**Predicting Stock Prices Using FbProphet**

Kanika Nama | Dinesh Deivamani 

[nama.k@husky.neu.edu](mailto:nama.k@husky.neu.edu)

[deivamani.d@husky.neu.edu](mailto:deivamani.d@husky.neu.edu)

**CSYE 7245 Big Data Sys & Intelligence Analytics, Spring 2018 Northeastern University**

**Abstract**

Stock market are key to technical analysis where the approach to investing is based on cycles. If we want positive return every time whenever we do investment, we can take help of a machine learning model that can help us to look deeper into these hidden cycles in stock market. Hence, as a part of this project, our main aim was to predict the Open price of stock by analyzing the yesterday’s Close price as accurately as possible based on given historical data. So, we are developing a FB Prophet model using Spark and Python. Furthermore, we extended our approach to spark clusters on google cloud data proc to improve the performance of our model. Based on our observations, Prophet generates a reliable model with accuracy of 80 percent. However, analysis needs to be extended to a greater number of parameters so that we can get the best prediction from our model.

**Introduction**

Forecasting is a data science task that is a central to many activities within an organization. For instance, large organizations must engage in capacity planning to efficiently allocate scarce resources and goal setting in order to measure performance relative to a baseline and produce high quality forecasts. Prophet was initially developed for the purpose of creating high quality business forecasts. This library tries to address the following difficulties common to many business time series: Seasonal effects caused by human behavior -weekly, monthly and yearly cycles, dips and peaks on public holidays, Changes in trend due to new products and market events and Outliers. Moreover, Prophet has a number of intuitive and easily interpretable customizations that allow gradually improving the quality of the forecasting model. What is especially important, these parameters are quite comprehensible even for non-experts in time series analysis, which is a field of data science requiring certain skill and experience.

### Features of Business Time Series - There is a wide diversity of business forecasting problems, however there are some features common to many of them. Fig. shows a representative time series for Amazon stock prices. There are several seasonal effects clearly visible in this time series: weekly and yearly cycles, and a pronounced dip around Thanksgiving, Christmas and New Year. These types of seasonal effects naturally arise and can be expected in time series generated by human actions. The time series also shows a clear change in trend in the last six months, which can arise in time series impacted by new products or market changes. Finally, real datasets often have outliers and this time series is no exception. This time series provides a useful illustration of the difficulties in producing reasonable forecasts with fully automated methods.

### 

### The Prophet Forecasting Model- We use a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

**y(t)= g(t)+s(t)+h(t)+εt**

g(t): piecewise linear or logistic growth curve for modelling non-periodic changes in time series

s(t): periodic changes (e.g. weekly/yearly seasonality)

h(t): effects of holidays (user provided) with irregular schedules

εt: error term accounts for any unusual changes not accommodated by the model

**Dataset**

The dataset is used from <https://www.yahoofinance.com/>

We are using close prices for Amazon from 2016/01/01 to 2019/03/01 as an example to have better understanding about what we are doing. With yahoo.fin ,we can access stock price easily. The documentation we can see here is a strong trend of growing price from 2016. However, there are still a lot of up and down or cycles during these years.

**Steps of Implementation-**

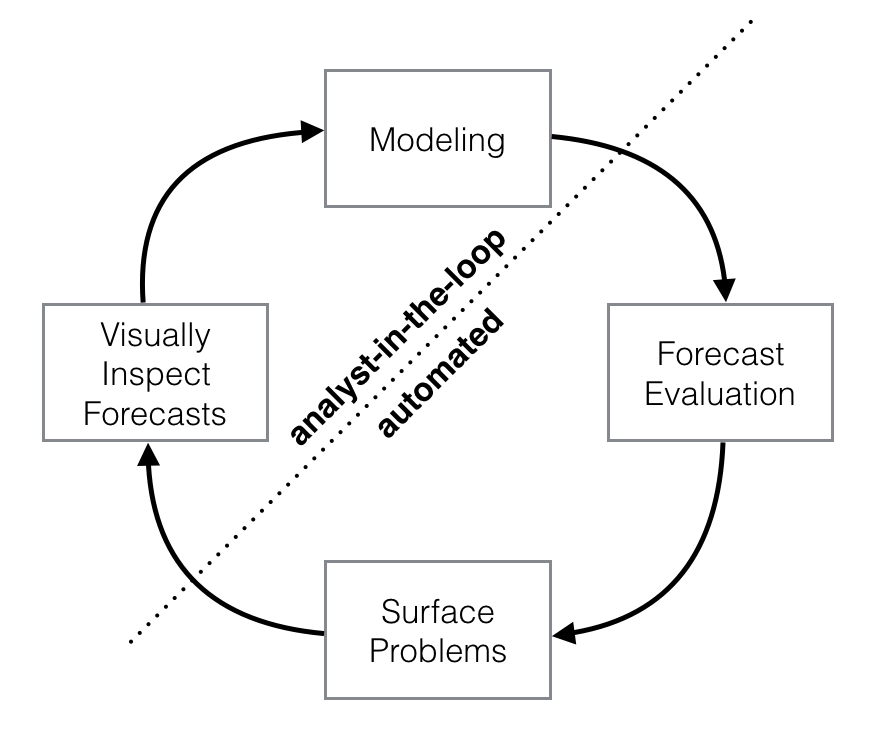
* Introduction and Installation
* Model Fitting
* Making Future Predictions
* Obtaining the Forecasts
* Plotting the Forecasts
* Plotting the Forecast Components
* Cross Validation
* Obtaining the Performance Metrics
* Visualizing Performance Metrics
* Conclusion

**Where Prophet shines**

Not all forecasting problems can be solved by the same procedure. Prophet is optimized for the business forecast tasks we have encountered at Amazon stock, which typically have any of the following characteristics:

* hourly, daily, or weekly observations with at least a few months (preferably a year) of history
* strong multiple “human-scale” seasonality’s: day of week and time of year
* important holidays that occur at irregular intervals that are known in advance (e.g. the Super Bowl)
* a reasonable number of missing observations or large outliers
* historical trend changes, for instance due to product launches or logging changes
* trends that are non-linear growth curves, where a trend hits a natural limit or saturates

We have found Prophet’s default settings to produce forecasts that are often accurate as those produced by skilled forecasters, with much less effort. With Prophet, you are not stuck with the results of a completely automatic procedure if the forecast is not satisfactory — an analyst with no training in time series methods can improve or tweak forecasts using a variety of easily-interpretable parameters. We have found that by combining automatic forecasting with analyst-in-the-loop forecasts for special cases, it is possible to cover a wide variety of business use-cases. The following diagram illustrates the forecasting process we have found to work at scale:



For the modeling phase of the forecasting process, there are currently only a limited number of tools available.We have frequently used Prophet as a replacement for the forecast package in many settings because of two main advantages:

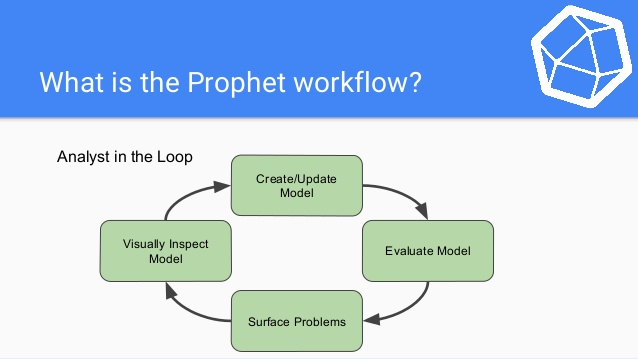
1. Prophet makes it much more straightforward to create a reasonable, accurate forecast. The forecast package includes many different forecasting techniques (ARIMA, exponential smoothing, etc), each with their own strengths, weaknesses, and tuning parameters. We have found that choosing the wrong model or parameters can often yield poor results, and it is unlikely that even experienced analysts can choose the correct model and parameters efficiently given this array of choices.
2. Prophet forecasts are customizable in ways that are intuitive to non-experts. There are smoothing parameters for seasonality that allow you to adjust how closely to fit historical cycles, as well as smoothing parameters for trends that allow you to adjust how aggressively to follow historical trend changes. For growth curves, you can manually specify “capacities” or the upper limit of the growth curve, allowing you to inject your own prior information about how your forecast will grow (or decline). Finally, you can specify irregular holidays to model like the dates of the New year, Thanksgiving and Black Friday.

**How Prophet works**

At its core, the Prophet procedure is an additive regression model with four main components:

* A piecewise linear or logistic growth curve trend. Prophet automatically detects changes in trends by selecting changepoints from the data.
* A yearly seasonal component modeled using Fourier series.
* A weekly seasonal component using dummy variables.
* A user-provided list of important holidays.

**Model Architecture and workflow**



**Important Hyperparameters:**

1. *Seasonality*- The seasonal component *s(t)* provides a flexible model of periodic changes due to weekly and yearly seasonality. Weekly seasonal data is modeled with dummy variables. Six new variables are added: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, which take values 0 or 1 depending on the day of the week. The feature Sunday is not added because it would be a linear combination of the other days of the week, and this fact would have an adverse effect on the model. And Yearly seasonality model in Prophet relies on Fourier series.
2. *Holidays and Events-* The component h(t) represents predictable abnormal days of the year including those on irregular schedules, e.g., Black Fridays.

To utilize this feature, the analyst needs to provide a custom list of events.

1. *Error-* The error term ϵ(t) represents information that was not reflected in the model. Usually it is modeled as normally distributed noise.
2. *Trend -* The Prophet library implements two possible trend models for g(t).The first one is called Nonlinear, Saturating Growth. It is represented in the form of the logistic growth model:

https://cdn-images-1.medium.com/max/1000/1*egEuwH9DTF_xa78AkRLEtg.png

where:

* C is the carrying capacity (that is the curve’s maximum value).
* k is the growth rate (which represents “the steepness” of the curve).
* m is an offset parameter.

This logistic equation allows modelling non-linear growth with saturation, that is when the growth rate of a value decreases with its growth. One of the typical examples would be representing the growth of the audience of an application or a website. Actually, CC and kk are not necessarily constants and may vary over time. Prophet supports both automatic and manual tuning of their variability. The library can itself choose optimal points of trend changes by fitting the supplied historical data.

Also, Prophet allows analysts to manually set changepoints of the growth rate and capacity values at different points in time. For instance, analysts may have insights about dates of past releases that prominently influenced some key product indicators.

The second trend model is a simple *Piecewise Linear Model* with a constant rate of growth. It is best suited for problems without saturating growth.

In short explanation-

**Seasonality & Holiday Parameters**

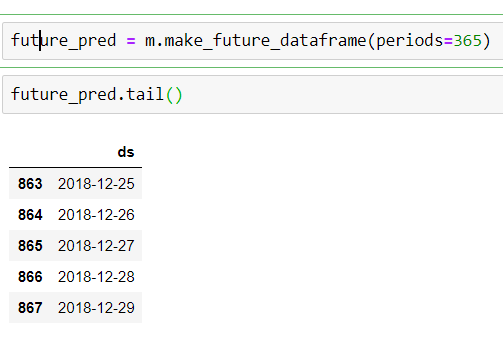
|  |  |
| --- | --- |
| **Parameter** | **Description** |
| yearly seasonality | Fit yearly seasonality |
| weekly seasonality | Fit weekly seasonality |
| daily seasonality | Fit daily seasonality |
| holidays | Feed data frame containing holiday name and date |
| Seasonality prior scale | Parameter for changing strength of seasonality model |
| Holiday prior scale | Parameter for changing strength of holiday model |

**Model Fitting-**

We start by creating an instance of the Prophet class and then fit it to our dataset by using “m.fit”.

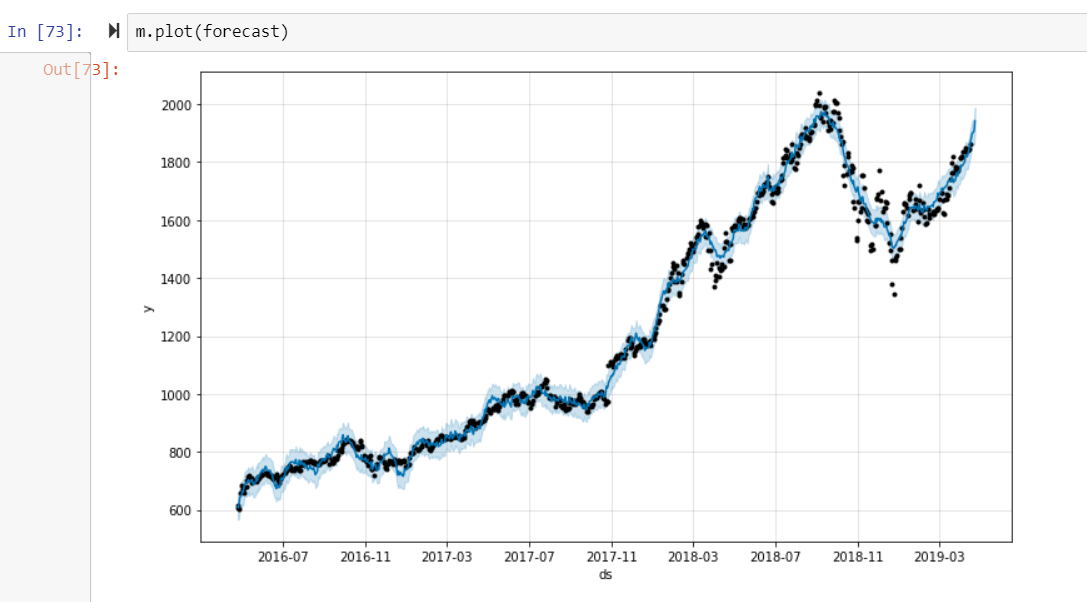
**Making Future Predictions**

The next step is to prepare our model to make future predictions. This is achieved using the Prophet.make\_future\_dataframe method and passing the number of days we’d like to predict in the future. We use the periods attribute to specify this. This also include the historical dates. We’ll use these historical dates to compare the predictions with the actual values in the ds column.



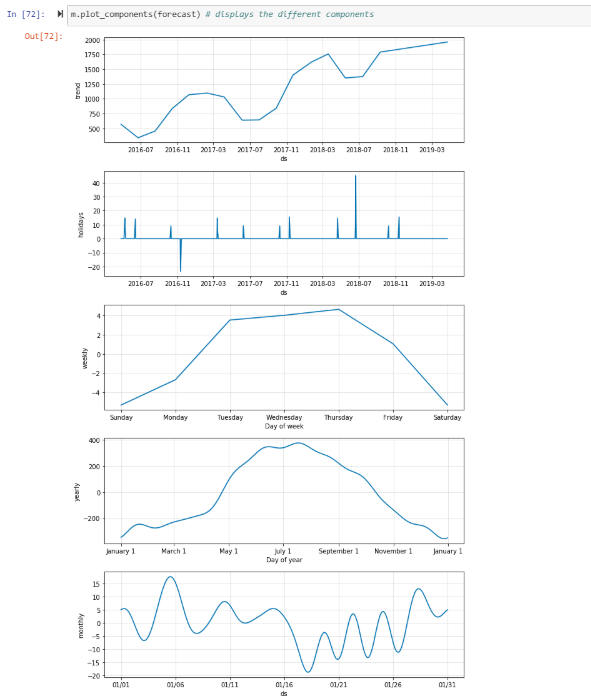
**Plotting the Forecasts** – (Forecast 7 days into future)

Prophet has an inbuilt feature that enables us to plot the forecasts we just generated. This is achieved using “m.plot()” and passing in our forecasts as the argument. The blue line in the graph represents the predicted values while the black dots represents the data in our dataset.



**Obtaining the Forecasts**

We use the predict method to make future predictions. This will generate a dataframe with a yhat column that will contain the predictions. However, we are mainly interested in ds, yhat, yhat\_lower and yhat\_upper. yhat is our predicted forecast, yhat\_lower is the lower bound for our predictions and yhat\_upper is the upper bound for our predictions.

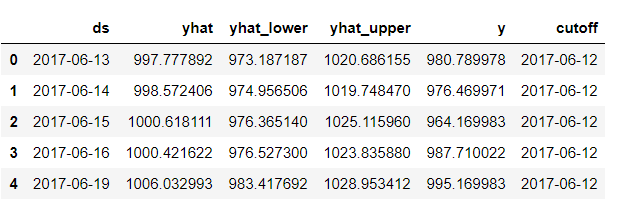
**For prediction models**

**Cross Validation**

Next let’s measure the forecast error using the historical data. We’ll do this by comparing the predicted values with the actual values. In order to perform this operation, we select cut of points in the history of the data and fit the model with data up to that cut off point. Afterwards we compare the actual values to the predicted values. The cross-validation method allows us to do this in Prophet. This method takes the following parameters as explained below:

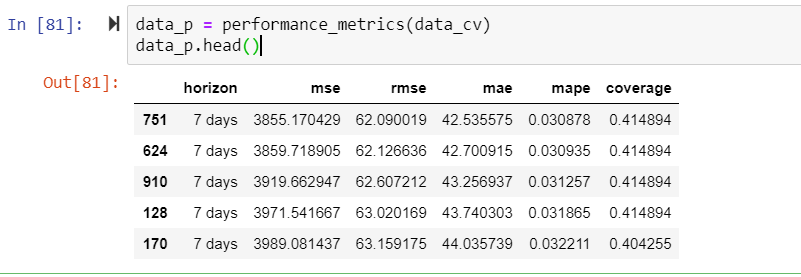
* Horizon the forecast horizon
* Initial the size of the initial training period
* Period the spacing between cutoff dates

The output of the cross-validation method is a data frame containing “y” the true values and “yhat” the predicted values. We’ll use this data frame to compute the prediction errors.



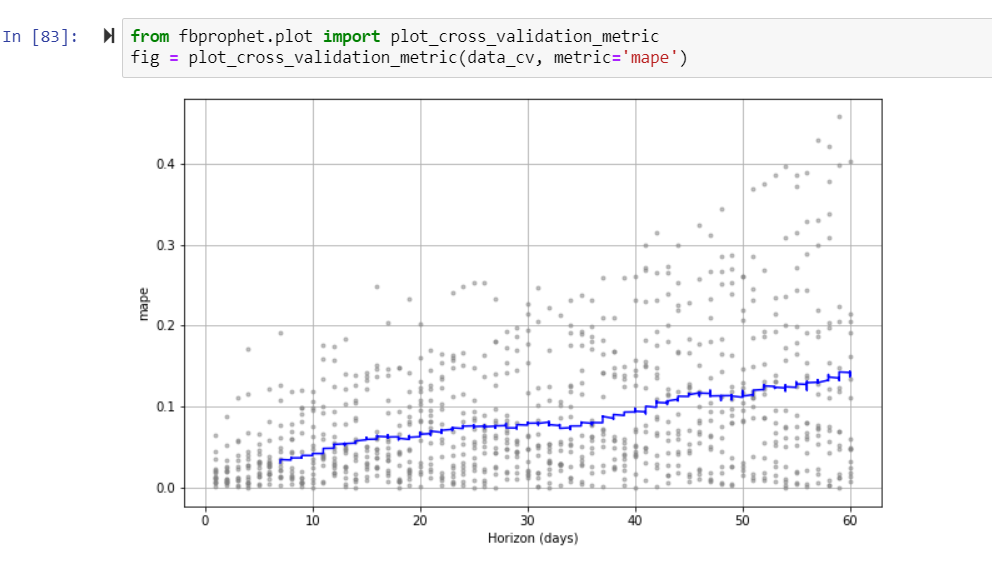
**Obtaining the Performance Metrics**

We use the performance metrics utility to compute the Mean Squared Error(MSE), Root Mean Squared Error(RMSE),Mean Absolute Error(MAE), Mean Absolute Percentage Error(MAPE) and the coverage of the the yhat\_lower and yhat\_upper estimates.



**Visualizing Performance Metrics**

The performance Metrics can be visualized using the plot cross validation metric utility. We visualize the RMSE as shown in the graph**.**



**Identifying Large Forecast Errors-** When there are too many forecasts for analysts to manually check each of them, it is important to be able to automatically identify forecasts that may be problematic.

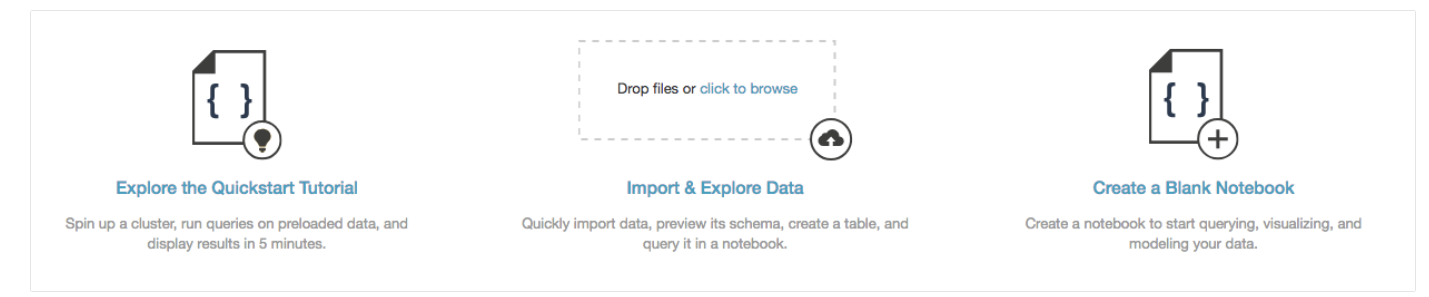
* When the forecast has large errors relative to the baselines, the model may be mis- specified. We can adjust the trend model or the seasonality, as needed.
* Large errors for all methods on a date are suggestive of outliers. We can identify outliers and remove them.
* When the SHF (simulated historical forecasts) error for a method increases sharply from one cutoff to the next, it could indicate that the data generating process has changed. Adding changepoints or modeling different phases separately may address the issue.

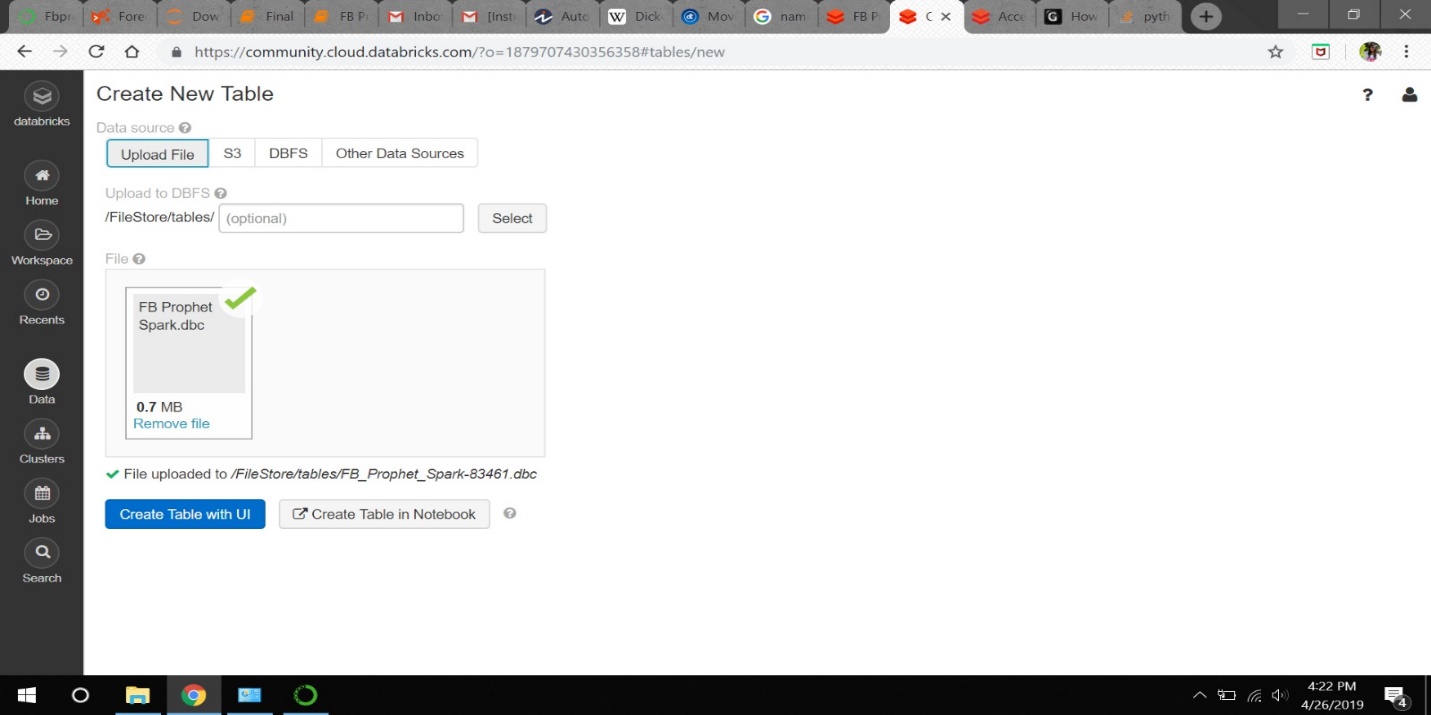
**3 How to run the code in Databricks-**

**Databricks**- Databricks is a managed platform for running Apache Spark and provides a host of features to help its users be more productive with Spark. It's a point and click platform for those that prefer a user interface. However, this UI is accompanied by a sophisticated API for those that want to automate aspects of their data workloads with automated jobs.



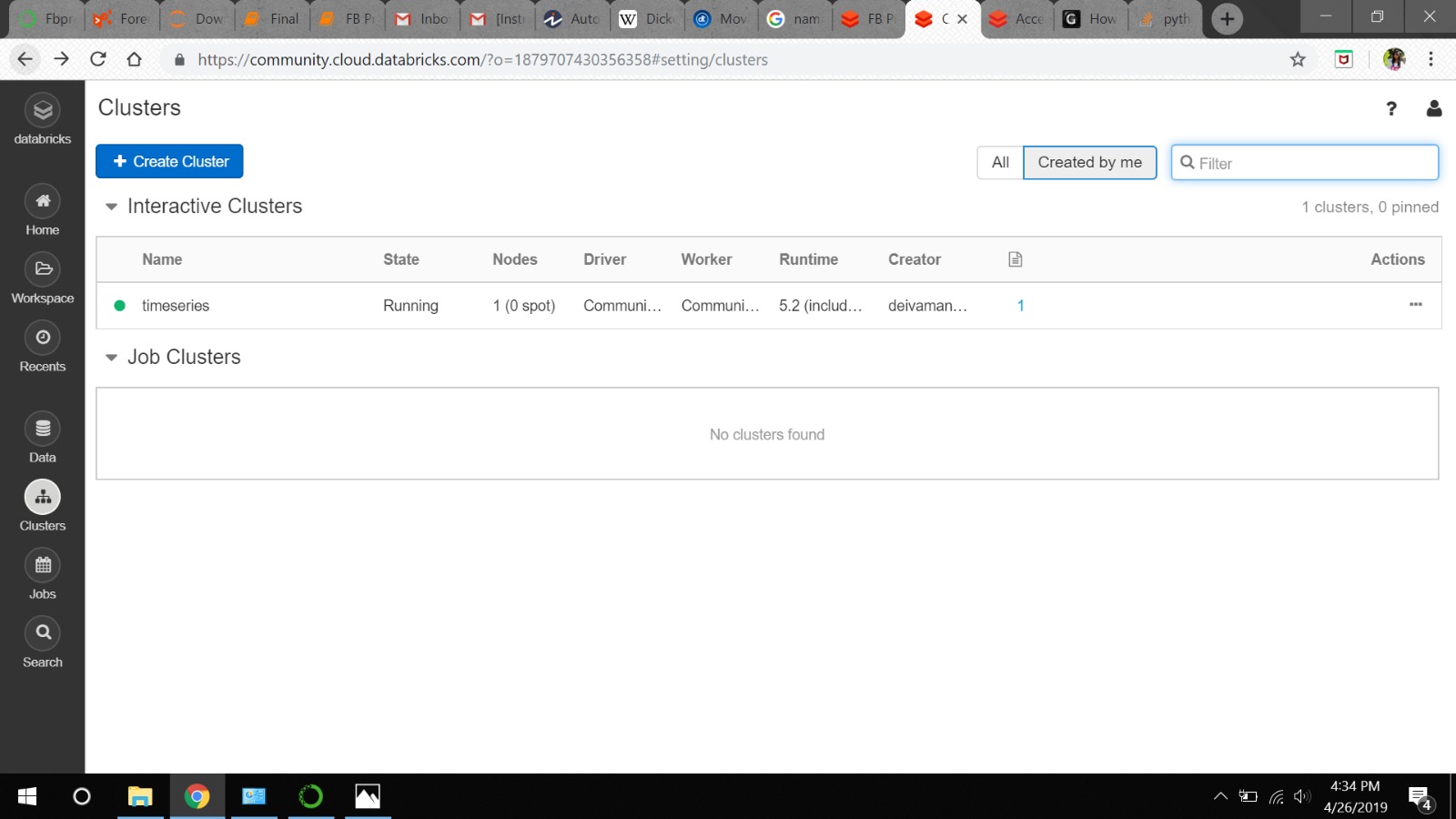
**3.1Import data**- Drop files into or click to browse in Import & Explore data box on landing page.



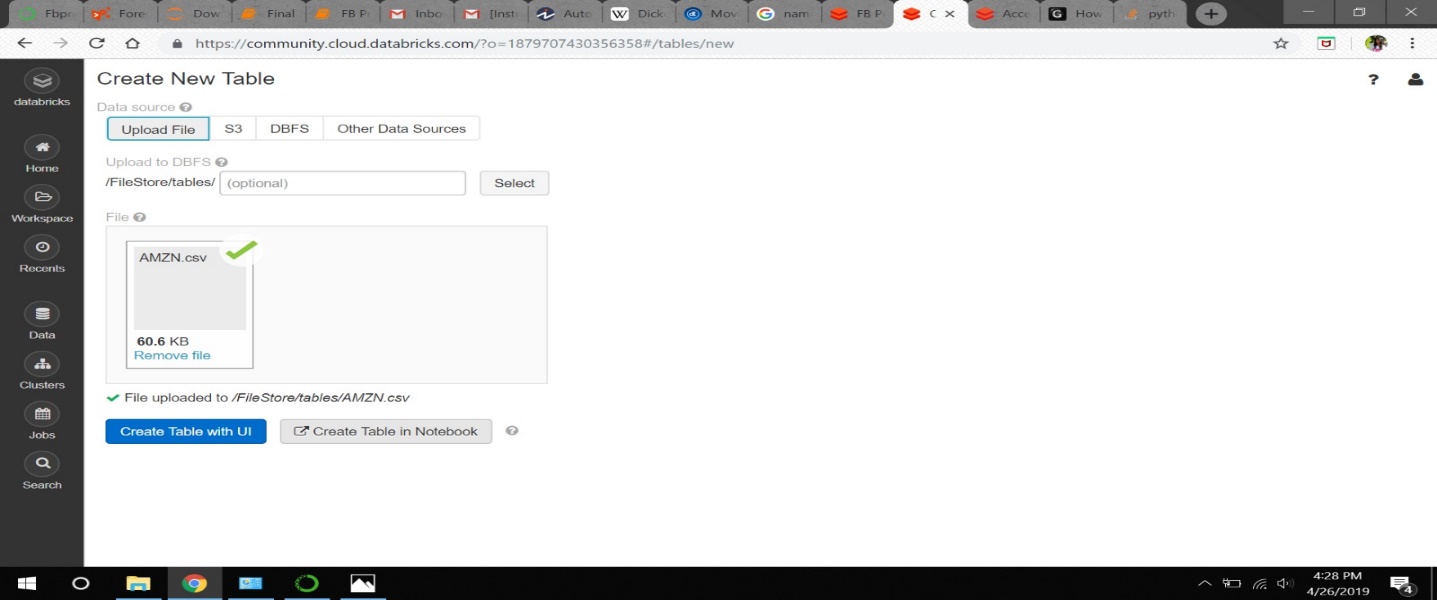


**3.2 Creating Cluster:**

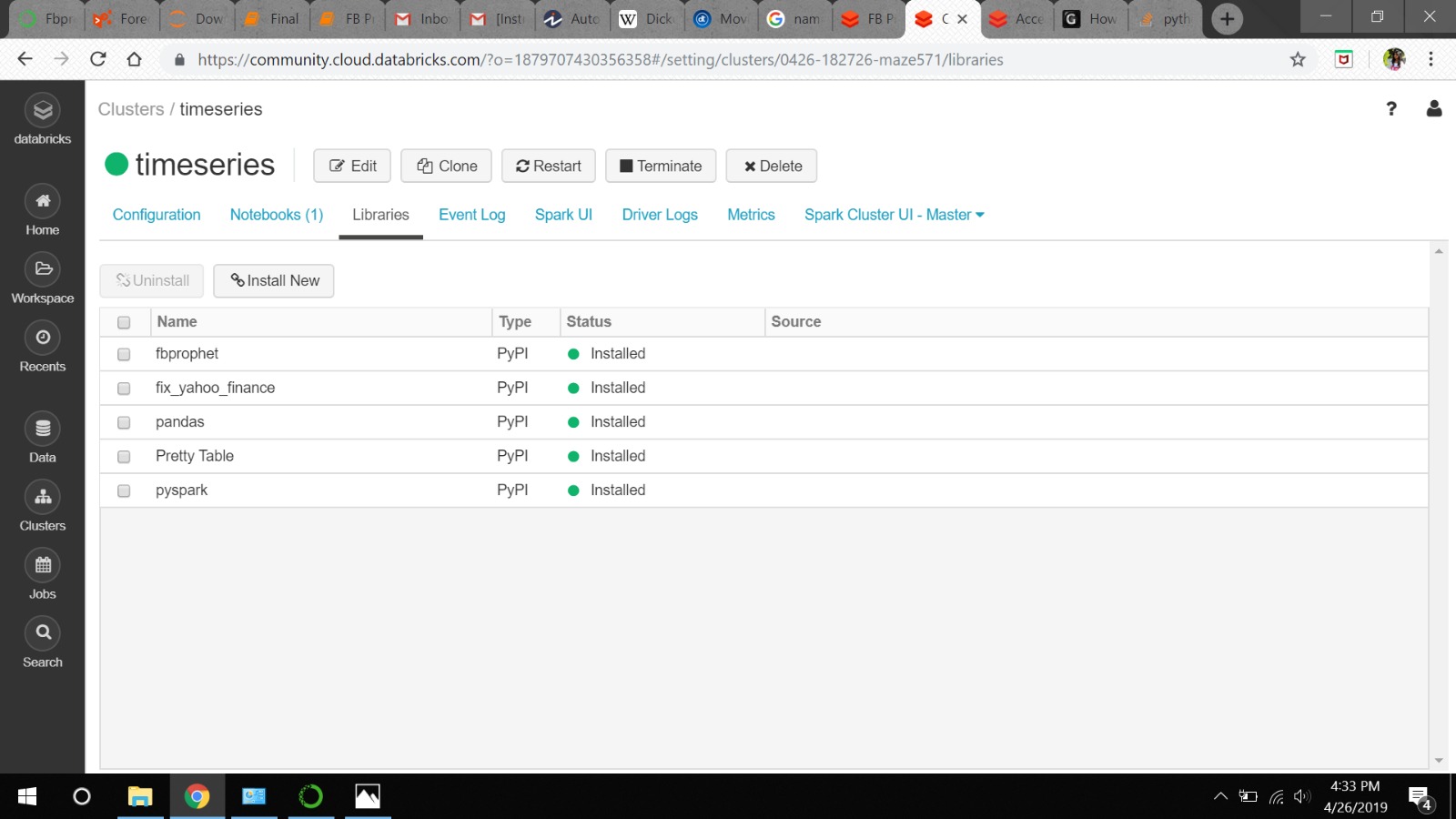
In Databricks you can create two different types of clusters: standard and high concurrency. Standard clusters are the default and can be used with Python, R, Scala, and SQL. High-concurrency clusters are tuned to provide the efficient resource utilization, isolation, security, and the best performance for sharing by multiple concurrently active users. High concurrency clusters support only SQL, Python, and R languages.



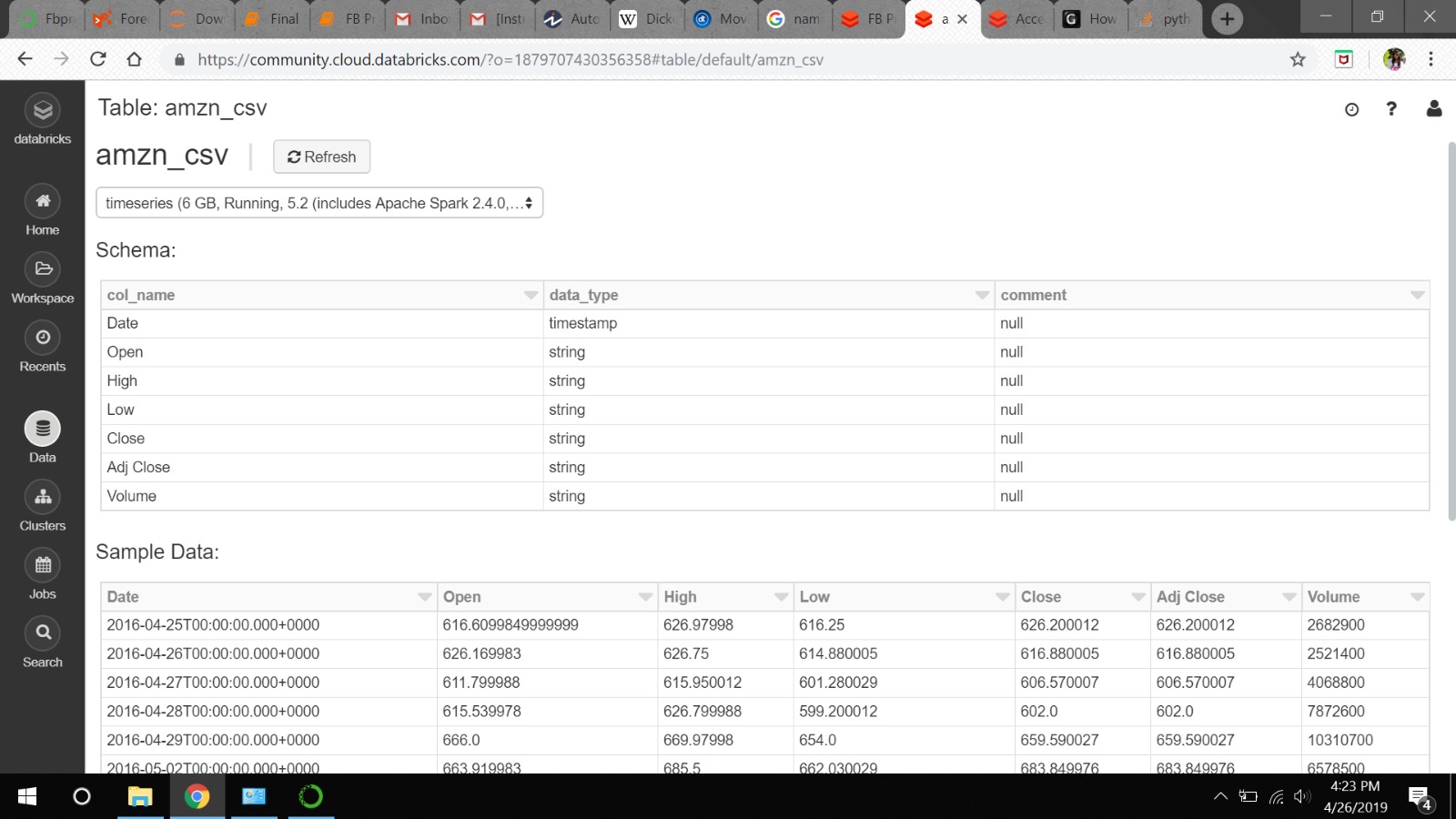
**3.3 Importing CSV file-**



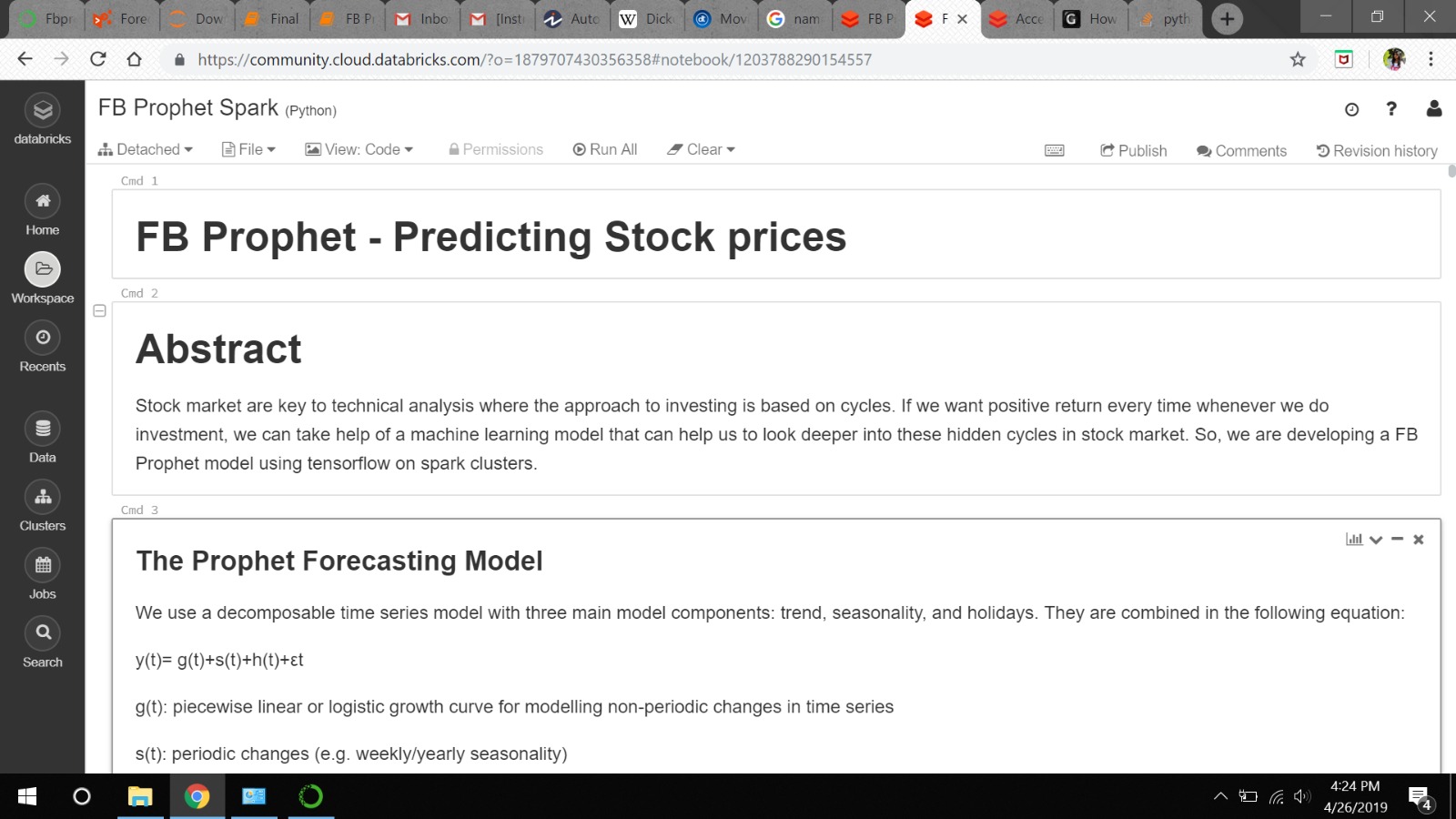
**3.4 Importing Libraries**- The below libraries are the required for our model. Steps to getting libraries installed- Select the running cluster which you have created earlier then click on the libraries you want to install for your model and done(required libraries installed)



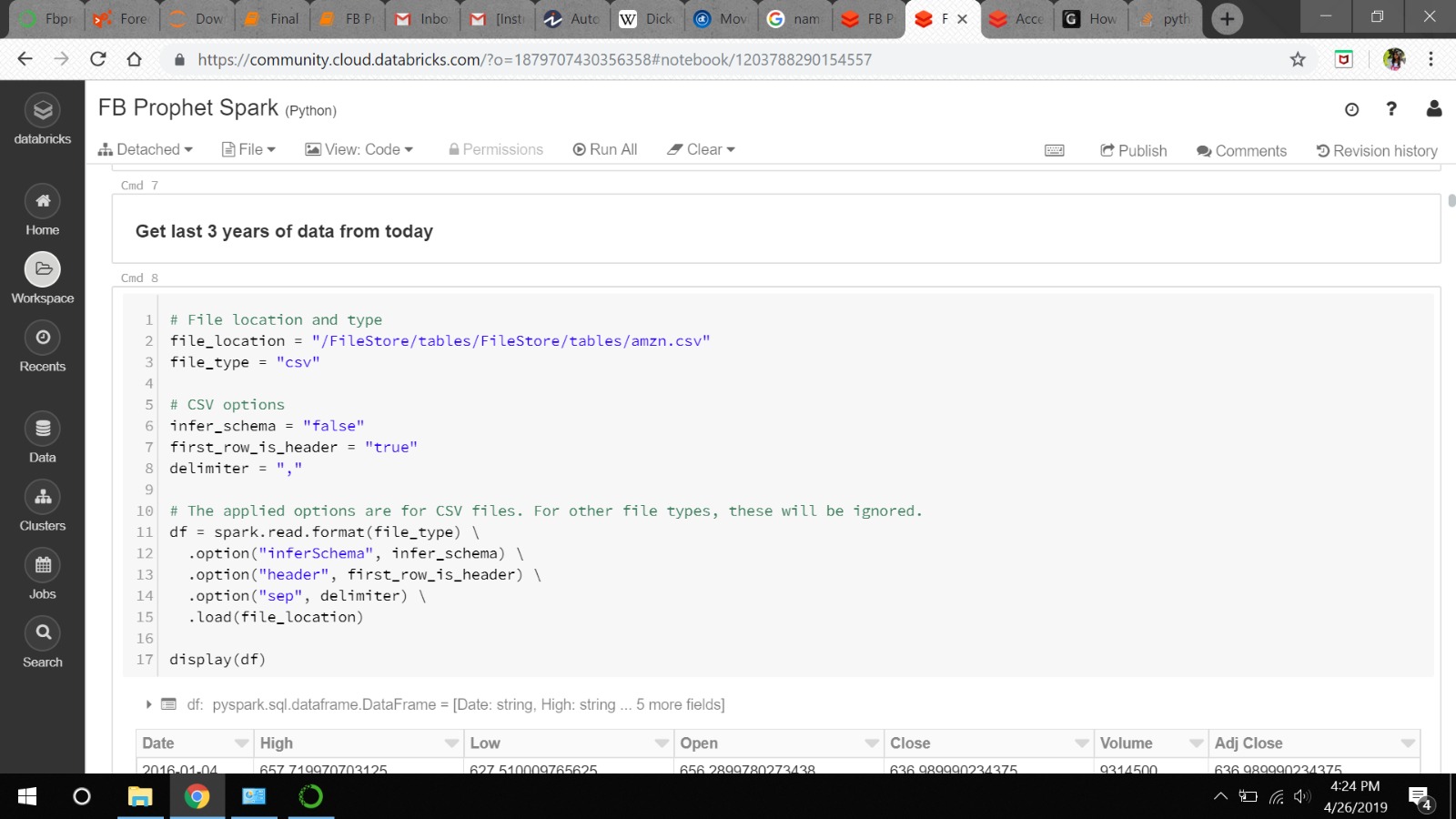
**3.5 View databases and tables**- Click Data Icon in the sidebar. Databricks selects any running cluster to which you have access. The Databases folder displays the list of databases with the default database selected. The Tables folder displays the list of tables in the default database.



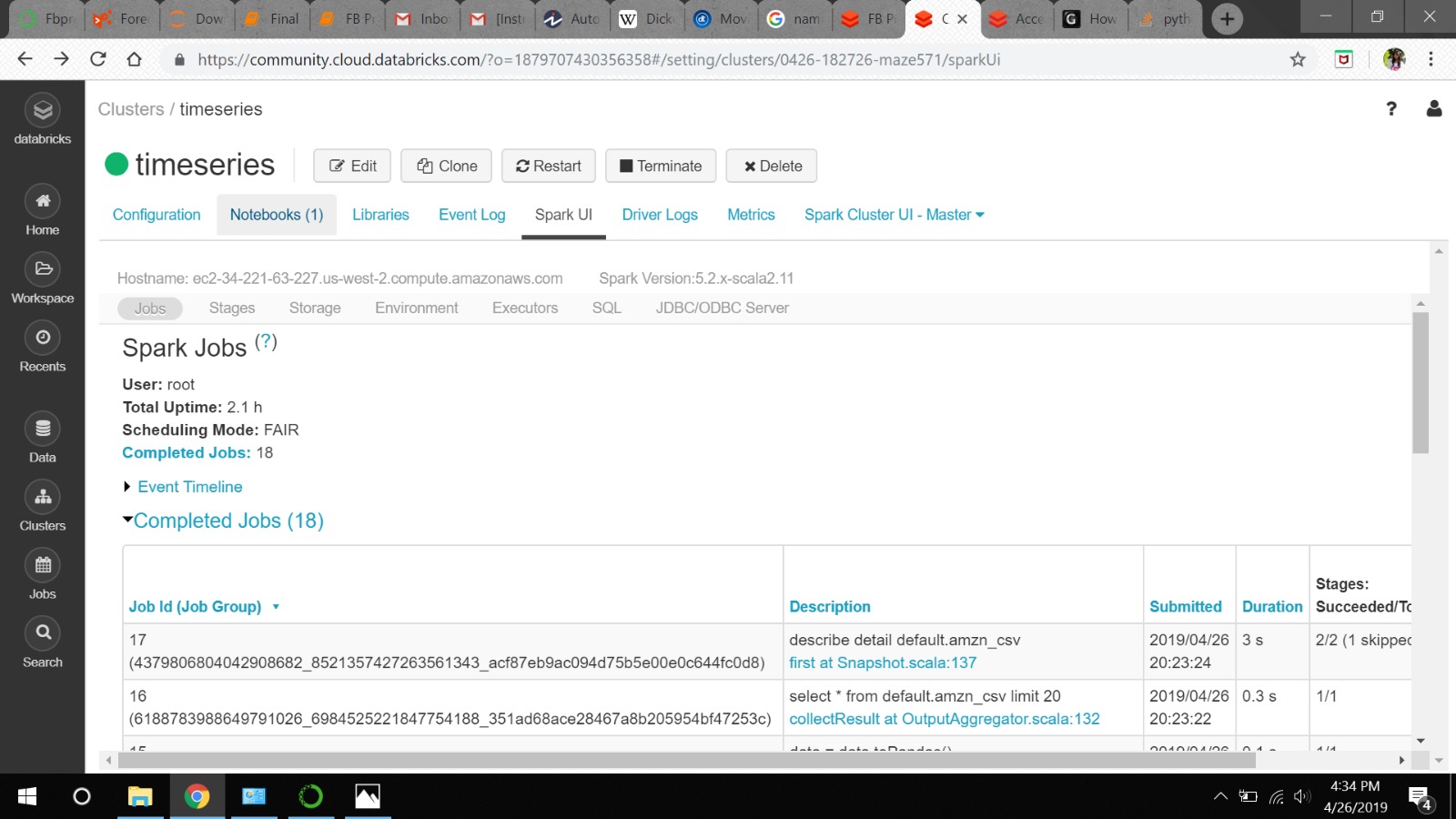
**3.6 Notebook in databricks environment-**

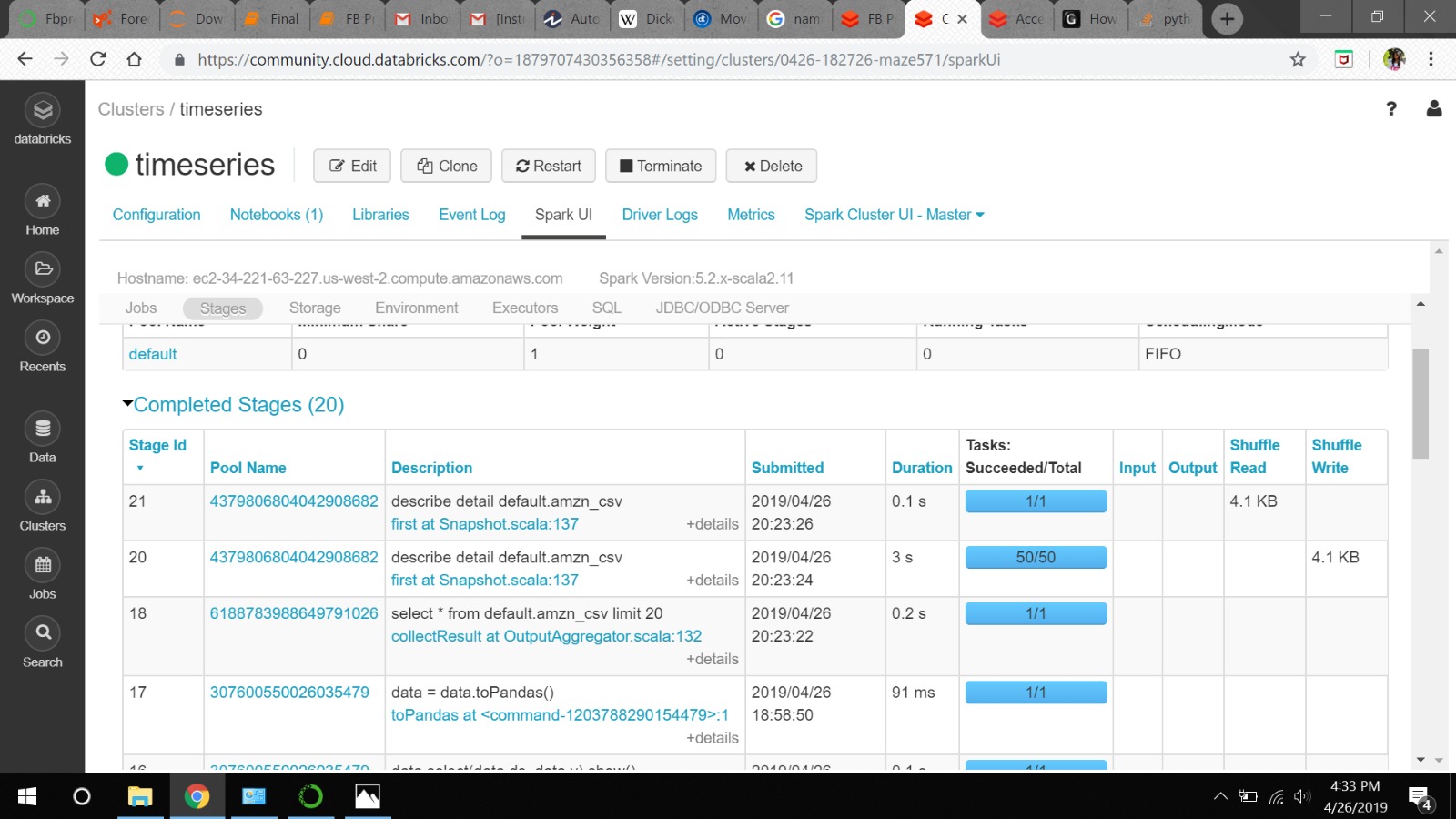
****

**3.7 Creating and accessing file path in databricks environment**

****

**3.8 Jobs completed-**

****

****

**LSTM Model**: To implement a deep learning technique in TensorFlow, we are implementing a simple LSTM model.

**Databricks Platform**: Databricks is a convenient platform to run Apache Spark. The code is deployed in Databricks Spark Cluster which has advantages of computation efficiency and handling bigdata.

**GCP Dataprocs**: We set up a storage bucket in GCP dataprocs in which our code was imported to. A cluster was created to run the code from the storage bucket. However, we were not successful in completion of GCP Dataprocs deployment.

**Conclusion** –

1.We created the FB Prophet model including the required seasonality’s for our timeseries data.

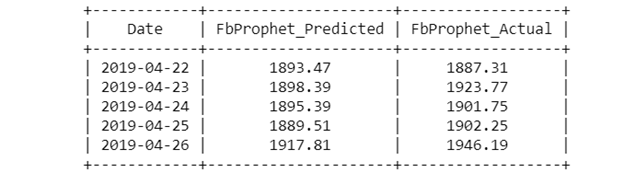
2.Holiday parameters are included which contains long weekend holidays and Prime days for our Amazon stocks as they can have an impact on the company's sales.

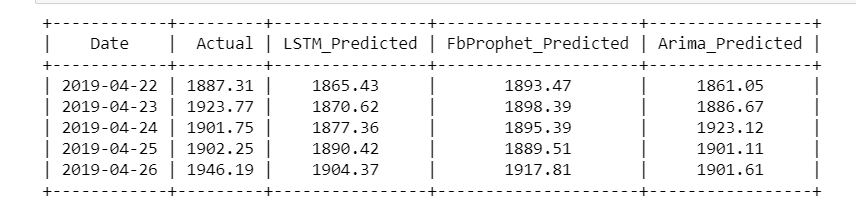
3.We have tuned different hyperparameters of different components and chose the set of parameters that gave better results for every component. Later we exported the predicted vs actual data in a csv file for comparison.

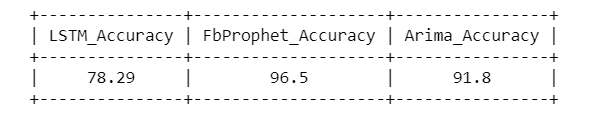
4.Also, we performed cross validation using performance metrics which gave MSE, RMSE, MAE, MAPE values.

5.We used MAPE value to check the performance of our model and we got approximately 0.08% error for a horizon of 30 days.

6.We have achieved an approximate accuracy of 96.5% for the dataset on our created model.

****



**Advantages:**

* Open Source
* Accurate and fast
* Allows for a large number of people to make forecasts, possibly without training in time series methods;
* Tunable parameters
* Available for both Python and R

**Acknowledgment**

We would like to show our gratitude to professor Nik Bear Brown and project manager Balaji Mudaliyar for guiding us during this project.

**References-**

1.Asidlo.“Asidlo/Sparkprophet.” *GitHub*, github.com/asidlo/sparkprophet/blob/master/sparkprophet.py.

*2.Facebook.github.io*, facebook.github.io/prophet/docs/.

3.Choudhary, Ankit, and IIT Bombay Graduate. “Time Series Forecasts Using Facebook's Prophet (with Python & R Codes).” *Analytics Vidhya*, 20 Sept. 2018, [www.analyticsvidhya.com/blog/2018/05/generate-accurate-forecasts-facebook-prophet-python-r/](http://www.analyticsvidhya.com/blog/2018/05/generate-accurate-forecasts-facebook-prophet-python-r/).

4.“Powerful Python Data Analysis Toolkit¶.” *Pandas*, pandas.pydata.org/pandas-docs/stable/.

5.“KDnuggets.” *KDnuggets Analytics Big Data Data Mining and Data Science*, [www.kdnuggets.com/2018/11/sales-forecasting-using-prophet.html](http://www.kdnuggets.com/2018/11/sales-forecasting-using-prophet.html).

6.Peerj.com. (2019). [online] Available at: https://peerj.com/preprints/3190.pdf [Accessed 27 Apr. 2019].

7.Towards Data Science. (2019). *Predicting the ‘Future’ with Facebook’s Prophet*. [online] Available at: https://towardsdatascience.com/predicting-the-future-with-facebook-s-prophet-bdfe11af10ff [Accessed 27 Apr. 2019].