Product-Market Fit Analysis
Using Big Data Processing With PySpark

A MINI-PROJECT REPORT

Submitted by

V RISHABH 211701036

in partial fulfilment for the course

CD19651 Mini Project

for the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND DESIGN

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR

THANDALAM

CHENNAI - 602 105

APRIL 2024

RAJALAKSHMI ENGINEERING COLLEGE CHENNAI -

602105

BONAFIDE CERTIFICATE

Certified that this project report **"PRODUCT-MARKET FIT ANALYSIS USING BIG DATA PROCESSING WITH PYSPARK"** is the bonafide work of **"V RISHABH (211701044)"** who carried out the project work for the subject CD19651 – Mini Project under my supervision.

Dr. Uma Maheshwar Rao, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Professor and Head
Department of
Computer Science and Design
Rajalakshmi Engineering College
Rajalakshmi Nagar
Thandalam
Chennai - 602105

Dr. Gunasekar S, M.Tech., Ph.D.,
SUPERVISOR
Assistant Professor (SG)
Department of
Computer Science and Design
Rajalakshmi Engineering
College
Rajalakshmi Nagar
Thandalam
Chennai - 602105

Submitted to Project and Viva Voce Examination for the subject

CD16651 – Mini Project held on _____.

Internal Examiner External Examiner

ABSTRACT

Finding the best product-market fit is essential for new players in the rapidly evolving e-commerce space who want to make wise investment choices. This project aims to find insights that are in line with consumer tastes and market expectations by processing large volumes of user data from a variety of sources using PySpark's big data analytics capabilities. This study ensures data accuracy, completeness, and timeliness by integrating user behaviors, market trends, and competition analysis through the establishment of a strong data processing pipeline. The resulting data platform makes it possible to create dynamic dashboards and visualizations that convert complicated statistics into insights that non-technical stakeholders can act upon. This method helps with strategic decision-making by emphasizing trends, patterns, and correlations, in addition to making analytics interpretation easier.

In the conclusion, the project provides the client with a thorough study of high-demand product categories and customer preferences along with data-driven recommendations for product-market fit. These suggestions aim to direct strategic investment choices in the e-commerce space, guaranteeing congruence with validated consumer patterns and market prospects, thus augmenting the probability of commercial success.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S.Meganathan, B.E, F.I.E.,** our Vice Chairman **Mr. Abhay Shankar Meganathan, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N.Murugesan, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. Uma Maheshwar Rao ,M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Design for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guides, **Dr.S.Gunasekar , M.E., PhD.,** Department of Computer Science and Design, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator **Dr. S. Gunasekar, M.E., Ph.D.,** Department of Computer Science and Engineering for his useful tips during our review to build our project.

V RISHABH (211701044)

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The e-commerce industry is a booming in the current digital era, marked by fierce competition and quickly shifting consumer tastes. Finding the best product-market fit is a crucial issue for emerging companies looking to make a name for themselves in this fast-paced industry. This means that in addition to identifying goods that meet consumer needs, one must anticipate new trends in order to take advantage of unexplored market niches. Advanced solutions that can manage big data with agility and precision are required since traditional analytical approaches are insufficient due to the complexity and volume of data involved in these decisions.

This project uses PySpark, an open-source framework that expands Apache Spark's capabilities, to solve the need for sophisticated data analytics in the e-commerce industry. PySpark is a popular tool for real-time data processing and analysis because of its exceptional efficiency in handling big datasets in clustered situations. The project's goal is to create a complete data processing pipeline that incorporates many data sources, such as user interactions, market trends, and competitor activity, by

utilizing PySpark. Making better decisions is made possible by having a comprehensive understanding of the e-commerce environment, which is made possible by this integration.

The project aims to create actionable insights that may be graphically represented through dashboards and interactive visualizations by methodically gathering, cleaning, and integrating data. Because of the user-friendly design of these tools, even non-technical stakeholders may successfully interpret and interact with the data. Providing accurate advice regarding product-market fit is the project's ultimate goal. It is anticipated that these suggestions will assist potential e-commerce players in making data-driven choices that satisfy present market needs as well as potential future expansions.

This project aims to improve the decision-making process for start-up e-commerce companies by utilizing PySpark to harness the power of big data analytics. Additionally, by showcasing useful applications of sophisticated analytical techniques in real-world settings, it advances the field of data science.

CHAPTER 2

LITERATURE REVIEW

1. ***Extreme Gradient Boosting Model-based Forecasting of Big Data Online Sales Record (Sharma & Patil, 2022):***

This paper explores the use of XGBoost, an advanced machine learning technique, in forecasting sales for online products like books and magazines. By employing PySpark for data analysis, the study demonstrates a more efficient and accurate forecasting model, highlighting the effectiveness of ensemble learning in handling large datasets typical of e-commerce platforms. The results showcase reduced error rates and improved prediction accuracy, making it a valuable reference for predictive analytics in e-commerce.

2. ***Potential customer mining application of smart home products based on LightGBM PU learning and Spark ML algorithm practice (Duan & Wang, 2020):***

This research applies LightGBM and PySpark to mine potential customers for smart home products within a large telecommunications company. By analyzing

big data collected from user interactions, the paper outlines methods for identifying likely buyers and enhancing targeted marketing strategies. The integration of machine learning and big data tools illustrates how similar methodologies can be adapted for analyzing consumer behavior in any e-commerce context.

3. A New Model for Collecting, Storing, and Analyzing Big Data on Customer Feedback in the Tourism Industry (Ho et al., 2023):

This study introduces a novel big data framework tailored to the tourism industry, focusing on customer feedback management using cloud computing and PySpark. The model facilitates scalable data collection, efficient storage, and sophisticated analysis, offering insights into customer preferences and trends. The framework's adaptability makes it relevant for e-commerce sectors looking to harness customer feedback for product development and market positioning.

4. ***Predictive Analytics System for Stock Data: methodology, data pre-processing and case studies (Myint & Hlaing, 2023):***

Focused on financial markets, this paper discusses a big data analytics framework that employs PySpark for predictive modeling of stock prices. It outlines the methodologies of data collection, pre-processing, and correlation analysis between different datasets. The techniques described can be directly applied to e-commerce analytics for predicting market trends and product demands, offering a comparative look at data handling and visualization in stock market analytics.

5. ***Big data application in functional magnetic resonance imaging using apache spark (Sarraf & Ostadhashem, 2016):***

Although primarily focused on medical imaging, this paper's relevance lies in its demonstration of PySpark's capabilities in managing and processing high-dimensional data efficiently. The authors designed a pipeline for analyzing complex imaging data, showcasing how big data tools can enhance processing speed and analytical depth—principles applicable in e-commerce for handling large-scale user data and product images.

6. ***Examining Amazon Customer Reviews using PySpark and AWS: A Data Lake Approach (S et al., 2023):***

This paper presents a practical application of PySpark in analyzing customer reviews on Amazon, utilizing AWS services to create a scalable data lake. The study focuses on constructing ETL (Extract, Transform, Load) pipelines to facilitate the analysis of diverse datasets. It's particularly useful for your project as it provides insights into handling and analyzing customer reviews to inform product-market fit decisions, demonstrating the power of big data tools in extracting actionable business insights from complex datasets.

These papers provide a strong theoretical and practical foundation for this project in product analysis by offering a wide range of applications and approaches that use PySpark for big data analytics.

CHAPTER 3

PRESENT TECHNOLOGY

1. Pandas:

Pandas is a powerful and flexible Python library primarily used for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series, making it an indispensable tool for data science, especially when working with structured data.

Description and Capabilities:

Pandas provides two main data structures: DataFrame, which can be likened to a relational data table, and Series, a single column. The power of pandas lies in its ability to perform complex data manipulations with simple commands. It can handle data from a variety of formats such as CSV, SQL databases, JSON, and Excel. Pandas allows for various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning techniques like handling missing data.

Key functions include powerful indexing to access and modify data, time series-specific functionality, and integrated handling of missing data. It supports

merging and joining of datasets, flexible reshaping and pivoting of data sets, hierarchical labeling of axes, and robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving/loading data from the ultrafast HDF5 format.

Performance and Limitations:

While Pandas is incredibly efficient for small to medium-sized data sets, its performance can diminish when working with very large datasets that don't fit into memory. For datasets in the range of gigabytes, Pandas might show a slowdown, and operations might become computationally expensive. This limitation stems from the fact that it is not inherently designed for distributed data or computations over a cluster.

For larger datasets, operations in Pandas can consume a lot of memory and processing power, sometimes leading to performance bottlenecks. However, for most typical data manipulation tasks in data science and analytics, especially on single machines, Pandas provides a high level of efficiency, ease of use, and a broad array of functionalities.

2. NumPy:

NumPy is another fundamental package for scientific computing with Python. It supports large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Description and Capabilities:

The core feature of NumPy is the powerful N-dimensional array object. It provides sophisticated functions that are crucial for performing efficient mathematical operations on arrays. NumPy arrays also use much less memory than built-in Python sequences, which can be a significant advantage when handling large datasets.

NumPy integrates well with other libraries, providing a solid foundation for implementing more complex algorithms. It offers comprehensive mathematical functions, random number capabilities, Fourier transforms, linear algebra routines, and tools for integrating C/C++ and Fortran code. Moreover, many other advanced data manipulation and machine learning libraries, including Pandas, are built on top of NumPy.

Performance and Limitations:

NumPy is highly optimized for performance. It executes operations in compiled code written in C, which provides a performance boost over traditional Python loops. However, like Pandas, NumPy operates on single-core processes and does not natively support distributed computing, making it less suitable for 'big data' applications that require parallel processing across a cluster.

Despite its performance optimizations, NumPy's requirement that all items in an array be of the same type can also be a limitation for certain types of data-intensive applications. When dealing with very large datasets, memory issues can arise, as operations typically load entire arrays into memory.

3. SciPy:

SciPy builds on NumPy by adding a collection of algorithms and high-level commands for data manipulation and analysis. Designed for scientific and engineering applications, it provides functions for optimization, stats, signal processing, and more.

Description and Capabilities:

SciPy extends the capabilities of NumPy with additional modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers, and other tasks common in science and engineering.

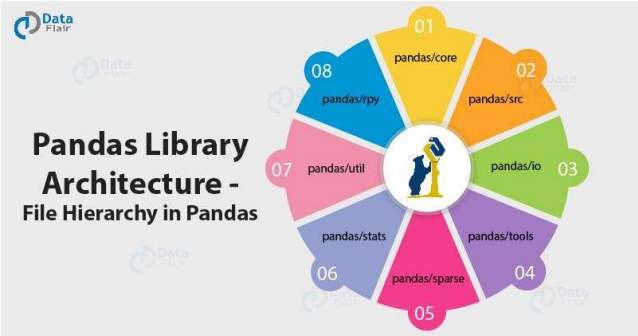
SciPy makes use of NumPy arrays as the basic data structure, and it brings to the table even more versatile operations. It is particularly suited for scientific computations where precision and problem-solving are crucial. It includes modules for minimization, regression, Fourier-transformation, and more.

Performance and Limitations:

SciPy leverages the capabilities of NumPy, ensuring high performance for operations on arrays and matrices. However, its use of dense matrices rather than sparse matrices in many algorithms makes it unsuitable for dealing with extremely large datasets or matrices where most elements are zero.

While it provides powerful tools for scientific calculations, it's generally not

used for distributed computing or handling data that doesn't fit into memory. Therefore, for large-scale data processing tasks typical of big data scenarios, alternatives like distributed computing frameworks become necessary.



<u>Current Technologies and libraries which are quite popular for data processing and analytics</u>

CHAPTER 4

PROPOSED TECHNOLOGY

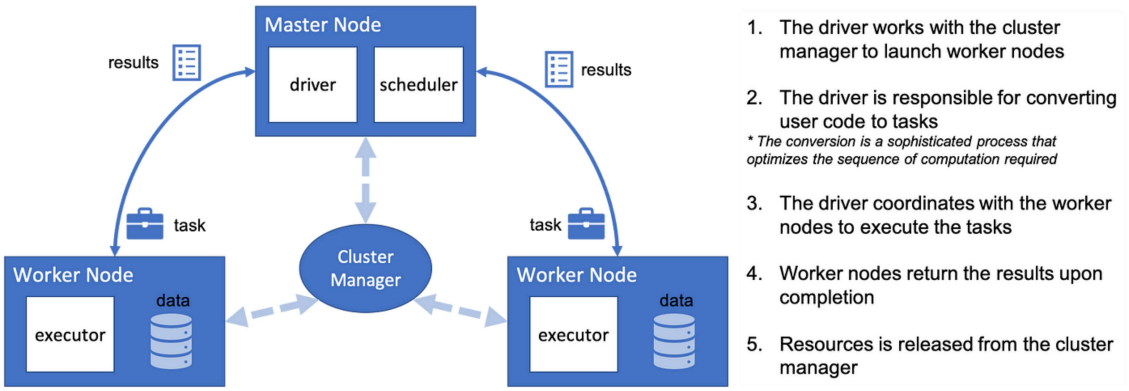
Apache Spark, with its advanced analytics capabilities and efficient data handling, stands out as a particularly effective technology for this purpose of this project of Big Data Analytics. The proposed system leverages Apache Spark, specifically PySpark, to develop a robust analytics platform that is superior to traditional methods such as Pandas in terms of scalability, processing speed, and ability to handle large datasets.

System Overview

Apache Spark is an open-source, distributed computing system that provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. Spark has been noted for its ability to handle petabyte-scale data collections and is significantly faster than traditional Hadoop-based systems and more conventional data analysis tools like Pandas, especially when processing data that fits into memory (Zaharia et al., 2010).

PySpark, the Python API for Spark, extends these capabilities, allowing data scientists to utilize the expressiveness of Python combined with the power of Apache Spark. The proposed system utilizes PySpark for various stages of data processing and analytics, from data ingestion and cleaning to advanced machine

learning.



Apache Spark Architecture which shows its working

Data Collection and Integration

As Sharma and Patil (2022) demonstrate, big data plays a crucial role in generating sales through advanced forecasting models in online platforms. In this system, data from various sources, such as user behavior logs, product catalogs, and competitor prices, are collected in real-time. PySpark facilitates the integration of these diverse datasets into a cohesive data lake, enabling seamless data processing and manipulation.

Data Cleaning and Transformation

The raw data collected often contains inconsistencies, missing values, and outliers. Leveraging PySpark's DataFrame API, the system performs data

cleaning operations such as filling missing values, removing duplicates, and applying transformations necessary for further analysis. This step ensures data quality, which is critical for reliable analytics (Duan & Wang, 2020).

Data Analysis and Machine Learning

Utilizing the MLlib library in PySpark, the system applies machine learning algorithms to uncover patterns and associations within the data. As demonstrated by Myint and Hlaing (2023), predictive analytics can be significantly enhanced using Spark's machine learning capabilities. For instance, the system can predict market trends, customer preferences, and product performance using classification, regression, and clustering techniques. This analysis not only helps in understanding the current market dynamics but also forecasts future trends, thus aiding in strategic decision-making.

Visualization and Dashboarding

The analysis results are visualized using PySpark’s integration with tools like Databricks notebooks or Apache Zeppelin. These dashboards provide real-time insights into key metrics such as sales performance, customer demographics, and market trends. Interactive visualizations enable stakeholders to make informed decisions quickly and effectively.

System Advantages

The proposed PySpark system offers several advantages over traditional data processing methods:

Scalability: PySpark inherently supports distributed computing, which means the system can scale horizontally by adding more nodes to the Spark cluster. This scalability is crucial for handling the growing volume of data in e-commerce (Ho et al., 2023).

Performance: PySpark executes operations in-memory, significantly speeding up data processing tasks compared to disk-based processing systems like Pandas, which is limited by single-node processing (Sarraf & Ostadhashem, 2016).

Fault Tolerance: PySpark uses Resilient Distributed Datasets (RDDs), which help in managing data redundancy and fault tolerance efficiently. This ensures that the system is robust against data loss (Zaharia et al., 2010).

Advanced Analytics: With MLlib, PySpark provides sophisticated analytical tools to perform complex algorithms necessary for deep insights, which are often challenging with Pandas or similar tools due to their limitations in handling large-scale data effectively.

Spark

In [1]:
```
# Importing the necessary libraries
import time
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.ticker as ticker

from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, StringType, DoubleType, IntegerType, DateType
from pyspark.sql.functions import col, sum, countDistinct, count, regexp_replace, split, month, year, size, element_at, stru
from pyspark.sql import functions as F
from pyspark.sql.window import Window
```

In [2]:
```
# Create SparkSession
spark = SparkSession.builder \
    .appName("AmazonElectronics") \
    .getOrCreate()
```

In [3]:
```
# Define the schema
schema = StructType([
    StructField("timestamp", DateType(), True),
    StructField("asin", StringType(), True),
    StructField("brand", StringType(), True),
    StructField("buybox_seller", StringType(), True)
```

Which detailed product categories have the most products?

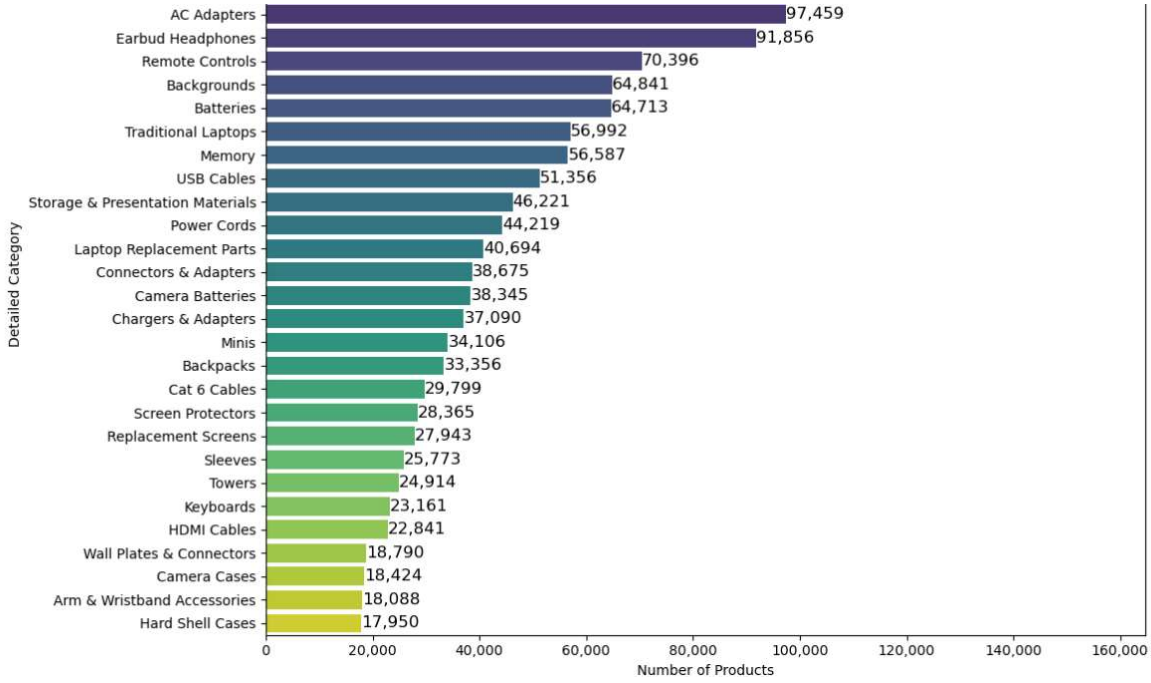
In [33]:
```
# Group by 'detailed_category' and count the number of rows for each category
grouped_df = df.groupBy("detailed_category").agg(count("*").alias("count"))

# Order by the count in descending order and limit to the top 20 categories
top_30_df = grouped_df.orderBy(col("count").desc()).limit(30)

# Convert the PySpark DataFrame to a Pandas DataFrame for plotting with seaborn
pandas_df = top_30_df.toPandas()

# Create a horizontal bar chart using seaborn
plt.figure(figsize=(12, 8))
ax = sns.barplot(x="count", y="detailed_category", data=pandas_df, palette="viridis")
ax.xaxis.set_major_formatter(ticker.FuncFormatter(lambda x, _: '{:,.0f}'.format(x)))

# Format the data labels with a thousands comma separator and add them to the bars
for index, value in enumerate(pandas_df["count"]):
    ax.text(value, index, f"{value:,.0f}", ha='left', va='center', color='black', fontsize=12)

ax.set_xlabel("Number of Products")
ax.set_ylabel("Detailed Category")
ax.set_title("Top 30 detailed categories with the most products")

plt.tight_layout()
plt.show()
```

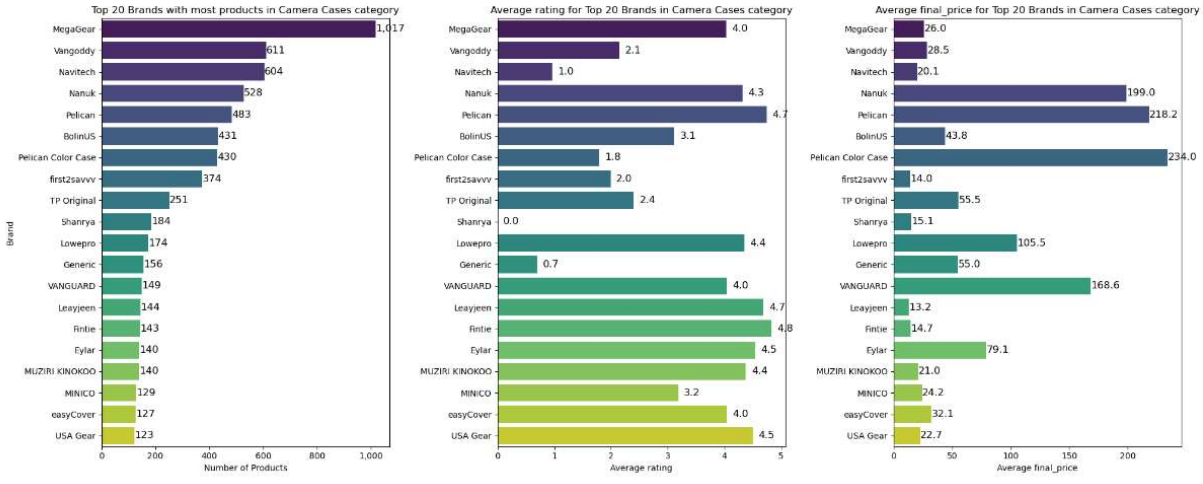
Codebase Screenshots

In conclusion, leveraging Apache Spark and PySpark for e-commerce data analytics offers an advanced, scalable, and efficient solution for analyzing big data. This system not only addresses the limitations of traditional data processing tools but also enhances the analytical capabilities, enabling businesses to achieve a precise understanding of product-market fit in a competitive landscape. Through continuous refinement and utilization of Spark's capabilities, e-commerce platforms can drive innovation and maintain a competitive edge in the market.

CHAPTER 5

OUTPUT & RESULTS :





CHAPTER 6

CONCLUSION:

The transition to Apache Spark, specifically leveraging PySpark, for data processing and analytics in the e-commerce sector represents a significant technological advancement over traditional methods like Pandas. This proposed system promises to address several crucial challenges faced by e-commerce businesses today, including the need to process vast amounts of data rapidly, the requirement for scalable and fault-tolerant infrastructure, and the imperative for sophisticated analytical tools that can drive strategic decision-making.

Firstly, the inherent scalability of PySpark, which allows for horizontal scaling by adding more nodes to the cluster, makes it exceptionally well-suited to the dynamic and data-intensive nature of e-commerce operations. Unlike single-machine based systems such as Pandas, which often struggle with large datasets, PySpark facilitates distributed data processing that effectively handles the increasing volume and variety of data generated by online consumer interactions and transactions. This scalability ensures that as the business grows and data accumulates, the system can expand seamlessly without performance degradation.

Secondly, PySpark enhances performance through in-memory data processing,

offering a substantial speed advantage over disk-based processing systems. This capability is crucial for performing real-time analytics, which enables businesses to make quicker, data-informed decisions that can significantly impact sales and customer satisfaction. For example, real-time insights into consumer behavior and market trends can help businesses adapt their strategies promptly, thus staying competitive in a fast-paced market.

Furthermore, PySpark's fault tolerance, provided by Resilient Distributed Datasets (RDDs), ensures data integrity and system reliability, which are vital in a commercial environment where data loss can lead to financial implications and damage to customer trust. This robustness enhances the overall security and stability of the data processing infrastructure, making it dependable for critical business operations.

Additionally, the advanced analytical capabilities of PySpark, supported by MLlib, offer sophisticated tools that go beyond basic analytics to include predictive modeling and machine learning. These tools allow businesses to not only understand current market dynamics but also to forecast future trends and consumer behaviors with a high degree of accuracy. Such predictive insights are invaluable for optimizing product offerings, personalizing marketing efforts, and improving inventory management, all of which contribute to better product-market fit and business success.

In conclusion, adopting Apache Spark and PySpark for data processing and analytics in e-commerce not only mitigates the limitations posed by traditional data processing tools but also significantly enhances analytical capabilities, scalability, and system performance. This transition supports e-commerce platforms in navigating the complexities of modern retail markets, driving innovation, and maintaining a competitive edge. The proposed system, therefore, is not merely an upgrade of technological infrastructure but a strategic enhancement that aligns with the evolving needs of the digital commerce landscape, ultimately enabling businesses to leverage data as a strategic asset.

REFERENCES:

[1] Extreme Gradient Boosting Model-based Forecasting of Big Data Online Sales Record *(Sharma & Patil, 2022).*

This paper applies the XGBoost model using PySpark to forecast online product sales, emphasizing its accuracy and efficiency in data analysis

[2] Potential customer mining application of smart home products based on LightGBM PU learning and Spark ML algorithm practice*(Duan & Wang, 2020).*

Discusses big data applications in customer mining using PySpark and machine learning algorithms, focusing on precision marketing and customer behavior prediction

[3] A New Model for Collecting, Storing, and Analyzing Big Data on Customer Feedback in the Tourism Industry

Proposes a new model for handling big data using PySpark, focusing on customer feedback in the tourism sector (Ho et al., 2023).

[4] Predictive Analytics System for Stock Data: methodology, data pre-processing and case studies *(Myint & Hlaing, 2023).*

Offers a comprehensive overview of using PySpark for predictive analytics in financial data, relevant for understanding methodologies applicable in e-commerce analytics

[5] Big data application in functional magnetic resonance imaging using apache spark *(Sarraf & Ostadhashem, 2016).*

Although focused on healthcare, this paper highlights techniques in big data processing with PySpark that can be adapted for analyzing e-commerce data

[6] Examining Amazon Customer Reviews using PySpark and AWS: A Data Lake Approach *(S et al., 2023).*

Discusses the use of PySpark in constructing ETL pipelines for analyzing customer reviews, directly applicable to e-commerce product-market fit analysis

[7] How I Work With Millions of Rows of Data (Mo Chen, 2023)

Discusses leveraging PySpark for Big Data Processing Analytics in depth.