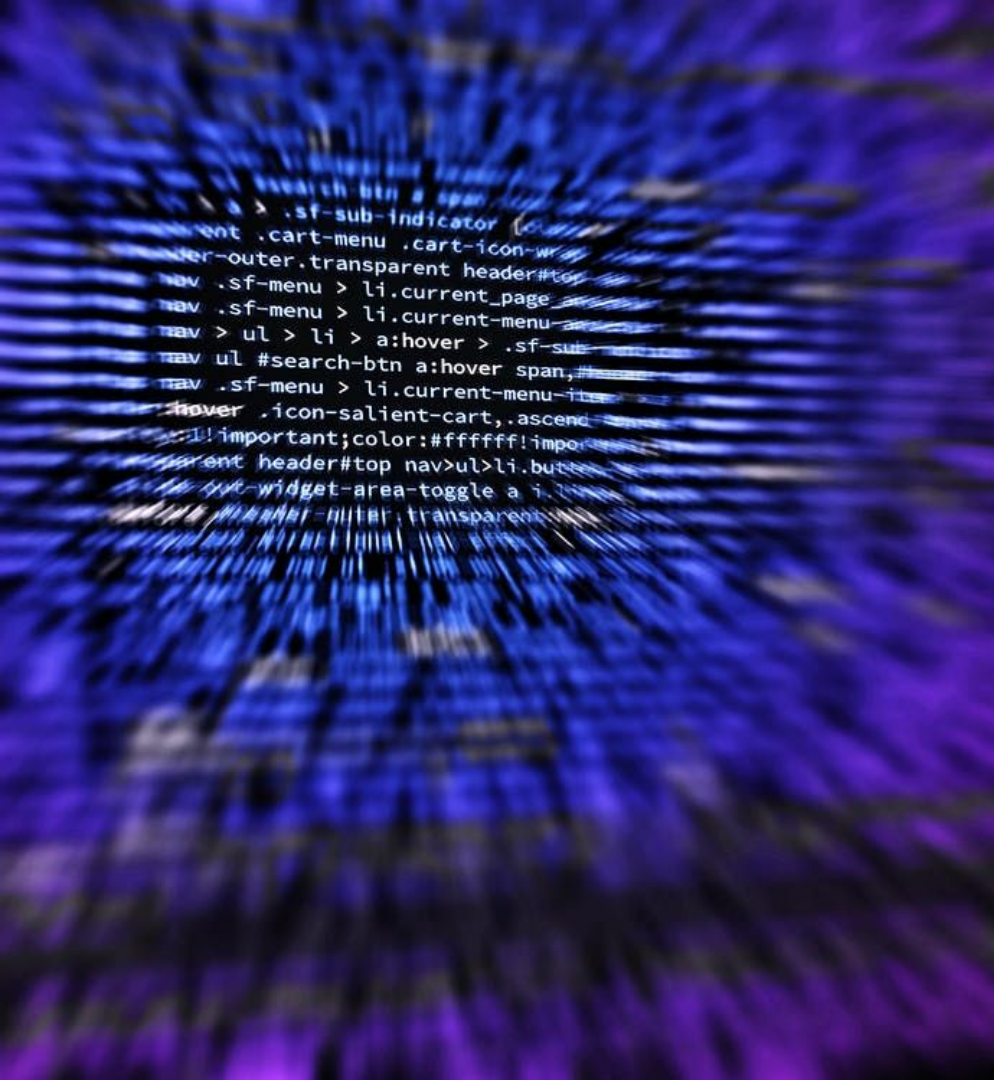
YASH
RANDIVE

## df_test

Test DF

```
df_test = review_text_test.copy()
df_test['vote_funny'] = review_meta_test.vote_funny
df_test['vote_cool'] = review_meta_test.vote_cool
df_test['vote_useful'] = review_meta_test.vote_useful
df_test
```

**Cleaning the data and checking the NaN values** is important for the analysis in order to ensure that accurate and significant information is used to make visualizations and predictions




YASH
RANDIVE

# Next Up

1. Text Preprocessing
2. Sentiment Analysis
3. Exploratory Data Analysis

YASH
RANDIVE

# Congratulations!

**You have reached the end of this part of the project and are all set to move ahead!**

**For queries or questions in the code reach me at :**

yashrandive.datascience@gmail.com

## All the Best!