**Installation and Running Instructions**

* **R Script**

For the data preprocessing and coding of our linear regression models, please refer to the file “Linear Regression.R” in our Github folder <https://github.gatech.edu/MGT-6203-Spring-2024-Canvas/Team-75/tree/main/Code/>

* **Python Scripts / Jupyter Notebooks**

The following scripts are also in our Code folder in the GitHub repository.

1. cleanData.py: This script handles data preprocessing and cleaning. It reads multiple CSV files containing different economic indicators and stock market data, merges them based on the 'DATE' column, renames columns, and performs necessary data transformations. The resulting merged and cleaned dataset is stored in the data variable.

2. LSTMmodel.py: This script implements a Long Short-Term Memory (LSTM) neural network model for predicting SPY prices. It uses the preprocessed data from cleanData.py, splits it into training and testing sets, scales the features using MinMaxScaler, and reshapes the data for LSTM input. The LSTM model architecture is defined, compiled, and trained on the data. The script evaluates the model's performance using various metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared Score, Explained Variance Score, Mean Absolute Percentage Error (MAPE), and Mean Absolute Deviation (MAD). Finally, it plots the actual and predicted SPY prices.

3. spy\_price\_prediction\_models.py: This script trains and evaluates three different machine learning models for SPY price prediction: Decision Tree, Random Forest, and Gradient Boosting. It uses the preprocessed data from cleanData.py, applies polynomial features to the selected features, splits the data into training and testing sets, and performs hyperparameter tuning using GridSearchCV. The script calculates various performance evaluation metrics for each model, such as MSE, R² Score, MAD, MAPE, and Theil's Inequality Coefficient (Theil's U). It also visualizes the feature importances for the Decision Tree model.

4. modelVisualizations.py: This script provides visualizations for the different models trained in LSTMmodel.py and spy\_price\_prediction\_models.py. It includes an actual vs. predicted prices plot for the Decision Tree model, a residual plot for the Random Forest model, a feature importance plot for the Gradient Boosting model, and a prediction vs. actual prices plot for the LSTM model.

5. TimeSeries\_forecasts\_sp500idx.ipynb: Time series analysis notebook. To run the Python notebook with the time series analysis portion of the project, a Python (version 3) installation is required. If you do not have Python installed, you can download it from the official Python website.

Additionally, you will need to install Jupyter Notebook, the platform on which the notebook will run. After Python has been installed. You can install Jupyter using pip, Python's package manager, with the following command:

*- pip install notebook*

Once Jupyter is installed, you can start the notebook server from the command line:

*- jupyter notebook*

To load the notebook, navigate to the directory where the code (.ipynb) file is located and run the jupyter notebook command. Your web browser will open with a list of available notebooks. Click on the notebook you wish to open.

All required libraries are listed in the requirements.txt file located within the code directory. This file lists all the libraries that are required for the project, and they can be installed using the following command:

*- pip install -r requirements.txt*

Usage

1. Ensure that you have the required Python library dependencies installed, including NumPy, Pandas, Matplotlib, Scikit-learn, and Keras.

2. Place the necessary CSV files containing the economic indicators and stock market data in the specified file paths.

3. Run cleanData.py to preprocess and merge the data.

4. Run LSTMmodel.py to train and evaluate the LSTM model for SPY price prediction.

5. Run spy\_price\_prediction\_models.py to train and evaluate the Decision Tree, Random Forest, and Gradient Boosting models for SPY price prediction.

6. Run modelVisualizations.py to generate visualizations for the trained models.

Dependencies

- NumPy, Pandas, Matplotlib, Scikit-learn, Keras, Statsmodels, Darts, Seaborn

**Note**

* Make sure to update the file paths in cleanData.py to match the locations of your CSV files.
* **Time Series Analysis**

For the time series analysis code, the data file should be placed in a "Data" folder contained within the parent folder containing the python notebook file

**Parent Folder (containing all project files project)**

- “Data” Directory

-- Data File **(sp500\_index.csv)**

- “Code” Directory

-- Python Notebook **(TimeSeries\_forecasts\_sp500idx.ipynb)**

**Overview of Directory Structure**

Our GitHhub repository strucure is as follows:

* The “Code” Folder

It contains a “requirements.txt” file and our R scripts, Python scripts, and ipynb files.

* The “Data” Folder

It contains raw data files that we collected. Due to small data file sizes, we upload the data csv files directly to this folder.

* The “Final Report” Folder

It has a copy of our final report, the same that we submitted via Canvas.

* The “Other Resources” Folder

We do not have any files in this folder.

* The “Progress Report” Folder

It has our progress report document.

* The “Project Proposal” Folder

It has our group proposal document.

* The “Visualizations” Folder

It has a Tableau workbook file, Python scripts, an iPython notebook, and visualizations we extracted as part of exploratory data analysis and modeling.

* The “README.docx” File (this file)