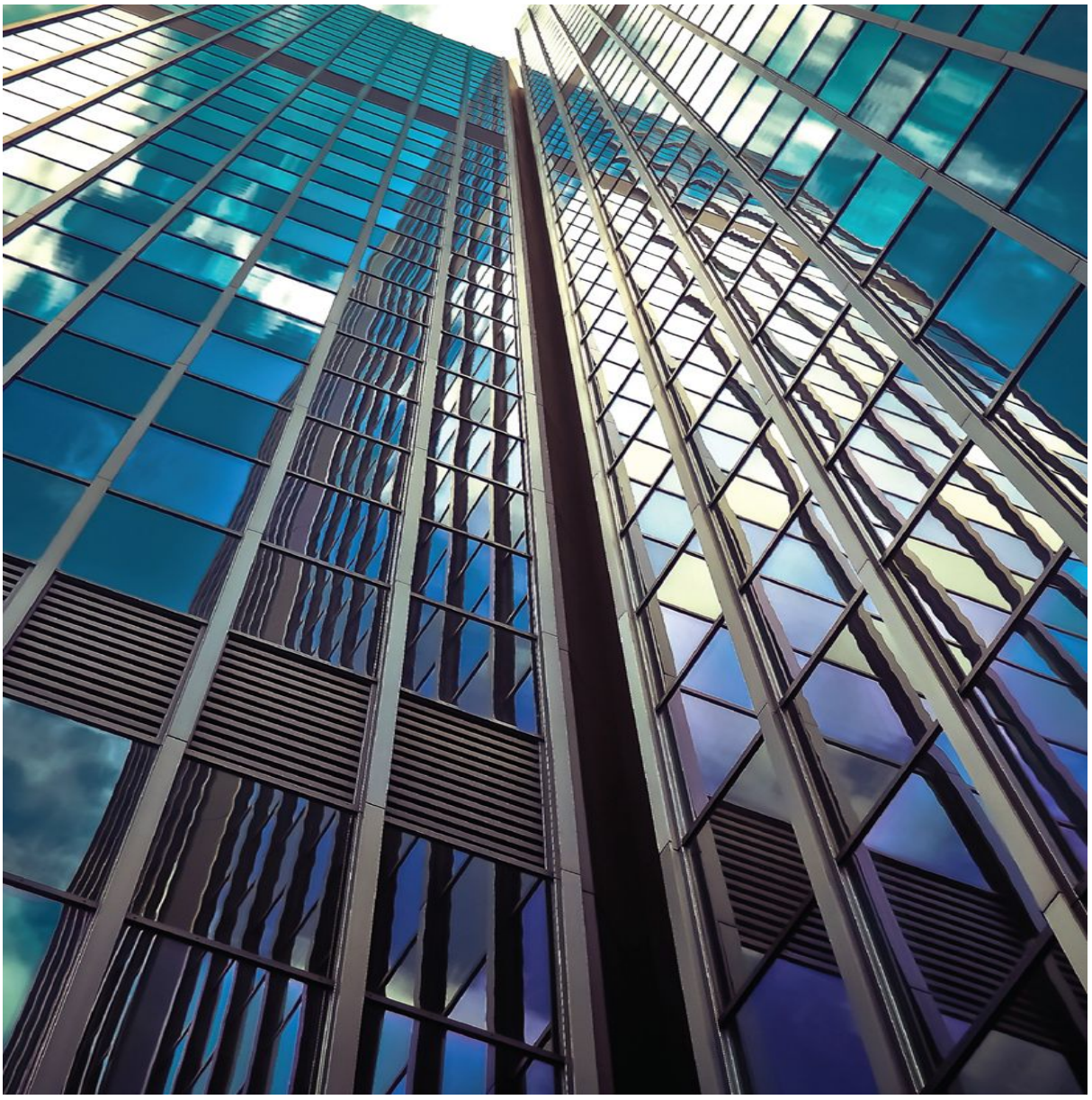
DATA ANALYSIS AMAZON

TRENDING STOCKS

PREDICTION STOCKS 2020



NAZARET SERRANO ROMERO

Introduction

The objective of this project is to find correlations between popularity of a company and the price of its share. I want to test this hypothesis:
"When the popularity of a company increase or decrease the value of its shares changes". If the company has an increase in the popularity of 10 and increase in the adjusted close of 40. These are data indicating that the frequency has increased 10 with respect previous day and the adjusted close value also increases with respect to the previous day. This is data that is going to be studied, if there is or there is not correlation between popularity and close adjusted, calculating their increases or decreases per share with respect to the previous day. Apply a more effective machine learning algorithm to be able to predict future data.
The company that I have chosen is: *Amazon (USA)*.

The data has been collected from the pages:

- Alpha Vantage - Free APIs for Realtime and Historical Stock, Forex (FX), Cryptocurrency Data, Technical Analysis, Charting, and More!
- Google Trends

Popularity data is collected by google, that is:
How many searches have been done that day on a specific topic? on topics general, without specificity. That's why I have my doubts that it can work, but I have to try to check my ideas.

INDEX

1. Tools and Libraries

In the first section of the jupyter notebook, I will load all the libraries that I will use during the project. The library are:
Pandas, Numpy, Matplotlib, Seaborn, Sklearn, Os, Source, Operator,Plotly, Interact and Fixed.

2. Data

In this section I will load the data in .csv format, analyze and explore the data and wrangle the data to apply the machine learning models. In other sections of the project I will also need to load data, but I will do it wherever I need it.
I am going to upload data from the company Amazon, to train the models and to testing I will upload 2020 data from each of the companies named above.

2.1 Data load

I Read csv files from Amazon, Tsla and Telefonica with pandas, analyzing data and preprocessing to be able to train machine learning models. Which are data that I will use to carry out study.Then check that the data has been loaded correctly.

What types of data are there? They are a data set corresponding to each of the companies to study. The datasets descriptions have the same columns in common with five columns.

	Timestamp	Adjusted_Close	Popularity	Increment_A_C	Increment_P
0	2017-12-29	1169.47	0.0	0.00	NaN
1	2018-01-01	1169.47	82.0	0.00	82.0
2	2018-01-02	1189.01	100.0	19.54	18.0
3	2018-01-03	1204.20	95.0	15.19	-5.0
4	2018-01-04	1209.59	81.0	5.39	-14.0

Description of each column:

- Timestamp: It is the date of data collection.
- Adjusted_Close: Stock prices values are stated in terms of it is 'closing price' and it is 'adjusted close'. The closing price is the 'raw' price which is just the cash value of the last transacted price before the market closes. The adjusted close price factors in anything that might affect the stock price after the market closes.
  Unit of measure of the closed share price:
  Amazon → USD price per share.
- Popularity: It is popularity of the company in data collected by google trends.
  It is represented in a range of (0-100) in relation to the maximum value in EEUU and year 2018-2019. A value of 100 indicates maximum popularity of a term while 50 and 0 indicate that a term is half popular in relation to the maximum value or that there was not enough data of the term, respectively.
- Increment_A_C: Increase or decrease of the price with respect to the previous day.
  Using the formula:

  Increment = Current price - last previous price
- Increment_P: Increase or decrease of the popularity with respect to the previous day.
  Using the formula:

  Increment = Current popularity - last previous popularity

Testing data 2020

I read a csv file of the company of the year 2020 with pandas. The data contained in the testing data set are:

- Date: It's the date of value.
- Popularity: It's the popularity of the company in the data collected by google trends. It's represented in a range of (0-100) in relation to the maximum value in EEUU, and from the months of January and February of the year 2020.
  A value of 100 indicates the maximum popularity of a term, while 50 and 0 indicate that a term is popular in relation to the maximum value or that there was not enough data for the term, respectively.
- Increment_Pop: Increase or decrease of the popularity with respect to the previous day.
  Using the formula:

  Increment = Current popularity - last previous popularity

  I'll use the Increment_Pop column to test and predict the increases or decreases of the actions of the year 2020. With these test data I will test the machine learning algorithms used.

2.2 Data cleaning

Training data

I check if I have Nan in the data set that I am going to use for training. We found one, because there was no data from the previous day and it was not possible to calculate increases or decreases.

I proceed to delete the row, because it corresponds to the date of 29/12/2017.

Testing data

I check for Nan. We found one, because there was no data from the previous day and it was not possible to calculate increases or decreases.

I proceed to delete the row, because it corresponds to the date of 31/12/2019.

I prepare the data column that I am going to use to test.

2.3 Data exploration

Training data

I am going to describe data:

There is a total of 730 data for the year 2018-2019.

The columns that I will use to analyze the hypothesis are:

- Increment_A_C.
- Increment_P.

Increment_P: With a mean 0.07 and standard deviation 6.43. The minimum value is -33 and the maximum value is 82. Q1 is -2.00, Q2 is 0.00 and Q3 is 2.00. I will use this column as an independent variable. I will call its later X.

Increment_A_C: With a mean 0.93 and standard deviation 26.10. The minimum value is -139.36 and the maximum value is 126.94. Q1 is -4.94, Q2 is 0.00 and Q3 is 10.97.I will use this column as a dependent variable. I will call its later y.

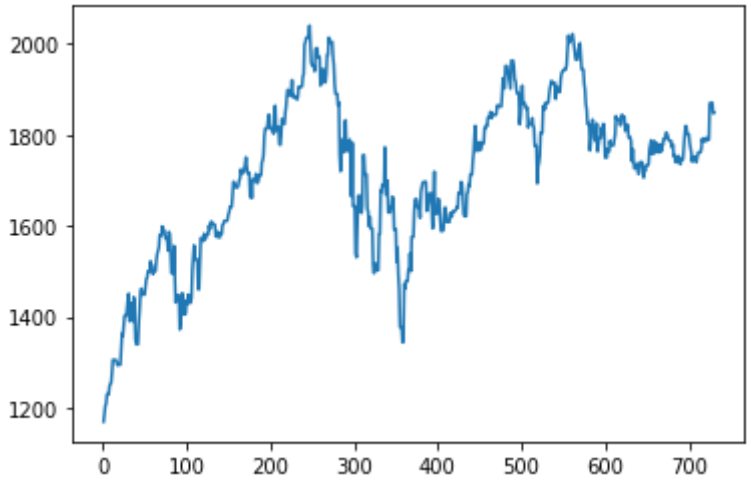
The columns:

- Adjusted_Close.
- Popularity.

I will not remove them, but for now I will not need them.

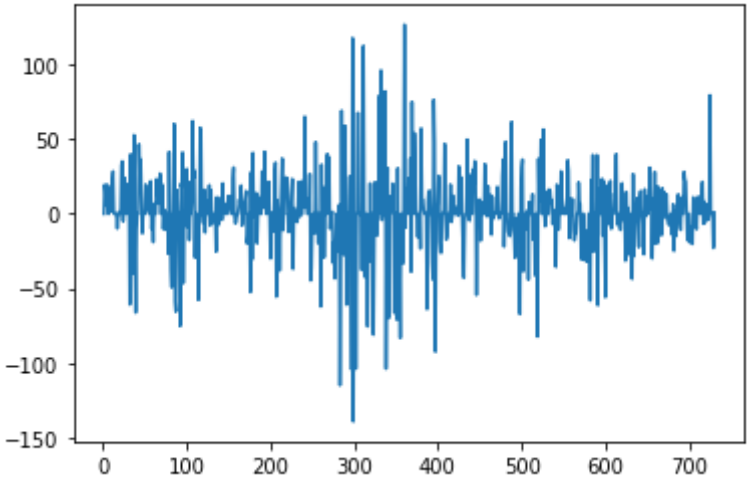
I am going to explore the data that the data set contains.

fig(1). **Variability of actions**.



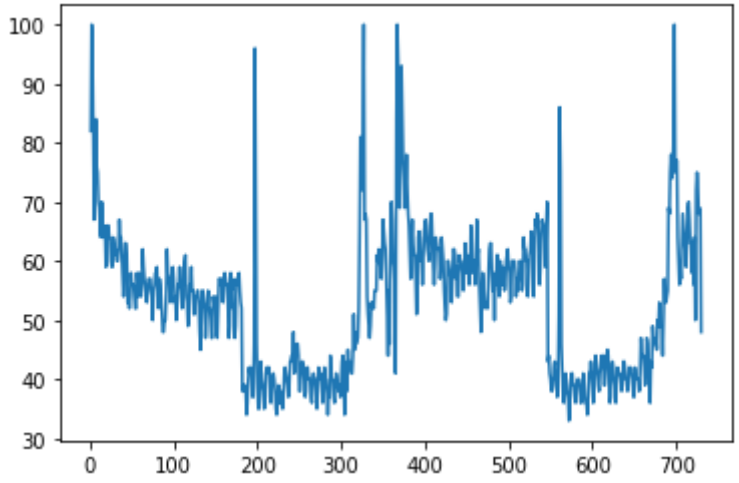
You can see in this order from left to right there is variability in adjusted closed. On the ordinary axis is price value adjusted close. In the abscissa axis it is representing data that our data frame contains. There is not missing data.

fig(2). **Variability of increases or decrement actions**.



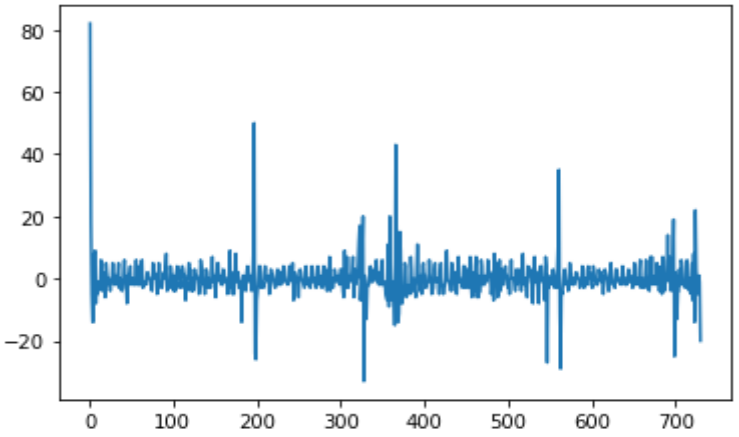
There is significant variation in daily increases and decreases, as can be seen in the graph.

fig(3). **Variability of popularity**.



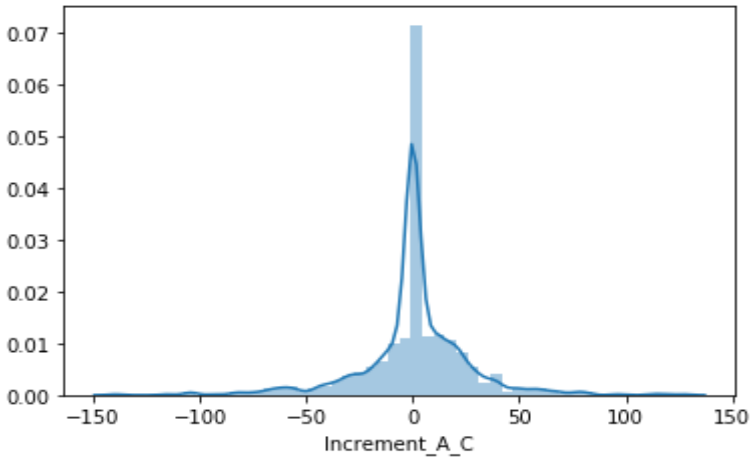
We have some peaks of increasing popularity, but overall it remains constant. It can be seen that the last stage of the year is when this company is less popular.

fig(4).**Variability of increases or decrement popularity**.



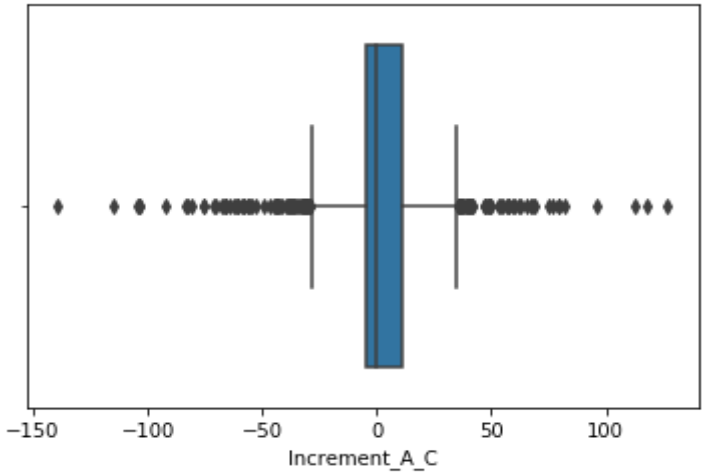
We can see the peaks we observed in the previous graph more clearly in the daily increments or decrements. It has variability in increases or decreases with the values of the shares.

fig(5). **Increases or decrement of the price per share**.



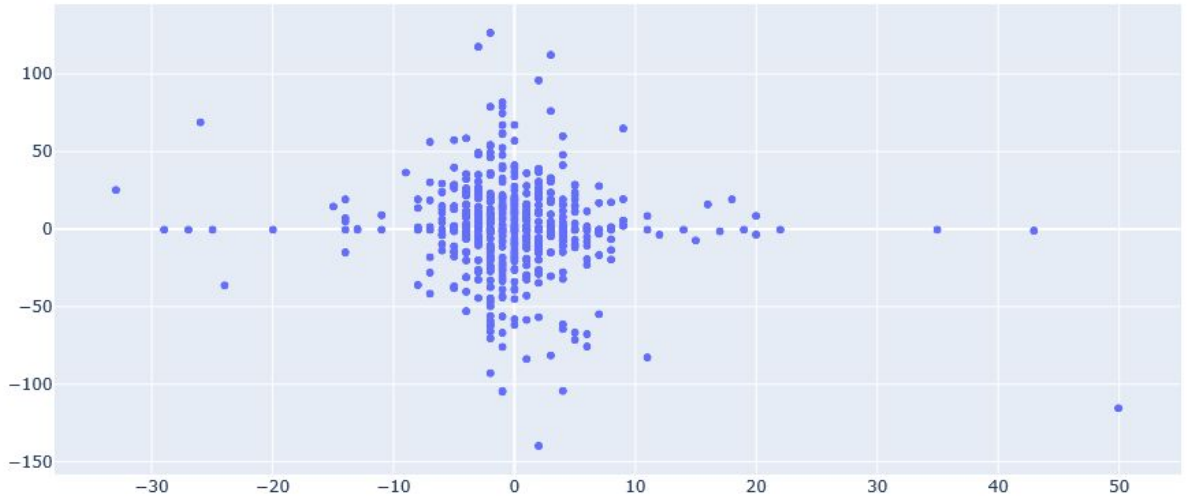
You can see that the distribution of the increment_A_C of data, it has a negative asymmetry, indicates values with more frequencies determined to the left of the median. With a positive kurtosis >0, indicating that data is very concentrated towards the mean of distribution.

fig(6). **Increment_A_C.**



You can see in the box plot from the graph that the median is centered at zero. You see so many outliers because I also have a lot of data that are below the minimum value and above the maximum value. I am not going to eliminate outliers at the moment.

Relation of variables



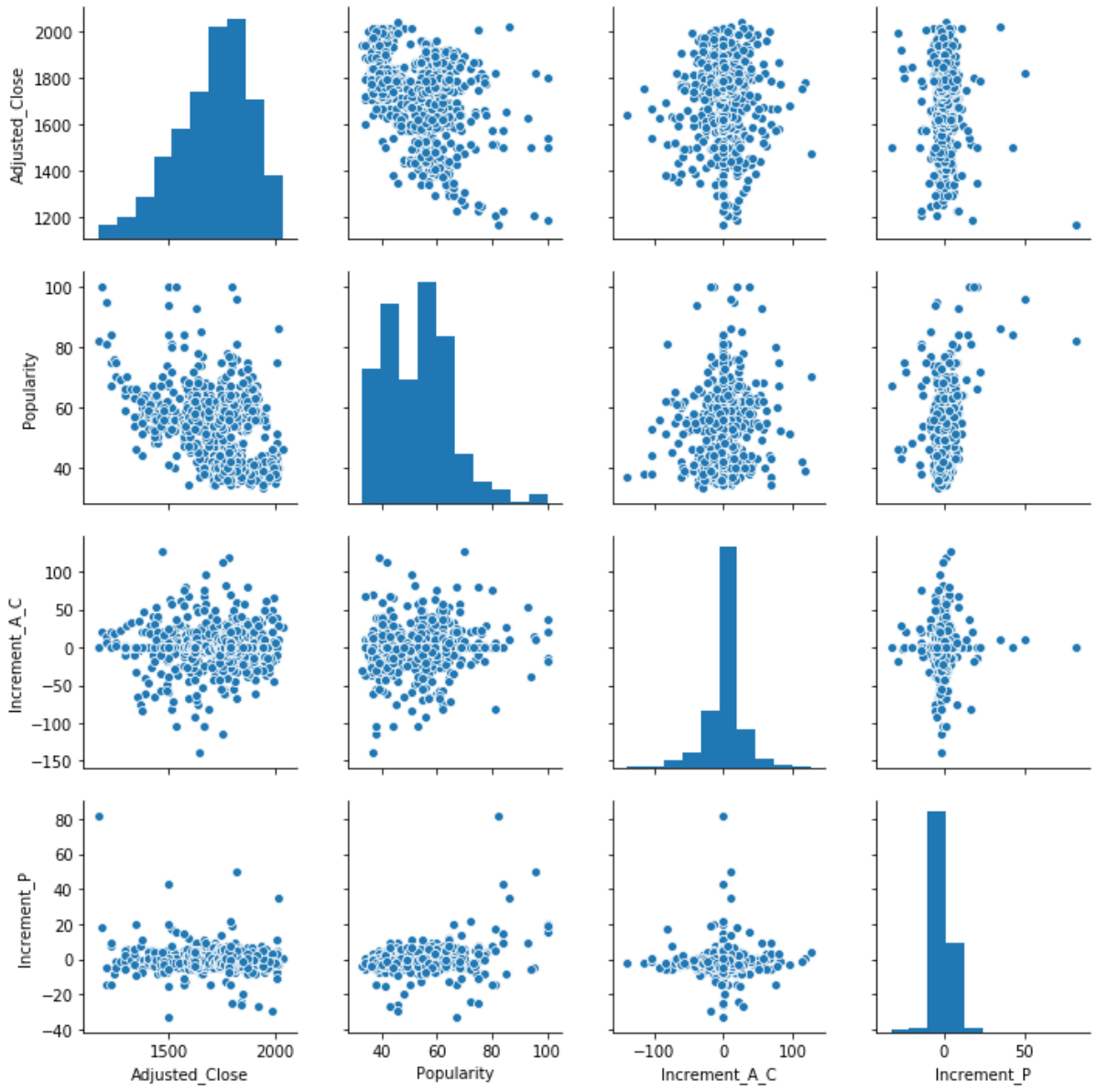
Amazon data doesn't follow a high correlation, it can be seen that the Increment_A_C variable is very scattered. Increment_A_C variable can take any value and the variable Increment_P can have values close to zero.

For example when the variable Increment_P takes value close to 0, it can be seen that the variable Increment_A_C can take the values from -70 to 60 , but in this case it takes a value anyone,without following a linear relationship.

I will prepare this data by filtering data to prevent this from happening.

I will filter the data, to try to study the points that can have an increase or decrease and are correlated, either by performing a unit analysis for each point or to apply some machine learning algorithm that tries to predict future data, with data of the year 2020.

Finally we will see the relationship that all data frame variables have.

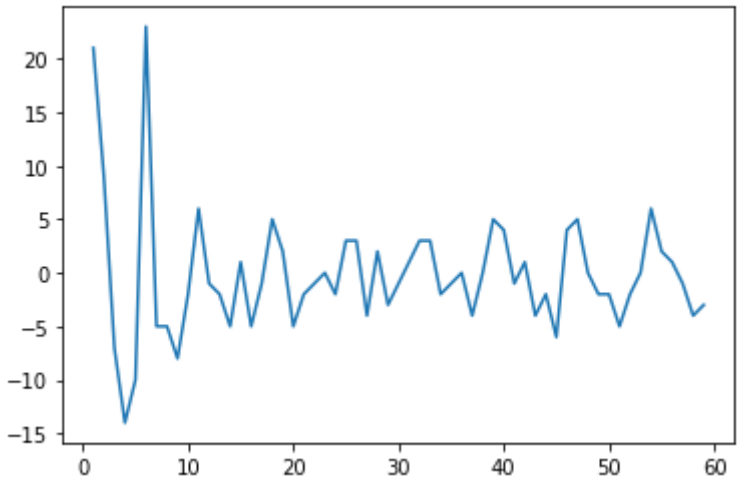


You can see in this graph the types of data that contain variables of the data frame, for my study I will use the variable that I named previously that are Increment_P as an independent variable, and Increment_A_C as a dependent variable.

We can see distributions of all the variables, and the relationship there is with each of them.

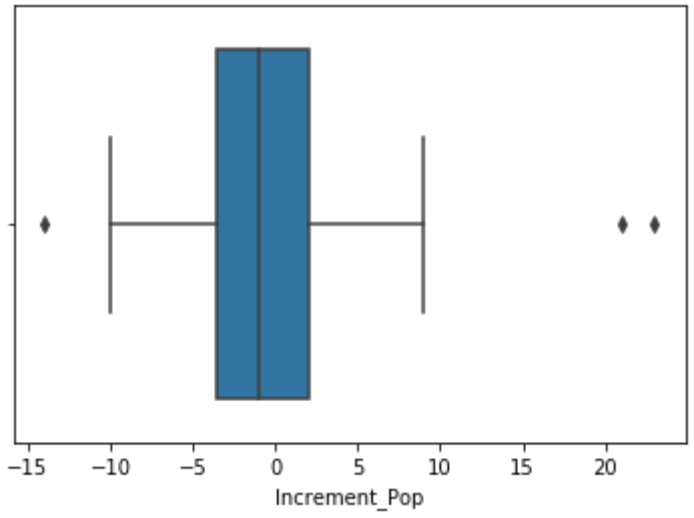
Testing data

fig(8). **Variability of Increment_Pop**.



You can see that at the beginning of the year there is a peak that decreases in days and increases again.
Later variability is observed but not as pronounced as at the beginning of the year.

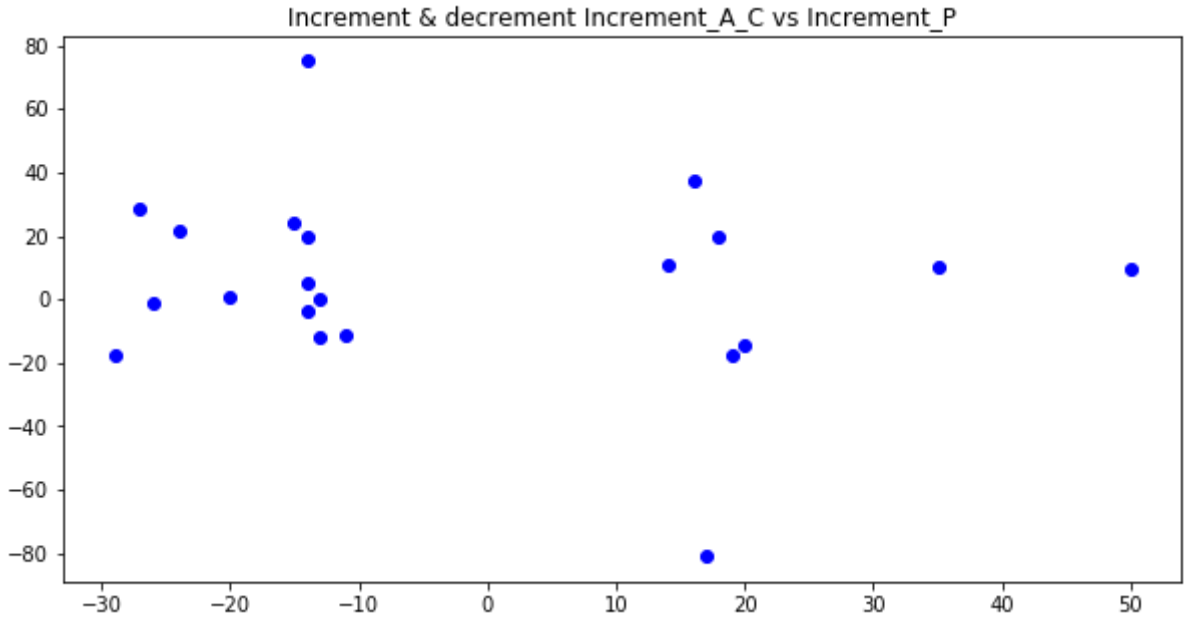
fig(9). **Increment_Pop**.



You can see in the box plot from the graph that the median is centered below zero. There are three outliers, but I'm not going to remove them.

2.4 Data wrangling

As I said before, I am going to proceed to data filtering Increment_A_C, I am going to keep all the data except the data that is equal to 0.00. After that, I delete the rows that have run out of values (Nans).
I keep the data of the variable Increment_P that have an increase> 10 and a decrease <-10.
I concatenate the filtered data and create a data frame (data3) with the filtered data.

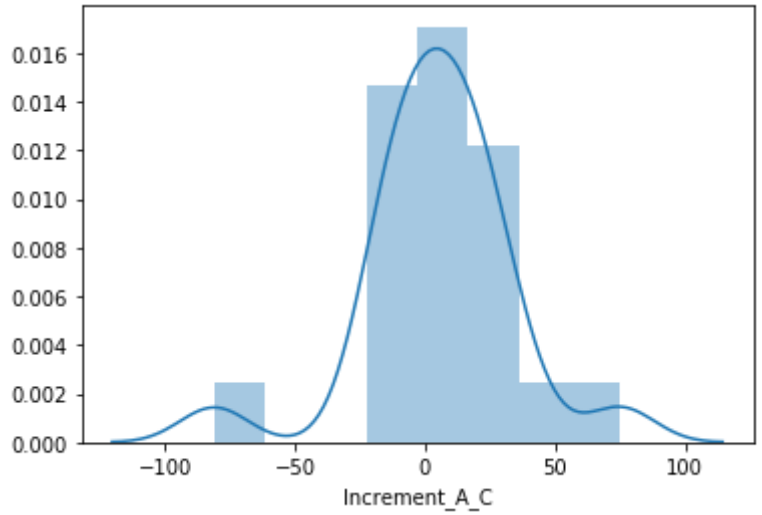


This graph is the result of the data that I have left after applying some filters explained in the previous point. You can see observe:

- **Increment_P decreases:**
  - When Increment_P decreases -81, Increment_A_C increases (81).
  - When Increment_P decreases -18, Increment_A_C decreases (-29).
  - When Increment_P decreases -18, Increment_A_C increases(19).
  - When Increment_P decreases -15, Increment_A_C increases(20).
  - When Increment_P decreases -12, Increment_A_C decreases (-13).
  - When Increment_P decreases -11, Increment_A_C decreases (-11).
  - When Increment_P decreases -4 , Increment_A_C decreases(-14).
  - When Increment_P decreases -1, Increment_A_C decreases (-26).

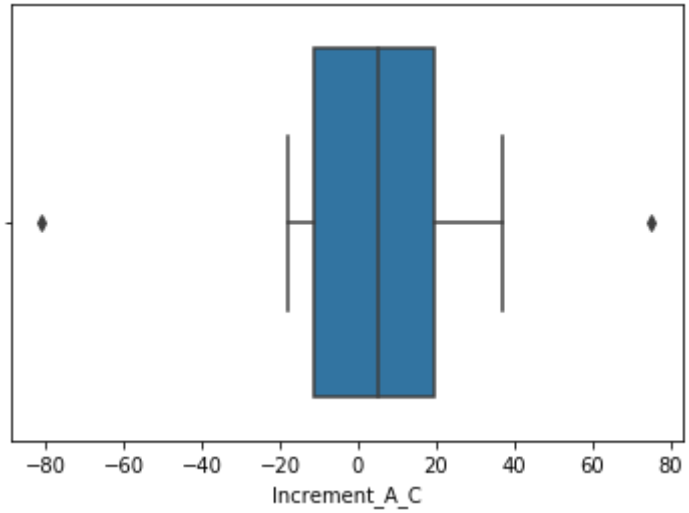
- **Increment_P increases:**
    - When Increment_P increases 75 the Increment_A_C decreases(-14).
    - When Increment_P increases 37 the Increment_A_C increases(16).
    - When Increment_P increases 29 the Increment_A_C decreases(-27).
    - When Increment_P increases 24 the Increment_P decreases (-15).
    - When Increment_P increases 21 the Increment_P decreases(-24).
    - When Increment_P increases 20 the Increment_P decreases(-14).
    - When Increment_P increases 20 the Increment_A_C increases(18).
    - When Increment_P increases 11, the Increment_A_C increases (14).
    - When Increment_P increases 10, the Increment_A_C increases (35).
    - When Increment_P increases 9. the Increment _A_C increases (50).
    - When Increment_P increases 5, the Increment_A_C decreases (-14).
    - When Increment_P increases 1, the Increment_A_C decreases (-20).
    - When Increment_P increases 0.1. the Increment_A_C decreases (-13).

fig(10). **Distribution of data Increment_A_C**



This is distribution of data after filtering. I've little data left. You can see that it does not follow a normal distribution of data. We have a negative asymmetry, leaving the data more centered to the left of the mean, the mean is 4.91.
A positive kurtosis indicating a higher concentration of the data around the mean.

fig(11). **Box Plot Increment_A_C**

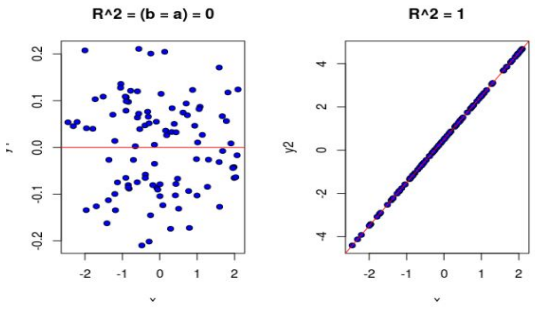


We can see in the box 75% of the data, with a median of 5.39. The other 25% are the "bigotes", the lines that extend from the box, extend to the maximum and minimum values of the series or up to 1.5 times the IQR. and the rest are outliers, or Outliers.

After that, I will proceed to export data3, to use it later.

3. Machine Learning Models

I am going to implement different regression algorithms, to keep the best percentage of successes.
The coefficient of determination is defined as the proportion of the total variance of the variable explained by the regression. The coefficient of determination, also called R squared, reflects the goodness of the adjustment of a model to the variable to be explained.
It is important to know that the result of the coefficient of determination ranges from 0 to 1.



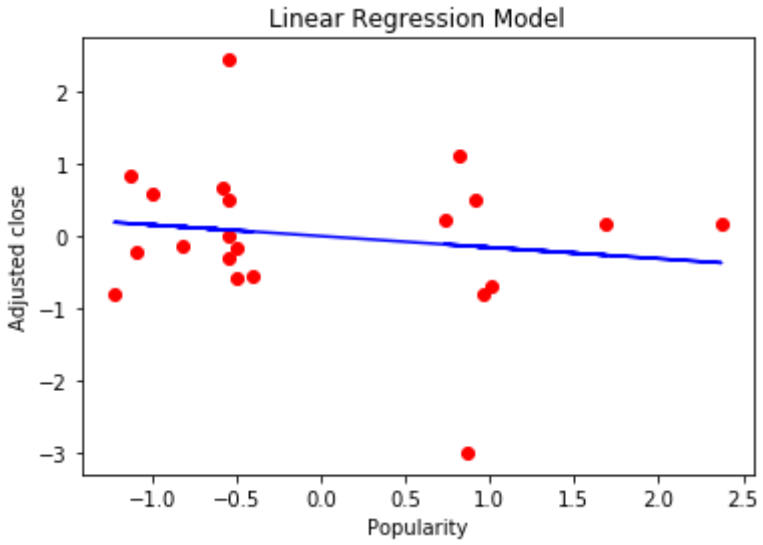
The closer to 1 its value is placed, the greater the adjustment of the model to the variable we are trying to explain.

Conversely, the closer to zero, the less tight the model will be and, so less reliable it will be.

I am not going to divide data into a training set and test set, because I have very little data. To train models I will use complete data without division, and to test I will use the data of popularity increases or decreases 2020, as an independent variable, and results will be increments in action per day, as a result of independent variables.

3.1 Linear Regression

I am going to select an independent variable (X) and dependent variable (y) to train the model.
I will standardize the variables. Standardization of datasets is a common requirement for many machine learning estimators implemented in scikit-learn.
I am going to create a linear regression model, and adjust to the training data of the model.
You can see in the next graph, that model is very bad, there isn't a linear relationship.



Blue line is the regression that best fits a model.
And red dots are data.

Prediction of the increases or decreases in actions with training data. In order to continue performing model validation calculations.
Prediction of the increases or decreases in actions with 2020 data.
I am going to calculate the coefficient of determination (R Squared), and other metrics.
I am going to calculate the coefficient of determination (R Squared)). Using metric score. The result is: 0.02, which means that it is a very bad model.

2% of the total data of stock increments or decreases can be explained by the increase or decrease in the company's popularity.
MSE: Is the overage of difference between actual data points and predicted output squared. MSE basically measures the average square error of our predictions. For each point, calculate the square difference between predictions and target then average those values.
The higher this value, the worse the model. It is never negative, since we are squaring individual prediction errors before adding them, but it would be zero for a perfect model.
RSE: It measures performance based on a comparison with a simple predictor performance. The RSE normalizes the total squared error of the tested model and divides it by the total squared error of the simple predictor.
Just like the other two measures, it ranges from 0 to infinite, being 0 the best value. If the values are very large, we can obtain approximately but relatively good RSE that depends on the size of the y. Regardless, the lower RSE values indicate a better fit. RSE, being an absolute measure and measured in units of y, it is not very clear what constitutes a good RSE.
Error: It is the mean of the residual standard error.
We can interpret the results of our model,Knowing than our standardized data, we have:

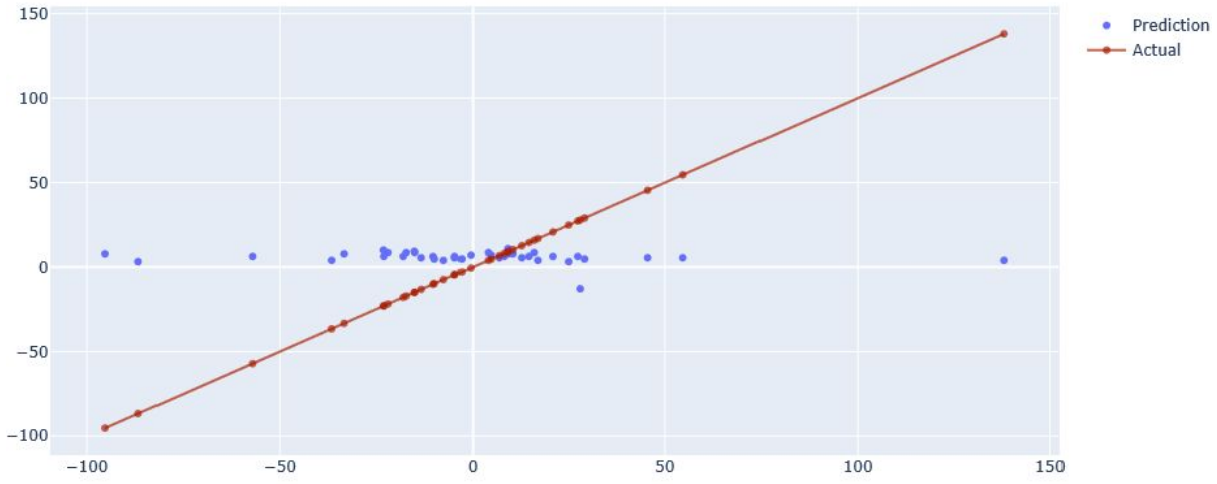
- SSD: 20 a very high value, so it indicates a bad model.
- RSE: 1, is far from the value 0.
- Error: 3.82896006e+16, a super high number.The model is bad.
- MSE: 0.98, the measure goes from 0 to infinite, being 0 the best value you can get. Measures the average of the squared errors, that is, the difference between the estimator and what is estimated.
- Root-mean-square error (RMSE): 0.99 ,the RMSE measures the quadratic mean of the differences between the predictions made by a model and the actual values (residuals). The measure is always positive or 0, being this last value the best one possible (but also an overfitting situation in many cases).
- R2: Is better the closer it is to 1.Previously calculated with another method.

The prediction model would be:
Share price = 2.72588481e-17 + -0.1560434 * Increment_P

Although the model is bad, I am going to predict the 2020 data. After that, I will call API REST to check actual results of the stock price. I will calculate your daily increases or decreases and compare with the results of our model.

I am going to calculate the coefficient of determination (R Squared) to see if the predictions fit the actual data, but obviously not. As you see, the result is: -0.07, when R2 is negative it is interpreted as if its value were 0, therefore our model is very bad.

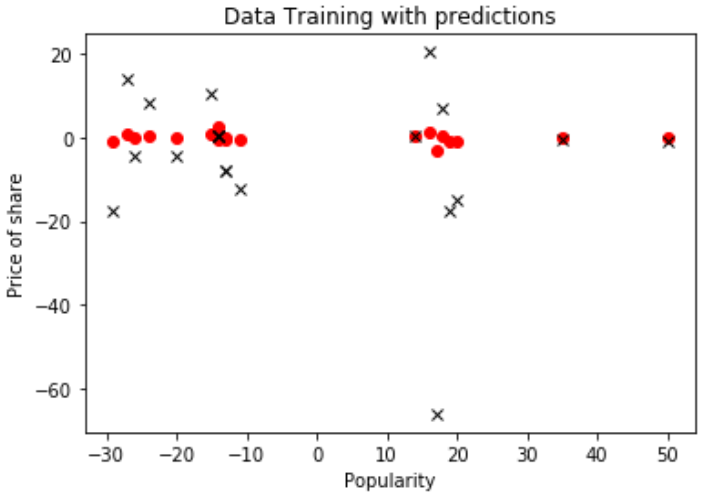
In this graph we can see the differences of the real values and the predictions. The red line represents real values of the increases or decreases of actions of the year 2020. The blue points represent the increases or decreases that our algorithm has predicted.



3.1 SVM (Regression)

I am going to adjust SVM regression with the dataset. I have created a function to implement the algorithm. Then I use an interactive map to be able to change the hyperparameters of the algorithm.
Once I have adjusted the hyperparameters, I call the function to implement the svm algorithm, obtaining a score of 0.71, quite good. I am going to give data its real value, to make its representation easier to understand.

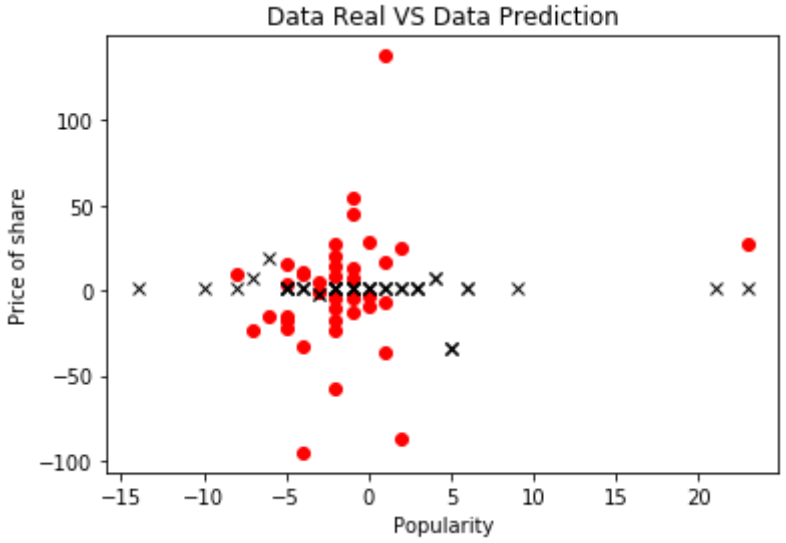
fig(12). **Data training vs prediction**



As you can see in this graph the black X corresponding to the data of the predictions and the red points are the real data.

I add the results SVR of the 2020 test prediction to the dataframe day_test.
I compare the predictions of our SVR algorithm with the real data of the increases or decreases of the actions of 2020.

fig(13). **Data real vs data prediction**



In this graph you can see that the predictions don't fit the real data very much, something is happening here. I am going to check the score, to see the result and be able to assess.

I check that the Validation_amazon data set created above contains Nans. I am going to proceed to eliminate them. That's because on weekends and holidays no data is taken from the value of the shares.
The result of R2 is 0.03, being a fatal result.

The algorithm doesn't correctly predict the testing data but in the training data I have obtained quite good results. This is because our model suffers from OVERFITTING .

Ways to solve it:

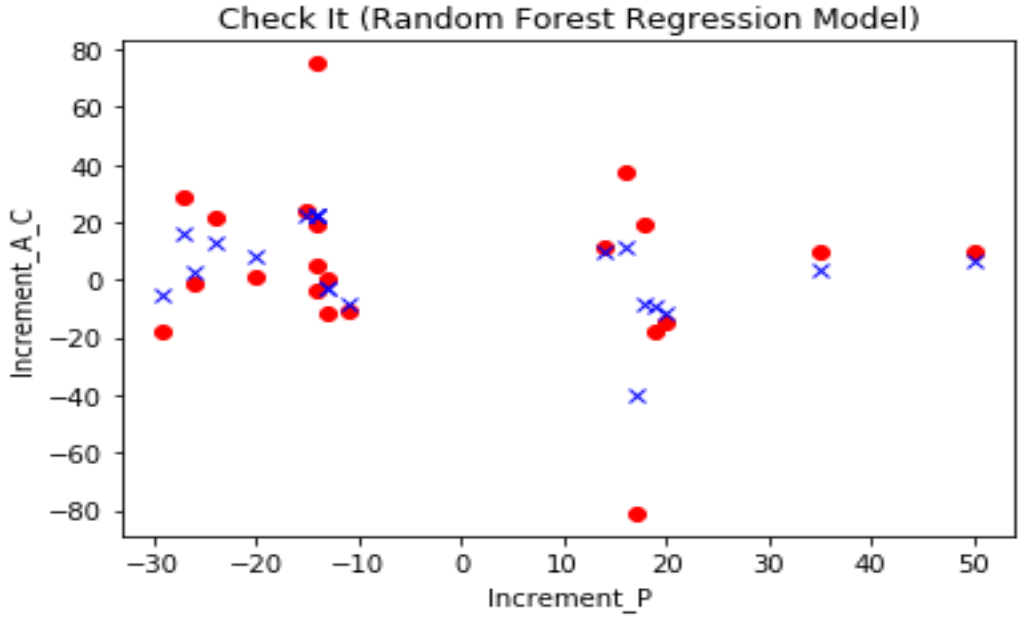
- Train the model with more data, but because I had to filter the data because I had a lot of data in the Increment_A_C variable when the Increment_P variable took the value 0 (I can't do this).
- Another thing that has a lot of influence is dividing the data into a training and testing set, being able to enter random data (cross validation) . I can't do it either because I have very little data due to filtering.

I am going to implement another algorithm, to see how effective they are, not ruling out going back to this point to make adjustments.

3.1 Random Forest Regression

I am going to adjust Random Forest Regression with the dataset. I have adjusted the model with data. Then I fit the hyperparameters of the algorithm and the data to the model. I make the predictions with the training set.

fig(14). **Random Forest Regression Model**



You can see in this graph that the red points are the values of the training set, and the blue X are the data that our model has predicted.

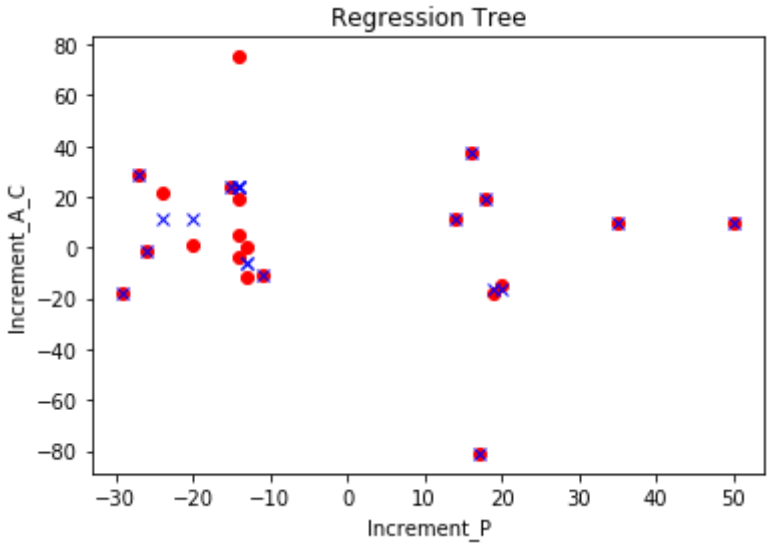
I have obtained a score of 0.57 and a RMSE of 18, not a good model.
But, I am going to implement it with data 2020.

You can see that once I pass the testing data (2020) to the model.The results I have obtained with the testing data are dire with a score of -0.07 and RMSE of 13. My model doesn't behave as well. In this case it suffers overfitting.

3.1 Regression Tree

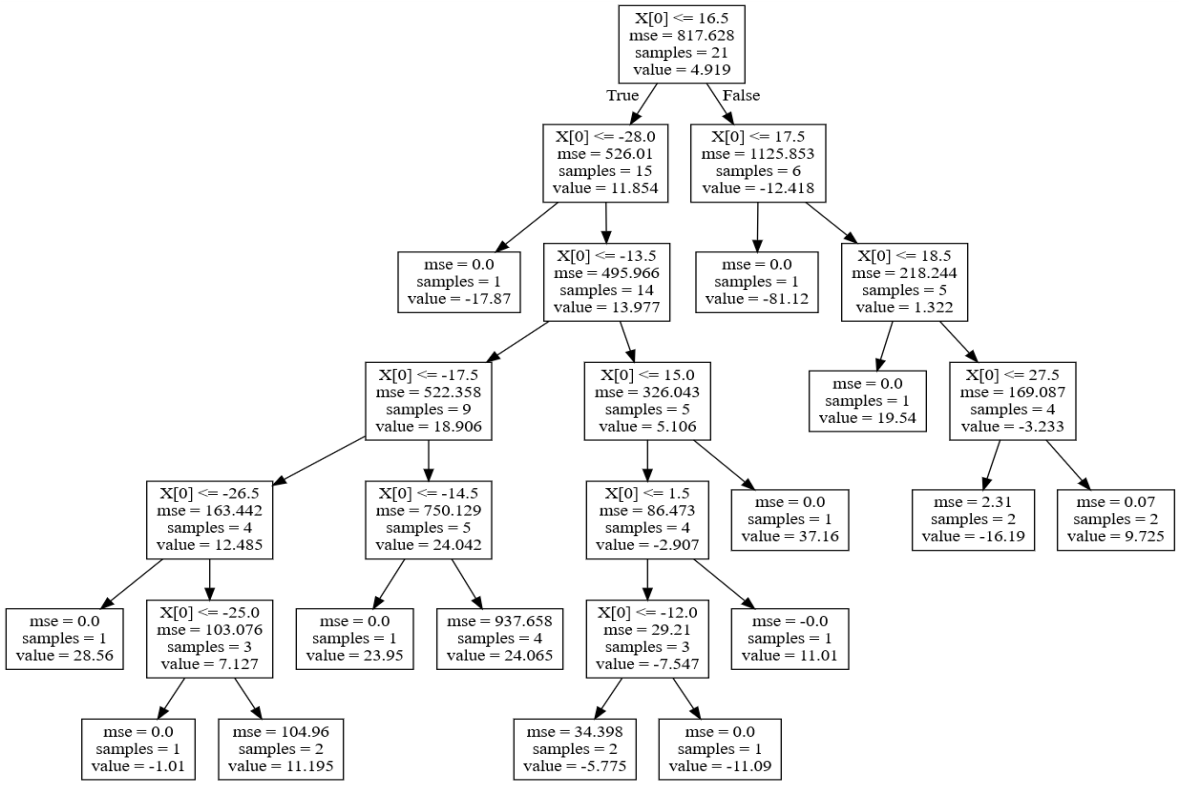
I am going to adjust the Regression Tree with the dataset. I have adjusted the model with data. Then I fit the hyperparameters of the algorithm and the data to the model. I make the predictions with the training set.

fig(15). **Regression Tree**



You can see in this graph that the red points are the values of the training set, and the blue X are the data that our model has predicted.
I have obtained a score of 0.77 and a RMSE of 13.86, much better than Random Forest Regression.

This is my tree:



I'm going to implement data 2020. I've obtained the testing data, it's a very bad model with a score of -0.14 and RMSE of 39.82. In this case again there is overfitting.

4. SUMMARY

AMAZON MODELS	R2 TRAIN	RMSE TRAIN	R2 TEST	RMSE TEST
LINEAL REGRESSION	0.02	6.6	-0.07	38.68
SVR ("kernel --> rbf")	0.71	0.53	-0.03	37.77
RANDOM FOREST REGRESSION	0.57	18.95	-0.07	38.66
TREE REGRESSION	0.77	13.86	-0.14	39.82

As it can be seen in the results, the best model in the training data has an R2 of 0.77, it could be translated that this model would be able to explain reality in 77% of cases, even though it is the most hopeful of the model presented, it should be improved in order to seek greater precision. An error in the stock market of 23% is too high.

In the models in which I have obtained an R2 greater than 0.5, when I have performed the validation of the models it has not been good, the pattern of the created model not being correct.
It has been tried to train the models with more historical data and the same thing happens.
Training has also been experimented by joining training and testing 2020 data and creating randomness to train and test, in this case the R2 dropped and they were not good models.
I don’t rule out in the future being able to add some variable that may give another meaning to the data.
They are not good models for the date I have.