**Q1 What is the fundamental differences between Spark ML and normal ML approaches? Please state your findings with examples (i.e. array vs distributed array). What are the fundamental approaches for ML Pipeline in distributed computing?**

Spark ML uses RDD’s to store values. RDD’s are partitioned among the nodes. Normal ML approaches uses datasets and arrays, which are locally stored in a single machine without having a capability of parallel computing.

ML pipelines are iterative as their nature. Protecting this nature in distributed approaches are tricky but two basic approaches are founded.

***Data Parallelism:***

Data is partitioned and distributed among the workers. Each worker applies the model to its own part. Synchronization is a critical issue, slow workers should catch the other ones.

***Model Parallelism:***

Different parts of the model are used to process the same dataset for each worker. Model parameters needed to split to use this approach.

Also several techniques are founded to model the worker communications and flow of data. Each of them has its own limitations, so should be used considering trade-offs.

***Bulk Synchronous Parallel***

Map-Reduce logic relies behind it. Finished workers are waiting for the others when they reach a barrier. This situation creates a cost surely but the correct solution is guaranteed with this type of ML programs.

<https://doi.org/10.1007/978-0-387-09766-4_311>

***Stale Synchronous Parallel***

Workers can access the model values from a cached version. Allowed to move forward for a certain number of iterations. There is no specific barrier.

***Approximate Asynchronous Parallel***

“Provides approximate processing by synchronizing each worker with a subset of workers at each iteration.” *(Asim Kadav and Erik Kruus* , <https://arxiv.org/pdf/1612.08608>*)* Reduces the staleness of the message (data) so workers can see them without a delay.

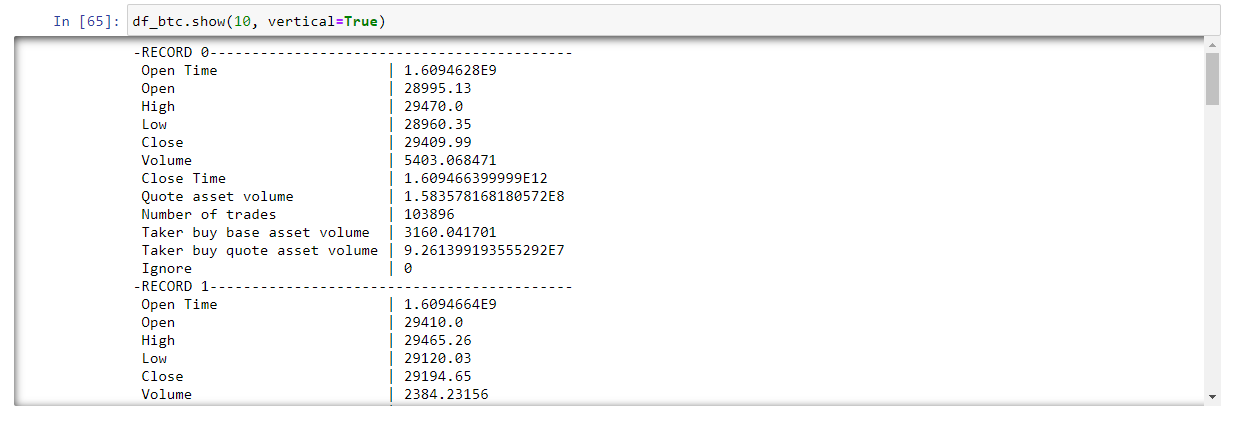
**Q2 What is dense vector and sparse vector in Spark? Why do we need to separate vectors?**

These are two different representations of a vector. Sparse vector is used when its components are mostly zero. So every element of it is not needed to be stored, and its represented with a technique that only non-zero elements are shown. Indices and their corresponding values are given. With this method, it can save us a lot of storage and computation power in some models. Dense vector is the traditional representation of a vector, it stores every entity in the matrix.

**Q3 Please use fundamental statistics commands (Summary and descriptive statistics, Sample covariance and correlationi, Cross tabulation, Frequent items, Mathematical functions) together with colStat (multivariate statistical summary) in your OWN data pipeline. You can also use your Projects data.**

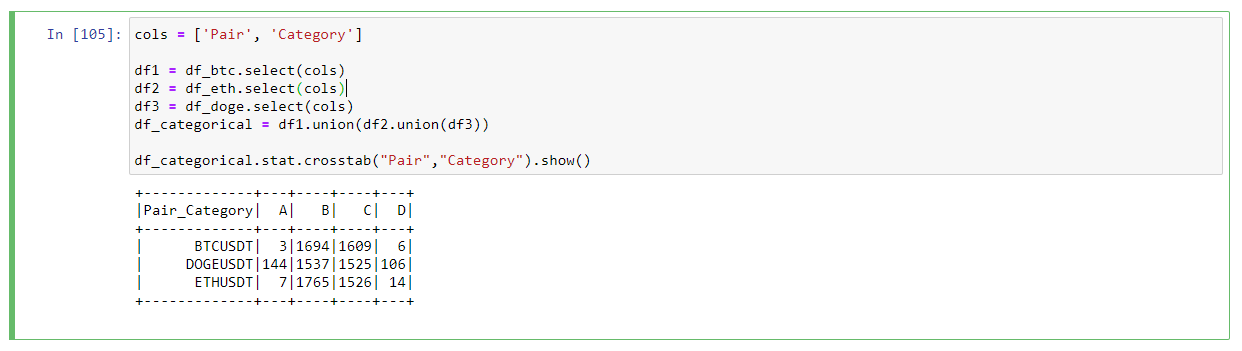
For this question I used Bitcoin, Ethereum and Dogecoin values with respect to Tether. Using Binance API, gathered them and stored in a csv. As can be seen below, Binance API gives 12 columns including OHLC values, Open and Close Time, Volume. These are the most important ones for now. But storing all is the desired option for now since in the future phases of the project they might be useful.





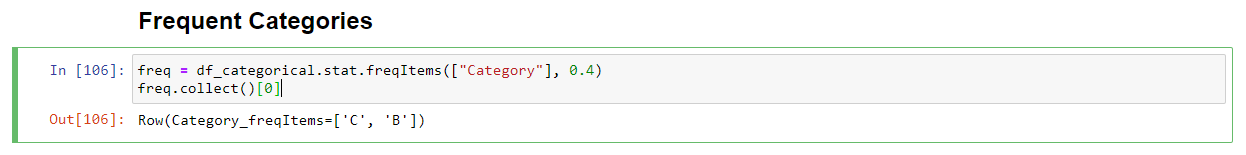
To use cross tabulation, I needed a categorical data. So I categorized the price movements of the pairs in a proper way. Movements bigger than 5 percent within a hour relies in category A. 1 to 5 percent changes goes in category B. C contains the movements between -1 and -5. Lastly, D has the changes lower than -10.





Using crosstab function, observed the category distribution of the pairs. Can be seen that DOGEUSDT has the most values in category A. But also in D ! It is far more volatile than the others.

After that I tried freqItems method to get the most frequent categories among all. Threshold was 0.4. And it gives the result as expected, similar with crosstab.



I have previously created a csv for comparison of the prices of BTC and ETH. Using that csv, constructed a dataframe and RDD with respect. Mathematical formulations used in the class were the reference point. Skewness, Kurtosis, Mean are calculated using these formulations.



Lastly, colStat was used to get the Norm L1, L2, max and min values of the BTC closing prices.

