In [ ]:

```
from sklearn.metrics import accuracy_score

def construct_json_input():
    df_training_subset = read_data().sample(n = 5000, random_state = 200)
    true_labels = df_training_subset["star_rating"]
    training_subset_json = json.dumps({
        "n_reviews" : len(df_training_subset),
        "reviews" : df_training_subset.drop(columns = "star_rating").to_dict("records")
    })
    return bytes(training_subset_json, encoding = "utf8"), true_labels

# test the deployed model with a subset of training data
def test_model_deployment():
    data, true_labels = construct_json_input()
    status_code, response = ping_endpoint(data)
    print("Status code", status_code)
    predicted_labels = json.loads(response)["predictions"]
    print("Model accuracy", accuracy_score(true_labels, predicted_labels))

test_model_deployment()
```

Part F: Follow-up questions

Now that you have completed the coding portions, answer the following questions. Write your answers as if you are composing a technical report for your supervisor; limit your responses to **one or two paragraphs** (5-10 sentences per paragraph) in each question.

You should write your answers in markdown within this notebook, below the **Your response** line for each question below.

Question 1 (6pts): In Part A (Data Cleaning), did you exclude any rows in the dataset? If you did, what are the exclusion criteria and how many rows were excluded? Did you remove or clean any column? Explain in sufficient details so that someone who reads your description could replicate the code from scratch.

Grading rubric:

Rubric item	Score
Describe all the row-based operations, e.g. whether and how you excluded rows	2pts
Describe all the column-based operations, e.g. whether and how you removed or cleaned columns	2pts
Provide sufficiently detailed descriptions that can be used to reconstruct the code	2pts

Your response:

No rows were excluded from the dataset from my operation. If I had more time, I would use the verified purchase column to filter out the reviews which are bad as `N` .Regarding to the columns such as 'marketplace', 'customer_id', 'review_id', 'product_id', 'product_parent', 'product_category', and 'review_date' were removed as they were not needed for the analysis. The `drop` method was used to remove these columns.

Question 2 (5pts): In Part B (Exploratory Data Analysis), which types of visualization did you employ (e.g., scatter plot, boxplot)? Why did you choose these types? Which columns or relationships between columns did you investigate? Summarize your findings.

Note: Even if you feel tabular explorations are enough or if you use EDA tools, you should still perform at least one visualization analysis, so that you can respond to this question.

Grading rubric:

Rubric item	Score
List all of the graph types used along with their goal (e.g., “I used violinplot to see the distribution of data in column X,” instead of simply “I used violinplot”)	1pt
List all the columns or relationships between columns that were examined.	2pts
Describe the findings from the exploratory data analysis in layman’s terms, so that a reader not familiar with the dataset or your code can still understand them.	2pts

Your response:

I first worked out the `value_count` of reviews by rating and visualized that with barchart. It showed that the rating count is not balanced.

I then chose a boxplot to visualize the distribution of star ratings within the data in order to clearly identify key statistical elements such as the median, quartiles, and any outliers. It showed that the graph is negatively skewed, indicating that most of the data values are concentrated on the higher end of the scale. This could signify a tendency towards higher star ratings.

Then I used Scatter Plot to visualize Helpful Votes and Total Votes Relationship to understand the correlation between

Question 3 (6pts): In Part C (Feature Construction), which feature construction method(s) did you employ? Specify the associated parameters (for example, if you used Sklearn's TfIdf, what are your preprocessor, tokenizer and analyzer?).

Grading rubric:

Rubric item	Score
List all of the feature construction steps, along with the state of the dataset before and after each step.	3pts
In each step, specify the input parameters in sufficient details so that the original code can be reconstructed from your descriptions.	3pts

Your response:

In Part C, I implemented Term Frequency - Inverse Document Frequency (TFIDF) Vectorization converted text documents into sparse matrices. I applied a Sklearn TFIDF vectorizer to the training reviews. This method was carried out without a specific preprocessor. Both the tokenizer and analyzer were configured to utilize a string split function. Then I fitted the vectorizer with test reviews, Finally, I applied the entire dataset to fit the vectorizer.

Question 4 (5pts): In Part D (Model Training and Evaluation), list all the models that you explored. How did you select the best model, or the best combination of models (if you used an ensemble method)? If you used train-test split or cross validation, describe them in detail (e.g., the training portion, the type of cross validation, etc.).

Note: For model selection procedure, you should report your selection criteria and hyperparameters considered, among other details. Imagine someone has to work off your description and find the exact same model you report as the best.

Grading rubric:

Rubric item	Score
Name at least two explored models.	2pts
Provide sufficiently detailed descriptions of the model selection procedure that can be used to reconstruct the code.	3pts

Your response:

1. Data Splitting: I initially divided the dataset into a training and a test set. I divided the data into training and test sets, with 20% testing reviews and 80% training reviews, at `random state=0` .
2. Cross-Validation: I employed 5-fold cross-validation within the training set. This method involved splitting the training set into five 'folds', and the model's performance was averaged over the five iterations.
3. Model Exploration:

- **Logistic Regression**: I experimented with logistic regression, where I fine-tuned the maximum number of iterations by trying values of 10, 100, and 1000. Through cross-validation, I determined that 100 iterations provided the best average accuracy with reasonably time, resulting in approximately 63% accuracy.

- **Support Vector Machine**: For the SVM, I used the RBF kernel with a regularization parameter `C=0.01` .I chose `multi_class=ovr` as there are mutiple classes to predict. I got an average accuracy score higher than 64%.
- **K-Nearest Neighbors**: I used the KNN algorithm with the default setting of 5 nearest neighbors. I did not receive satisfactory results, but the model still didn't converge.

4. The best model was Linear SVC at `LinearSVC(C=0.01, multi_class='ovr',class_weight="balanced")`

Question 5 (3pts): Did you complete Parts A-E in one pass, as an ordered sequence of steps, or was it necessary to revisit some earlier parts of your solution in order to achieve the desired outcome when working on a later part? If you proceeded sequentially, did you perform any data planning to identify the best steps for each part? If you had to revisit some parts of your solution, which part did you revisit most, and why?

Grading rubric:

Rubric item	Score
Mention whether you went through the parts A-E sequentially.	1pt
Describe your data planning if you proceeded sequentially, OR	2pts
Specify the part that you revisited most recently and the reason for doing so.	2pts

Your response:

I completed A-E in order. This is a logically order to preprocess the data,do exploratory data analysis, contract feather vectors, then model training and last deployment. The section I had to visit the most is the model training, because it takes trials and errors to find out the optimal parameters. I had to repeated the preprocessing Part A because I need to get the reviews in the raw form ready to be trained for the models.

Question 6 (4pts): Did you identify any imbalance in the dataset? If you did, describe it here. Which method did you apply to address this imbalance? Describe the state of the dataset after you applied this method; was the identified imbalance mitigated or completely resolved?

Note: even if your final model did not address the data imbalance issue, you should still implement a method to address it for evaluation purpose, so that you can respond to this question.

Grading rubric:

Rubric item	Score
Describe the imbalance in the dataset quantitatively.	2pts
Describe the steps used to mitigate the imbalance and the state of the dataset after applying these steps.	2pts

Your response:

I've identified an imbalance in the dataset, in the distribution of `star_ratings` from 1 to 5. There were significantly more ratings at the higher end (5), with 53K instances, compared to the lower ratings (e.g., 17K for rating 4 and 14K for rating 1).

I've applied stratified k-fold cross-validation to train the model on different splits of the data. This ensured that each fold was a good representative of the overall class distribution, balancing the ratio of the classes. It's hard to definitively state whether the imbalance was completely resolved or just mitigated. Additional resampling techniques such as oversampling the minority classes or undersampling the majority class could be considered if the imbalance still has a substantial impact on model performance.

Question 7 (5pts): Name two features that are not present in the given dataset but may help improve the accuracy of your models. For each feature, briefly describe how you would go about collecting data for it (e.g., by scraping a particular website, or using a particular API / database). Also explain why you think having that feature would improve your models.

Note: Here you can assume that all public websites can be scraped.

Grading rubric:

Rubric item	Score
Name the two proposed features that are not present in the dataset.	1pt
For each of the two features, name a specific data source (e.g., Amazon listing page) that may contain the data for it.	2pts
Explain how the proposed features would improve your models.	2pts

Your response:

1. Customer Demographics Information:

How: Demographics information such as age, gender, location, and income level could be collected through user surveys or by integrating with third-party APIs that have access to this information (with user consent). Why: Understanding the demographic information of the customers might help in building more personalized models. For instance, preferences and buying behaviors could be different across different age groups or locations, and including these variables could allow the model to discern patterns that are more nuanced and tailored to specific customer segments.

2. Historical Purchase Behavior:

How: This information could be obtained from Amazon's internal purchase history database or by tracking users' interactions with the site over time (with proper permissions). If you don't have direct access, you may also use a web scraping tool to collect data on similar products bought together or user purchasing patterns from public profiles. Why: Historical purchase data would allow the model to understand the customer's previous interactions, preferences, and buying habits. By incorporating this context, the model can make more informed predictions and recommendations. For example, understanding that a customer often buys eco-friendly products might influence the model's recommendations or sentiment analysis of a review for an eco-friendly product.

Question 8 (2pts): Which parts in the entire pipeline took the most development time? If you were to do a similar assignment in the future, how would you reduce the time taken for these parts of the solution?

Grading rubric:

Rubric item	Score
Describe the time-consuming portions of the pipeline and the reasons for them being time-consuming.	1pt
Describe methods to reduce time in the future assignments.	1pt

Your response:

The model training phase took the most development time in this pipeline. Training a machine learning model often involves many iterations to find the optimal hyperparameters, data preprocessing, feature engineering, and so on. These processes can be very time-consuming, especially if the parameter tuning is done manually or randomly.

To reduce the time taken for the model training part of the solution in the future, I would implement Grid Search Method which will find the best parameters can automate and streamline the process of tuning hyperparameters. This method systematically searches through a grid of parameter values and allows you to find the optimal combination without the need for random or manual tweaking.

Question 9 (4pts): Are there differences in the language used in reviews with high ratings and those with low ratings? You can decide your own threshold for high vs. low and compare the review texts between the two groups. For example, did reviews with high ratings tend to contain more positive words, or have greater lengths?

Grading rubric:

Rubric item	Score
Clearly define “low” and ‘high” reviews.	1pt

Rubric item	Score
Clearly characterize one difference between the language used in low and high reviews in layman’s terms.	2pts
Provide some numerical evidence to back up your observation (e.g., count, mean, std). No detailed tables or plots are required.	1pt

Your response:

High reviews are ones that have ratings of 4 stars or more. **Low reviews** are the ones that have ratings of 2 stars or less. The number of words varies for the high reviews and low reviews.

I used `len(review.split())` to count the number of words in each review in the column of `review_body`. Word count is slightly higher for the low rating reviews, which is 68 and lower for high rating reviews, which is 63. The slightly