Capstone Three – Final Report

Sentiment Analysis for Amazon Product Reviews

Problem Statement

The goal of this project is to develop a sentiment analysis model using advanced Natural Language Processing (NLP) techniques and benchmarking variety of machine learning models, specifically leveraging Large Language Models (LLM), to classify Amazon product reviews as positive, negative or neutral based on the review text. This model aims to provide valuable insights for product manufacturers, sellers, and customers to understand public sentiment towards various products, thereby enabling informed decision-making. So, any seller or amazon itself can crawl through the customer messages in social media platforms (Instagram, twitter) or public forums (i.e. reddit) about certain products. This helps to extract the sentiment to measure the general perception of the product and services from all available platforms not limiting sellers to comments provided in amazon website only.

Context

With millions of products available on Amazon, understanding customer sentiment is crucial for sellers to enhance their offerings, manufacturers to improve product quality, and customers to make informed purchasing decisions. Automated sentiment analysis accelerates the process of deriving meaningful insights from large volumes of reviews, allowing for scalable measurement of customer satisfaction and identification of areas for improvement.

Dataset

The dataset used in this project is sourced from the “Amazon Review’23” dataset collected by University of California San Diego (UCSD) McAuley Lab from Amazon website.

The original dataset contains 571M Product Reviews from 34 Product Categories (33 known product categories and 1 unknown category) including review title, review text and rating information in English language only. For the training of modules approximately 1M Product Reviews are sampled from the original dataset by using PySpark Framework to decrease overall training cost and time for the project.

Data Source Link: Amazon Reviews’23
Data Source Name: Amazon Reviews’23
Collected by: University of California San Diego (UCSD) McAuley Lab

Collected from: Amazon website (1996 - 2023)
Downloaded files: Just review files (no meta data files)
Language: English
Data Source Origin: Amazon website

Dataset Highlights:

- **Larger Dataset:** 571.54M reviews, 245.2% larger than the last version
- **Newer Interactions:** Current interactions range from May 1996 to Sep 2023
- **Richer Metadata:** More descriptive features in item metadata
- **Fine-grained Timestamp:** Interaction timestamp at the second or finer level
- **Cleaner Processing:** Cleaner item metadata than previous versions
- **Standard Splitting:** Standard data splits to encourage RecSys benchmarking

Dataset Comparing to Previous Versions

Year	#Review	#User	#Item	#R_Token	#M_Token	#Domain	Timespan
2013	34.69M	6.64M	2.44M	5.91B	–	28	Jun'96 - Mar'13
2014	82.83M	21.13M	9.86M	9.16B	4.14B	24	May'96 - Jul'14
2018	233.10M	43.53M	15.17M	15.73B	7.99B	29	May'96 - Oct'18
2023	**571.54M**	**54.51M**	**48.19M**	**30.14B**	**30.78B**	**33**	**May'96 - Sep'23**

Sample Review Data


```
{
        'rating': 5.0,
        'title': 'Such a lovely scent but not overpowering.',
        'text': "This spray is really nice. It smells really good, goes on really fine, and does the
        trick. I will say it feels like you need a lot of it though to get the texture I want. I have a
        lot of hair, medium thickness. I am comparing to other brands with yucky chemicals so
        I'm gonna stick with this. Try it!",
        'images': [],
        'asin': 'B00YQ6X8EO',
        'parent_asin': 'B00YQ6X8EO',
        'user_id': 'AGKHLEW2SOWHNMFQIJGBECAF7INQ',
        'timestamp': 1588687728923,
        'helpful_vote': 0,
        'verified_purchase': True
}
```

- rating: Float value from 1.0 to 5.0 representing the product rating
- title: String containing the title of the user review
- text: String containing the full text body of the user review
- images: List of images posted by users after receiving the product, with different size options (small, medium, large)
- asin: String ID of the specific product
- parent_asin: String parent ID of the product (products with different colors, styles, sizes often share a parent ID)
- user_id: String ID of the reviewer
- timestamp: Integer representing the review time in unix format
- verified_purchase: Boolean indicating if the user's purchase was verified
- helpful_vote: Integer count of helpful votes for the review

Methodology

The project followed a structured data science methodology, including the following steps:

Data Sampling and Formatting

The ‘sample_data_generation.ipynb’ notebook is implemented for down sampling of the huge dataset into 1M reviews and some early data cleaning operations for the reviews with broken formats.

The downloaded Amazon Reviews'23 dataset is a huge dataset with 571 million reviews inside from 34 different product categories.

To make more rapid experiments with lower resource requirements (GPU, memory and disk space etc.) and cost, below actions are taken:

- Make a wise down sampling from the huge data to ensure there are large enough samples to have a rating and category agnostic sentiment analysis,
- Make some smart feature engineering actions (rather than storing images, just extracted has_image feature and dropped images and categories added by using file names of the reviews with no processing requirement of product meta files)
- Build the customized model to benefit from transfer learning by fine-tuning of fundamental LLM models trained with super big data
- Utilize a cloud environment to benefit from flexible and free resources as much as possible

There were multiple steps to follow after downloading the dataset. Rather than processing all files together which requires huge resources, the processing is executed per file separately and saved the outputs as csv files. Below steps are followed:

1. Using Pyspark to open and index each review file per product category as Spark Data Frame by benefiting Spark's parallel processing capabilities

2. Generating two new features ('product_category', 'has_image') and dropping one feature column ('images' which contains image URLs)
3. Taking required number of samples from each rating of each product category and saving as temporary csv files.
4. Opening each csv folder with Pandas to get single partition file which is in csv format and removing the rows with more than expected columns (For example record is expected to have 11 columns but there are more or less columns) and the rows with unexpected value types in columns (for example there are Boolean type columns with number inside for some rows).
5. Merging all cleaned data for each category into a single clean data file and saving it as parquet file for future steps.

Sampled dataset size: ~1 million reviews
Sampling method: Approximately equal number for per rating class per product categories
Targeted sample per rating per product category: 1 million / (34 * 5) = 5882 sample

The final dataset includes approximately 1 million reviews from 34 product categories including Unknowns for each rating score (1-5 stars).

Data Wrangling

The ‘exploratory_data_analysis.ipynb’ notebook consists of both initial data wrangling operations followed by exploratory data analysis steps. Some partial data cleaning operations are also executed during sampled dataset file generation in file ‘sample_data_generation.ipynb’ as well.

1- Discovery: Explored and understood the collected raw data. Identified data sources, assessed data quality, and gained insights into the structure and format of the data.

Since we collect approximately equal number of ratings for each product category, it is obvious to have 3.01 as mean with 1.42 standard deviation.

2- Structuring: Organized and formatted the raw data to facilitate efficient analysis by handling missing values, converting data types. For example, timestamp column is in UNIX Time format in milliseconds originally converted into human readable datetime format for more efficient EDA analysis.

3- Cleaning: Addressed inconsistencies, errors, and outliers within the dataset by removing inaccurate data, addressing anomalies, standardizing inputs (converting timestamps) and removing irrelevant data.

There are 1035 records with empty review fields which are removed.
Remaining test set size is 951,769 reviews.

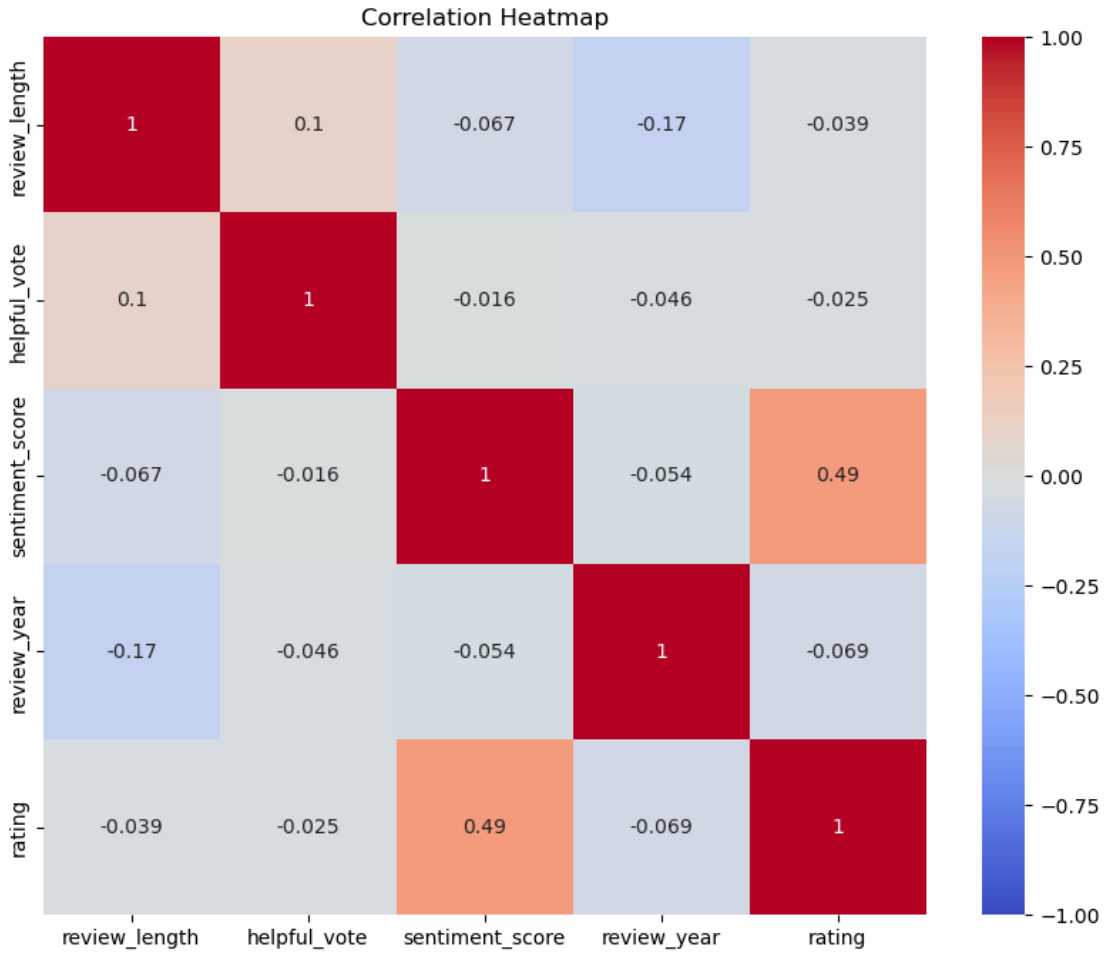
4- Enriching: Enhanced the dataset with additional information to provide more context or depth. For example, timestamp information is converted to ‘review_year’, ‘review_month’

columns for more efficient EDA analysis. Or sentiment analysis function of TextBlob library is utilized to enrich data with a new ‘sentiment_score’ column.

Exploratory Data Analysis (EDA)

The ‘exploratory_data_analysis.ipynb’ notebook focuses on thoroughly exploring the dataset to understand its structure, patterns, and relationships.

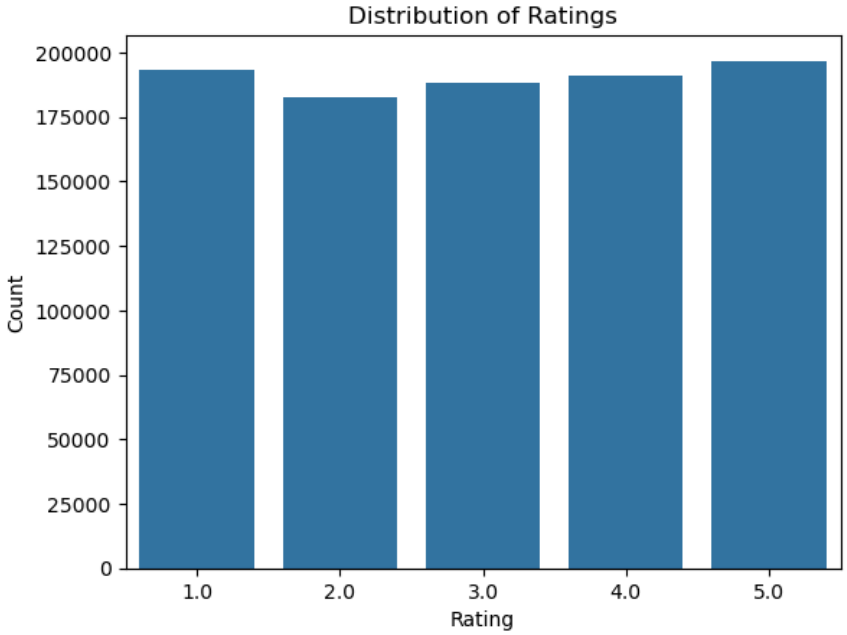
1- Descriptive Statistics: Summary statistics such as mean, median, standard deviation, and quartiles were calculated for various features. These statistics provided insights into the central tendency and variability of the data.
2- Visualization: Visual representations, including histograms, box plots, and scatter plots, were created to explore the distributions, correlations, and trends among variables. These visualizations helped in understanding the data more intuitively and identifying potential patterns.
3- Feature Importance:
    a. Correlation Analysis: Analyzed correlations between features and target variable (‘rating’) to identify potential influencers.’sentiment_score’ feature has the highest positive correlation with rating values as 0.49.



b. Feature Importance Techniques: Utilized various methods to identify the most significant features that may influence rating, laying the groundwork for modeling.

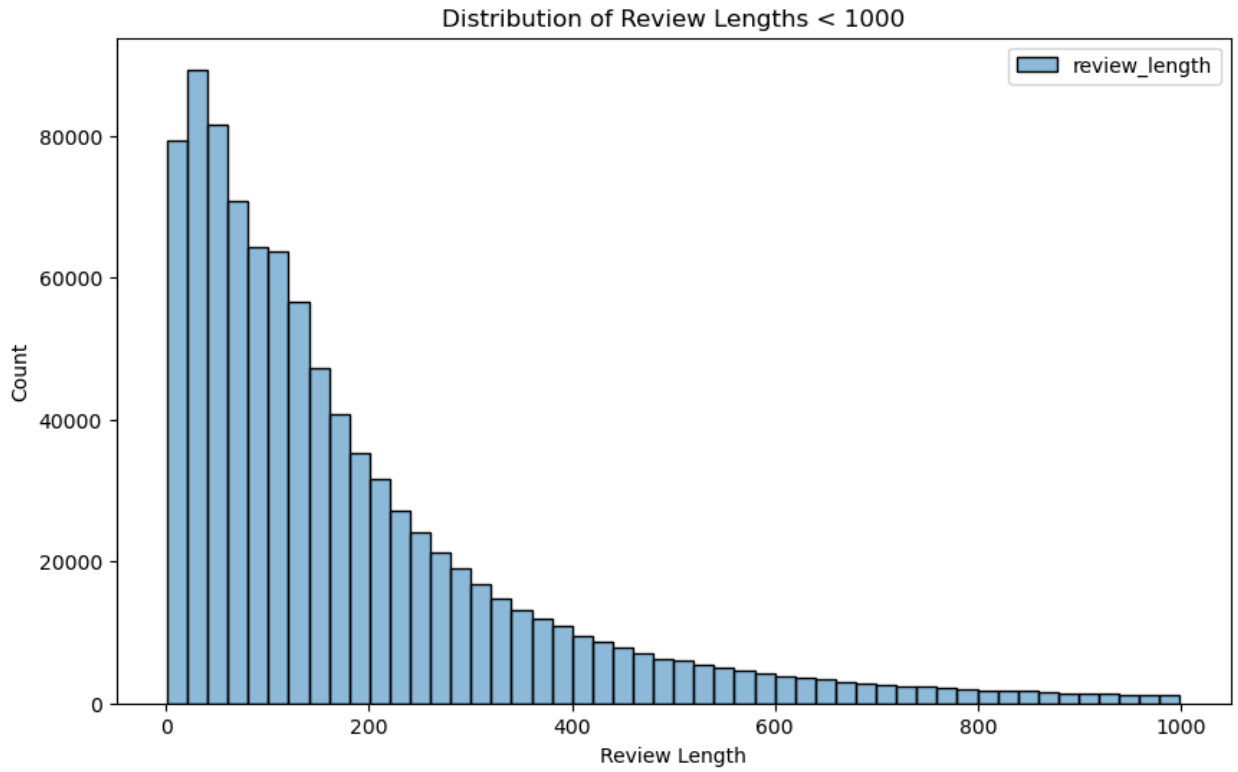
Next, the notebook explores the dataset by analyzing the distribution of ratings, counting the number of reviews for each rating, and visualizing the rating distribution using a bar plot.

From the distribution of ratings, it is obviously indicating the equal number of ratings in the sampled database. This will help us to have a balanced dataset when training the ML model for sentiment classes (positive, neutral or negative).

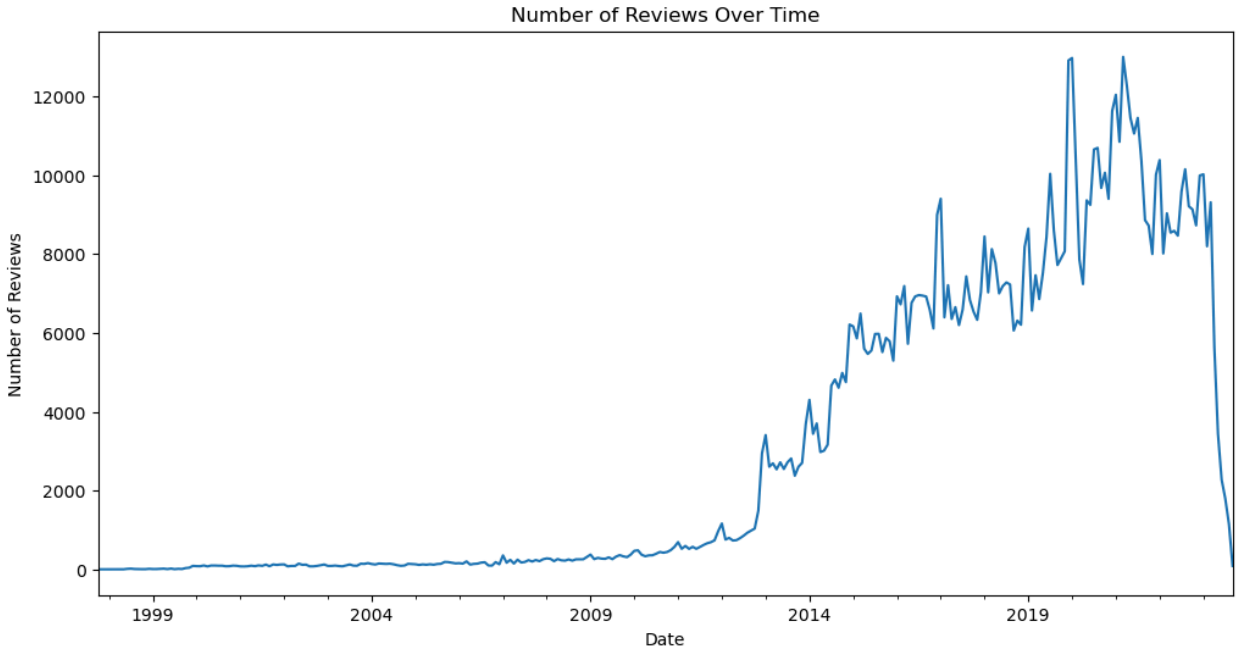


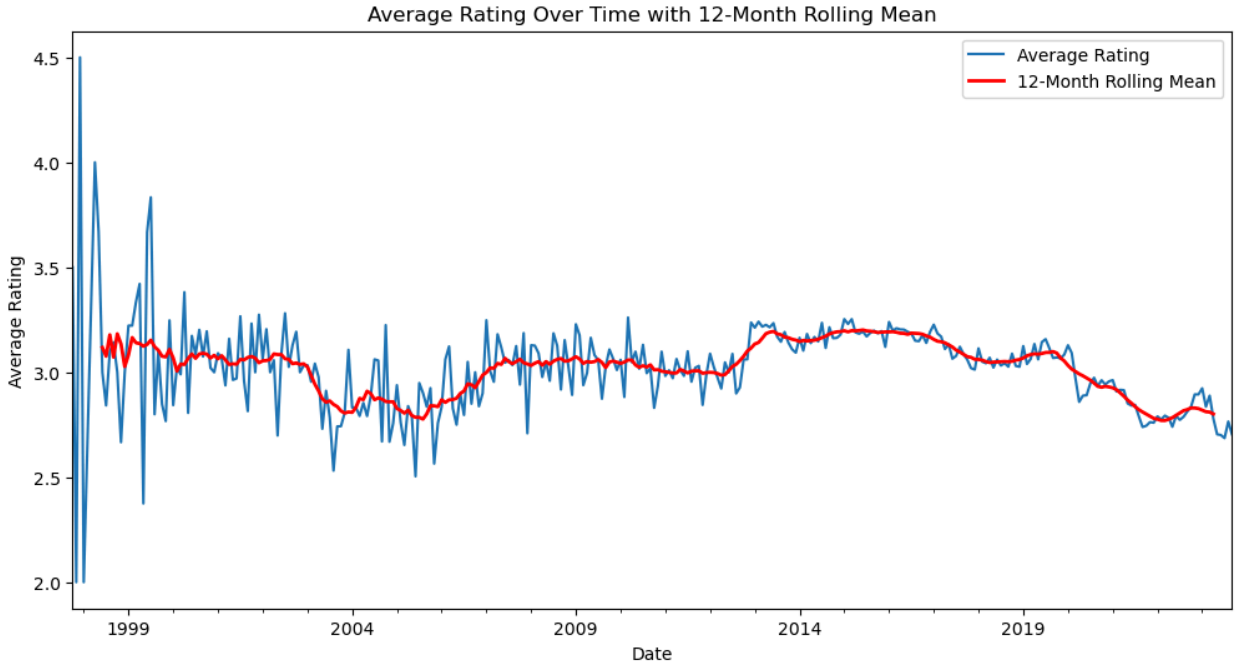
When we check for the review length distribution, the major amount of the reviews are short reviews. Definitely we look for the correlation of length with the positive or negative reviews in further steps but at the first glance, it gives the impression that since all ratings are distributed equally but there is more of short reviews, the correlation should not be strong between rating and review length. Meanwhile, there are more than 1000 reviews with more than 1000 characters and 395 of reviews has more than 5000 characters up to review texts with 28K character. It is crazy, but data is data!

Correlation between review length and rating: -0.04 which means no correlation basically.

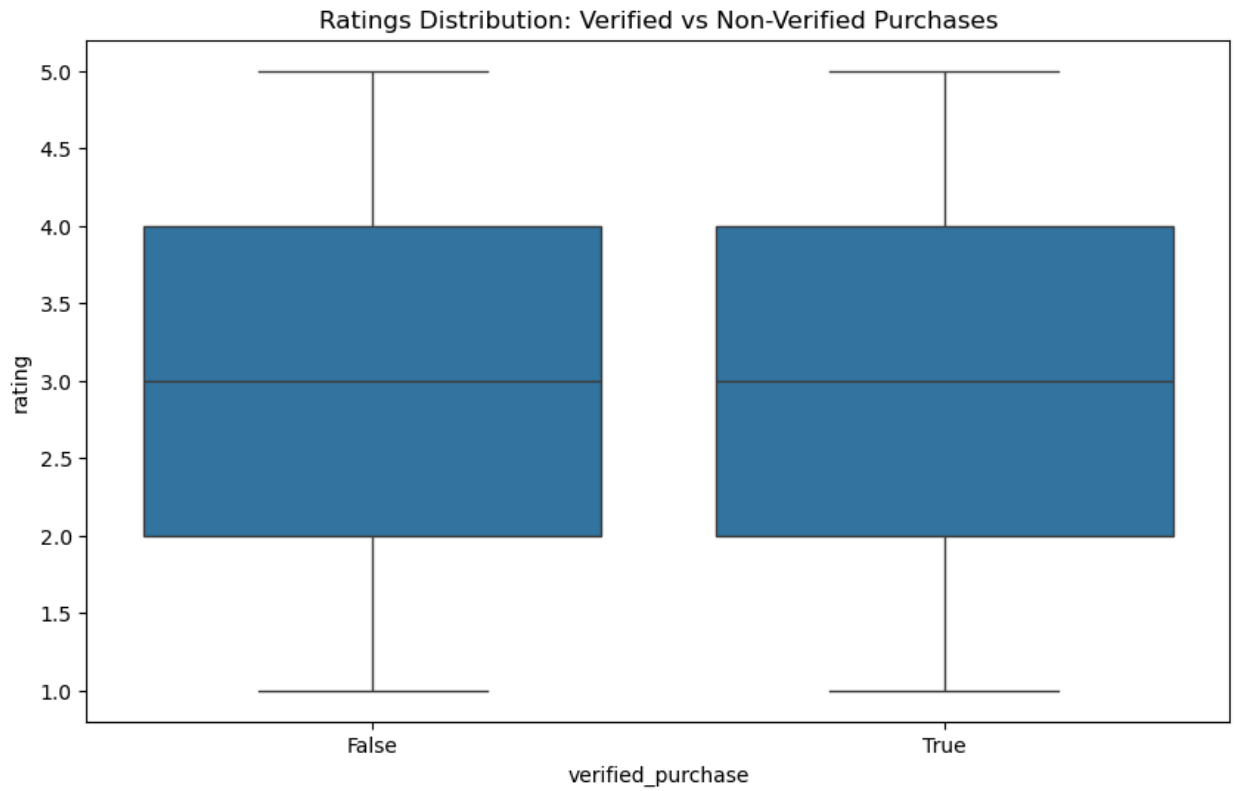


In compliance with the original large dataset, the sampled dataset also has more review records recently after 2018. There exist few samples in the past. We looked at the correlation between rating and time. Since the ratio of samples are more from recent years, the average ratings by time have less deviation as shown in the second figure below. The average rating is more likely to decrease over time in recent years with an increased number of reviews, but we should keep in mind that we sampled an equal number of ratings for each product. Maybe the ratio of positive reviews was higher in history.





The ratio of verified purchases is quite large comparing to unverified ones in the dataset.
Percentage of verified purchases: 88.73%
But regardless from the difference in ratio, below boxplot diagram indicates that there is no difference of being verified or unverified over rating.



Correlation between helpful votes and rating: -0.03

To get a high-level understanding of the frequency of some words for positive and negative reviews, generated word clouds. Even though there are some very common words such as "use" or "like" (probably as don't like in negatives), there are some certain words in two categories. Negative reviews consist of expectation, disappointment and price etc. Positive reviews consist of more love, greatness and easiness.

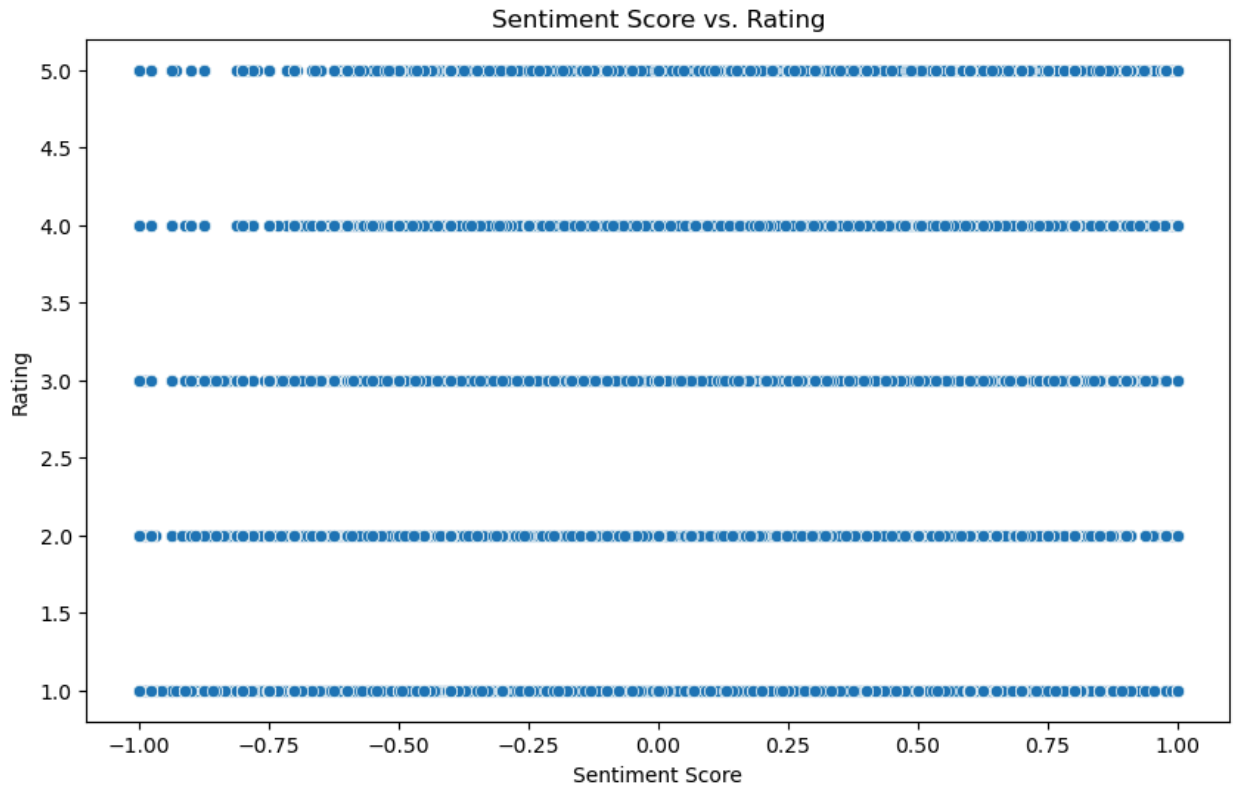
Word Cloud for Positive Reviews

Word Cloud for Negative Reviews

TextBlob package has sentiment analysis functionality inside that is a good way of building a base function to benchmark with DL based solutions. But as you see in the scatter plot below, it does not label the sentiment correctly for many reviews. There are many 5 start reviews with negative sentiment category, and vice versa for 1-star reviews.

It is obvious that an LLM model can provide a better solution than word scoring based lexicon analysis solutions as similar to TextBlob.

But in any case, the correlation between sentiment score and rating: 0.49 which means it is working partially.



The percentage of reviews with images is 5.26%. The people who share their comments with images tend to give less rating (around 1s) compared to reviews with no images as seen in below boxplot. Because in general users have tendency to upload images when they are complaining about products or packaging.

Pre-processing and Training Data Preparation

In the ‘base_statistical_ML_modeling.ipynb’ and each LLM Models training files which are ‘amazon-sentiment-analysis-bert.ipynb’ and ‘amazon-sentiment-analysis-distilbert.ipynb’ there exist both pre-processing operations to make data ready for model expectation and training operations.

There are five different models trained and benchmarked with the evaluation set which is approximately 190K of dataset with a ratio of 20 percent of whole reviews. Below models are trained and benchmarked:

- Logistic Regression Modeling with 761K Reviews,
- Random Forest Modeling with 761K Reviews,
- XGBoost Modeling with 200K Reviews,
- Pre-trained BERT Finetuned with 200K Reviews LLM Model,
- Pre-trained DistilBERT Finetuned with 400K Reviews LLM Model,
- Pre-trained DistilBERT Finetuned with 761K Reviews LLM Model

The following steps are taken to make the data ready for training.

1- **Feature Encoding:** Categorical variables were encoded using one-hot encoding, replacing the original columns with binary indicator variables to enable the machine learning model to process categorical data effectively.

For logistic regression, random forest and XGBoost models are required text transformer via TfidfVectorizer (limited to 10000 feature), numeric_transformer and categorical_transformer. For LLM models text column is required to be tokenized by using the relevant pre-trained tokenizers (with dynamic padding) of the models retrieved from Hugging Face Hub.

Also, the target variable column is generated as ‘sentiment_label’ and ‘labels’ columns by converting rating columns into categories in respect to the schema below.

Rating Star	Sentiment Mapping	Encoded Numeric Value
1 or 2	Negative	0
3	Neutral	1
4 or 5	Positive	2

2- **Feature Scaling:** Numeric features were standardized using the StandardScaler from SciKit-Learn, ensuring that all features were on the same scale, which is crucial for many machine learning algorithms to perform optimally.
3- **Dataset Splitting:** The data was divided into training and testing sets, allocating 80% for training and 20% for testing. This step ensures that the model is trained on a sufficient amount of data while also having unseen data for evaluation.

Modeling and Benchmarking

The models are trained in the notebooks below with different training set sizes due to model training cost. Pre-trained LLM Models are retrieved from Hugging Face Hub and finetuning training with new sample dataset are done on Google Colab Service due to dependency of more powerful GPU environment.

Model	Training Set	Test Set	Notebook
Logistic Reg. 761K	761K	200K	base_statistical_ML_modeling.ipynb
Random For. 761K	761K	200K	base_statistical_ML_modeling.ipynb
XGBoost Model 200K	200K	200K	base_statistical_ML_modeling.ipynb
BERT Finetun. 200K	200K	190K	amazon-sentiment-analysis-bert.ipynb
DistilBERT Finetun. 400K	400K	190K	amazon-sentiment-analysis-distilbert.ipynb
DistilBERT Finetun. 761K	761K	190K	amazon-sentiment-analysis-distilbert.ipynb

1- **Model Selection and Hyperparameter Tuning:** A range of machine learning models are trained and benchmarked as listed in table above. Each model is initialized with default parameters, and hyperparameter tuning is performed using GridSearchCV to find the optimal set of parameters for Logistic Regression, Random Forest and XGBoost models. For LLM models different batch size parameters are used for training. Trained models with the best hyperparameters are saved to file storage along with feature encoding classes such as tokenizers or transformers (text, numeric, categorical).
For BERT LLM model finetuning, ‘bert-base-uncased' model is utilized as foundational model and tokenizer. For DistilBERT LLM model, ‘distilbert-base-uncased’ model is utilized as foundational model and tokenizer. All LLM Model finetuning trainings are executed for 3 epochs with different batch values since the batch size is one of the top

parameters that is impacting GPU memory consumption. High values of batch size cause out of memory exceptions.

2- **Model Evaluation:** After training each model with hyperparameter-tuned settings, the performance is evaluated on the test set. Metrics such as accuracy, precision, recall, F1-score are calculated as overall average of all classes and scores per class type (negative, neutral and positive). Classification reports and confusion matrices are generated to provide detailed insights into model performance.

Training of the models requires different time intervals and different hardware requirements accordingly. LLM Models are trained on Google Colab Service due to their dependency on GPU hardware. For statistical models also, they take long hours of training with 16 core Intel i7 3100Ghz CPU environment.

Hardware and time requirements and output model size for each model are listed in the table below.

Model	Hardware	Training Time	Model Size
Logistic Reg. 761K	Intel i7 16-Core CPU	~4 Minutes	237KB
Random For. 761K	Intel i7 16-Core CPU	~1.0 Hour	8.9GB
XGBoost Model 200K	Intel i7 16-Core CPU	~2.4 Hours	1752KB
BERT Finetun. 200K	Nvidia L4 24GB GPU	~9.5 Hours	418MB
DistilBERT Finetun. 400K	Nvidia A100 48GB GPU	~4.2 Hours	256MB
DistilBERT Finetun. 761K	Nvidia A100 48GB GPU	~7.0 Hours	256MB

3- **Comparison of Model Performance:** The performance of different models is compared based on their accuracy values and class-based Precision, Recall and F1 Score performance.

All Models are tested with 190K-200K unseen evaluation dataset and below results are recorded for each of them.

		Negative			Neutral			Positive		
Model	**Acc**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
Logistic Reg. 761K	0.69	0.69	0.82	0.75	0.45	0.17	0.25	0.74	0.83	0.78
Random For. 761K	0.68	0.66	0.84	0.74	0.49	0.08	0.13	0.72	0.82	0.77
XGBoost Model 200K	0.70	0.70	0.83	0.76	0.48	0.16	0.29	0.75	0.82	0.79
BERT Finetun. 200K	0.80	0.82	0.86	0.84	0.55	0.49	0.52	0.88	0.89	0.89
DistilBERT Finetun. 400K	0.78	0.78	0.88	0.83	0.52	0.39	0.45	0.88	0.87	0.87

DistilBERT Finetun. 761K	**0.82**	**0.83**	**0.89**	**0.86**	**0.61**	**0.51**	**0.55**	**0.91**	**0.91**	**0.91**

Especially LLM Models are far ahead of the statistical ML models in terms of all accuracy metrics. But once we considered the difference between BERT and DistilBERT which is lightweight version of the BERT actually with less parameters but faster training times and lower resource utilization, **the DistilBERT can be a better choice because training times are lower and in terms of accuracy it is similar to BERT .**

- BERT-base-uncased:
  - 12 layers, 768 hidden units, 12 attention heads
  - Total parameters: 110 million (110M)

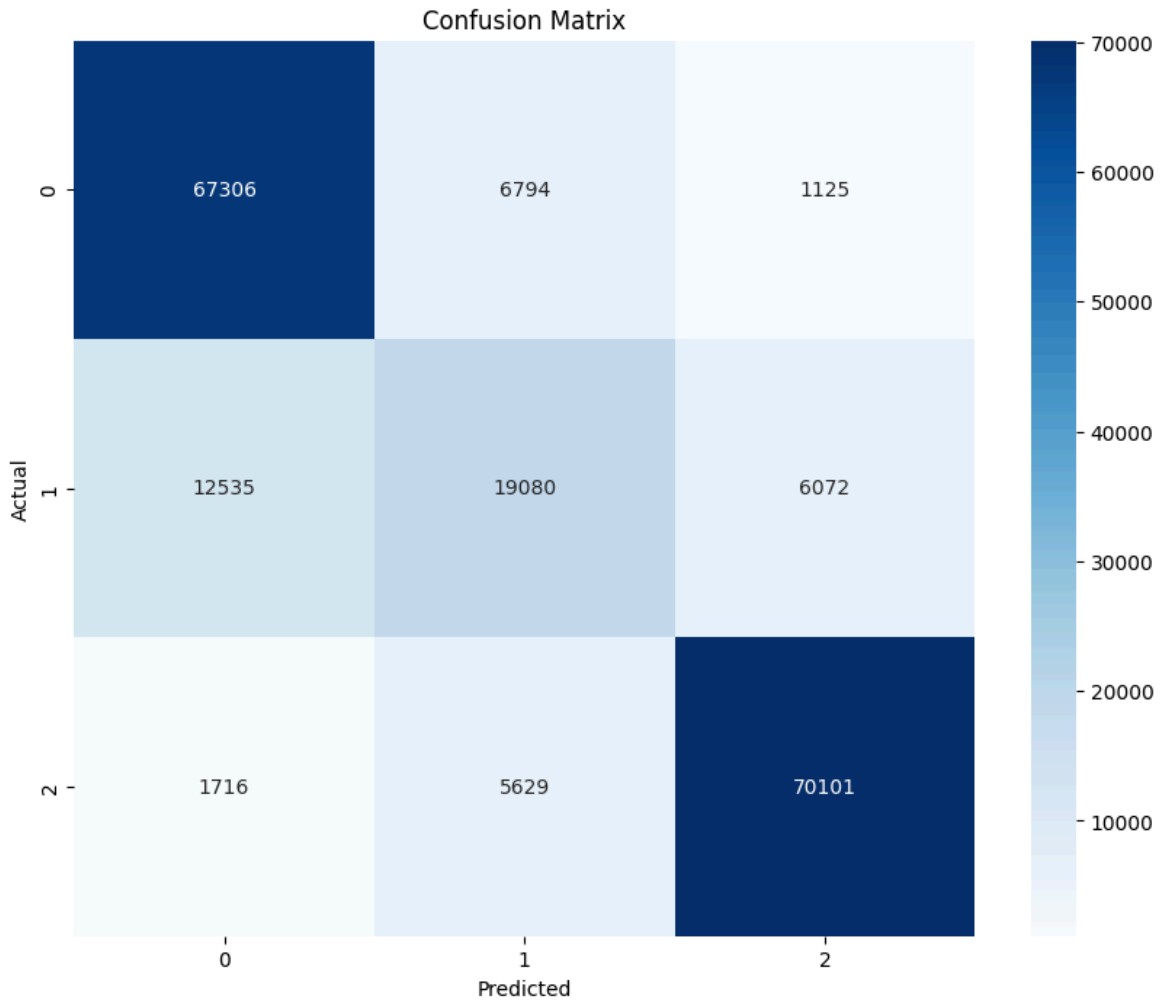
- DistilBERT-base-uncased:
  - Known to have approximately 40% fewer parameters than BERT-base
  - Estimated total parameters: ~66 million (66M)

The most score impacting class type is neutral reviews for all models. This is because of the high subjectivity of rating star assignment and weak correlation between sentences and assigned star for the review. So, the recall values for neutral reviews are suffering for all the models. But especially the statistical models are drastically bad on this class type compared to LLM models which takes the advantage of attention mechanism under the hood to have longer context memory to predict.

Training arguments for the DistilBERT 761K Model are below. It is important to be aware that batch size 100 is defined by considering the training will be executed on A100 GPU. Lower segment GPUs may send out memory exceptions if this batch size is not decreased. During the training of LLM models the output models are saved to Google Drive by mounting the drive inside the notebook.

- output_dir='./results',
- num_train_epochs=3,
- per_device_train_batch_size=100,
- per_device_eval_batch_size=100,
- warmup_steps=500,
- weight_decay=0.01,
- logging_dir='./logs',
- logging_steps=10,
- evaluation_strategy="epoch",
- save_strategy="epoch",
- load_best_model_at_end=True

The confusion matrix retrieved with the evaluation of DistilBERT 761K model is below.



The hyper-parameters of statistical models are listed as below

Best parameters for XGBoost:

{'model__learning_rate': 0.1,

'model__max_depth': 6,

'model__n_estimators': 200}

Best parameters for Random Forest: {'model__max_depth': None, 'model__n_estimators': 200}

Best parameters for Logistic Regression: {'model__C': 1}

4- **Best Model Identification**: The DistilBERT Model is the best model once considered the training time and required hardware resources. BERT Model is able to perform very close to DistilBERT Model with very limited training set, but it takes far longer by time and more resource hungry.

   DistilBERT finetuned with 761K reviews is able to provide 0.55 F1 score for neutral reviews and 0/82 overall accuracy is quite good for the project.

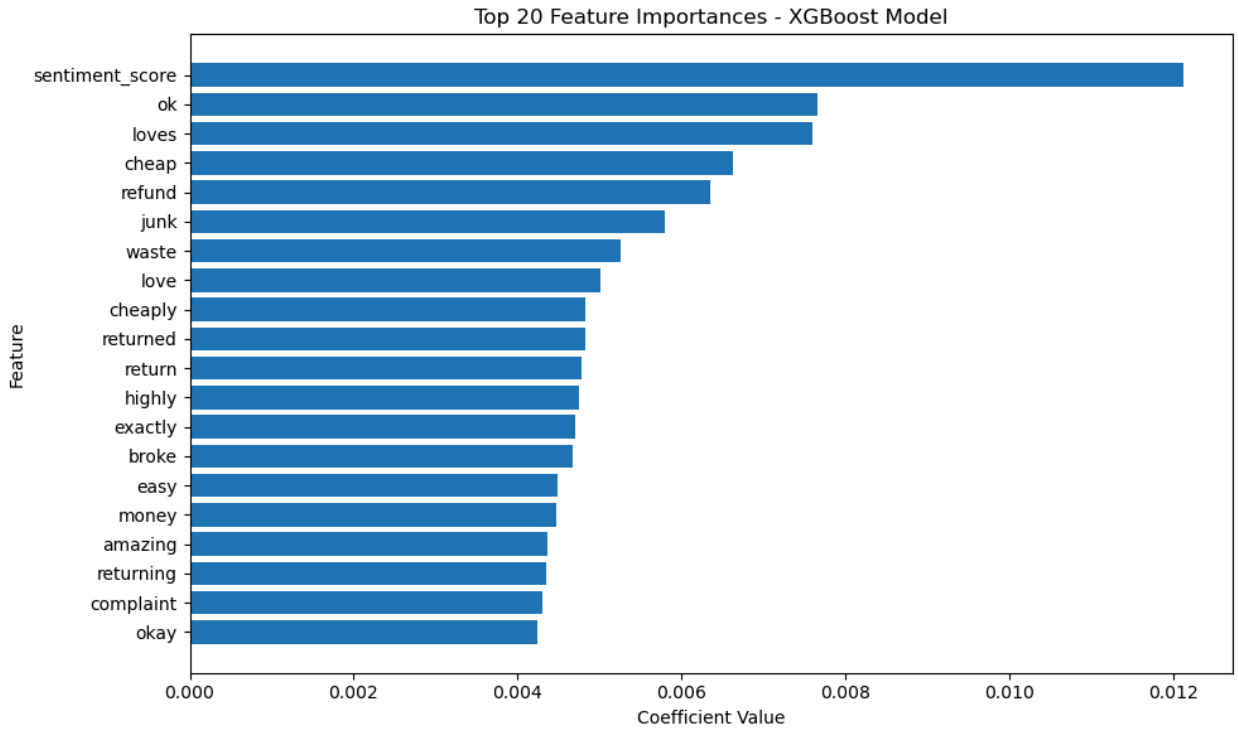
5- **Feature Importance Analysis & Visualization**: The selected finetuned DistilBERT model has a big architecture with changed header to classify the sentiment in three groups. It takes the tokenized review text as input and generates 3 element logits at the output layer.

   DistilBERT is trained using a process called knowledge distillation, where it learns to mimic the behavior of the larger BERT base model. This involves training DistilBERT to reproduce the output probabilities of the BERT base model, effectively compressing the knowledge of the larger model into a smaller one.

   DistilBERT consists of 6 transformer layers, compared to BERT base's 12 layers. Each layer has 12 attention heads, and the hidden size is 768 dimensions.

   - Configuration Parameters
   - Vocabulary Size: 30,522 tokens.
   - Maximum Sequence Length: 512 tokens.
   - Hidden Size: 768 dimensions.
   - Intermediate Size: 3,072 dimensions (feed-forward layer).
   - Number of Attention Heads: 12.
   - Number of Layers: 6.
   - Dropout Rate: 0.1 for both dropout and attention dropout

   Among statistical ML models, the relatively better performing XGBoost model’s top 20 feature importance values are visualized below. The most important feature is ‘sentiment_score’ feature as expected which consists of sentiment scoring of overall review text by using TextBlob library and this feature had highest correlation compared to other raw and extracted features.



6- **Publishing the Model:** The selected finetuned DistilBERT model has been published as a new model in Hugging Face Hub as user tuback.in repository

https://huggingface.co/tuback/distilbert-finetuned-amazon-product-reviews-sentiment

BERT model finetuned with 200K reviews is also published in Hugging Face Hub.

https://huggingface.co/tuback/bert-finetuned-amazon-product-reviews-sentiment

Key Findings and Recommendations

Influential Factors: The analysis revealed that factors LLM models are performing much more accurate compared to statistical models. Especially neutral reviews are to be a breaking point among both categories of ML models. But LLM models are more resource hungry and needs bigger data even for finetuning, The DistilBERT model is enough for our project since it is more lightweight compared to BERT model. It is obvious that if BERT model is trained same training dataset with DistilBERT, it outperforms DistilBERT. But to remain in certain boundaries in respect to environmental resources, DistilBERT fits well enough to resolve our problem.

Predictive Model Performance: Advanced LLM Models are outperforming the statistical models significantly especially on neutral sentiments.

1. **Neutral Sentiment Classification:** All statistical models struggle significantly with neutral sentiment. The Random Forest model, in particular, has a very low recall (0.08) for neutral sentiment, indicating it rarely identifies neutral reviews correctly. Logistic Regression shows similar weaknesses, with an even lower recall (0.17) and XGBoost with another low recall (0.16). But for DistilBERT LLM Model the recall value reaches 0.51 level.
2. **Negative Sentiment Classification:** All models achieve moderate performance in classifying negative reviews. Random Forest performs slightly better in terms of recall (0.84) compared to Logistic Regression (0.82) and XGBoost Model (0.83). But for DistilBERT LLM Model the recall value reaches 0.89 level.
3. **Positive Sentiment Classification:** All models perform reasonably well with positive sentiment, but there's room for improvement, especially in balancing precision and recall. But for DistilBERT LLM Model the recall, precision hence F1 score values reach 0.91 level.

Need for Advanced NLP Techniques

The current models —Logistic Regression, Random Forest and XGBoost— rely on traditional feature engineering and straightforward machine learning approaches. However, their performance highlights several limitations, particularly in capturing the nuances of sentiment in text data. Here's why more advanced NLP techniques are warranted:

1. **Contextual Understanding:** Logistic Regression, Random Forest and XGBoost lack the ability to understand context and semantics deeply. Advanced models like BERT (Bidirectional Encoder Representations from Transformers) offer contextual embeddings that capture the meaning of words in context, leading to better performance in sentiment classification.
2. **Handling Ambiguity:** The models struggle with ambiguous or nuanced language, especially in the neutral class. BERT and similar models can handle such ambiguities better due to their deep understanding of language context and subtleties.
3. **Feature Learning:** The existing models rely on handcrafted features like TF-IDF and basic text preprocessing. Advanced models like BERT learn rich, dense representations of text automatically, improving the capture of semantic and syntactic information crucial for accurate sentiment analysis.
4. **Imbalanced Classes:** The imbalance in class distribution affects model performance, particularly in predicting the neutral class. Advanced models like BERT can leverage transfer learning and pre-trained embeddings to better manage class imbalances and improve overall classification performance.

Moving to Advanced Techniques: Leveraging BERT for Enhanced Sentiment Analysis

This analysis has been a preliminary step to establish a baseline for sentiment analysis. To enhance model performance and leverage more sophisticated NLP techniques, the following steps are planned:

- **Utilize Larger Dataset:** Scaling up to a larger dataset will provide more comprehensive coverage of diverse sentiments and improve the model's ability to generalize.

- **Implement BERT with Hugging Face and Colab:** Transition to using BERT for sentiment analysis. BERT's ability to capture contextual information and nuances in text will significantly improve performance. Leveraging Hugging Face's Transformers library and Google Colab will facilitate efficient experimentation and fine-tuning on larger datasets.
- **Fine-Tuning BERT:** Fine-tune the pre-trained BERT model on the specific sentiment analysis dataset to adapt it to the nuances of the data, improving accuracy and handling of complex sentiment classifications.

These planned steps aim to achieve a more accurate and nuanced sentiment analysis model, addressing the limitations observed with traditional machine learning approaches. enhance the results, especially for neutral reviews.

Future Works

Below further research and development tasks are good to work on:

- There are some other foundational models which can be trained and benchmarked for improved accuracy such as T5, GPT, XLNet.
- Rather than staying with Amazon only dataset, different product sales platform reviews can be collected, and the project can be enriched,
- It worths to focus improvement on neutral sentiments. Since generally neutral reviews may include highly positive or negative statements inside irrelevant from its 3-star label, an NLP based method such as TextBlob sentiment scoring can be used initially to mark them for human supervision.
- There can be new data features like subscriptions for renewable products on Amazon. Since the consumers keep their subscription active to the products that they really love, this can be used as a feature for some subscription-based products to improve statistical model accuracy.
- Model performance can be benchmarked per product category if there is a difference between.

Conclusion

The project provides valuable insights about model performances between statistical ML models and LLM models for text-based sentiment analysis and showcase caveat situations. So, any seller or amazon itself can crawl through the customer messages in social media platforms (Instagram, twitter) or public forums (i.e. reddit) about certain products. This helps to extract the sentiment to measure the general perception of the product and services from all available platforms not limiting sellers to comments provided in amazon website only.

Contributing

Contributions to this project are welcome.