Title: Predicting Amazon product rating

Introduction

This project focuses on analyzing Amazon product reviews using machine learning techniques. The primary objective is to predict the rating score (1 to 5) given by users based on the review text and other features. By leveraging the power of PySpark MLlib and AWS services.

Data Source

The dataset chosen for this project is a collection of consumer reviews from Amazon, sourced from Kaggle. The dataset contains 568,454 records with 10 attributes, totaling 300 MB in size. The dataset was selected due to its rich textual content, diverse range of products, and the potential to provide meaningful insights into customer preferences and opinions.

Attributes

- Id: Unique identifier for each review.
- ProductId: Identifier for the reviewed product.
- UserId: Identifier for the user who wrote the review.
- ProfileName: Name of the user profile.
- HelpfulnessNumerator: Number of users who found the review helpful.
- HelpfulnessDenominator: Number of users who indicated whether the review was helpful.
- Score: Rating given by the user (1 to 5).
- Time: Timestamp for when the review was posted.
- Summary: Brief summary or title of the review.
- Text: Full text of the review.

Source: https://www.kaggle.com/datasets/arhamrumi/amazon-product-reviews/data

Data Preprocessing

The data preprocessing steps are crucial for ensuring the dataset is clean and suitable for machine learning modeling. The following steps were undertaken:

1. **Reading Data**:
   - The dataset was read from an Amazon S3 bucket into a Spark DataFrame.

2. Cleaning Data:
   - Removed non-numeric values in columns that should contain numeric data.
   - Dropped rows with missing values to ensure data quality.
3. Normalization and Feature Engineering:
   - Normalized text columns to lowercase and removed special characters.
   - Created additional features such as HelpfulnessRatio and TextLength to enhance the model's predictive power.

Storing Transformed Data in S3

The cleaned and transformed data was saved back to an Amazon S3 bucket for further processing and analysis.

Creating Table in Athena

An external table was created in Amazon Athena to allow for efficient querying of the transformed data stored in S3.

Machine Learning Model Development

The machine learning pipeline was designed to tokenize the text data, remove stop words, convert text to numerical features using TF-IDF, and then apply a logistic regression classifier to predict the review scores.

1. **Pipeline Stages**:
    - **Tokenizer**: Splits the text into words.
    - **StopWordsRemover**: Removes common words that do not contribute much to the sentiment.
    - **CountVectorizer**: Converts the text into a vector of term frequencies.
    - **IDF**: Transforms the term frequencies into a TF-IDF representation.
    - **Logistic Regression**: Classifies the reviews into one of the five score categories.

The trained logistic regression model achieved high accuracy on the training data (99.63%), indicating a good fit. However, its performance on unseen test data was lower (71.01%), suggesting some overfitting. Precision, recall, and F1-score metrics also reflect reasonable model performance in classifying Amazon product review scores based on textual content.

Challenges

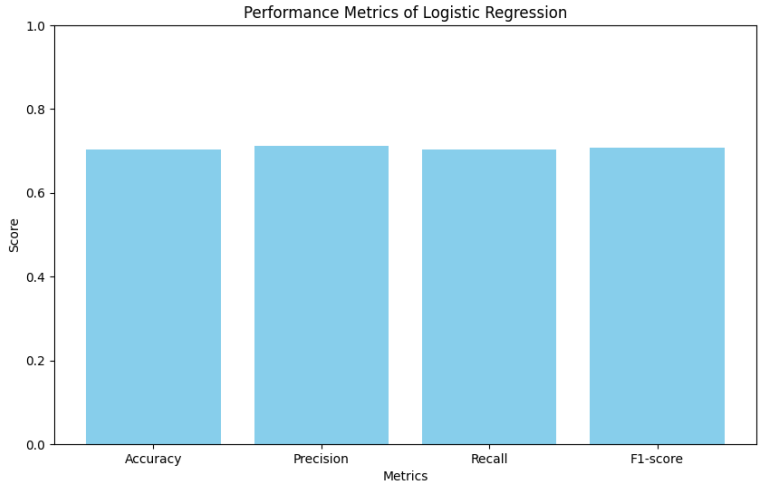
1. **High Dimensionality**:
    - Text data results in high-dimensional feature space.
    - **Solution**: Used TF-IDF to reduce dimensionality and focus on important words.
2. **Computation Time**:
    - Processing a large dataset can be time-consuming.
    - **Solution**: Leveraged distributed computing with PySpark to parallelize tasks and reduce computation time.

Benefits of Distributed Computing

- **Scalability**: Ability to handle large datasets efficiently.
- **Speed**: Faster processing times due to parallel execution.
- **Flexibility**: Easier to implement complex machine learning pipelines.

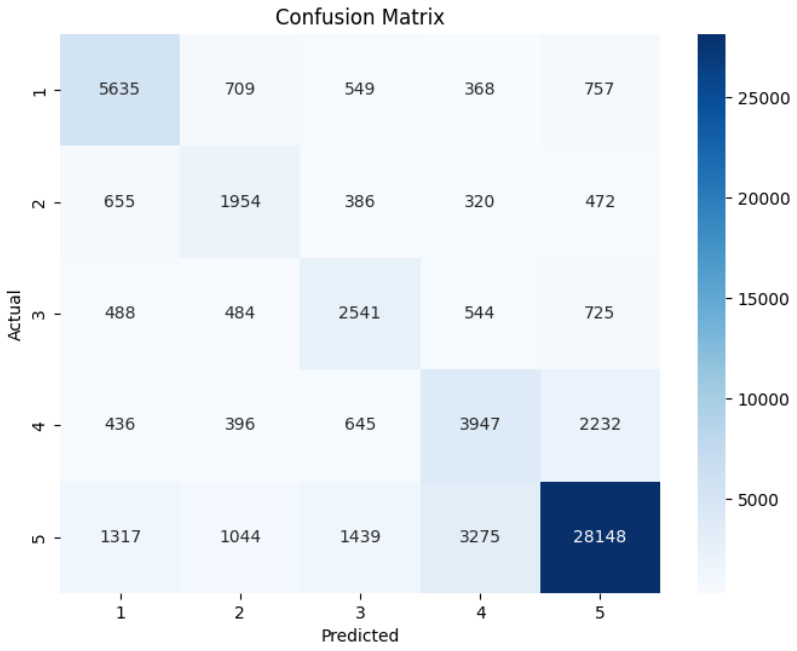
Visualizations

Performance Metrics



Confusion Matrix

The confusion matrix revealed that the model was particularly good at predicting reviews with a score of 5, but it struggled with lower scores.



Architecture Diagram

Diagram 1:

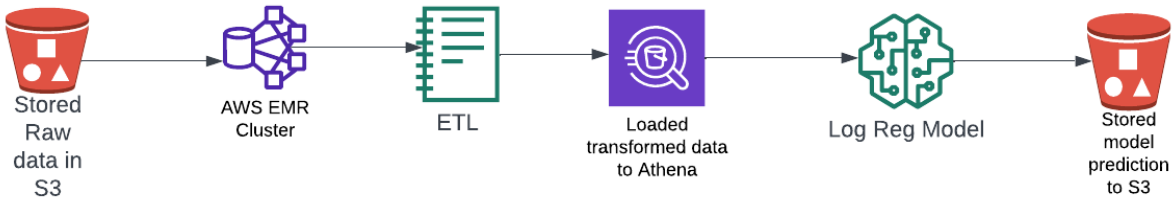
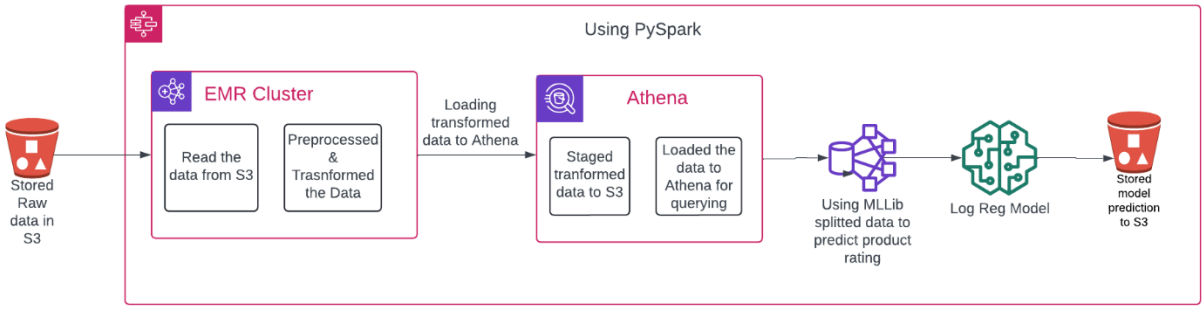


Diagram 2:



Conclusion

In conclusion, this project successfully processed and transformed a vast Amazon review dataset using PySpark, ensuring data quality and consistency through rigorous preprocessing steps. The cleaned data was efficiently staged in Amazon S3 and structured into an Athena table.

While the logistic regression model demonstrated strong predictive capabilities for certain review scores, it showed limitations in distinguishing between lower ratings (Classes 1 and 2). This suggests potential areas for model improvement, such as incorporating additional features or exploring different algorithms better suited for handling imbalanced classes.

Furthermore, the insights gained from this analysis can assist Amazon in understanding customer sentiment and improving product offerings based on customer feedback.

Overall, the project highlights the application of machine learning techniques using pySPark MLlib in analyzing large-scale review datasets, offering valuable insights into customer preferences and opinions.

Future Work

- **Improving Model Performance**: Experimenting with more advanced machine learning models such as Gradient Boosting or Neural Networks.
- **Feature Engineering**: Including additional features like user profile information and product metadata.