**Project Name:** Bitcoin Price Prediction

**Github Link:** https://github.com/projectsforstudents2022/Bitcoin\_Price\_Prediction.git

**Why was this project created?**

Due to its phenomenal growth, unexpected volatility, and intriguing uses, Bitcoin has recently gained notoriety. The number of classifiers already developed to predict changes in the price of bitcoin is the finest indication of how this increasing interest in the public sphere has boosted interest in the data science field.

**What problem is it solving?**

In order to anticipate the volatility of bitcoin the next day, we plan to use NLP algorithms to evaluate news stories over the previous two years (2016–2020). The model was able to achieve high test accuracy in predicting the sign of the change in bitcoin's price the next day even with a small training/test set of only 640 samples.

**Entire explanation of project**

* **PROPOSED APPROACH**

Many of the canonical categorization models that are at our disposal were used by me. I first attempted to classify the sign change of bitcoin for the upcoming day by using a logistic regression on my collection of word features. A supervised learning algorithm called logistic regression seeks out the weights that model a link between a set of input features and the corresponding set of labeled outputs most accurately (of which there are two possible options: 1 or 0). This model's excellent training outcomes outperformed its bad testing results, which indicates strong bias and overfitting.

I made use of TensorFlow's Deep Neural Network classifier. This Deep Neural Network consists of several interconnected "neurons" or layers, each of which has weights from the input features or layers before it. The values of the input feature values or the value of the preceding layer are then added to these weights, which are subsequently subjected to a "activation function" to make them normal. The goal of this model is to identify a relationship that is more complex than a simple linear relationship and can be expressed using a more detailed language.

Algorithm for creating next word prediction model :

**Step 1:** Import Libraries & Load Dataset

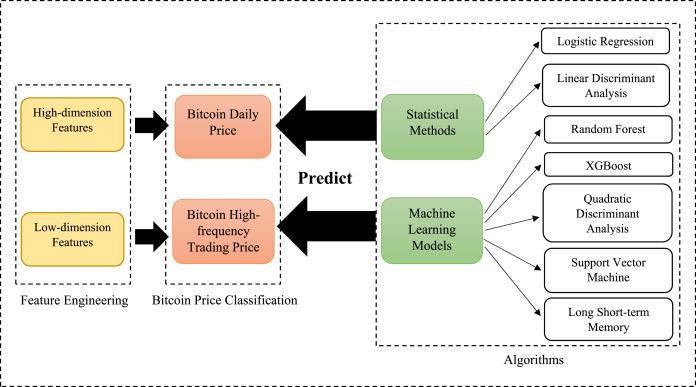
**Step 2:** Data Cleaning

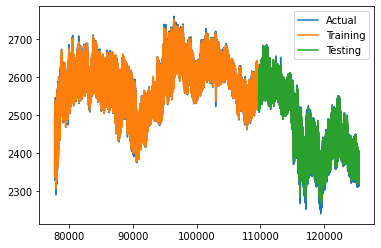
**Step 3:** Label Encoding

**Step 4:** Build Neural Network

**Step 5:** Train Model

**Step 6:** Testing & Visualization

* **DATA FLOW DIAGRAM**
* **RESULT**



* **CONCLUSION**

In comparison to simpler logistic and linear regressions, I was able to achieve the lowest loss on the test set and the highest accuracy when I used a deep neural network, which was to be expected given that this deep network can be thought of as layering several of these simpler algorithms together and combining their results. Accuracy on test case is 68%.