Project Technical Documentation

Capstone Project – Big Data Analytics (24MBMB03)

Author: Abhishek Gantana

Title: "Advanced Sales Forecasting: A Machine learning approach to Market Integrated predictions"

Repository URL: https://github.com/AbhishekGantana/Final_Capstone_project_bigdata_24MBMB03_Abhishekgantana_MBABA.git

1. Introduction

This capstone project demonstrates the application of big data analytics methods to derive insights from a large volume of data. The solution follows the full lifecycle: data acquisition, cleaning, transformation, storage in a scalable architecture, analytical processing, and visualization of results. The project, titled 'MarketMind Analytics', is an AI-driven retail forecasting and sentiment intelligence platform built using Big Data Analytics on Databricks.

2. Objectives

- To handle large-scale data ingestion and processing using big data technologies.
- To perform data cleaning, transformation, and enrichment to prepare a usable dataset.
- To apply analytical techniques (descriptive, diagnostic, predictive, and prescriptive) to extract meaningful insights.
- To create visualizations and dashboards to communicate findings to stakeholders.
- To demonstrate proficiency in big-data tools and analytics workflows, aligning with MBA analytics/operations competencies.

3. Architecture & Technologies

Technologies Used:

- Programming/Scripting: Python (Jupyter Notebook, Databricks)
- Big Data Platform: Apache Spark (Databricks Community Edition)
- Data Storage: Databricks File System (DBFS)
- Data Processing: Spark DataFrames, SQL, Pandas, NumPy

- Visualization: Plotly, Seaborn, Prophet
- Automation: Databricks Jobs (for sequential notebook execution)

Architecture Overview:
1. Ingestion Layer – Raw data ingested from CSV sources.
2. Processing Layer – Data cleaning, sentiment scoring, and feature engineering.
3. Modeling Layer – Random Forest for predictive modeling.
4. Forecasting Layer – Prophet for 30-day trend predictions.
5. Visualization Layer – Interactive dashboard generation via Databricks.

4. Dataset & Data Sources

The dataset used is a synthetic retail market dataset integrating sales, marketing, and sentiment variables.
- Source Format: CSV
- Size: ~50 MB (post-cleaning)
- Schema: date, region, product_category, sales_volume, marketing_spend, sentiment_score
- Pre-processing removed missing entries and standardized column data types.

5. Data Ingestion & Pre-processing

Data Ingestion:
- Tools: pandas, os
- Process: CSV data read and validated into a structured DataFrame.

Pre-processing:
- Removal of duplicates and nulls.
- Type casting and renaming for schema consistency.
- Sentiment data merged post-NLP processing.
- Data exported to DBFS (/Workspace/Users/.../outputs).

6. Data Storage & Big Data Platform

Storage Layer:
- Location: Databricks File System (DBFS)
- Format: CSV (Intermediate), Parquet (Optimized)
- Scalability ensured through Databricks distributed compute cluster.
- All intermediate outputs version-controlled in GitHub repository.

7. Analytics & Processing

Processing & Modeling Steps:
- Sentiment Analysis: NLTK's VADER used for text polarity scoring.
- Feature Engineering: Created lag, rolling average, and ratio-based features.
- Model Training: RandomForestRegressor model trained with R^2 score of 0.87.
- Forecasting: Prophet generated 30-day demand projections.

Performance Optimization:

- Spark caching and optimized joins.
- Vectorized Pandas operations and selective filtering.

8. Visualization & Reporting

Dashboards built using Plotly and Prophet’s interactive visualization functions.

- KPIs displayed: forecasted sales, sentiment index, campaign ROI.
- Figures: bar plots, heatmaps, line trends, interactive Prophet forecasts.
- Unified Databricks Dashboard integrated all visuals for presentation.

9. Project Structure (Repository Layout)

```
/
├── data/            # Raw & processed datasets
├── notebooks/         # Data processing and analysis notebooks
│   ├── step1_ingestion.ipynb
│   ├── step2_sentiment.ipynb
│   ├── step3_feature_engineering.ipynb
│   ├── step4_model_training.ipynb
│   ├── step5_forecast_and_intelligence.ipynb
│   ├── step6_dashboard.ipynb
├── scripts/          # Python scripts for automation
├── output/           # Results, visualizations, dashboards
├── docs/            # Documentation and architecture diagrams
└── master_pipeline.ipynb  # Executes all modules sequentially
```

10. How to Run / Deployment Instructions

Prerequisites:

- Python 3.10, Databricks Community Edition account.
- Required Libraries: pandas, numpy, scikit-learn, nltk, prophet, seaborn, plotly.

Steps:

1. Clone repository: git clone <repo-url>
2. Upload files to Databricks Workspace.
3. Run 'master_pipeline.ipynb' to execute all stages sequentially.
4. View results in Databricks Dashboard under /outputs directory.

11. Key Results & Findings

- Model R^2 Score: 0.87
- Predicted Sales Growth: +8.3%
- Sentiment Correlation: r = 0.46 with sales
- Investment Mix: 62% Buy, 28% Hold, 10% Sell
- Unified dashboard enabled business-ready insights.

12. Challenges & Learnings

Challenges:

- Handling null sentiment values during feature merging.
- Cluster latency during heavy visualization renders.

Learnings:

- Optimizing join operations and caching improves Spark performance.
- Visual storytelling is essential for stakeholder communication.

13. Future Work

- Integrate real-time data streaming from APIs.
- Automate incremental data loads.
- Deploy dashboards via Power BI integration.
- Extend forecasting to multi-seasonal and multivariate time series.

14. References

- Apache Spark Documentation
- Prophet by Meta AI – Forecasting Guide
- NLTK VADER Sentiment Analysis Papers
- Databricks User Documentation

15. Appendix

- Glossary of Terms
- Data Dictionary
- Snippets of key code from each notebook
- Architecture Diagram: End-to-End Big Data Pipeline