

Amaron Review Project

Said Lfagrouche

Rahat Fahim

Kazi Farhan Hoque





Goal:

Predict Rating, from 1 to 5, based on what review that you give.





Project Overview

Dataset(from Kaggle): All Amazon product Reviews (over (54.41 GB), 10% sampled for this project)

Methods: Utilizing advanced Neural Networks and Natural Language Processing (NLP) techniques, with a focus on integrating the Hugging Face NLP model.



Customer reviews

★★★★☆ 4.3 out of 5

314 customer ratings



Advantages of NLP:

- Understand Human language.
- Text and Speech processing.
- Sentimental Analysis.
- Information Extraction.
- Question Answering System.



Tools Utilized

Frontend Technologies:

- HTML, CSS, JavaScript:
 - Crafted a user-friendly interface with HTML for structure, CSS for styling, and JavaScript for dynamic elements and interactivity.



Flask

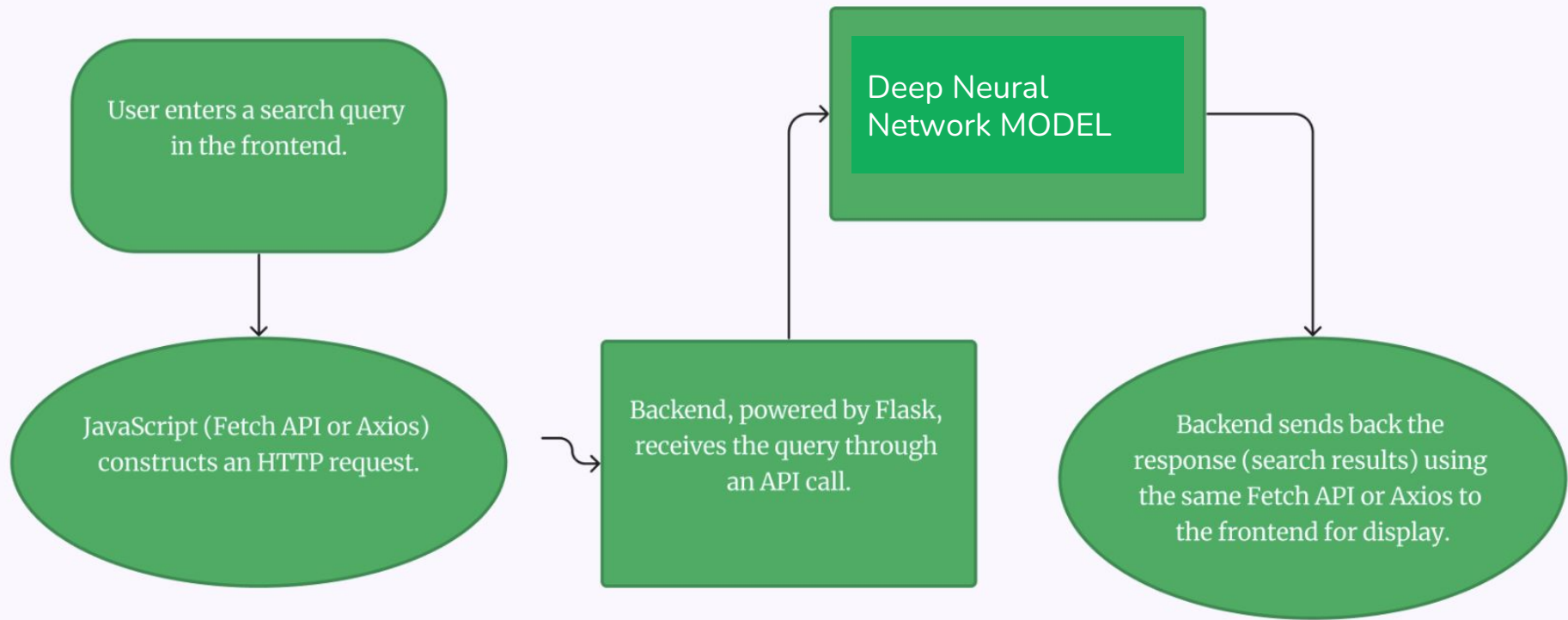


Libraries and Frameworks:

- Flask(bridge between the model and front end)
- **Data Cleaning and Exploration:** Pandas, NumPy
- **Data Visualization:** Matplotlib and Seaborn
- **Scikit-Learn:** - Utilized Scikit-Learn for machine learning algorithms, model evaluation, and data preprocessing.
- **Hugging Face Transformers:** - Leveraged pre-trained models for advanced tokenization and transfer learning in NLP.




Hugging Face





Approach and Data Handling

- **Sampling:**
 - Reduced dataset to 10% for manageable training times.
- **Tokenization:**
 - Converting text into distinct tokens.
 - Removing stopwords to enhance data quality.
- **Normalization:**
 - Cleaning and transforming text data for consistency.
- **Vectorization:**
 - Representing text as numeric matrices for training and testing.
- **Hugging Face's NLP Model:**
 - Utilizing advanced tokenization capabilities for enhanced text processing.



Document	the	cat	sat	in	hat	with
<i>the cat sat</i>	1	1	1	0	0	0
<i>the cat sat in the hat</i>	2	1	1	1	1	0
<i>the cat with the hat</i>	2	1	0	0	1	1



- **Tokenization:**

```
from nltk import word_tokenize
```

```
tokens = [word_tokenize(sen) for sen in sample.reviewText]  
# sample['raw_tokens'] = tokens
```

Then I removed all stop words from the tokenized word list, like "this" and "it's"

```
from nltk.corpus import stopwords
```

```
stoplist = stopwords.words('english')
```

```
def removeStopWords(tokens):  
    return [word for word in tokens if word not in stoplist]
```

```
filtered_words = [removeStopWords(sen) for sen in tokens]
```

```
sample['raw_tokens'] = filtered_words #join to dataframe
```

Normalization:

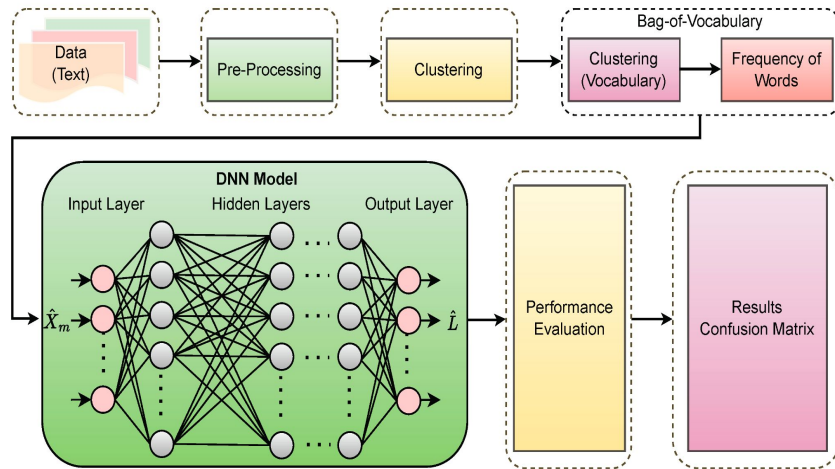
```
def remove_non_ascii(words):  
    """Remove non-ASCII characters from list of tokenized words"""  
    new_words = []  
    for word in words:  
        new_word = unicodedata.normalize('NFKD', word).encode('ascii', 'ignore').decode('utf-8', 'ignore')  
        new_words.append(new_word)  
    return new_words  
  
def to_lowercase(words):  
    """Convert all characters to lowercase from list of tokenized words"""  
    new_words = []  
    for word in words:  
        new_word = word.lower()  
        new_words.append(new_word)  
    return new_words  
  
#remove all punctuation  
def remove_punctuation(words):  
    """Remove punctuation from list of tokenized words"""  
    new_words = []  
    for word in words:  
        new_word = re.sub(r'^\w\s', '', word)  
        if new_word != '':  
            new_words.append(new_word)  
    return new_words  
  
def replace_numbers(words):  
    """Replace all interger occurrences in list of tokenized words with textual representation"""  
    p = inflect.engine()  
    new_words = []  
    for word in words:  
        if word.isdigit():  
            new_word = p.number_to_words(word)  
            new_words.append(new_word)  
        else:  
            new_words.append(word)  
    return new_words  
  
def normalize(words):  
    words = remove_non_ascii(words)  
    words = to_lowercase(words)  
    words = remove_punctuation(words)  
    words = replace_numbers(words)  
    return words
```




Exploring Model Options

Models Tested:

- **DNN with Bag of Words (BOW):**
 - Utilized a neural network architecture with BOW vectorization. 90% accuracy
- **DNN with Manual Word Embedding:**
 - Created word embeddings manually within the model. - 69% accuracy



Lessons Learned and Challenges Overcome

- A project idea that is relevant to businesses.
- Lack of sufficient data. (Kaggle saved us)
- Attaining a high accuracy.
- Project management throughout the project.
- The right machine learning algorithm to use.



Looking Ahead

Consider using the entire dataset:

- Explore the feasibility of utilizing the complete dataset (1.6 million records) for more comprehensive model training and evaluation.

Cloud Computing:

- We could use a lot more data at a virtual machine compared to a local computer.





TRY IT NOW

- product [here](#)
- product [here](#)



EXAMPLES



LEAKS

Reviewed in the United States on December 2, 2023

Style: 64oz | Color: Lapis | **Verified Purchase**

I read the reviews, and ignored them. "Couldn't be an issue with me" I thought...Well it was. I cleaned it, filled it with water, tilted it...and boom! Leaks immediately. Tightened as tight as it would go and it still leaked. Super disappointed. You'd think after so many reviews, this seller would halt selling them temporarily until the issue is fixed. Guess not?



Very happy with this Stanley

Reviewed in the United States on December 8, 2023

Style: 30oz | Color: Cream | **Verified Purchase**

I love this Stanley . It's now my favorite Stanley cup . I even prefer it over the 40 oz tumblr . The color is very beautiful and pleasing . The flip straw is very convenient and the size is absolutely perfect . The cup is built very well and LEAK free which is great .

Thank You

Any question? 

GitHub Link [here](#)