Capstone Project Proposal: SPY Price Prediction using Machine Learning

Objective:

The objective of this project is to develop a predictive model to forecast the price movement of SPY (S&P 500 ETF) using historical price data, technical indicators, and macroeconomic factors. The goal is to generate actionable insights that can guide trading strategies and investment decisions, helping investors optimize portfolio allocation.

Project Overview:

SPY is a widely traded ETF that tracks the performance of the S&P 500 index, serving as an excellent proxy for the overall stock market. This project focuses on forecasting the future price or returns of SPY based on historical data, technical indicators, and relevant macroeconomic factors. The aim is to build a model that can predict the price of SPY accurately and provide insights for informed decision-making in trading and investment.

Data Sources:

- **Yahoo Finance API**: For historical SPY price data (daily, weekly, and monthly), including open, high, low, close prices, and trading volume.
- **Business Insider**: For historical news data related to the economy that could influence market sentiment.
- **Additional Sources**: Optional data sources such as Twitter (for sentiment analysis) and news outlets for market-moving headlines (using NLP techniques).

Key Features:

1. **Price Data**:
    - Opening, closing, high, and low prices.
    - Volume of trades.
2. **Technical Indicators**:
    - **Simple Moving Averages (SMA)**: 20-day, 50-day, 200-day moving averages.
    - **Exponential Moving Averages (EMA)**: 20-day, 50-day moving averages.
    - **RSI (Relative Strength Index)**: Indicator of overbought/oversold conditions.
    - **MACD (Moving Average Convergence Divergence)**: For trend-following and momentum.
3. **Macroeconomic Factors**:
    - **GDP Growth Rate**.
    - **Inflation Rates**.
    - **Interest Rates**.
    - **Unemployment Rate**.
4. **Sentiment Analysis** (Optional):
    - **Twitter Sentiment Analysis**: Using an API like Tweepy to gauge market sentiment regarding SPY or the S&P 500.
    - **News Sentiment Analysis**: Using Natural Language Processing (NLP) to analyze market-moving headlines.

Approach:

1. Data Collection and Preprocessing:

- **Collect Historical Data**: Gather SPY price data for at least 5 years, as well as macroeconomic data and sentiment data if applicable.
- **Calculate Technical Indicators**: Use stock data to calculate indicators like SMA, EMA, RSI, and MACD.
- **Data Synchronization**: Align macroeconomic data with SPY data, ensuring consistency in timeframes (e.g., aligning weekly data).
- **Data Cleaning**: Handle missing values and outliers through imputation, removal, or other methods as necessary.

2. Exploratory Data Analysis (EDA):

- **Trend Identification**: Visualize SPY price data to identify trends, seasonality, and volatility.
- **Correlation Analysis**: Examine correlations between SPY prices and technical indicators, economic factors, and sentiment data.
- **Time-Series Decomposition**: Decompose the series to analyze trends, seasonality, and residuals.

3. Model Selection:

- **ARIMA (Auto-Regressive Integrated Moving Average)**: A traditional time-series forecasting method.
- **LSTM (Long Short-Term Memory)**: A deep learning model for time-series forecasting, useful for capturing long-term dependencies.
- **Random Forests or XGBoost**: These models handle large datasets with numerous features, such as technical indicators and macroeconomic data.
- **Linear Regression or SVM (Support Vector Machine)**: To predict price direction or returns based on technical indicators and macroeconomic features.

4. Model Evaluation:

- **Regression Metrics**: Evaluate the models using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2.
- **Classification Metrics**: If predicting price direction, use metrics such as Precision, Recall, and F1-Score.
- **Backtesting**: Simulate trading strategies using historical data to evaluate the profitability of the predictions.

5. Hyperparameter Tuning:

- **Grid Search** or **Random Search**: Use these methods to optimize hyperparameters for models like Random Forests, XGBoost, and LSTM.

6. Model Interpretation:

- **Feature Importance**: For tree-based models, identify the most important features influencing predictions.
- **SHAP or LIME**: For deep learning models, use SHAP values or LIME for interpreting the model's predictions and improving trust in the results.

7. Deployment:

- **Visualization Dashboard**: Create a dashboard (using tools like Streamlit or Flask) to visualize SPY price forecasts and technical indicators over time.
- **Web Service**: Deploy the model as a web service that can be accessed for real-time predictions or integrated into a trading platform if needed.

Deliverables:

- **Code**: Python scripts for data collection, preprocessing, model development, evaluation, and visualization.

- **Report**: A comprehensive report detailing the methodology, results, and performance of the models.
- **Presentation**: A slide deck summarizing the problem, solution, methodology, findings, and the impact of the model on trading strategies.
- **Optional**: A deployed web dashboard or app for visualizing predictions and trading insights.