Steps taken (code)

Libraries imported:

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

import warnings

warnings.filterwarnings('ignore')

from nltk.corpus import stopwords

from sklearn.feature\_extraction.text import TfidfTransformer, CountVectorizer

#from sklearn.metrics import classification\_repo

#rt, confusion\_matrix

from sklearn.model\_selection import train\_test\_split

import string, nltk

from nltk import word\_tokenize

from nltk.stem import PorterStemmer

from nltk.stem import WordNetLemmatizer

https://regenerativetoday.com/exploratory-data-analysis-of-text-data-including-visualization-and-sentiment-analysis/

**Exploratory data analysis**

The chosen dataset was sourced form outscraper.com . It consists of x data columns and x data rows or google maps reviews using the queries with ‘Dublin’ as a location and ‘restaurant’ as a filter. The csv data was read into a panda’s dataframe.

Exploratory data analysis showed a range of review ratings from x to x providing a good spread of positive and negative reviews. Rows with missing data in review\_text column were deleted as these are not useful to the analysis. Other columns such as review\_img\_url and owner\_answer with missing data were left, these will not be considered in the dataset. The dataset consists of quantitative numerical data such as review rating, review id and number of review\_likes and descriptive quantitive data in the review\_text column. Wordclouds provided visualisations of the frequency of certain word occurances in the dataset.

Visualisation of review length, the distribution of the word count, and the sentiment polarity were generated to judege Iid the dataset is skewed in an way.

Exploratory data analysis consisted analysing the column :

-looking at the number of rows and columns using ‘.shape’

-checking for missing data

-obtaining summary statistics such as count, mean , standart deviation, minimum and maximum values and quantiles of the data using ‘.describe’ function

-checking object types

-counting values in the ‘ratings category’ to ensure a range of views is visible in the dataset

-visualisations to provide an overview of the data

**Data cleaning consisted of:**

-clean text function to remove punctuation from the input text, convert it to lowercase, and return a cleaned version of the text. Applied to dataset

- set of stopwords specific to the English language was created using the stopwords.words('english') method from the nltk.corpus module

-wordcloud was generated based on provided set of stopwords list filtered out

-a preprocessor function performs HTML tag removal, extracts emoticons, converts the text to lowercase, removes non-word characters, and combines the preprocessed text with extracted emoticons

-the column [‘review\_text’] was converted to string object

- the column [‘review\_text’] was lemmatized using WordNetLemmatizer to reduce words to base form

the column [‘review\_text’] wasstemmed using PorterStemmer to reduce words to root form

-wordclouds generated of positive and negative reviews based on rating value

-counter class imported from ‘collections’ module used to process and store word counts for analyzing word frequency and generate visualization of common words

**Data processing consisted of:**

-word\_tokenize from nltk.tokenize was used for word tokenization, which splits text into individual words

-FreqDist from nltk.probability was used to calculate the frequency distribution of words, identify common rare words

- pos\_tag function to perform part-of-speech tagging on a list containing only the last token. The result is a list with a single tuple containing the last token and its part-of-speech tag, the pos\_tags list generated, which contains tuples of words and their corresponding part-of-speech tags

-chunking and vader sentiment to add a vader compound and sentiment (positive/negative/neutral) to the dataset as columns