Assignment Report

Module 1

- **Objective:**

    1. Explain the text processing pipeline.
    2. Generate term statistics:
        a. Vocabulary size with word frequency.
        b. N-grams.
        c. POS Collection.
    3. Verify Zipf's law and find the best fitting curve.
    4. Find the set of terms which best describes the corpus.

- **Dataset Description:**

    The Amazon Instant Video dataset contains product reviews from Amazon. It has 37126 reviews.The attributes defined are as follows:

    - reviewerID - ID of the reviewer
    - asin - ID of the product
    - reviewerName - name of the reviewer
    - helpful - helpfulness rating of the review
    - reviewText - text of the review
    - overall - rating of the product
    - summary - summary of the review
    - unixReviewTime - time of the review (unix time)
    - reviewTime - time of the review (raw)

- **Text Processing Pipeline:**

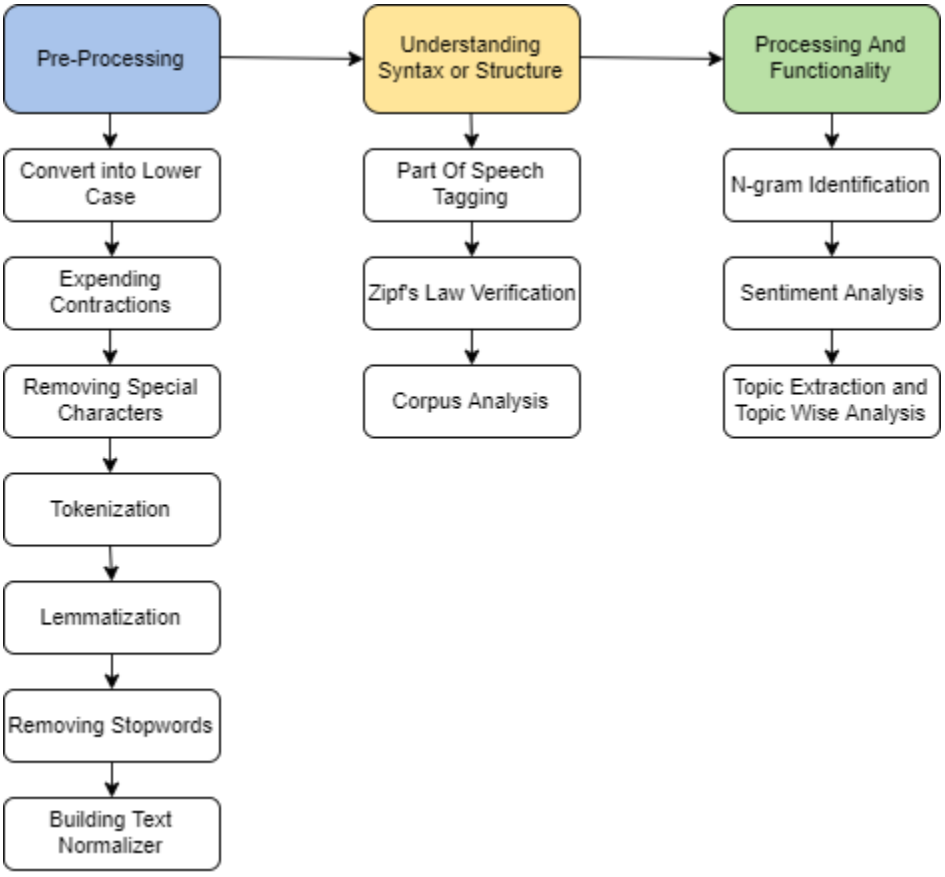

Fig1. Text Processing Pipeline

- **Procedure:**

1. **<u>Data Loading</u> :** This step involves the following sub-steps:
    - Read the dataset from the json file and generate the dataframe.
    - Extracting the reviewText from the dataframe and converting it to a numpy array.
    - Extracting the overall attribute as rating of the product from the dataframe and converting it to a numpy array. Rating is used as labelled feature where:
        i. [overall > 3.0] - positive
        ii. [overall <= 3.0] - negative
        iii. If the rating is greater than 3 label it as 1.
        iv. If the rating is less than or equal to3 label it as 0.

2. **<u>Text Preprocessing</u>:**
    - Expanding Contractions : Process of expanding the little literary shortcuts we use while writing, for instance using should've instead should have .
    - Tokenization and converting into lowercase.
    - Removing all non word characters, whitesaces etc.
    - Lemmatization:It is the process of grouping together the different inflected forms of word so they can be analysed as a single term.
3. Building the corpus by following the above process.
4. **N-gram Identification**: Calculated the n-grams from the corpus with n=2 i.e Bigrams and n = 3 trigrams.
5. Frequency calculation for each term in the corpus and finding the corresponding rank.
6. Plotting the log(frequency) vs log(rank) graph to analyse which terms are influential in the corpus.
    - ➢ The best fit line is drawn with Linear Regression.
    - ➢ The terms which are least deviated from the best fit line are the most useful terms in the corpus.
    - ➢ We extract those useful words and remove the others from the corpus.
7. **Parts of Speech:**
    - ➢ Tag the terms in the updated corpus with the parts of speech, which signify whether the word is noun, verb, pronoun, adjective etc.
    - ➢ For this we have used **averaged_perceptron_tagger.**

● Observations:

- The vocabulary obtained from the documents in the dataset with the frequency of word is shown below as a tuple (word, frequency)[only few selected examples]:
    - Few most frequent words are :
      ('the', 187293),('a': 112623), ('and': 102688), ('to': 83845), ('is': 82293).
    - Few Moderately frequent words are:
      ('pathetic': 69),('confusion': 55), ('garcia': 55), ('contest': 55), ('stupidity': 55), ('accuracy': 55).
    - Few Low frequent words are:

('horrorhowever': 1), ('grotesqueness': 1), ('storethe': 1), ('headstomping': 1), ('notright': 1), ('actionfear': 1).

- The N-grams(for n= 2 i.e Bi-gram) obtained from the documents in the dataset with the frequency is shown below as a tuple ((Bi-gram), frequency)[only few selected examples]:
  - Few most frequent Bi-grams are :
    (('of', 'the'), 20946), (('it', 'is'), 16592), (('in', 'the'), 11476), (('is', 'a'), 11442).
  - Few Moderately frequent Bi-grams are:
    (('i', 'like'), 1790), (('out', 'of'), 1781), (('story', 'line'), 1772), (('i', 'did'), 1749).

- The N-grams(for n= 3 i.e Trigram) obtained from the documents in the dataset with the frequency is shown below as a tuple ((Trigram), frequency)[only few selected examples]:
  - Few most frequent Trigrams are :
    (('it', 'is', 'a'), 2340), (('one', 'of', 'the'), 2237), (('this', 'is', 'a'), 2203).
  - Few Moderately frequent Trigrams are:
    (('a', 'good', 'a'), 424), (('going', 'to', 'be'), 424), (('a', 'bit', 'of'), 417).

- For the verification of Zipf's Law, the obtained Log(Frequency) vs Log(Rank) is shown below:

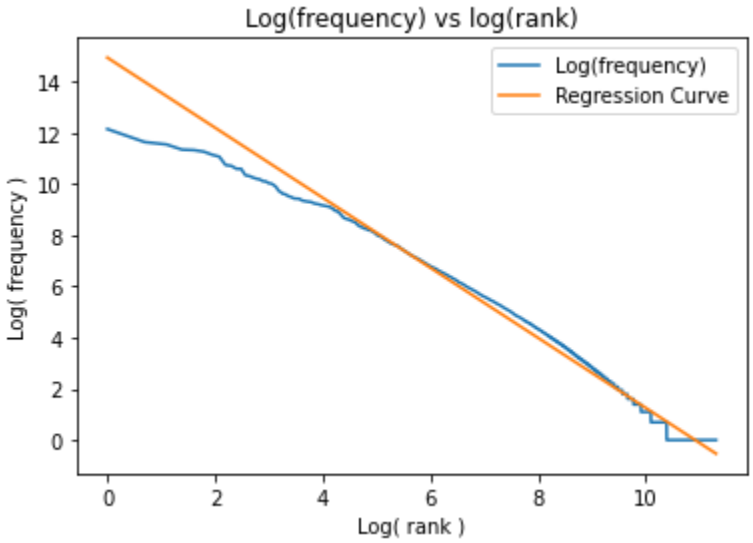

Fig2. Zipf's Law

- The corpus obtained after removing the non-useful words is as follows(few examples):
  ('entertaining': 1916), ('excellent': 1892),('recommend': 1861), ('unbelievable': 186), ('thriller': 416), ('crime': 1288)

- A sentence example from the dataset and its corresponding sentence after pre-processing is shown below:

  Original Sentence: 'I highly recommend this series. It is a must for anyone who is yearning to watch "grown up" television. Complex characters and plots to keep one totally involved. Thank you Amazin Prime.'
  Modified Sentence: highly recommend must anyone yearning grown television complex totally involved thank prime.

- The POS Collections observed for the above given original sentence is as follows:

```
[('i', 'NN'), ('highly', 'RB'), ('recommend', 'VB'), ('this',
'DT'), ('series', 'NN'), ('.', '.'), ('it', 'PRP'), ('is',
'VBZ'), ('a', 'DT'), ('must', 'MD'), ('for', 'IN'), ('anyone',
'NN'), ('who', 'WP'), ('is', 'VBZ'), ('yearning', 'VBG'),
('to', 'TO'), ('watch', 'VB'), ('``', '``'), ('grown', 'VBN'),
('up', 'RP'), ("''", "''"), ('television', 'NN'), ('.', '.'),
('complex', 'JJ'), ('characters', 'NNS'), ('and', 'CC'),
('plots', 'NNS'), ('to', 'TO'), ('keep', 'VB'), ('one', 'CD'),
('totally', 'RB'), ('involved', 'JJ'), ('.', '.'), ('thank',
'VB'), ('you', 'PRP'), ('amazin', 'JJ'), ('prime', 'JJ'),
('.', '.')]
```

● Conclusions:

- From above graph shown in fig2. ,we can conclude that Zipf's law is verified.
- From the best fit linear regression curve and our corpus curve we can say that words with very low rank and very high rank are not significant, they are much deviated from the best fit line.

- The words with rank 5 to 9.6 on logarithmic scale(Approx. 150-15000 rank on linear scale) are the most useful and when taken in the corpus gives the best accuracy.
- Initially in the corpus there were a lot of words, some with very high frequency and some with very low frequency, after removing the words which are not useful according to the Zipf's Law the modified sentence is more precise than the original sentence as shown above.
- The corpus obtained after removing the non-useful words in shown in the observation and this modified corpus contains more useful words like "loving", "Entertaining" etc which describes the sentiment precisely, also the modified corpus contains the words related to the genre of the videos reviewed for e.g "Crime", "thriller" etc.

Module 2

- **Objective:**

Perform Sentiment analysis task while considering the following combinations:

1. vector space models:
    a. CountVectorizer.
    b. TF-IDF.
    c. Any external vectorizer
2. Classical ML techniques
    a. Naive Bayes Model.
    b. Decision Tree.
    c. Logistic Regression.

- **Procedure:**

1. Building the Text Normalizer using the vector space models.The used models are as follows:

- CountVectorizer: It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text.
- Tf-Idf
- HashVectorizer:The limitations of the above methods is that the vocabulary can become very large.The hashingVectorizer implements the approach to use a one way hash of words to convert them to integers, and then tokenize and encode the document as needed.

2. Doing sentiment analysis on the vectors obtained in previous step using ML techniques such as:
    - Naive Bayes.
    - Decision Tree.
    - Logistic Regression.

- **Observations:**

    - The Accuracy and F1 score results obtained for the exercise are:

Vector Space Model	Classifier	Accuracy	F1 Score
Countvectorizer	Decision Tree	0.758	0.848
Countvectorizer	Logistic Regression	0.844	0.905
Countvectorizer	Naive bayes	0.346	0.365
Tf-Idf	Decision Tree	0.763	0.851
Tf-Idf	Logistic Regression	0.846	0.909
Tf-Idf	Naive bayes	0.416	0.487
Hash vectorizer	Decision Tree	0.678	0.795
Hash vectorizer	Logistic Regression	0.789	0.882
Hash vectorizer	Naive bayes	0.789	0.882

- The confusion matrix obtained for the above combinations is as follows:

Vector Space Model	Classifier	Confusion Matrix
Countvectorizer	Decision Tree	[[1200 1924] [1668 10059]]
Countvectorizer	Logistic Regression	[[1532 1592] [717 11010]]
Countvectorizer	Naive bayes	[[2344 780] [8930 2797]]
Tf-Idf	Decision Tree	[[1236 1888] [1626 10101]]
Tf-Idf	Logistic Regression	[[1138 1986] [300 11427]]
Tf-Idf	Naive bayes	[[2062 1062] [7611 4116]]
Hash vectorizer	Decision Tree	[[784 2340] [2440 9287]]
Hash vectorizer	Logistic Regression	[[0 3124] [0 11727]]
Hash vectorizer	Naive bayes	[[10 3114] [17 11710]]

● Conclusions:

- From the above observation, the best performance is obtained on the combination: Tf-idf with logistic regression.
- The vector space model which gives better results on a particular classifier is as follows:

Classifier	Vector Space model
Decision Tree	Tf-Idf
Naive bayes	Hash vectorizer
Logistic Regression	Tf-Idf

Module 3

- **Objective:**

  - ➢ Perform topic extraction technique on the reviews.
  - ➢ Report sentences under each topic.
  - ➢ Analysis of the correctness of the topics extracted.
  - ➢ Topic wise sentiment distribution for the whole repository.

- **Procedure:**

  1. Convert the text to vectors using countVectorizer.
  2. We have used a search based approach to find the correct estimation of the number of topics in the whole corpus.
  3. From the perplexity evaluation measure we can say that three number of topics is the most appropriate as it has minimum perplexity.
  4. Performing the topic extraction with the optimal number of topics.
  5. Extracting the most common words from each topic.
  6. Extracting the sentences corresponding to each topic.
  7. Performing the topic-wise sentiment analysis by observing the number of good and bad reviews in each topic.

- **Observations:**
    - The perplexity variations with the number of topics observed is as follows:

Number of topics(n)	Perplexity
1	4057.99
2	4070.20
3	4042.68
4	4054.70
5	4087.19

    - The common words under each topic(n =3) is as follows:
        - **Topic 0 :** 'superb', 'fantastic', 'justified', 'interested', 'exciting', 'wonderful', 'loved', 'funny', 'entertaining' etc.
        - **Topic 1 :** 'week', 'hate', 'long', 'wrong' etc.
        - **Topic 2 :** 'audience', 'fight', 'someone', 'based', 'person', 'excellent' , 'strong' etc.

    - The sentences reported under each topic are:
        - **Topic 0 :**
            - 'I highly recommend this series. It is a must for anyone who is yearning to watch "grown up" television. Complex characters and plots to keep one totally involved. Thank you Amazin Prime.'
            - 'if this had to do with Dat Phan, he was hilarious, I enjoyed his comedy and would watch him alone numerous times'
        - **Topic 1 :**
            - "I had big expectations because I love English TV, in particular Investigative and detective stuff but this guy is really boring. It didn't appeal to me at all."
            - "This one is a real snoozer. Don't believe anything you read or hear, it's awful. I had no idea what the title means. Neither will you."
        - **Topic 2 :**

- 'Mysteries are interesting. The tension between Robson and the tall blond is good but not always believable. She often seemed uncomfortable.'
- 'Enjoyed some of the comedians, it was a joy to laugh after losing my father whom I was a caregiver for'

○ The topic-wise sentiment analysis observations are given below:

Topic	Good Reviews Count	Bad Reviews Count	Bad to good review percentage
Topic 0	13369	1037	7.75%
Topic 1	9704	4979	51.30%
Topic 2	6263	1774	28.32%

- **Conclusion**:

1. The optimal number of topics found in the corpus are three (n= 3), as it has a minimum value of perplexity.
2. From the common words under each topic, we can say that:
    - Topic0 :contains words which are mostly positive i.e. reflecting good reviews
    - Topic1: contains words which are mostly negative i.e. reflecting bad reviews.
    - Topic3: contains both kinds of words i.e positive and negative..
3. The sentences under each topic are as:
    - Topic0 : more positive reviews
    - Topic1: more negative reviews.
    - Topic3: Mixed reviews.
4. Topic-wise sentiment analysis :
    - As expected the number of positive reviews belonging to topic0 is higher as compared to the very less value of bad reviews.
    - For the topic1, the number for bad reviews was high as expected.
    - Topic 2 contains both positive and negative reviews. It may be possible that topic 2 contains mostly reviews corresponding to rating 3 to 4 or average rating.

- **References:**
    - https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/
    - https://medium.com/@suneelpatel.in/nlp-pipeline-building-an-nlp-pipeline-step-by-step-7f0576e11d08
    - https://medium.com/@yanlinc/how-to-build-a-lda-topic-model-using-from-text-601cdcbfd3a6
    - https://stackabuse.com/python-for-nlp-topic-modeling/

- **Google Colab Link:**
    - https://colab.research.google.com/drive/16oEt9E_CPXGvdx1GPRVn97YA7uDEi-L6#scrollTo=16G2rYnXcK6B

- **Group Member Contribution:**
    - Anupriya Pal(M20CS053): Module 2 complete , Module 3 - topic wise sentiment analysis
    - Nishit Bhardwaj (M20CS067): Module 1 complete, Module 3 - optimal number of topic, topic extraction and analysis.