

Interpretation: Model starts by having DataVisualizer.py download data from 1 of 5 csv files corresponding to the stock, adding a date per line. File itself has columns Close, High, Low, Open, and Volume (in order). A new DataFrame, data2, is then created with 4 columns (taking data from the imported file): the stock's closing value of today (close), the stock's closing value of yesterday (yestClose), the stock's trading volume (volume), and the prediction column which is future closing price shifted by the indicated amount on shift (Prediction). Two training arrays (x and y) are then made, with x being a training set modeled after the existing data and y being a set of the stored value predictions. These x and y arrays are then put into a variable tree, which runs DecisionTreeRegressor().fit(), a method that takes the array of learned data, trains the algorithm to predict patterns, and splits the data into an accurate model. An array for future closing values (x_future) is then created, in which the tree data is used to predict future closing values to be stored in tree_prediction. A new DataFrame, future, is then created to store the predicted prices by the amount of predicted days in shift, where the file then plots historical data (blue), predicted data (red), and a connecting line between the historical and predicted data. The model is then fully created with the proper formatting, with historical closing prices (historical_data) and the predicted closing prices (future) being output as well.

Analysis: Overall, the model performs quite accurately for the limited 6 months (roughly) of data that is provided. Data acquisition and translation into the code runs quite smoothly, while the wild fluctuations with some of the stock prices seem to have no ill-effects on the accuracy of the model. Additionally, the use of machine learning and testing environments gives our algorithm plenty of test space to find the most accurate market trends from the data provided, giving the figures and extra boost in reliability.

Key Findings: The algorithm's output matched quite closely with what the team initially found when studying the stock patterns for previous weeks. Interestingly, the rapid market shifts in Carvana and Celsius provided the model with wider variations in predicted closing values, signaling the stock's volatility and adaptability of our model.

Data Acquisition Methods: Translating data by means of an API request through Yahoo Finance and implemented into a csv file, which is then analyzed by the algorithm.