Harnessing the Power of Big Data: A Big Data Analysis of E-commerce Reviews using PySpark

Kartik Gupta[1]**, Shreya Gupta**[2]**, and Varad Pimpalkhute**[3]

[1,2,3]**College of Information & Computer Sciences, University of Massachusetts Amherst**

ABSTRACT

In the contemporary business landscape, online reviews have become a critical factor in shaping consumer behavior. Online platforms such as Amazon, Yelp, and TripAdvisor host an overwhelming number of customer reviews daily, providing invaluable insights into customer satisfaction levels. Customer reviews can be broadly categorized as Positive or Negative and are essential in understanding the general sentiment towards a product or service. With the vast amount of feedback available online, manual review analysis is impractical, necessitating an automated system for sentiment classification of customer reviews. This project aims to develop such a system using PySpark, MongoDB, Natural Language Processing (NLP) techniques. The proposed system will enable businesses to gain valuable insights into customer satisfaction levels, thereby allowing them to make informed decisions regarding product/service enhancements and marketing strategies. The study is expected to contribute to the existing body of knowledge on sentiment analysis and provide insights into the feasibility of using PySpark and MongoDB for natural language processing tasks.

Keywords: PySpark, SparkML, Machine Learning, Big Data

1 INTRODUCTION

In today's highly competitive business environment, understanding customer feedback and preferences is vital for the success of any organization. The proliferation of online platforms has made it easier for customers to express their opinions and feedback, making customer reviews a valuable source of information for businesses. However, with the sheer volume of feedback available online, manually analyzing and classifying customer reviews is an overwhelming task, necessitating the need for automated systems for sentiment analysis.

Sentiment analysis, which involves the automatic classification of text into positive, negative, or neutral categories, has emerged as a popular technique to analyze customer feedback. While several studies have been conducted on sentiment analysis, most of them have focused on analyzing social media data, such as tweets and Facebook posts. There is a paucity of research on the sentiment analysis of customer reviews from e-commerce platforms such as Amazon, Flipkart, etc.

The ability to automatically analyze customer reviews and classify them into positive, negative or neutral categories can provide valuable insights into customer preferences and satisfaction levels. It can help businesses identify common customer pain points and address them, leading to improved customer satisfaction and loyalty. Furthermore, it can help identify reviews where the sentiment of the feedback mismatches the provided rating, which can be flagged for further observation.

Therefore, this project aims to develop a sentiment classification system for customer reviews using PySpark, MongoDB, natural language processing, and machine learning techniques. The proposed system will enable businesses to gain a better understanding of customer opinions and preferences, leading to improved decision-making regarding product/service enhancements and marketing strategies. Additionally, the study will contribute to the existing body of knowledge on sentiment analysis and provide insights into the feasibility of using PySpark and MongoDB for natural language processing tasks.

At a high level, the pipeline for sentiment classification of customer reviews will involve data collection, pre-processing, feature extraction, model training, and sentiment classification stages. Customer reviews will be collected and stored in a MongoDB database. Pre-processing will involve text normalization, stop word removal, and tokenization. Feature extraction will be performed using techniques such as Bag of Words, TF-IDF, and Word2Vec. Machine learning models such as Logistic Regression, Naive Bayes, and SVM will be trained, evaluated, and the best performing model will be selected. The selected model will be used for sentiment classification, and the results will be stored in the MongoDB database. In the subsequent sections, we will discuss about the methodology, as well as the proposed timeline for the project.

2 METHODOLOGY

This project aims to develop a sentiment classification system for customer reviews using PySpark, MongoDB, NLP techniques. We will use the Amazon Review Dataset [1] to train and evaluate our model. The dataset comprises millions of customer reviews from various categories such as books, electronics, and home appliances. A brief description of the data fields in the dataset is described in Table 1.

reviewerID	ID of the reviewer	A2SUAM1J3GNN3B
asin	ID of the product	0000013714
reviewerName	Name of the reviewer	J. McDonald
helpful	Helpfulness rating of the review	[2, 3]
reviewText	Free form long feedback	I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think the book was published for singing from more than playing from. Great purchase though!
overall	Numerical rating from 5	5.0
summary	Free form short feedback	Heavenly Highway Hymns
unixReviewTime	Time of review in unix time	1252800000
reviewTime	Time of review	09 13, 2009

Table 1. Meta-fields of Amazon Review Dataset.

To ensure that our model is robust, we will split the dataset into three categories: train, validation, and test. We aim to split the data into 70/10/20 ratios between these categories. We will store the dataset in MongoDB to ensure scalability and high performance. MongoDB provides efficient data distribution across different machines using replication and database sharding. Data pre-processing is essential in text analytics to extract meaningful insights. We will perform several pre-processing techniques such as tokenization, stop-word removal, stemming, and lemmatization. Additionally, we will remove special characters, URLs, and numbers from the text data using PySpark.

We will experiment with several machine learning algorithms such as Support Vector Machines (SVM), Random Forest, and Naive Bayes, to develop our sentiment classification model. We will use SparkML, a library that supports custom APIs to develop efficient machine learning models. Spark provides versatility while maintaining efficiency for big data systems, making it ideal for our sentiment analysis pipeline.

Lastly, we would evaluate the performance of the sentiment classification model using metrics such as accuracy, precision, recall, and F1-score. Additionally, we will evaluate the performance of the pipeline in terms of latency and throughput. Our ultimate goal is to develop a robust sentiment classification system that can help businesses gain valuable insights into customer satisfaction levels and preferences.

2.1 Machine Learning Algorithms

The success of any sentiment classification system largely depends on the machine learning algorithm used to build the model. In this research, we will experiment with several machine learning algorithms, including Support Vector Machines (SVM), Random Forest, and Naive Bayes.

Support Vector Machines (SVM) is a popular algorithm used in text classification. It works by finding a hyperplane in a high-dimensional space that separates the different classes in the data. The algorithm tries to maximize the margin between the hyperplane and the nearest data points. SVM is known for its high accuracy and works well in datasets with high dimensionality.

Random Forest is an ensemble learning algorithm that uses decision trees to build the model. It works by creating multiple decision trees and using a voting mechanism to classify the data. Each decision tree is trained on a different subset of the data, which helps to reduce overfitting. Random Forest works well in datasets with both categorical and numerical data and has good performance in both small and large datasets.

Naive Bayes is a probabilistic algorithm that is commonly used in text classification. It works by assuming that the features are independent of each other, given the class. Naive Bayes is simple, fast, and works well in datasets with a large number of features. However, it can be sensitive to irrelevant features and can suffer from the curse of dimensionality in datasets with high dimensionality.

[1] https://cseweb.ucsd.edu/ jmcauley/datasets.html#amazon_reviews

3 DELIVERABLES

The project will be completed in two milestones, which are:

- System setup, downloading and storing the data, data preprocessing;
- Training the data using SparkML, testing the model, analysis, and evaluation.

3.1 First Milestone

To begin with, we will set up the system with all the required software, including PySpark, MongoDB, and SparkML. We will download the Amazon Review Dataset and store it in MongoDB. The dataset contains millions of customer reviews and ratings for various products, which will serve as our training data. We will split the dataset into train, validation, and test categories, with a split ratio of 70:10:20.

The next step will be data preprocessing, where we will use techniques such as tokenization, stop-word removal, stemming, and lemmatization to preprocess the data. We will also remove special characters, URLs, and numbers from the text data. This will be done using PySpark.

3.2 Second Milestone

Once the data is preprocessed, we will train the data using SparkML. We will experiment with several machine learning algorithms, including Support Vector Machines, Random Forest, and Naive Bayes, to develop a sentiment classification model. SparkML provides a range of algorithms for binary and multi-class classification, making it a suitable choice for our project.

After training the model, we will test the model's performance on the test data and analyze the results. We will evaluate the performance of the sentiment classification model using metrics such as accuracy, precision, recall, and F1-score. We will also evaluate the performance of the pipeline in terms of latency and throughput.

3.3 Final Deliverables

In the final step, we will make a final report and video summarizing our findings and the methodology used. The report will include a detailed description of the dataset, the preprocessing techniques used, the machine learning algorithms used, and the results obtained. The video will provide a visual representation of our project, showcasing the pipeline and the results obtained.