About The Data:

Dataset: contains reviews and ratings user gives on Flipkart.

About Flipcart: Online Shopping; Flipkart is primarily known as an online marketplace where customers can browse, select, and purchase a diverse array of products. Flipkart Private Limited is an Indian e-commerce company. The company initially focused on online book sales before expanding into other product categories such as consumer electronics, fashion, home essentials, groceries, and lifestyle products. As of March 2017, Flipkart held a 39.5% market share in the Indian e-commerce industry.

About the Variables: Reviews are strings and Ratings are categorized by 1 to 5 to corresponding reviews..

Goal: to predict whether the review given is positive or negative (0 for negative, 1 for positive)

Method: machine learning (**Train-Test** Split: 33% of the data for testing; 67% of the data for training; **Decision Tree** for the prediction: accuracy rate is 67.72%)

Python Code:

Libraries:

Pandas : For importing the dataset.

Scikit-learn : For importing the model, accuracy module, and TfidfVectorizer.

Warning : To ignore all the warnings

Matplotlib : To plot the visualization. Also used Wordcloud for that.

Seaborn : For data visualization. (Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.)

NLTK: for text analysis

Tqdm: It provides an intuitive and visually appealing progress bar that shows the percentage of completion, estimated time remaining, and other relevant information.
Re: let you check if a particular string matches a given regular expression

Step 1:
Import the libraries and dataset

Step 2:
Preprocessing the data

Funtions:

- pd.unique(): return the unique values based on the the table without order.
- re.sub(): searches for all the instances of pattern in the given string, and replaces them
- ' '.join(...): used to join the individual words of a sentence back together into a single string with spaces in between.
- token.lower(): converts each word (token) to lowercase
- word_tokenize(): is a function in Python that splits a given sentence into words

Visualization the data

- Plot(variable ratings):
    - Barplot: x = count of ratings and y = levels of ratings

Due with variable Rating:

- Change the rating variable into a new variable with only 0 represent negative(rating 4 or less) and 1 represent postive(rating 5).

Create a function to preprogress the variable Reviews:

- Remove puncutations(period, comma, apostrophe, quotation, question, exclamation, brackets, braces, parenthesis, dash, hyphen, ellipsis, colon, semicolon) from the text strings.
- Conver all into lowercase letters.
- Removing stopwords in English. (stop words in English are “a,” “the,” “is,” “are,” etc)

Step 3: Analysis of the dataset

- Check counts for the positive and negative ratings
- Visualization the important words which shows up frequently in the reviews

Step 4: Converting text into Vectors

TF-IDF calculates that how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (data-set). We will be implementing this with the code below.

TF-IDF: Term Frequency-Inverse Document Frequency

- a numerical statistic that reflects the importance of a word within a document or a collection of documents
- used in natural language processing and text analysis to measure the relevance of words to specific texts while considering their importance in a larger context

Vectorization the text:

TfidfVectorizer(): part of scikit-learn and is used to transform a collection of text documents into a TF-IDF matrix.

TF-IDF matrix = feature matrix: Term Frequency-Inverse Document Frequency Matrix; a numerical representation of a collection of text documents that is used to measure the importance of words or terms within each document relative to a larger corpus; helps capture the significance of words in individual documents while considering their frequency in the entire document collection.

Step 5: Model Training, Evaluation, and Prediction

- **Train-Test** Split:
    - train_test_split function: used for splitting your dataset into training and testing subsets
    - 33% of the data for testing
    - 67% of the data for training

- Train/Evaluate/Predict the model:
    - **Decision Tree** for the prediction: a supervised machine learning algorithm that is used for both classification and regression tasks; Decision trees are widely used in various fields, including finance, healthcare, marketing, and natural language processing. They are particularly useful when you need a transparent and interpretable model to make decisions based on complex data.
        - Set a decision tree classifier
        - Train the model on training data
        - Testing the model on the training data
        - Calculate the accuracy of the model on the training data
    - **Confusion matrix**:
        - A 2x2 matrix that represents true positives, true negatives, false positives, and false negatives.

Conclusion:

Decision Tree Classifier is performing well with this data. In future, we can also work with large data by scraping it through the website.