Stock Price Analysis and Prediction using PySpark and Machine Learning

Project Idea

The project aims to develop models using PySpark to predict stock prices based on historical data. It involves preprocessing the dataset, training various machine learning algorithms, evaluating model performance, and deploying the best-performing model for prediction.

Technology Summary

- Python
- PySpark
- Apache Spark
- Pandas
- Matplotlib
- Jupyter Notebook

Architecture Diagram

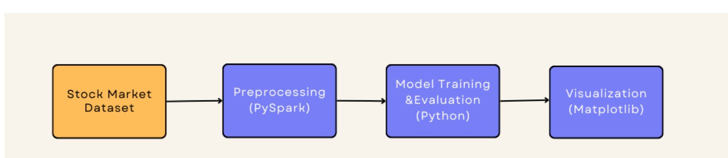


Figure 1: System Architecture

Architecture Summary

- **Stock Market Dataset**: Raw dataset containing historical stock market data.
- **Preprocessing (PySpark)**: Data preprocessing step using PySpark to handle missing values, outliers, and feature engineering.
- **Model Training & Evaluation (Python)**: Machine learning models are trained using PySpark MLlib. Model performance is evaluated using metrics like RMSE (Root Mean Squared Error).
- **Visualization (Matplotlib)**: Visualizations are created using Matplotlib to communicate the findings and insights derived from the analysis.

Project Goals

- **Goal 1:** Investigate the historical trends of stock prices
  - Analyze the historical data by plotting the stock prices over time.
  - Use statistical techniques like moving averages to identify trends and fluctuations.
- **Goal 2:** Investigate the historical trends of trading volumes
  - Analyze the historical data by plotting the trading volumes over time.
  - Use statistical techniques like moving averages to identify trends and fluctuations.
- **Goal 3:** Develop predictive models to forecast future stock prices
  - Preprocess the historical data by handling missing values and scaling the features.
  - Engineer relevant features such as moving averages, technical indicators, and lagged variables.
  - Train machine learning models like Linear Regression, Decision Trees, and Random Forests using the historical data.
  - Validate the models using cross-validation techniques such as k-fold cross-validation.
  - Evaluate the models using metrics of Root Mean Squared Error (RMSE).
- **Goal 4:** Develop predictive models to forecast future trading volumes
  - Preprocess the historical data by handling missing values and scaling the features.

  - Engineer relevant features such as moving averages, technical indicators, and lagged variables.
  - Train machine learning models like Linear Regression, Decision Trees, and Random Forests using the historical data.
  - Validate the models using cross-validation techniques such as k-fold cross-validation.
  - Evaluate the models using metrics of Root Mean Squared Error (RMSE).

- **Goal 5:** Identify patterns and correlations in the data for stock price prediction
  - Conduct correlation analysis to identify relationships between stock prices and other variables such as trading volumes, economic indicators, or news sentiment.
  - Explore patterns and correlations using techniques of correlation matrices.

- **Goal 6:** Identify patterns and correlations in the data for trading volume prediction
  - Conduct correlation analysis to identify relationships between trading volumes and other variables such as stock prices, market volatility, or industry news.
  - Explore patterns and correlations using techniques of correlation matrices.

- **Goal 7:** Evaluate the performance of different machine learning algorithms for stock price prediction
  - Train multiple machine learning algorithms like Linear Regression, Decision Trees, Random Forests, and Gradient Boosting Machines using the historical data.
  - Evaluate the performance of each model using metrics of Root Mean Squared Error (RMSE).
  - Compare the performance of different models to determine the most effective algorithm for stock price prediction.

- **Goal 8:** Evaluate the performance of different machine learning algorithms for trading volume prediction
  - Train multiple machine learning algorithms like Linear Regression, Decision Trees, Random Forests, and Gradient Boosting Machines using the historical data.
  - Evaluate the performance of each model using metrics of Root Mean Squared Error (RMSE).
  - Compare the performance of different models to determine the most effective algorithm for trading volume prediction.

Implementation of Project Goals

- **Goal 1:** Investigate the historical trends of stock prices

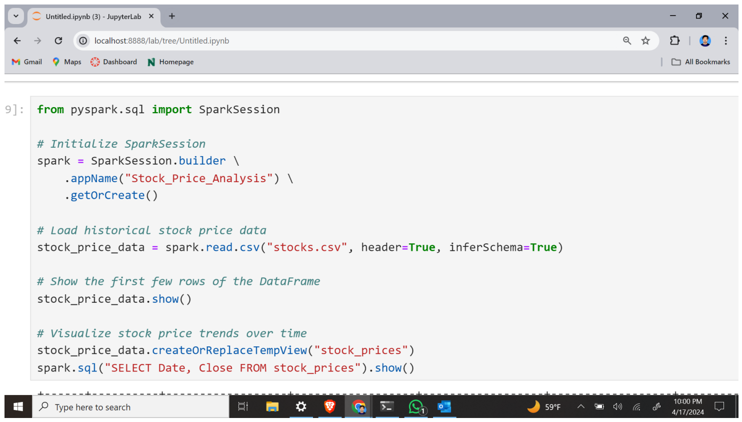


Figure 2: Goal 1 Implementation

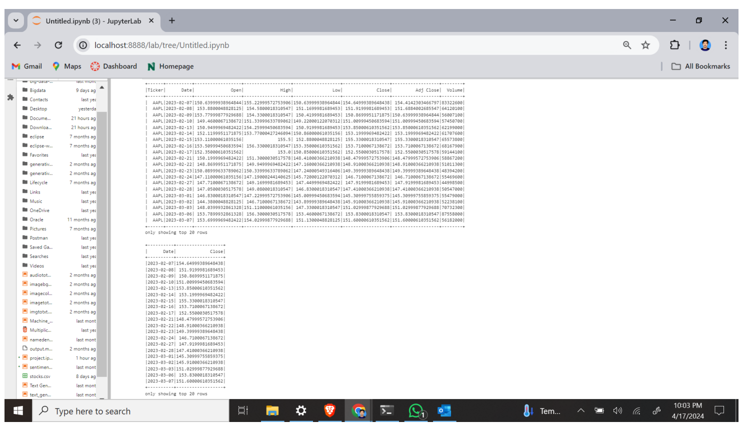


Figure 3: Goal 1 Output

- **Goal 2:** Investigate the historical trends of trading volumes
- **Goal 3:** Develop predictive models to forecast future stock prices
- **Goal 4:** Develop predictive models to forecast future trading volumes
- **Goal 5:** Identify patterns and correlations in the data for stock price prediction
- **Goal 6:** Identify patterns and correlations in the data for trading volume prediction

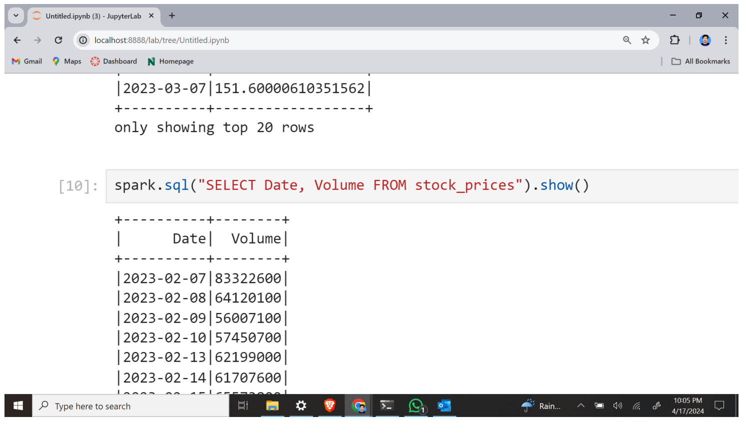


Figure 4: Goal 2 Implementation

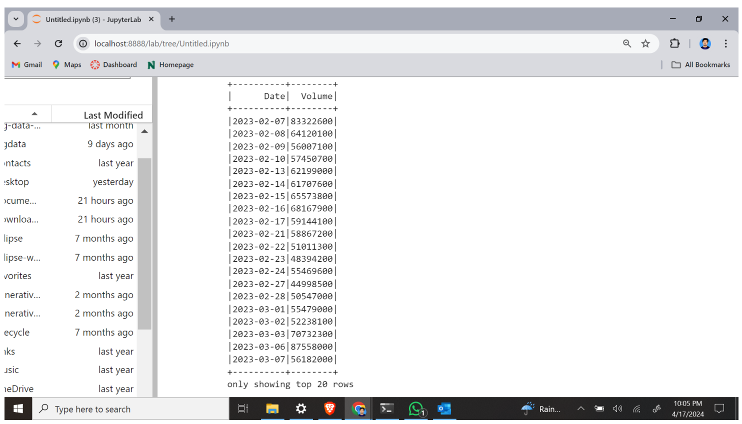


Figure 5: Goal 2 Output

- **Goal 7:** Evaluate the performance of different machine learning algorithms for stock price prediction
- **Goal 8:** Evaluate the performance of different machine learning algorithms for trading volume prediction

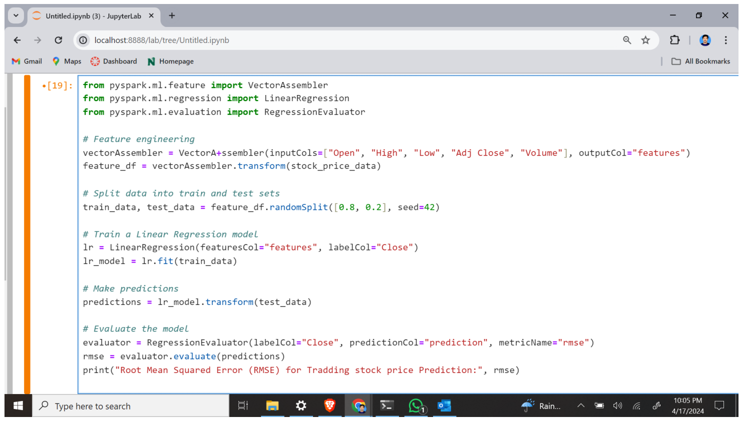


Figure 6: Goal 3 Implementation

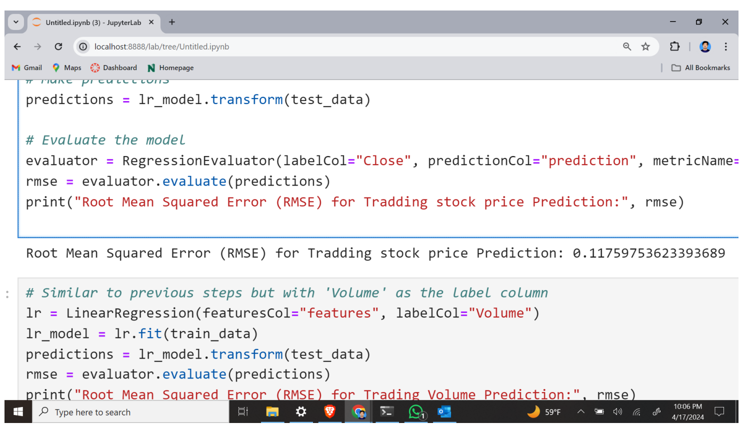


Figure 7: Goal 3 Output

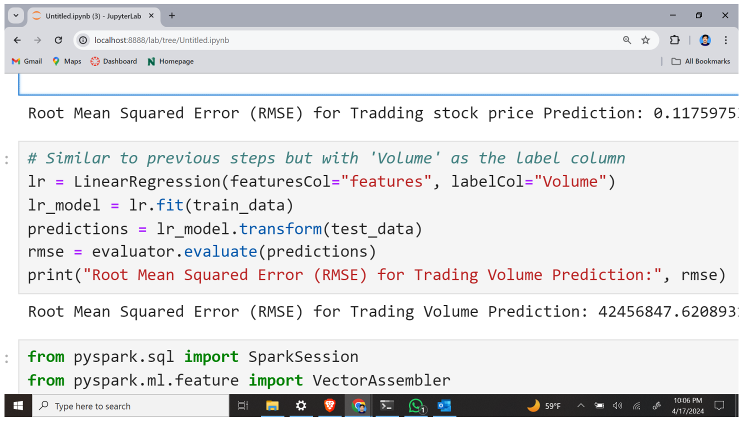


Figure 8: Goal 4 Implementation

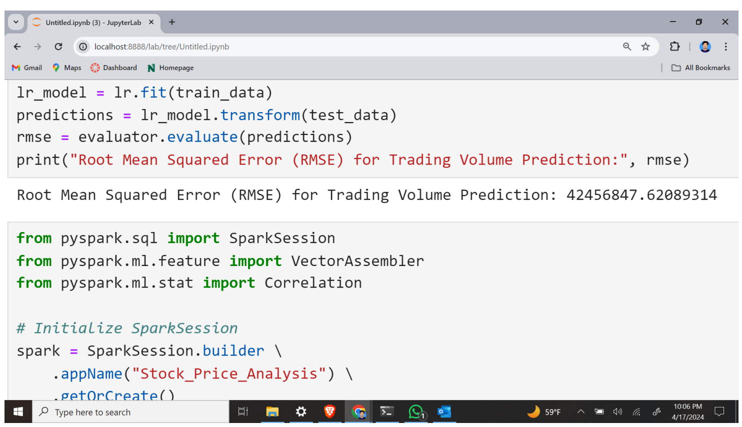


Figure 9: Goal 4 Output

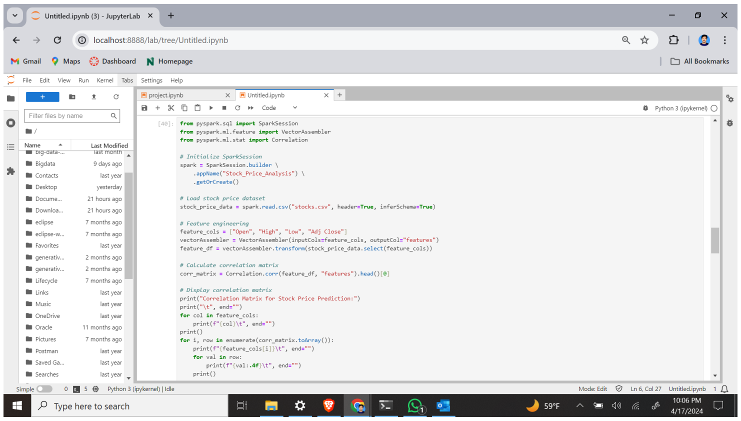


Figure 10: Goal 5 Implementation

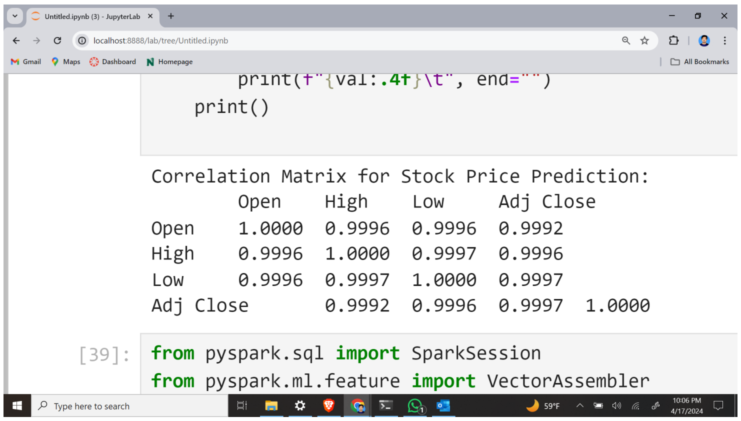


Figure 11: Goal 5 Output

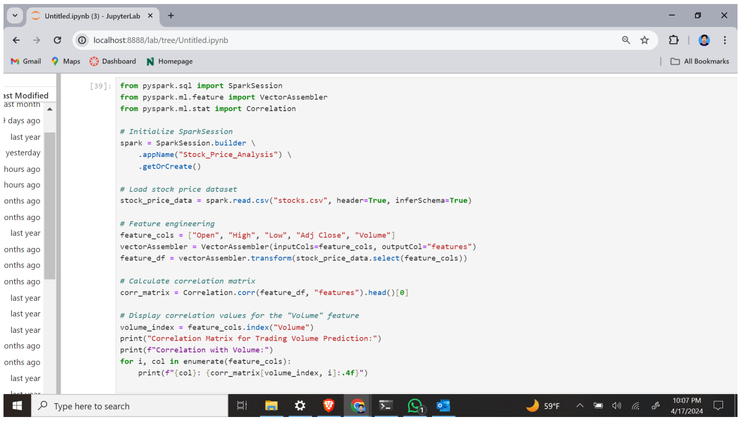


Figure 12: Goal 6 Implementation

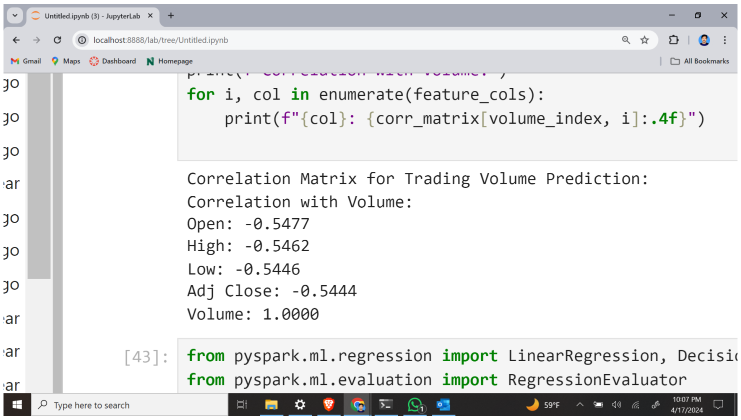
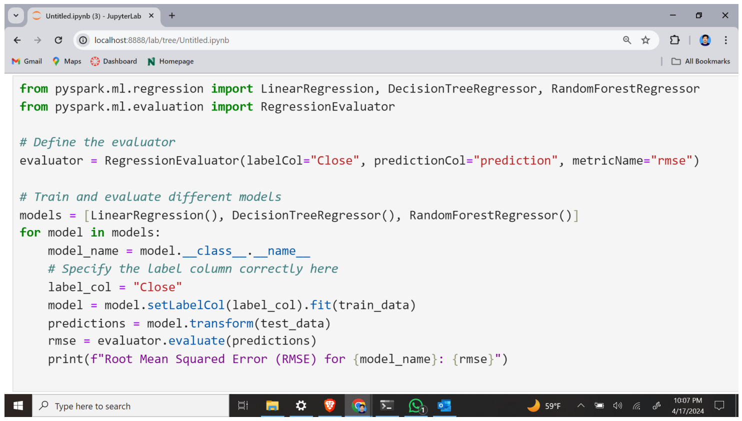


Figure 13: Goal 6 Output

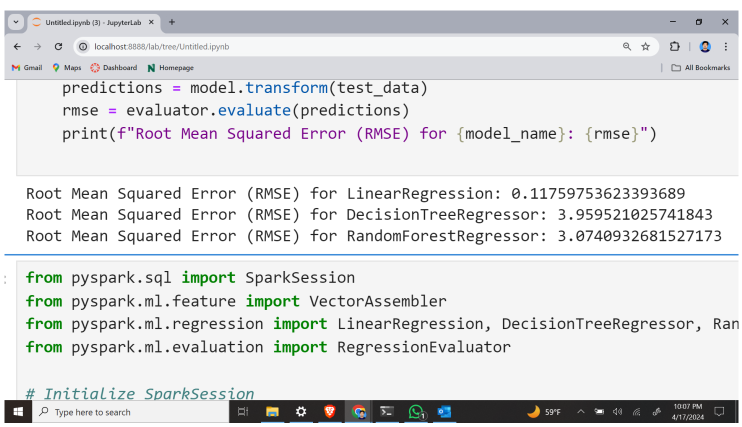


```
from pyspark.ml.regression import LinearRegression, DecisionTreeRegressor, RandomForestRegressor
from pyspark.ml.evaluation import RegressionEvaluator

# Define the evaluator
evaluator = RegressionEvaluator(labelCol="Close", predictionCol="prediction", metricName="rmse")

# Train and evaluate different models
models = [LinearRegression(), DecisionTreeRegressor(), RandomForestRegressor()]
for model in models:
    model_name = model.__class__.__name__
    # Specify the label column correctly here
    label_col = "Close"
    model = model.setLabelCol(label_col).fit(train_data)
    predictions = model.transform(test_data)
    rmse = evaluator.evaluate(predictions)
    print(f"Root Mean Squared Error (RMSE) for {model_name}: {rmse}")
```

Figure 14: Goal 7 Implementation



```
    predictions = model.transform(test_data)
    rmse = evaluator.evaluate(predictions)
    print(f"Root Mean Squared Error (RMSE) for {model_name}: {rmse}")

Root Mean Squared Error (RMSE) for LinearRegression: 0.11759753623393689
Root Mean Squared Error (RMSE) for DecisionTreeRegressor: 3.959521025741843
Root Mean Squared Error (RMSE) for RandomForestRegressor: 3.0740932681527173

from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.regression import LinearRegression, DecisionTreeRegressor, Ran
from pyspark.ml.evaluation import RegressionEvaluator

# Initialize SparkSession
```

Figure 15: Goal 7 Output

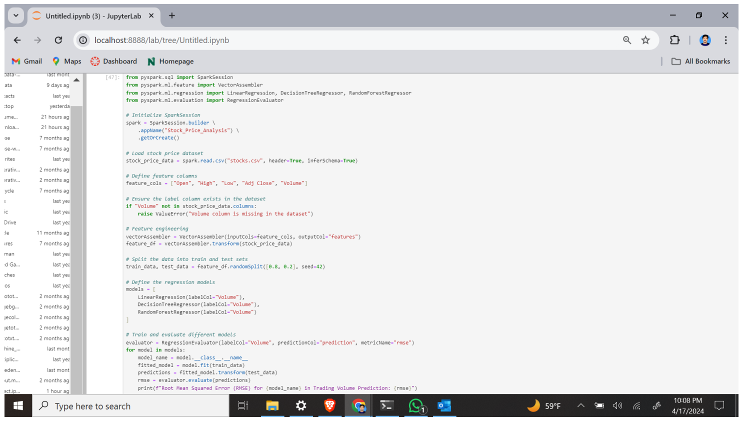


Figure 16: Goal 8 Implementation

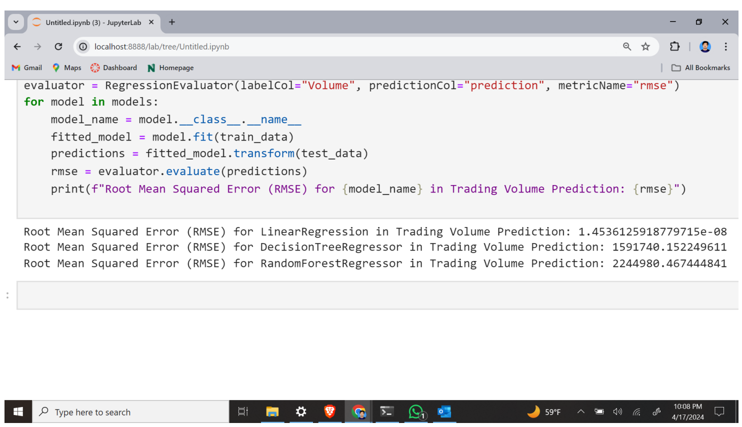


Figure 17: Goal 8 Output

Conclusion

Overall, the project demonstrates the effectiveness of pySpark in analyzing financial data and building predictive models for stock price prediction. The combination of data exploration, machine learning, and visualization techniques provides a holistic approach to understanding and forecasting stock market trends. Moving forward, further enhancements could involve refining the predictive models, integrating additional data sources, and enhancing the interpretability of the results through interactive visualizations.

Historical Trends Analysis

The project successfully analyzed the historical trends of stock prices and trading volumes, providing insights into patterns, fluctuations, and correlations over time. Through statistical analysis we gained a deeper understanding of how stock prices and trading volumes behave in different market conditions.

Predictive Model Development

Several machine learning algorithms, including Linear Regression, Decision Trees, and Random Forests, were trained to forecast future stock prices and trading volumes. By preprocessing the historical data and engineering relevant features, we created robust predictive models capable of making accurate predictions.

Evaluation and Validation

The performance of each predictive model was evaluated using appropriate evaluation metrics of Root Mean Squared Error (RMSE). Cross-validation techniques were employed to validate the models and ensure their reliability in real-world scenarios.

Insights and Interpretations

The project facilitated the identification of patterns and correlations in the data that can inform investment decisions. By conducting correlation analysis we uncovered valuable insights that can help investors make more informed choices in the stock market.

Dataset linkhttps://www.kaggle.com/datasets/amirmotefaker/stock-market-analysis-data
Git Hub repo linkhttps://github.com/supreeth1011/DataDives