# CASSANDRA

## TEAM Name- Sigmoids

**Members** – 1) Ajinkya Rajendra Mohale, Roll No - 20095067

2) Ankit Kumar, Roll No - 20095008

3) Shubham Kumar, Roll No - 20095111

System Description-

Firstly, we imported the required libraries and did load the train and test datasets separately.

Then Checked for the nan values in data frame.

Then we plotted graphs. The first one was the vendor names with their frequency. The others were vendor name vs no of days until payment, etc.

Based on these graphs, we decided to add new columns, namely the difference between invoice date and due date, another, the difference between invoice date and Created date. We also separated the columns of all dates into Date, month, year and added an additional feature of weekday, ranging from 1 to 7.

We also encoded the year columns into numbers from 1 to 7.

Then we plotted a heat map and looked for some more possible relations.

Then we spitted our train dataset into training and testing further, with split size 0.3

Finally, we made our model ready using the Gradient Boosting Regressor with estimators=1000, random state=1.Also, other regressors were used but gave less accuracy than GBR.

Then we calculated the mean squared error loss on train data to get idea of performance of our model, and manually tuned the hyper parameters according to them.

Finally, we merged the name data frame and our predictions and submitted.

Firstly, we completely dropped the Vendor_Names, Description columns, and only separated the columns of date, created new columns of difference between invoice and due date. Then, we got score of 30.

After that we did some tuning of parameters and tried other regression algorithms like random forest, xgboost, linear regression, but no significant improvement in score occurred.

The first breakthrough occurred when we encoded the year column to integers from 1,2,3,4,5,6,7, and our score dropped to around 26, all other conditions kept same.

After this, we included the dropped columns of Vendor_Name and descriptions, and encoded them with numbers while compiling both the test and train datasets to cover all entries and prepare a universal dictionary. After this significant improvement occurred and score further dropped below 25.

We also tried normalising various data columns, individually and combined, but it did not reduce the score.

At last, all the improvements till the final score were brought by parameter tuning only.