# Technical Design Document: Machine Learning Model for Stock Price Prediction using Apache Spark and Random Forest

## I. Introduction

This document outlines the technical design for developing a machine learning model to predict stock prices using the Random Forest algorithm. The project will be implemented in Python using Google Colab for development, Apache Spark for data processing, and Power BI for visualization.

## II. Project Components

### II.i Data Collection and Preparation

Obtain the dataset from Kaggle: "Global Stock Market 2008-Present".

Dataset: https://www.kaggle.com/datasets/pavankrishnanarne/global-stock-market-2008-present

Load the dataset into Google Colab and explore its structure.

Utilize Apache Spark for distributed data processing and feature engineering.

### II..ii Data Preprocessing

Clean the dataset by handling missing values, outliers, and inconsistencies.

Convert the 'Date' column to a datetime format.

Perform feature engineering to extract relevant features like moving averages, technical indicators, and sentiment scores.

### II.iii Model Development

Choose the Random Forest algorithm for stock price prediction due to its ensemble nature and ability to handle non-linearity.

Split the dataset into training and testing sets, maintaining chronological order.

Implement Random Forest regression using Spark's MLlib library.

Train the model on the training set, adjusting hyperparameters as needed.

Evaluate the model's performance using Root Mean Squared Error (RMSE) on the testing set.

### II.iv Model Deployment

Serialize and save the trained Random Forest model using Spark's model-saving capabilities.

Prepare a deployment script to load the model for future predictions.

### II.v Visualization with Power BI

Export the predicted results from the trained model into a format compatible with Power BI (e.g., CSV or Parquet).

Import the results into Power BI for visualization.

Design Power BI reports and dashboards to visualize the historical stock prices, predicted prices, and model performance metrics.

Create interactive visuals like line charts, time series plots, and error distribution plots.

## III. Technical Implementation

### III.i Data Preparation and Preprocessing

Load the dataset using Spark's DataFrame API.

Apply data cleaning and transformation operations using Spark SQL and DataFrame transformations.

Calculate additional features such as moving averages and technical indicators using Spark functions.

### III.ii Model Development

Split the dataset into training and testing sets using Spark's randomSplit method.

Configure the Random Forest regressor with relevant hyperparameters.

Train the Random Forest model using the training dataset.

Evaluate the model's performance using RMSE on the testing dataset.

### III.iii Model Deployment

Serialize and save the trained Random Forest model using the MLlib's save method.

Create a deployment script that loads the saved model and makes predictions on new data.

### III.iv Visualization with Power BI

Export the predicted results from the model into a format that Power BI can read.

Open Power BI and connect to the exported data.

Design visualizations using Power BI's intuitive interface, selecting appropriate chart types and layouts.

Create interactive filters and slicers to allow users to explore the data and predictions.

## IV. Workflow and Collaboration

Use Git for version control to track code changes and collaborate with team members.

Utilize Google Colab's notebook sharing and collaboration features to facilitate joint development and troubleshooting.

Share relevant Jupyter notebooks, scripts, and data files with team members.

## V. Conclusion

By leveraging Apache Spark for data processing, implementing the Random Forest algorithm for prediction, and using Power BI for visualization, this technical design outlines a comprehensive approach to developing a robust machine learning model for stock price prediction. The project aims to provide accurate predictions and meaningful insights to various stakeholders in the financial domain.