Project Checkpoint Report

Nikhil Gupta	ngupta22
Krishnakumar Chavan	kchavan
Venkatakrishnan Veeraraghavan	vveerara

1. Dataset Acquisition and Preprocessing

Dataset Description

Describe the dataset you have acquired for your project, including its source, size, and key features. Mention any data cleaning and preprocessing steps performed on the dataset.

There are 522,517 recipes in the recipes dataset, divided into 312 categories. Each recipe's components, preparation timings, serving sizes, nutritional information, directions, and other details are included in this dataset.
There are 1,401,982 reviews in the reviews dataset, written by 271,907 distinct users. The author, rating, review text, and other details are all included in this dataset.

There are two alternative formats available for the recipes dataset:

recipes.parquet and reviews.parquet was advised since it maintains the original data's schema.
recipes.csv is designed to be parsed in R while reviews.csv does not contain any list-columns so it can be easily parsed.

Data Cleaning

During the data cleaning process, missing values were handled in various columns including 'CookTime', 'Description', 'Images', 'RecipeCategory', 'Keywords', 'RecipeIngredientQuantities', 'AggregatedRating', 'ReviewCount', 'RecipeServings', and 'RecipeYield'. Missing values were replaced with medians or placeholders where appropriate. Rows with missing essential information or where data imputation was not feasible were dropped from the dataset. Additionally, duplicate entries based on unique identifiers ('RecipeId' for recipes and 'ReviewId' for reviews) were removed to ensure data integrity and accuracy.

Preprocessing

In the preprocessing stage, we converted time duration strings into a standardized format and numerical values. We also tokenized and standardized text data in the reviews dataset for tasks like sentiment analysis. Furthermore, we extracted keywords and ingredients for modeling and ensured consistent data formats across columns for easier analysis.

2. Exploratory Data Analysis (EDA)

Overview

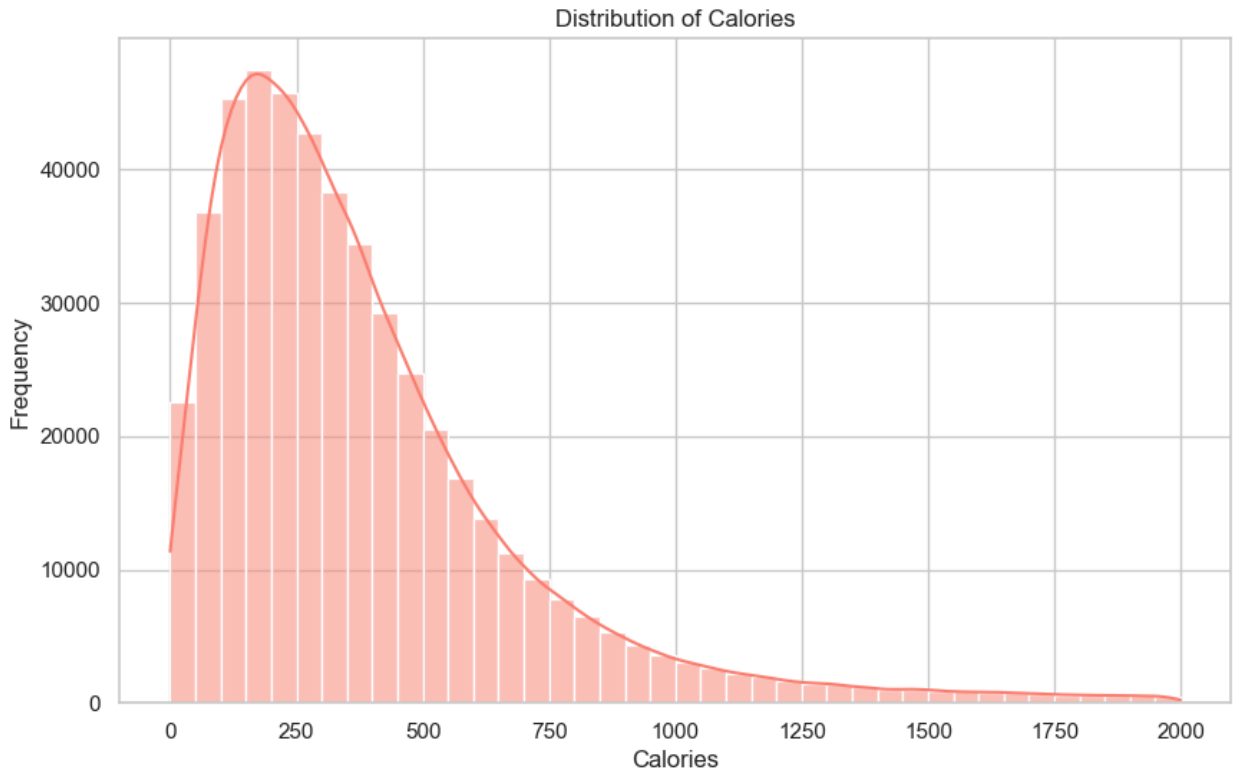
The Exploratory Data Analysis (EDA) was conducted to gain insights into the datasets "recipes_df" and "reviews_df." This involved tasks such as identifying missing data, analyzing data distributions, investigating variable relationships, and obtaining useful insights for future analysis.

Key Findings

Recipes Dataset:
Total entries: 522,517
Missing values in several columns, such as CookTime, Description, Images, RecipeCategory, Keywords, AggregatedRating, ReviewCount, RecipeServings, and RecipeYield.
Significant missing values in AggregatedRating (253,223) and ReviewCount (247,489) columns.
Numeric columns like Calories, FatContent, SaturatedFatContent, etc., have varying ranges and distributions.
Categorical columns like RecipeCategory, AuthorName, and Keywords show diverse categories and text data.
Reviews Dataset:
Total entries: 1,405,421
Only one column with missing values: Review (214 missing values).
Numeric columns like Rating show a range of values from 1 to 5, indicating user ratings.
DateSubmitted and DateModified columns suggest temporal information about when reviews were submitted and modified.

Visualizations

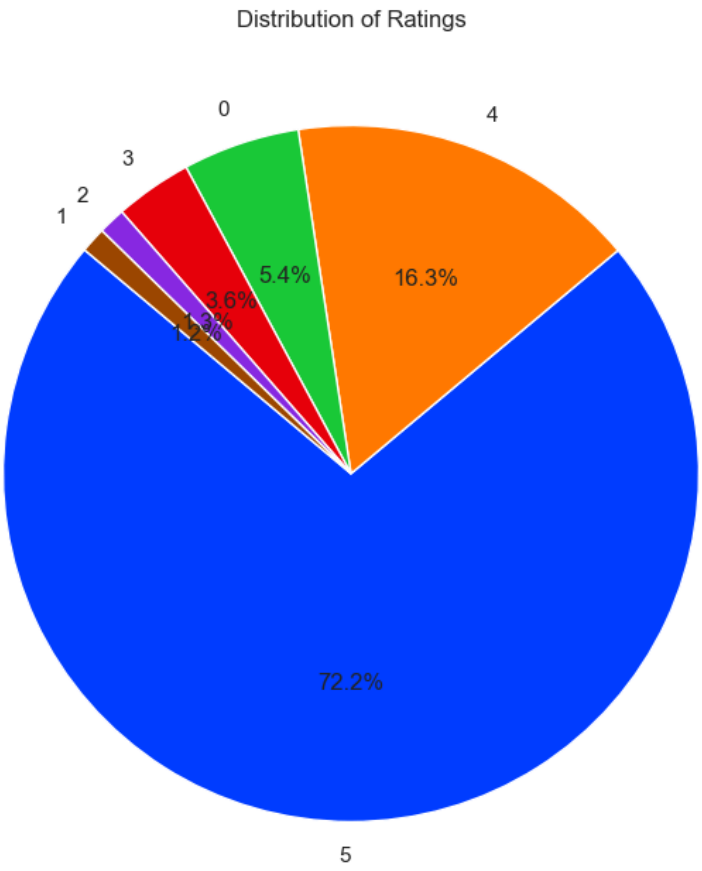
Distribution of Calories:



The histogram visualizes the distribution of calorie content across recipes in the dataset. The x-axis represents the calorie range, while the y-axis shows the frequency or count of recipes. To better represent the distribution, the histogram is plotted with a kernel density estimate (KDE) curve, which provides a smooth representation of the underlying probability density function.

From the histogram, we can observe that the calorie distribution is right-skewed, with a higher concentration of recipes having lower calorie counts. However, there are also recipes with very high calorie values, as indicated by the long tail of the distribution. This insight could be useful for identifying and analyzing recipes based on their calorie content, such as categorizing them as low-calorie, moderate-calorie, or high-calorie options.

Distribution of Ratings:



The pie chart illustrates the distribution of user ratings for recipes in the reviews dataset. Each slice of the pie represents a rating value (e.g., 1, 2, 3, 4, or 5), and the size of the slice is proportional to the number of reviews with that particular rating.

By analyzing the pie chart, we can gain insights into the overall sentiment and satisfaction of users towards the recipes. For example, if a large portion of the pie is occupied by higher ratings (4 or 5), it suggests that users generally had positive experiences with the recipes. Conversely, a significant portion of lower ratings (1 or 2) could indicate areas for improvement or recipes that did not meet user expectations.

3. GPT-2 Model Development

Model Architecture

At the core of our project lies the powerful GPT-2 (Generative Pre-trained Transformer 2) model, a state-of-the-art language model developed by OpenAI. Built upon the Transformer architecture, GPT-2 leverages self-attention mechanisms to capture long-range dependencies in text, enabling it to understand and generate natural language with remarkable fluency.

We harnessed the pre-trained GPT-2 model as a foundation and fine-tuned it on the vast Food.com Recipes and Reviews dataset. This allowed the model to specialize in the culinary domain, learning the intricacies of recipe instructions, ingredient combinations, and user preferences.

Training Process

To tailor the GPT-2 model for personalized recipe recommendation, we embarked on a comprehensive training process. First, we curated a representative subset of the Food.com dataset, carefully preprocessing the text data by tokenizing and encoding it using the GPT-2 tokenizer.

We then defined a custom PyTorch dataset class to streamline the loading and preprocessing of the data, and employed PyTorch's DataLoader to efficiently batch the input sequences.

We used various resources to fine tune several hyperparameters, which includes batch size, no of epochs, learning rate, etc to optimize model’s performance.

With our computational resources primed, we fine-tuned the pre-trained GPT-2 model on the prepared dataset, leveraging the PyTorch implementation of the model (GPT2LMHeadModel). The training process unfolded on powerful NVIDIA GPUs, where the model learned to generate personalized recipe recommendations through the Adam optimizer and cross-entropy loss minimization.

After an extensive training regime, we preserved the fine-tuned model and its accompanying tokenizer, ready to deploy in the generation of tailored recipe recommendations.

Results

We are currently taking only 5000 rows out of 1.6M rows.
This is producing following problems:
Training Data Quality: Fewer samples of training data is heavily impacting model’s training ability to produce gibberish outputs as it has not received enough data to completely train the model. The model is lacking various diverse samples of different data, so the mode is failing to generate sensible outputs.

Fine-Tuning: For now, the model might be only fine-tuned on a specific domain or task, allowing it to produce biased and irrelevant results. Training the data with more samples could potentially improve model’s performance.

Tokenization: The input text might not be tokenized properly or the samples tokenized do not produce enough numbers for the model to train on. We'll ensure that model data is properly tokenized for final submission.

Below are the results we are getting.

```
# Decode and print recipe suggestions
for i, recipe in enumerate(output):
    print(f"Recipe suggestion {i+1}: {tokenizer.decode(recipe, skip_special_tokens=True)}")
```

```
Recipe suggestion 1: I want to make cranberry burritos. I want want make make MAKE MAKE Make Make Made Make Making
Making Make Get Get Got Got Get

1919202070707171 71 71 70 70704704704694694714714694664664464464646 46 464648483232 32 32 33 33 32 31 31 30 30
...-.-.
 ( (((<<
)) ) ) ( (( (((((( (( ((( ()(((((((( ((( (()(( (///././ and/ (.//(/(/ / //*//?/?/.,,,,,., 8 8 9 9 8 10 10 12 12 13
13 14 14 15 151515 15 2015 201520152015 2015 2016 2016 16 16 8 7 7 8 4 4 88899 9 10 20 20 25 25 26 26 24 24 25 27 2
7 28 28 29 29 30 31 32
 or
Recipe suggestion 2: I want to make cranberry burritos.

I'm not going to take take. I III I ( I ) ( ( ) )
 () ((()()(((((( (( (((( ( (( ( [ [ ] ] [ 1 1 2 2 3 3 4 4 5 5 4 6 6 66 66 68 68 69 69 70 70 71 71 7070702020 20 20
,,,..!!!!''
//... [,,..... - - ( -      -  * *

Recipe suggestion 3: I want to make cranberry burritos.   I I
   - - - ---

I IIIIIIIIIIII III III  IIIIIIIIIVIVIIIIVIIIIIVIIIXIXIVIXIVIIII II II I

        '' " " "" """""""""""""" """ """""""" ""="""="""="/"/"="…"…""…"……"……"……"…)…)…….…….…… …
……)...)…)…]…)!]!]']']''',',,, and and in in In In IN IN in the in a a an an a as as a of of a for for in at at in
between between among among amongst amongst among amidst amidst amid amidst amongst amidst among amid amid amongst
amid among throughout throughout amidst throughout amid throughout Throughout Throughout Through Through Throughout
During During Throughout Despite Despite Throughout Additionally Additionally Furthermore Additionally Moreover Add
itionally Accordingly
```

4. Challenges and Next Steps

Challenges Faced

One of the key challenges we faced was training the large GPT-2 model on the dataset, which required significant computational resources and time. Due to the high complexity and memory requirements of the model architecture, we had to carefully manage the training process and employ techniques such as gradient checkpointing and mixed-precision training to optimize resource utilization.

Another challenge was dealing with the large scale and high dimensionality of the datasets, particularly the recipes dataset with over 500,000 entries and numerous features. Processing and analyzing such a large dataset required optimizations to ensure efficient data handling and model training.

Next Steps

In the next phase, we will focus on fine-tuning the GPT model using transfer learning techniques on the Food.com dataset to generate personalized recipe recommendations tailored to user preferences and dietary restrictions. Simultaneously, we will experimentally develop image generation algorithms based on Conditional Generative Adversarial Networks (CGANs) to create realistic visual representations of the recommended recipes, including step-by-step images and final dish images. User testing and feedback will be conducted to evaluate the accuracy of recommendations, quality of generated images, and overall system usability. A user-friendly web interface will be designed to allow users to input their preferences

and receive personalized suggestions with visual guides. Throughout this phase, we will continuously refine our methodologies, explore additional features, and optimize the system's performance.

5. Conclusion

In this checkpoint report, we have documented the progress made thus far in our project focused on recipe generation and analysis using natural language processing techniques. We acquired and preprocessed two datasets: a recipes dataset containing over 500,000 recipes with detailed information, and a reviews dataset with over 1.4 million user reviews.

We have gained valuable insights into Transformers and Generative Adversarial Network's architectures and characteristics through exploratory data analysis of datasets we've chosen, which not only includes filling in missing values, data visualizations and examining various variable relationships such as calorie content, nutritional components, and user ratings.

We have also provided initial visualizations to illustrate key aspects of the data, such as the distribution of recipe categories, calorie content, and user ratings. These visualizations will serve as a foundation for further analysis and modeling tasks.

Moving forward, our primary objective is to develop and refine a GPT-2 model capable of generating coherent and relevant recipes based on the dataset. We plan to experiment with different model configurations, integrate user review data, and implement comprehensive evaluation strategies to assess the quality and usefulness of the generated recipes.

6. References

1. https://pandas.pydata.org/docs/
2. https://numpy.org/doc/
3. https://matplotlib.org/stable/index.html
4. https://seaborn.pydata.org/
5. https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp/
6. https://pytorch.org/tutorials/
7. https://huggingface.co/transformers/
8. https://huggingface.co/transformers/model_doc/gpt2.html