Functional Forecast Project Report

Our project is a stock predictor application that takes advantage of hvega and numeric.linearalgebra libraries, and the yahoo-prices api. The user provides the program with a stock ticker, start and end date as well as an optional filePath from which a csv can be read. (formatted as such: cabal run functional-forecast22300 -- MSFT --start-date 01-01-2023 --end-date 12-31-2023 --file FILE_PATH) through the command line. From here, we read in data either from the file path or request it from the api, perform a regression on it and plot our predicted values along with the actual close values. Below is what each file has:

- Main.hs - main module for the program; generates closing prices chart then performs linear regression and creates second chart of predictions
- ApiRequests.hs - function for requesting from yahoo-prices api
    - requestInformation - takes ticker, start and end dates and requests api for requisite information
    - PriceResponse is the api's version of StockPerformance
- ProcessCSVData.hs - reads and parses csv data into StockPerformance object list
    - custom StockPerformance object
    - readFromCSV - reads csv into StockPerformance object list
    - parseDateString - helper for testing, parses DateString into Day object
- TimeSeriesRegression.hs - regression function and utility functions for predicting and creating new features
    - RegressionModel, PolynomialRegression WIP
    - laggedValues - generates lagged values for PriceResponse list
    - linearRegression - performs linear Regressiona and returns model given features, variable of interest and a lag value
    - performLinearRegression - takes list of stock information and returns a model
    - predict - takes a model and set of new features, returning a predicted close value
    - calculateLaggedValuesForNewDay - takes a new day, list of stock information, lag value and a new day and returns the lagged value for the new day
- Plotting.hs - plots data
    - saveChartAsHtml - saves vegalite image to given filepath as html
    - toVegaData - converts PriceResponse object to Value
    - createClosePriceChart - generates chart of close values
    - toVegaData2 - converts PriceResponse object to Value for predictions and close value graph
    - generatePlot - take a list of PriceResponse, a regression model, ticker, today and lag, generate plot of predictions and actual values

Furthermore, below are some screenshots of the output.

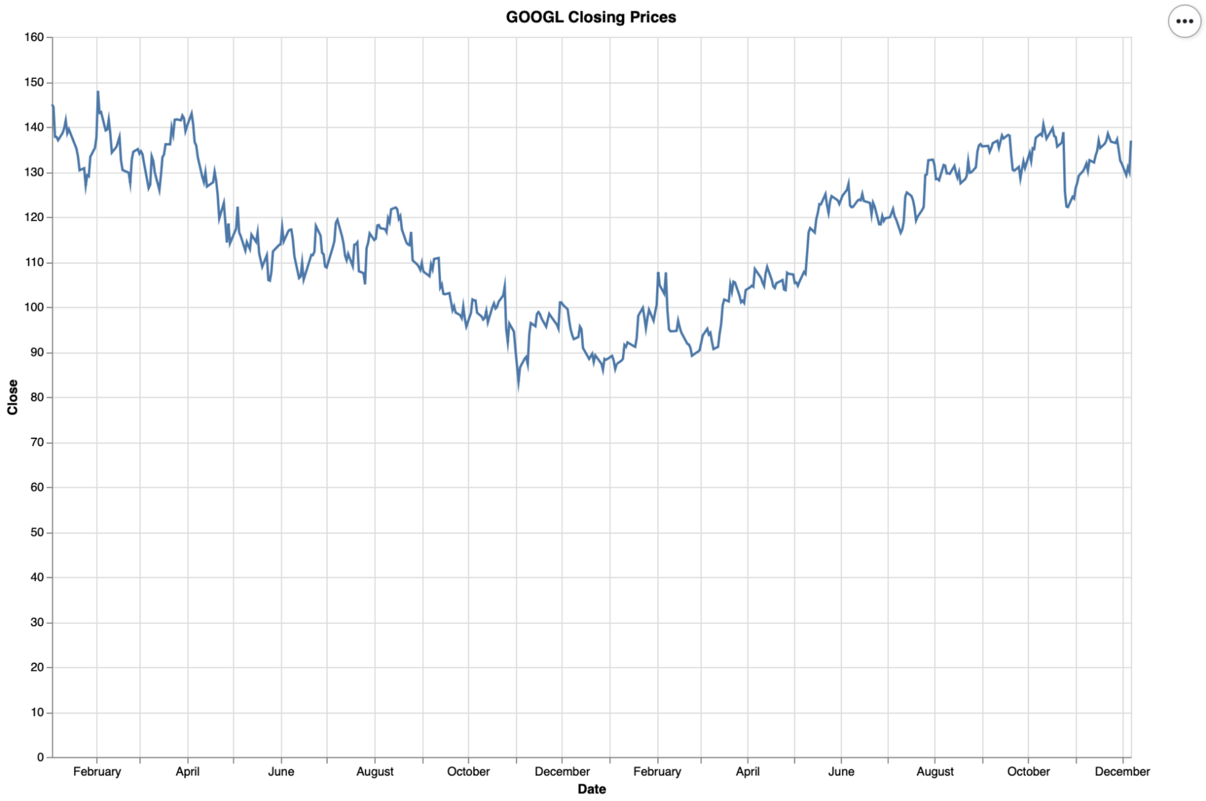


Figure 1: Graph of Closing Values

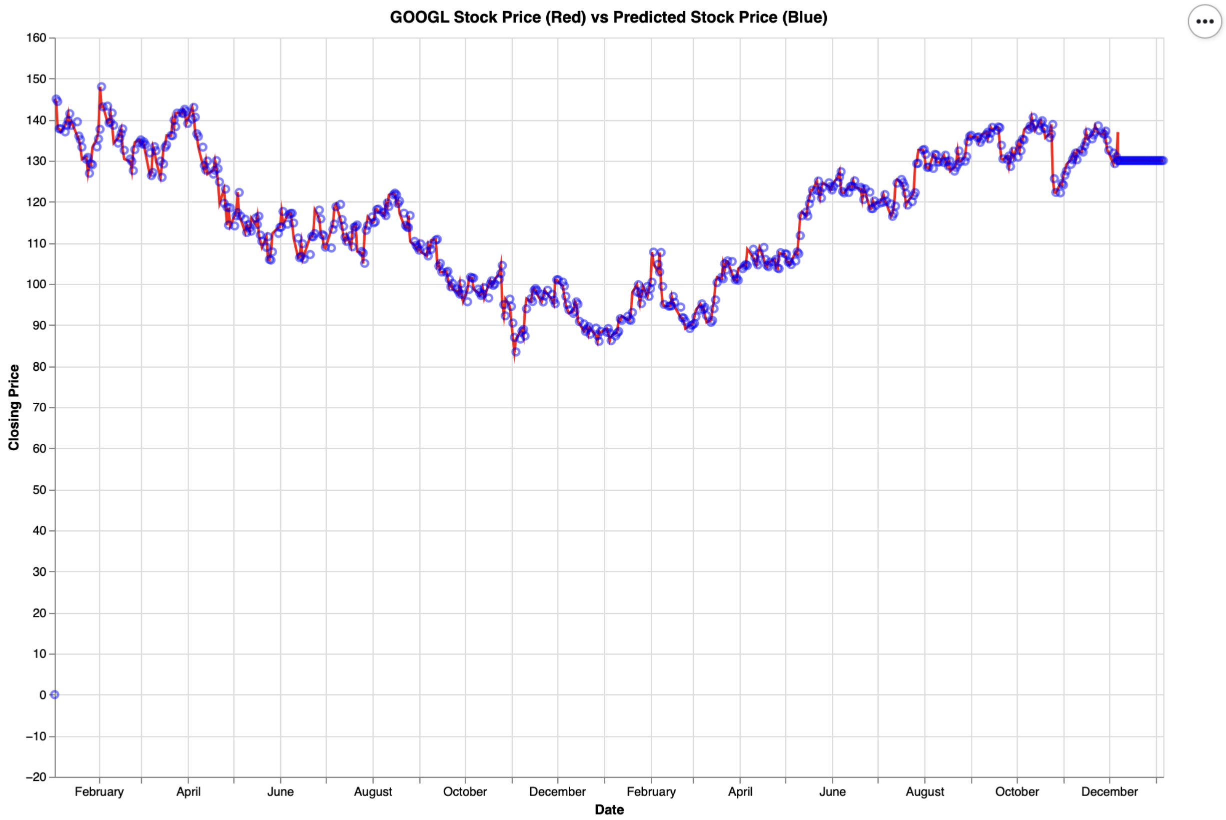
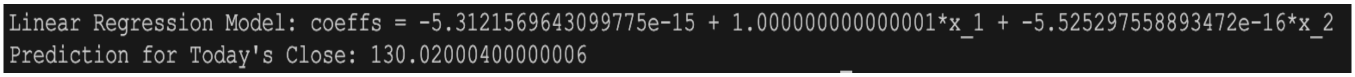


Figure 2: Graph of Actual vs Predicted Values



```
Linear Regression Model: coeffs = -5.3121569643099775e-15 + 1.000000000000001*x_1 + -5.525297558893472e-16*x_2
Prediction for Today's Close: 130.02000400000006
```

Figure 3: Regression output