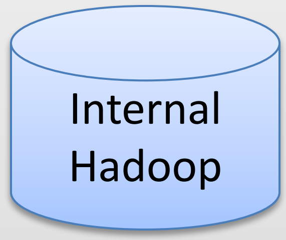
Cash Withdrawal Use Case

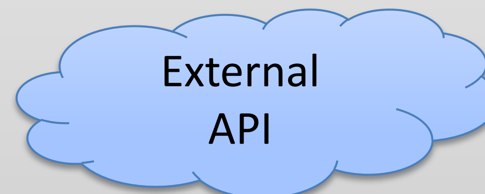
Internal Hadoop data used for our analysis:

- tb_gbase_balance_sheet
- cust_attribute_code_table
- industry_code_table



External data used for our analysis:

- **Structured Data**
  - Historical daily SGX price from Yahoo finance: SGX
  - Historical daily Nasdaq Exchange Price from Yahoo finance:NASDAQ
  - Historical daily S&P Exchange Price from Yahoo finance: S&P
  - Historical daily Nikkie Price from Yahoo finance: NIKKIE
  - Customer Equity price from Bloomberg terminal for two customers
- **Unstructured Data**
  - News related to customers OG30, YW75, IK16, LE80 from newsapi.com using API call from python and kafka(hadoop)
  - Top 20 news headline around the worldwide from KAGGLE



The period of study was from **2015-01-02 - 2018-05-31(YYYY-MM-DD)**
Total no of customer **458**
Account which was studied was **'TIME DEPO CUST'**
Currency Taken was **USD**
Total No of Raw Transactions: **530110**
Total No of Transactions: **191353(Grouped by each customers on each day)**

The period of study was from **2015-01-02 - 2018-05-31(YYYY-MM-DD)**
Total no of customer **392**
Account which was studied was **'TIME DEPO CUST'**
Currency Taken was **SGD**
Total No of Raw Transactions: **533372**
Total No of Transactions: **169789(Grouped by each customers on each day)**

What is Happening ?

- First we analysed the existing data to know what is the current situation in the bank. Who are **our good, loyal, risk and lost customers**.

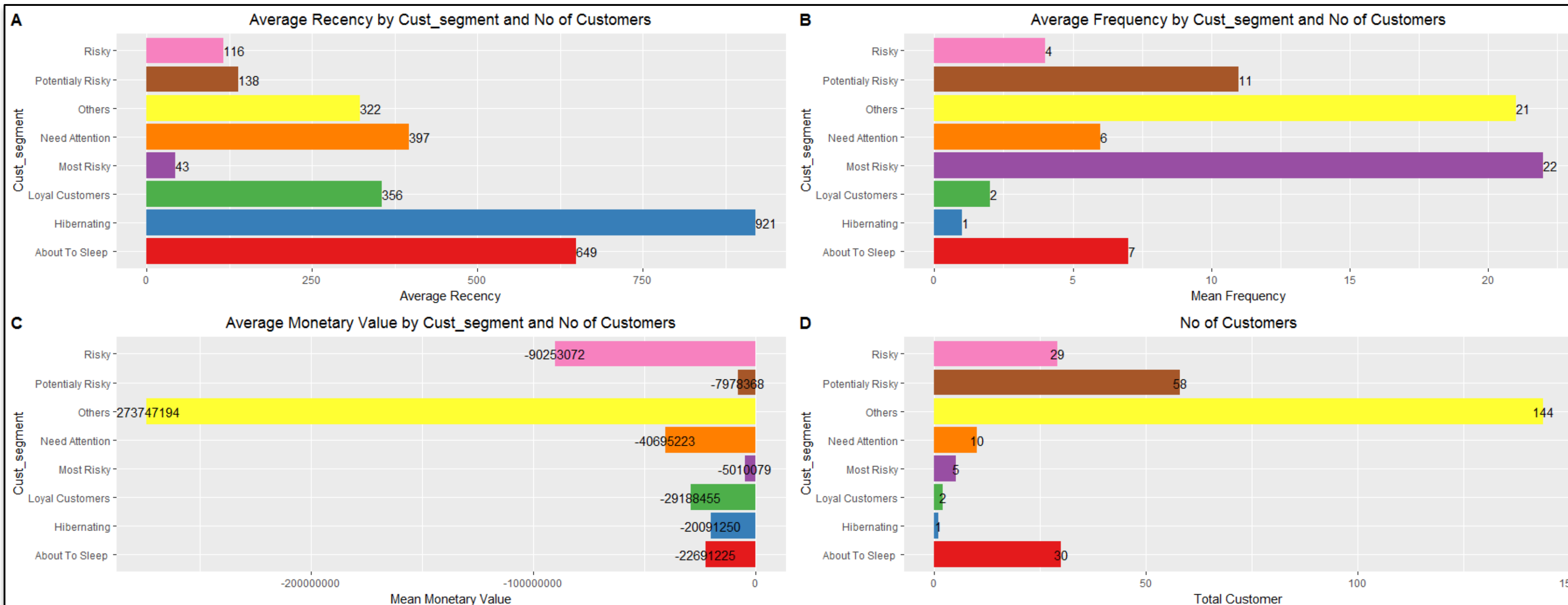
- We decided to use **RFM** for this analysis which is classification of customers based on their **recency, frequency and monetary value**. The next step of our analysis will be unsupervised machine learning to classify these customers using **K-means clustering**.

RFM & Clustering

RFM Analysis(SGD)

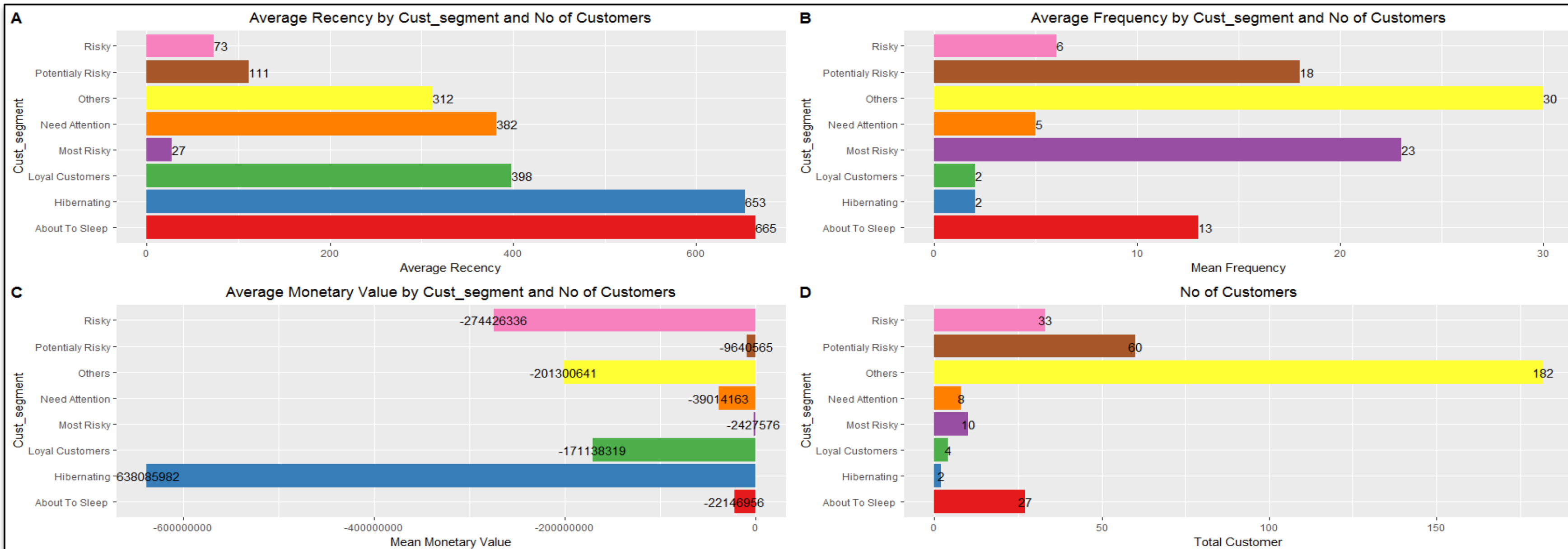
The period of study was from **2015-01-02 - 2018-05-31(YYYY-MM-DD)**
Total no of customer **392**
Account which was studied was **'TIME DEPO CUST'**
Currency Taken was **SGD**
Total No of Raw Transactions: **533372**
Total No of Transactions: **169789(Grouped by each customers on each day)**

- **RFM** (recency, frequency, monetary) analysis is a behaviour based technique used to segment customers by examining their transaction history how recently a customer has withdrawn/purchase (recency)
- how often the customer withdrawn/purchase (frequency)
- how much the customer withdrawn/spends (monetary)
- It is based on the marketing axiom that **80% of your business comes from 20% of your customers** which is 112 customers as per below.

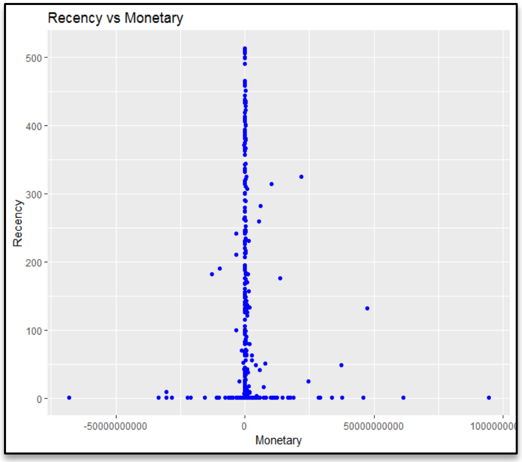


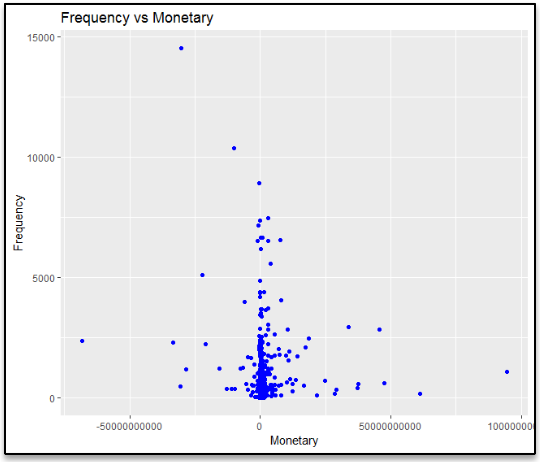
RFM Analysis(USD)

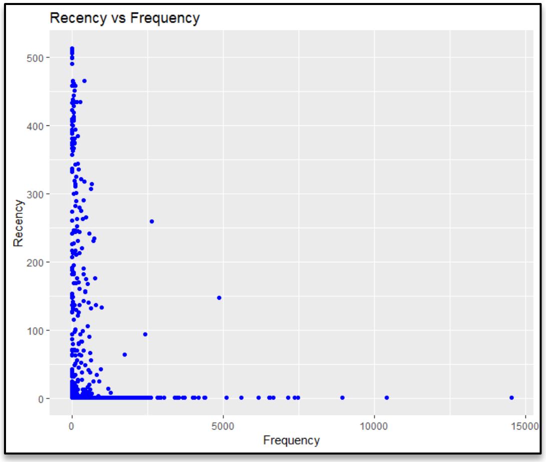
The period of study was from **2015-01-02 - 2018-05-31(YYYY-MM-DD)**
Total no of customer **458**
Account which was studied was **'TIME DEPO CUST'**
Currency Taken was **USD**
Total No of Raw Transactions: **530110**
Total No of Transactions: **191353(Grouped by each customers on each day)**



RFM Analysis







It tells that as the no of days increases from the last day a customer made any transactions the monetary value also decreases. The graph shows that after 350 days no customer does any transactions. Also within 80 days of one transactions many customers they tend to do another transactions.

The graph shows that as the no of transactions done by the customers increases the amount of deposit s increases and withdrawals decreases. So if a customer is doing less transactions then he has a higher chance of withdrawals.

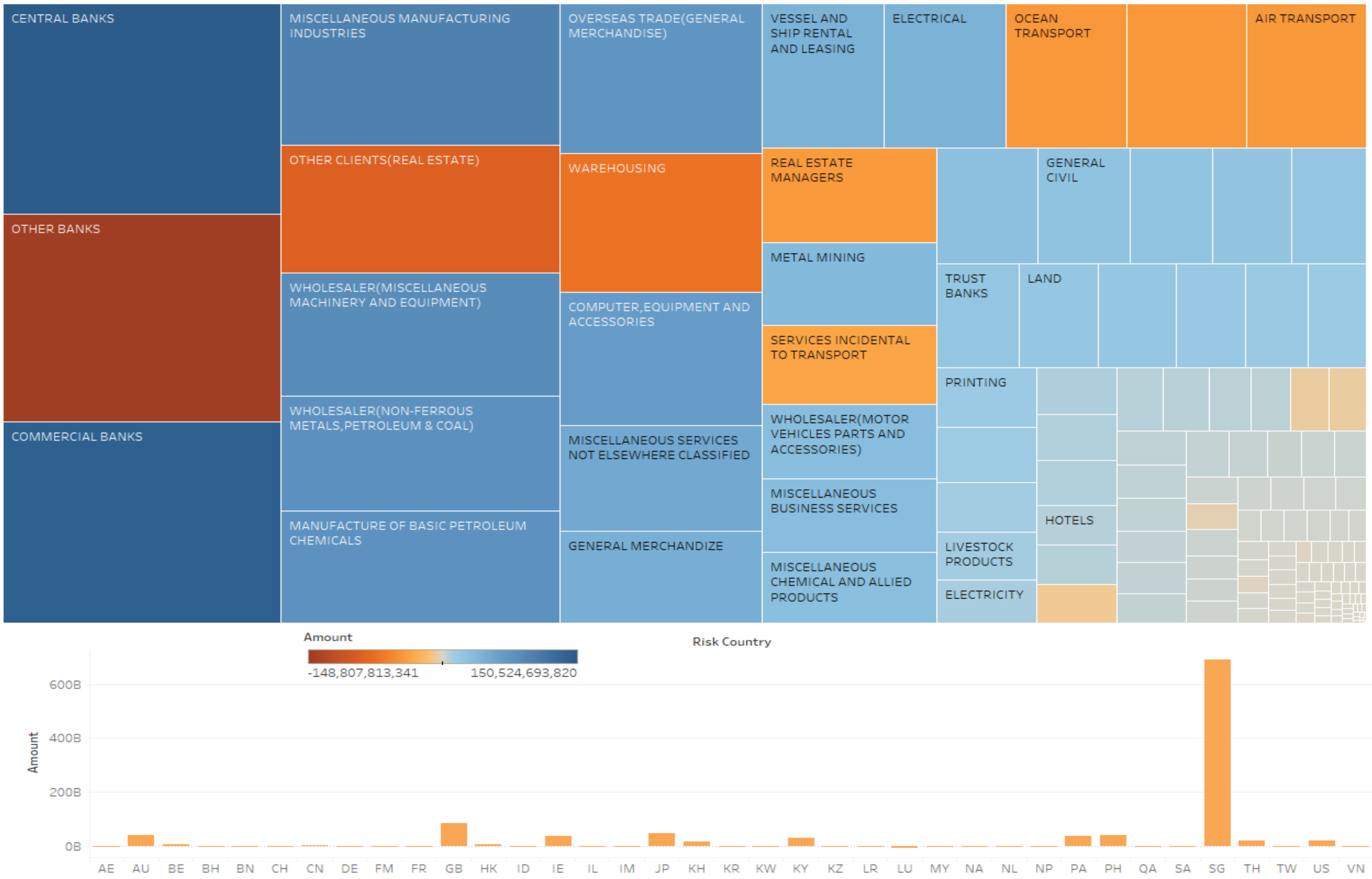
The graph shows that customer with high no of transactions have done transactions recently where as customer with less no of transactions have done transactions later. If a customer remains idle for about 300 days then the no of transactions done will be an average of 90.

What can be done ?

- From RFM analysis we decided to look into two groups of customer segments customer at **risk** and customer who **need attention**.
- We studied these customers and found out what are the **industries** they belong and the revenue they generate from these industries.
- **Download Unstructured data** related to company like news related to a particular company. Calculate the sentiment.

Exploration/ Sentiment Analysis

Industries and Risk Country



Sentiment Analysis

Unstructured Data from different source was downloaded :

1. **Apache nifi** to download news stream from newsapi.com
2. Using python and R to directly download data from newsapi.com, kaggle

Step 1:
Download unstructured data in Hadoop

Step 2:
Convert the unstructured data to structure data

Step 3:
Calculate the sentiment and use it as one of the independent variable in prediction.

Bloomberg

YAHOO! FINANCE

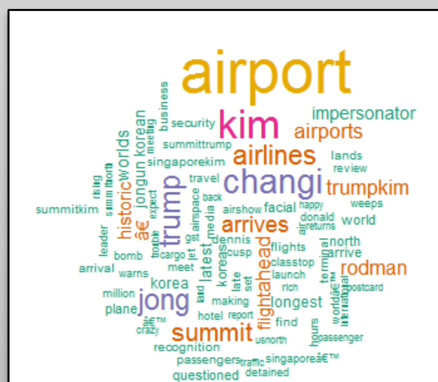
Quandl

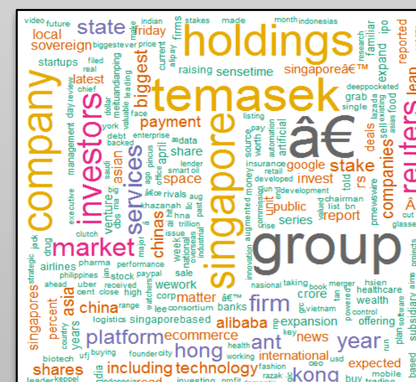
News API

Algorithm used for calculating sentiments are as below:

- **Text Blob using a pre-trained NaiveBayes classifier:** https://textblob.readthedocs.io/en/dev/
- **Valence Aware Dictionary for Sentiment Reasoning(VADER):** http://datameetsmedia.com/vader-sentiment-analysis-explained/

As an example we scrapped the news related to **Changi Airport ,Tamasek Holding** and calculated the sentiment which was used as a regressor for prediction.





What will Happen ?

- Our main objective in this case was to predict and forecast the amount of possible transaction a customer will do.

- We used different **forecasting model** and supervised **machine learning algorithm** to forecast as well as predict customer transactions.

Forecasting & Prediction

Forecasting(ARIMA)

Training Period: 2015-01-02 to 2018-04-30 (3 Year 4 Month)
Validation Period: 2018-05-01 to 2018-05-31 (1 Months)
Total no of customer 395
Account which was studied was 'TIME DEPO CUST'
Currency : SGD

Note: SVD- Singular Value Decomposition

Different Forecasting Models that was tested in our analysis:

1.stlf.nn-Standard scaling + stlf/ets + averaging - The data was standard scaled, and a correlation matrix was computed. Then forecasts were made and several of the closely correlated series were averaged together, before restoring the original scale. Forecast was done with stlf(), using an exponential smoothing model (ets).
2.tslm.basic- Computes a forecast using linear regression and seasonal dummy variables
3.SVD + stlf/arima - this model applied SVD to the training data as pre-processing, and then forecast each series with stlf(), using an ARIMA model for the non-seasonal forecast
4.SVD + stlf/ets - this model applied SVD to the training data as pre-processing, and then forecast each series with stlf(), using an exponential smoothing model (ets) for the non-seasonal forecast.
5.non-seasonal arima with Fourier series terms as regressors - This also used auto.arima(), but as a non-seasonal ARIMA model, with the seasonality captured in the regressors.
6.regressor.arima.sgx_stock- Auto ARIMA model with regressors as weekends and sgx stock price
7.regressor.arima- Auto ARIMA model with regressors as holidays
8.SVD + seasonal arima - Replaces the training data with a rank-reduced approximation of itself and then produces seasonal ARIMA forecasts with weekends as regressors.

Forecasting(ARIMA)

	(SGD)No Of Customers Observed within the F1 Score Range(15 Days of Forecasting)										Removing Customers Not present in Testing Data			
F1 %	0-10 %	10-20 %	20 -30 %	30-40 %	40-50 %	50-60 %	60-70 %	70-80 %	80-90 %	90-100 %	Outliers	Zero Transactions	**Total**	**Total Customers**
stlf.nn	14	4	1	1	1	0	1	0	0	0	0	192	214	392
tslm.basic	14	5	2	2	0	0	1	0	0	0	0	190	214	392
stlf.svd - arima	18	10	5	3	1	0	0	0	0	0	0	177	214	392
stlf.svd - ets	21	13	4	3	0	0	0	0	0	0	0	173	214	392
fourier.arima	4	22	3	1	0	0	0	0	0	0	0	184	214	392
regressor.arima.sgxstock	3	3	1	0	1	0	0	0	0	0	0	206	214	392
regressor.arima	2	7	3	0	0	0	0	0	0	0	0	202	214	392
seasonal.arima.svd													0	392

	(SGD)No Of Customers Observed within the MAPE Score Range(15 Days of Forecasting)										Removing Customers Not present in Testing Data			
MAPE %	0-10 %	10-20 %	20 -30 %	30-40 %	40-50 %	50-60 %	60-70 %	70-80 %	80-90 %	90-100 %	Outliers	Zero Transactions	**Total**	**Total Customers**
stlf.nn	37	10	3	3	1	1	1	0	0	1	3	154	214	392
tslm.basic	38	8	4	2	1	1	0	2	0	1	3	154	214	392
stlf.svd - arima	37	11	4	2	2	0	1	1	0	0	2	154	214	392
stlf.svd - ets	39	10	3	3	1	0	2	0	0	0	2	154	214	392
fourier.arima	36	12	3	2	0	1	0	1	0	0	5	154	214	392
regressor.arima.sgxstock	40	6	5	3	1	1	1	0	0	0	3	154	214	392
regressor.arima	41	10	4	3	0	1	0	0	0	0	1	154	214	392
seasonal.arima.svd													0	392

Out of the 395 customers that we put through our algorithm we were able to capture around 200 to 240 customers without any significant loss of MAPE . Most of the customers withdraw balance were predicted with an accuracy of less than 100 % to 40%.

Evaluation Matrix: MAPE
(Will tell you how closely our forecasting compared to actual value for 15 days in advance)

Evaluation Matrix: F1 Score
(Will tell you how many of the people will do withdrawals with a probability percentage)

$$\left(\frac{1}{n}\sum\frac{|Actual - Forecast|}{|Actual|}\right)*100$$

$$F_1 = \frac{2}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

Conclusion:
It was noticed that the accuracy was improving with addition of external regressors like holidays , external stock market price i.e. SGX ,end of the day closing market price of a company.

Prediction

For model building using prediction we used 3 machine learning models.
Training Period *: 2015-01-02 to 2018-02-28 (3 Year 2 Months)
Validation Period: 2018-03-01 to 2018-05-31 (3 Months)
Total no of customer 824
Account which was studied was 'TIME DEPO CUST','TIME DEPO BANKS'
Currency Taken was USD and SGD
Independent Variables used are **Sentiment Score** of the news, **SGX stock market price**, **Credit-Debit Amount shifted to 1 day**
*The Training period was dynamically changed in the code to accommodate those customers whose data was very less.

	No Of Customers Observed within the R2 Value											
R2 %	0-10 %	10-20 %	20 -30 %	30-40 %	40-50 %	50-60 %	60-70 %	70-80 %	80-90 %	90-100 %	**Total**	**Total Customers**
SVM Regressor	71	52	25	9	4	4	1	1	0	0	167	500
LSTM	58	51	51	33	17	3	2	0	0	0	215	500
Linear Model	41	58	38	24	14	7	3	1	0	0	186	500

	No Of Customers Observed within the F1 Score											
F1 Score %	0-10 %	10-20 %	20 -30 %	30-40 %	40-50 %	50-60 %	60-70 %	70-80 %	80-90 %	90-100 %	**Total**	**Total Customers**
SVM Regressor	6	14	6	9	9	7	12	10	7	0	80	500
LSTM	10	8	4	6	16	9	14	8	9	0	84	500
Linear Model	3	13	9	11	12	10	15	7	8	0	88	500

Evaluation Matrix: R^2
It tells us how much of the variance in the dependent variable is explained by the independent variables

Evaluation Matrix: F1
Will tell you how many of the people will do withdrawals with a probability percentage

- The total sum of squares (proportional to the variance of the data):
$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2,$$
- The regression sum of squares, also called the explained sum of squares:
$$SS_{\text{reg}} = \sum_i (f_i - \bar{y})^2,$$
- The sum of squares of residuals, also called the residual sum of squares:
$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

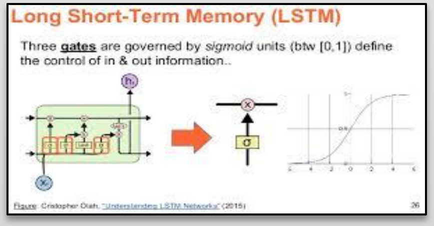
The most general definition of the coefficient of determination is
$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}.$$

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Prediction

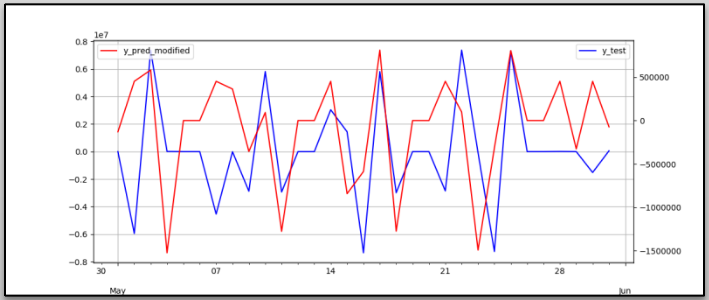
Customer withdrawal prediction was done using both external and internal data. External data that was used

- Structured Data
    - Historical daily **SGX** price from Yahoo finance
    - Historical daily **Nasdaq** Exchange Price from Yahoo finance
    - Historical daily **S&P** Exchange Price from Yahoo finance
    - Historical daily **Nikkie** Price from Yahoo finance
    - Customer Equity price from **Bloomberg** terminal for two customers
- Unstructured Data
    - **News** related to YW75 (Changi Airport)from newsapi.com using API call from python and Kafka(hadoop)
    - **Top 20 news** headline around the worldwide from kaggle.

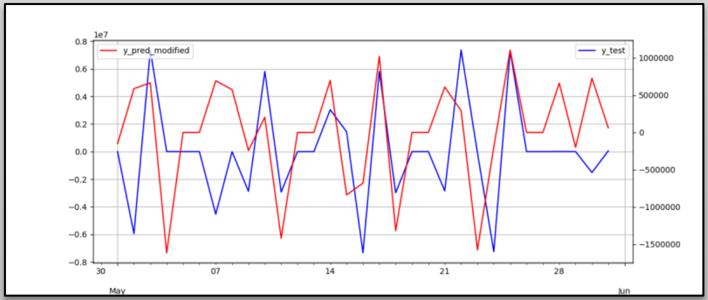


Different Algorithm that we tried:

1. LSTM Neural Network.



VS

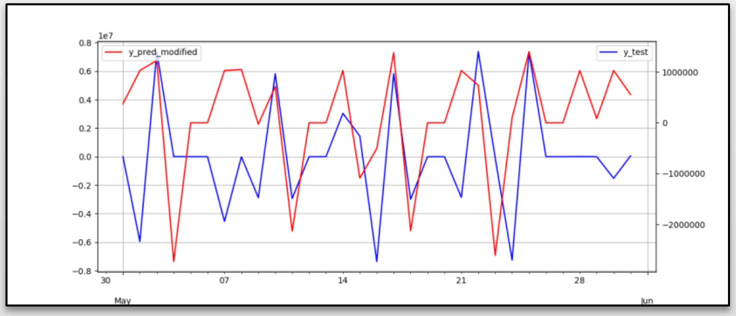


R-squared is LSTM: **0.085246(Without Sentiment)**

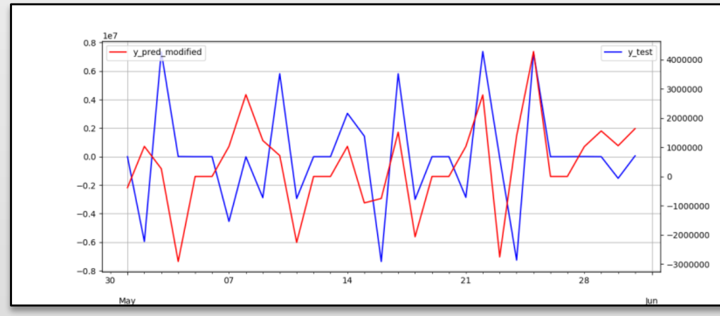
R-squared is LSTM: **0.09551(With Sentiment)**

Prediction

2. SVM Regressor



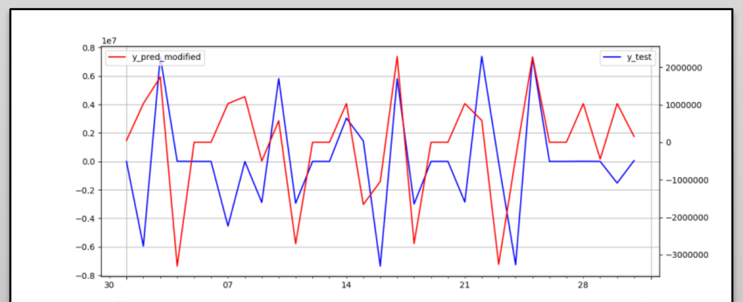
VS



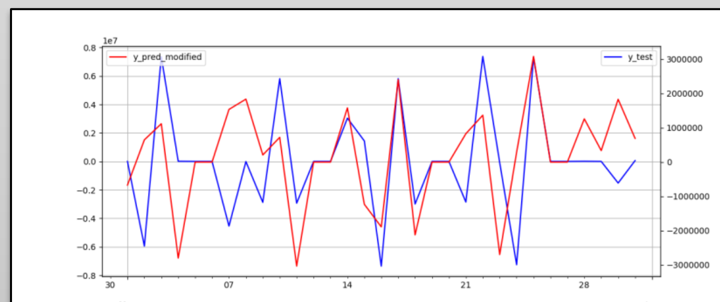
R-squared is SVM: **0.088602**

R-squared is SVM: **0.086131**

3. Linear Regressor



VS



R-squared is Linear: **0.149722**

R-squared is Linear: **0.175620**

Conclusion:
The prediction of all the model showed an **improvement** in **R^2** value when we added external data like SGX and sentiment scores obtained from news API

Recommend

RMs will be able to better understand their customers and take necessary action