# Machine Learning - Bond Liquidity Prediction

Team: as_dg_sc

Historical data for 3 month time period is given. But, upon analyzing the data, we realized that the minimum date for each transaction takes place is 16th March 2016 and maximum date for which it happens is 9th June 2016. So, effectively time duration for which data is given is 86 days.

- First of all, we converted nominal features to integer values, i.e., industryGroup10 was converted to 10 as it should be evaluated by the machine learning algorithm we later apply.
- All the values containing **Nan** were replaced with 0.
- We have removed features containing dates like **issueDate**, **maturity**, **ratingAgency1EffectiveDate** .
- We have categorized features into two parts: ***Static Features*** and ***Time-Dependent Features***. Static Features contains features which are time invariant like **issuer**, **market**, **amtOutstanding**, **collateralType** etc. Time-Dependent features contains features that are vary on daily basis.
- For each bond, we have calculated the sum of buy volumes for a given day and considered it as a time-dependent feature for that bond. This way, we come up with 86 time-dependent features for each bond which contain sum of values of buy volumes for each day. Similarly, we can calculate 86 time-dependent features for sell volumes. We aim to evaluate the buy and sell volumes separately i.e., train a regressor separately for buy and sell volumes.
- We have assumed that sequence of buy/sell volumes for each day follows a time series i.e., the volumes are dependent on values of buy/sell volumes of last few days. We have taken 85 days into account. So, value for buy/sell volumes for 86th day will depend on values of last 85 days.
- After predicting the value of buy/sell volumes for 87th day using 2nd to 86th day, we multiplied it by factor of 3 assuming average will be same on 3 days. This assumption has been taken due to lack of accurate answers, i.e., 87th day has been predicted by our algorithm. So, we do not want to use it to predict for 88th day as it would lead to erroneous prediction, Rather, we considered it to be same and multiplied the result of 87th day by 3.

After preparing and preprocessing the data, we generated 6 Data Frames:

B_train_x : contains static features and buy time dependent features from *Day1* to *Day85*
B_train_y: A vector of values which represent the sum of buy volumes of each bond for *Day86* which is used to train the model.
S_train_x : contains static features and sell time dependent features from *Day1* to *Day85*
S_train_y: A vector of values which represent the sum of sell volumes of each bond for *Day86* which is used to train the model.

B_test_x: contains static features and time dependent buy features from *Day2* to *Day86*
S_test_x: contains static features and time dependent sell features from *Day2* to *Day86*

We have applied 4 different approaches and combined their results.

*Approach 1:* **Gradient Boosting Algorithm** for Regression was applied. After tuning its parameters, we arrived at conclusions, learning rate = 0.001, n_estimators= 1000
*Approach 2:* **Random Forest Regressor** was applied as it uses random trees in large numbers to predict for a given instance. Each random tree is a decision tree which is trained to predict Yes or No based on a specific property. Its parameters were taken as n_estimators = 1000.
*Approach 3:* **Linear Regression** was applied as its ensemble gives good results when combined with Random Forest Regressor.
*Approach 4 (Statistical Approach):* We have calculated the summation of buy volumes and sell volumes for each bond for all days i.e. 90 days. Then, we assumed that the distribution of sum of volumes for bonds should be equal, i.e., the average sum of buy/sell voulmes for each day is constant. So, sum of buy volumes for 3 days for each bond can be thought of as summation of buy volumes for 90 days divided by 30.

Sum for 3 days = 3 *(Sum for 90days) / 90

After predicting the values for bonds, we checked whether the value becomes negative or not. If the algorithm had predicted it as negative, we overrode it to 0 as sum of volumes for a bond cannot be negative.

After computing sum of buy/sell volumes for 3 days by each approach, we take the average of all 4 approaches and create output set which contains values of buy/sell volumes for each bond.

```
output_set[i]['buy']=(mean_B[i]+3*(Btest_y1[i]+Btest_y2[i]+Btest_y3[i]))/4
output_set[i]['sell']=(mean_S[i]+3*(Stest_y1[i]+Stest_y2[i]+Stest_y3[i]))/4
```

where *Btest_y1, Btest_y2, Btest_y3* represent values by approaches 1, 2, 3 for buy volumes.
*Stest_y1, Stest_y2, Stest_y3* represent values by approaches 1, 2, 3 for sell volumes.
*mean_B / mean_S* represents values predicted by statistical approach for buy/sell volumes.

After running the algorithms several times, the best accuracy achieved was nearly 65%.
This accuracy can be justified as for several bonds, number of transactions taking place within 3 months is low or almost zero. So, the predicted values of volumes for several bonds are zero which may not the real scenario.