Best Stock Prediction Method Utilizing Regression

***Note: For the datasets utilizing the time-series module, the user must click the tab and hit the download button in order to download the data onto their local machine.***

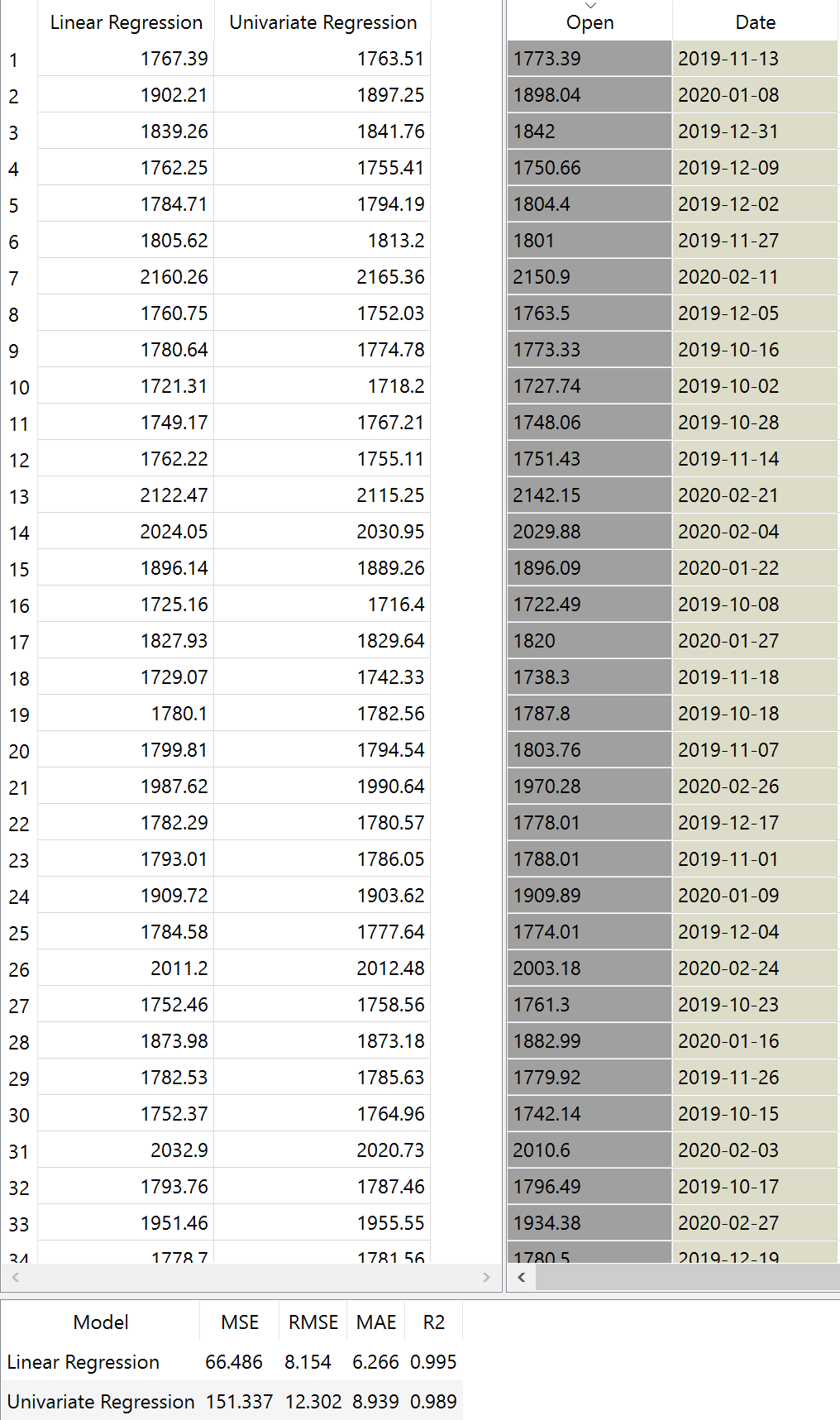
We chose to create a pipeline in Orange that would have the capabilities of predicting stock levels for various companies in the broad field of technology. Both Ritvik and I have been interested in the stock market for a few years. We have had some experience trading stocks and following the market, however, we both agreed that dedicating the time and effort to trade stocks could get very tedious if one was trying to effectively trade in an efficient manner. If someone expects to see growth in their stocks along with some degree of returns, they must spend countless hours analyzing trends in the stock market in order to ensure that they don’t make any hasty, potentially detrimental decisions. With this in mind, we sought to create a pipeline that would do some of the jobs for us so that we could focus our time elsewhere while still taking advantage of the stock game. Thus, we began assembling a stock-market trend predictor. A tool like this would not only help us with our casual-trading careers, but it has the potential to help stock traders everywhere. First, we decided to choose three different prominent innovative companies from which we would gather our stock data. The companies chosen were Apple, Amazon, and Google, because they are some of the world’s top most popular stocks to invest in. We concurred that utilizing the tools of regression would provide us with the most accurate results with regards to predicting future prices. Our project, rather than actually predicting future stock prices, will predict the best regression model for predicting data, which is the first step for applying the actual prediction methods to stock data.

When looking at stock data for a company, the numbers that you most likely are looking at are the companies’ prices per each individual share they sell, as well as the volume or number of shares that are being traded at a given time. On a daily basis, stock prices are known to fluctuate up and down, but this time frame does not provide us with enough information to create a reliable model that can accurately project stock trends over weeks, months, or years. Conversely, we made sure not to include all the public stock data available for each company in efforts to reduce data overfitting. Ultimately, we decided to use our data sampler widget to collect stock-data from each company over a six-month period. We tinkered with the pipeline in order to come up with different predictions and observations regarding which type of regression would most accurately model our data. After profuse testing, we discovered that using multivariate regression would allow us to create a plausible model for our data while also having the least error in comparison to the other data regression methods tested.

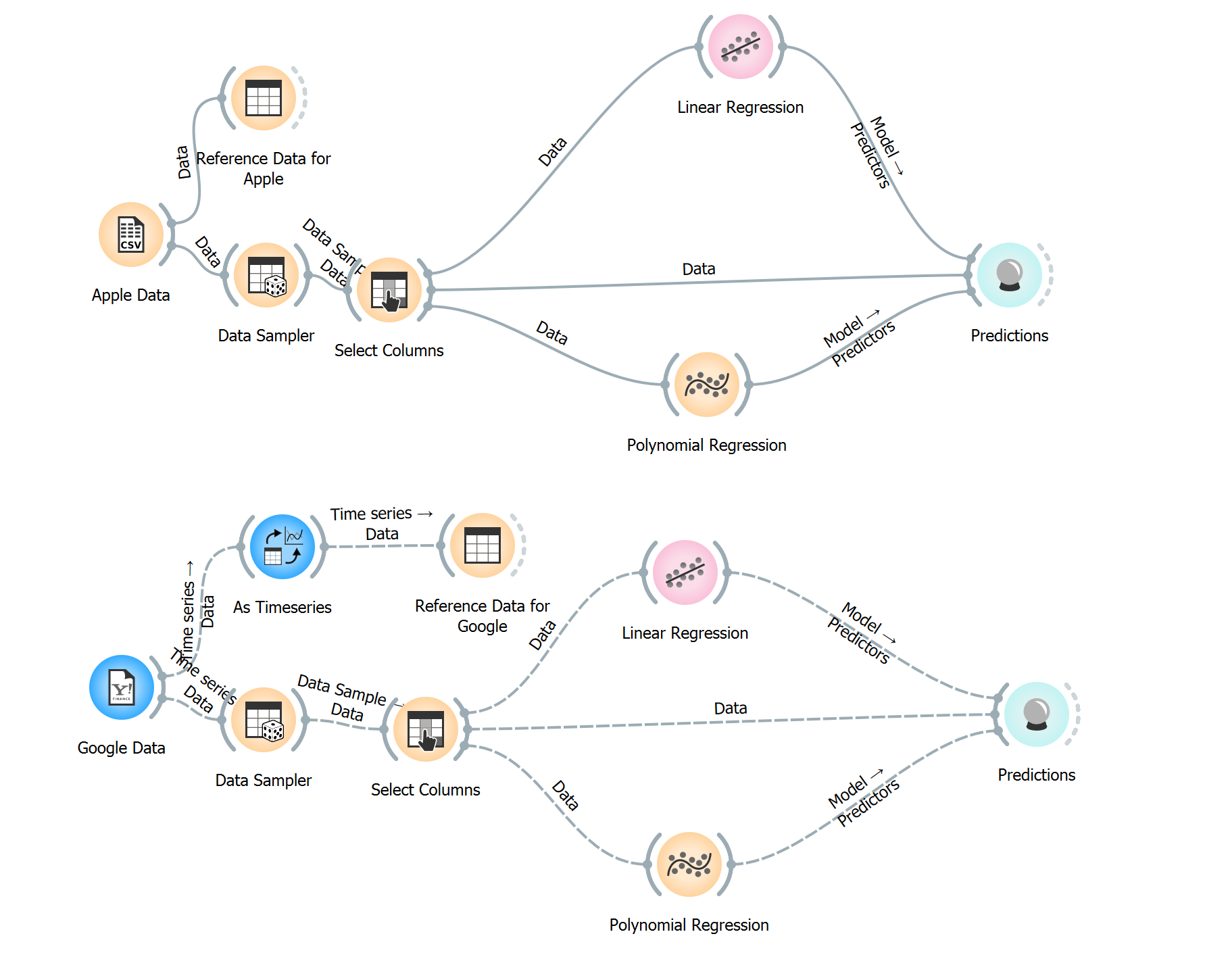
Since our pipeline’s purpose was to evaluate stock data for a company over a period of time, we decided to use the time-series module widget. This allowed us to chronologically combine our stock values with their corresponding dates so that we could properly analyze data from a strict six month period. The time series module allowed us to import stock data directly, without the use of enumerating the entire data set to associate dates to numerical values that can be used for regression and other numerical analysis methods used to predict future stock portfolios. The “as time-series” widget takes our input stock data (as a simple data table) and redefines it into a time-series table that associates each stock value with a specific date. Using this widget allowed us to have a new model of our data with more capabilities since further evaluation can be done with other useful widgets. This widget was essential to include because it ensures that the pipeline can conveniently be used by other people without them having to meticulously gather data and pair it with dates on their own. The difference in data collection can be seen with the variation of widgets used, the first pipeline uses a CSV input, which we took from an online API that had the stock data necessary and then associated number values for chronological order. The time-series module is another variation we used to collect the data into a dataset; The specialty of this widget is that it automatically enumerates the dates so we can apply mathematical functions to the data associated with each time ticker value. The next widget we used was the data sampler widget which serves the primary purpose of sampling a portion of the data, that is user-changeable. We use this data sampler to take a proportion of data, rather than just use the whole data set, in order to avoid overgeneralization of the predictions we come up with. If the predictions are overgeneralized, our data analysis would not properly fit other time periods of these companies. The “select columns” module is utilized in our pipeline for the purpose of choosing the target variable and the meta attributes, which serve in basic terms as the independent and dependent variables of the data that is being analyzed and modeled. The specific theme of our project entails utilizing the “open” pricing of the market as the target variable and the meta attribute being the enumerated “date”. The “linear regression” widget fits the data onto an x-y plane in a linear fashion. We use a lasso regression model that fits stock data the best, which we realized through trial and error. This widget takes our selected columns of sample data and performs regression analysis to produce an interactive data plot with a regression line included. The polynomial widget also allowed us to create different models for any nth degree. With regard to finding the appropriate degree, after manipulating the degree, we concluded that polynomial regression to the fourth degree best modeled our subsets of data. This resulted in the least error, but also counteracted against overfitting. Although increasing the degree further did show some slight improvement in some cases, we decided it would be best to keep the degree somewhat low in order to avoid the overfitting of our data. From these widgets, we then create links to the “predictions” widget, which allows us to review the errors and R^2 values. For our pipeline, we aim to explore which type of regression best fits the data using the MSE (Mean squared error) and the R^2 value which is the coefficient of determination. The “reference data” tab I have is just a data table that contains the overall data of all stock data values, which the user can change as targeted values.

After testing out our data using different regression methods, we concluded that linear regression modeled our data the most accurately. This conclusion was based on the fact that our data gave us a lower Mean Square Error (MSE) as well as a higher R-Squared value than using univariate regression (univariate because we are utilizing one target variable). For Amazon, our MSE for linear regression was 66.486, while the MSE for univariate regression was 151.338, which is a difference of 84.852. This difference is a primary indicator for our data sample that a linear regression offers a better prediction model than the univariate regression. This is the same trend modeled for the other two companies as well. For Google, our MSE for linear regression was 27.659, while the MSE for univariate regression was 76.544, which is a difference of 48.885. Finally, for Apple, our MSE for linear regression was 1.709, while the MSE for univariate regression was 3.799, which is a difference of 2.09. In all instances, the univariate regression seemed to fit our data less than linear regression. The MSE, when applied to the ideology of stocks, offers a localized error calculation, which is why the MSE changes drastically from one company to another since share valuations fluctuate that much as well. The coefficient of determination is a more applicable value that lets the user know how well the data fits the predicted models. The R-Squared values for Amazon were .995 and .989 for linear regression and univariate regression respectively, showing that the linear R-Squared value was greater by .006. The R-Squared values for Apple were .998 and .996 for linear regression and univariate regression respectively, showing that the linear R-Squared value was greater by .002. The R-Squared values for Google were .997 and .991 for linear regression and univariate regression respectively, showing that the linear R-Squared value was greater by .006. This furthers our argument by showing that in the case of all companies, the linear regression has a stronger coefficient of determination than that of the univariate regression, which serves to show that the sample of the data we took has a higher chance of being applicable to the linear regression prediction model rather than that of the univariate regression model.

Screenshots



This is a screenshot of the predictions tab in orange which gives users a view of the predicted regression data values. More importantly, at the bottom, it outputs the error values associated with each method of regression. We, however, chose the MSE and R^2 values as our comparison factors.



This is a screenshot showing the differences in using the time-series module to grab stock data vs utilizing an API and downloading the API. In theory, both practices work the same, however, when importing the CSV the user must manually input the enumeration values for the chronological ordering of the data. This screenshot also serves to show the backbone of our pipeline, which involves the ideas of gathering data, and evaluating it through mathematical operations and concatenating the data from these operations into the predictions tab, which measures the error as well as prints out the predicted values as well.