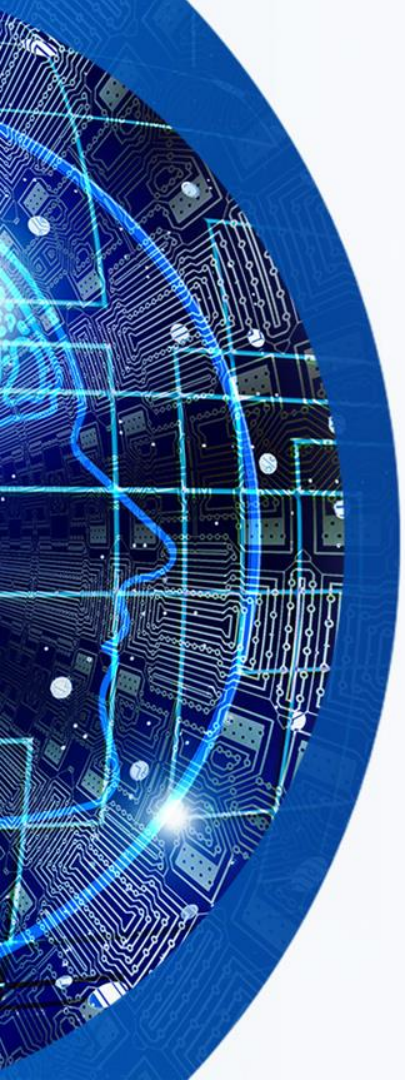
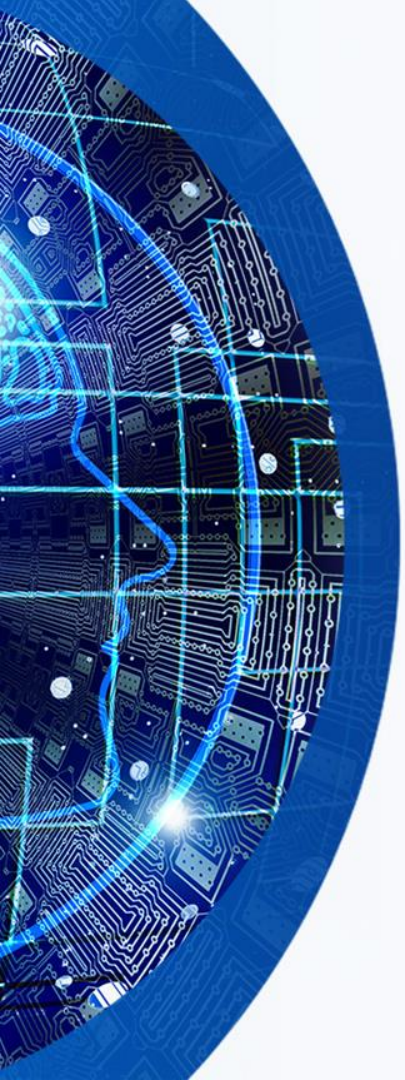
Home Credit Default Risk Prediction

SURESH BABU T



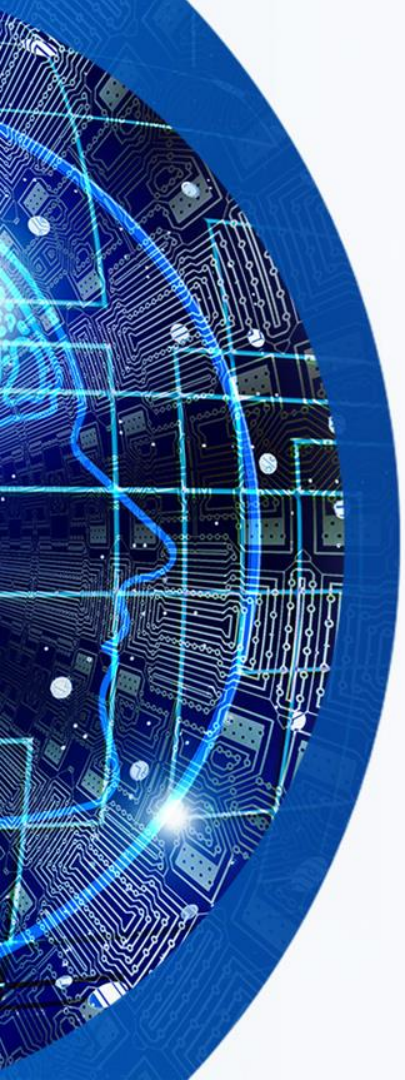
Agenda

- Literature survey & Data Acquisition
- Metric to Validate the Model
- Exploratory Data Analysis (EDA)
- Feature Engineering
- Vectorization
- Modeling
- Why Should I trust You (AI)
- Summary



Literature survey & Data Acquisition

- Problem definition
  - Enhance the lending experience to be positive for insufficient or non existent credit histories.
- Need for data science
  - decision made by Expert opinion Vs Rule based engine Vs Data driven approach ?
    - Need a scientific approach and able to adapt future changes.
- Business Constraints
  - How accuracy is?
  - Cost of misclassification
    - Revenue loss Vs Liability?

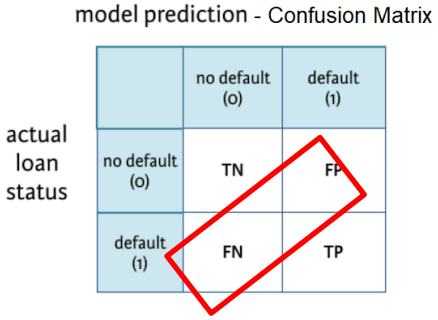


Literature survey & Data Acquisition

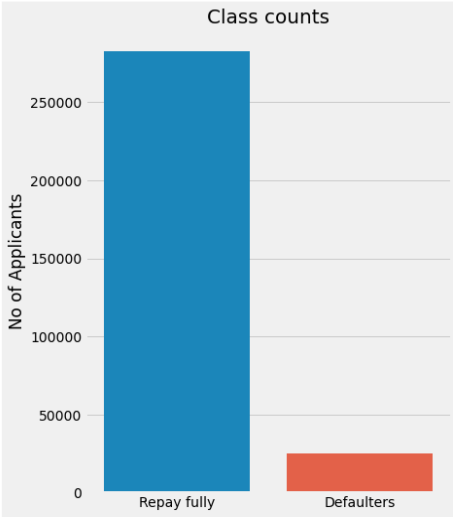
- Data sources shared by Home credit in Kaggle[publically available data]
  - Overall dataset size is 2.5GB, 300k application taken into consideration
    - Less than 9% of the applications are defaulters, 91% applications can repay. Highly imbalanced data.
  - Application details [Training and Testing]
    - Training dataset contains the class label, Testing without class labels. Test data can be validated only after uploading it to Kaggle site. Hence, Training data only used for training and validating the model.
  - Credit Bureau details
    - Credit Bureau and Bureau balance of the borrower
  - Point of Sale / Cash Transaction
    - Cash transaction done by borrower in various places
  - Customer Credit Card Transaction details
    - Credit card transactions done by the borrower
  - Previous application details
    - Previous application of borrower and contains application status
  - Installments payments details
    - EMI, balance and due details

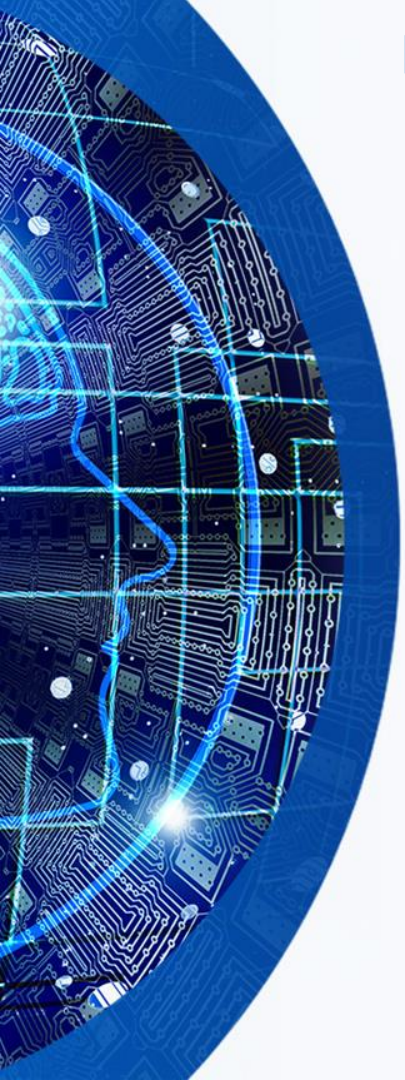
Metric to Validate the Model

- How to validate the Model is performing well?
  - Given that the dataset is highly imbalanced
- False Positive Vs False Negative scenario?
  - Precision considers FP and TP whereas Recall considers FN and TP.
  - F1Score is a harmonic mean of precision and Recall. Hence F1Score is better overall.
- Confusion Matrix gives information about actual vs prediction.
- AUC is the metric home credit uses to validate the model.
- Hence, AUC, confusion matrix and F1Scrore are important metrics to validate the model performance.
  - Also, metrics to help maximise the True predictions (TP, TN) and minimises the False predictions (FN, FP). False Negative is liability to the company. Within the misclassification, FN to be even more less to reduce the liability.



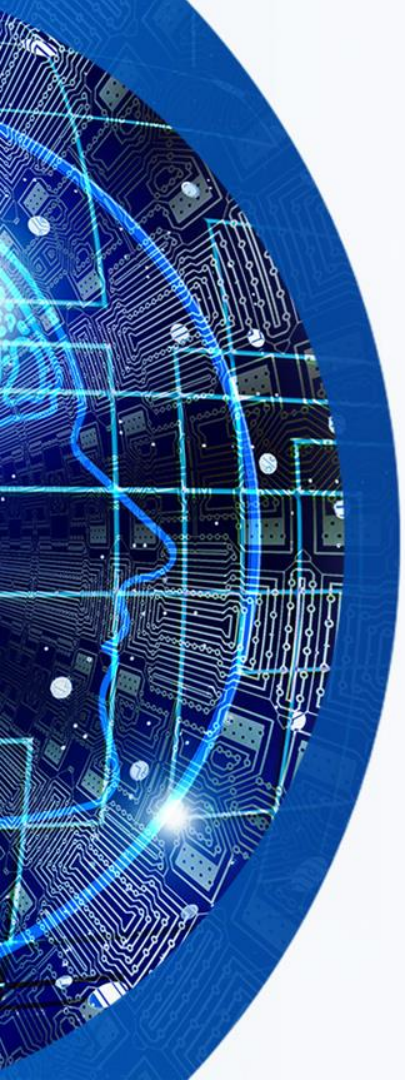

```
Defaulters examples = 24825
Repay fully examples = 282686
Proportion of Defaulters to Repay examples = 8.78%
```





Exploratory Data Analysis (EDA)

- **Data quality check**
  - Find and Impute / remove missing values ?
    - Numerical features – Impute with Median to avoid outliers
    - Categorical features – Impute with most frequent value
    - Text Data – Unfortunately Home Credit didn't share any text related data.
    - Try to keep all the information as much as possible to have higher prediction rate.
- **Check data duplication**
  - Does any data is duplicated /repeated?
    - None of the applications are repeated.



EDA – Continued.

- **Feature wise Analysis**
  - Why should we analyse the data ? – Helps to fix the issues.
  - Understand the data from statistical point of view : Ex: Days_Employed.

```
df_train['DAYS_EMPLOYED'].describe()
```

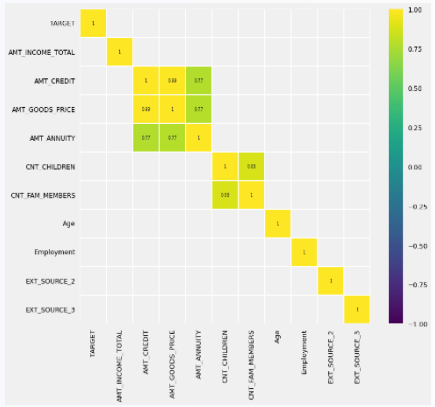
```
count    307511.000000
mean      63815.045904
std      141275.766519
min      -17912.000000
25%       -2760.000000
50%       -1213.000000
75%        -289.000000
max      365243.000000
Name: DAYS_EMPLOYED, dtype: float64
```

- ✓ This numerical column has negative values. Days employed can't be negative. Hence, find all negative numerical columns and if appropriate convert them into positive values.
- ✓ DAYS_EMPLOYED having a max value of 365243 which is Error and hence filled with median value.
- ✓ Mean Vs Median: Mean can be influenced with outlier. Hence Median used.

  - Likewise, every feature to be validated for better results.

EDA – Continued.

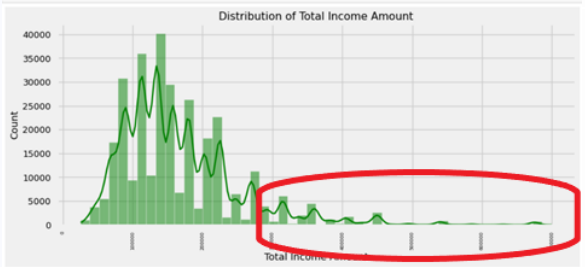
- **IQR – to fix the outliers**
  - Outliers, can worse the model performance
  - IQR [Interquartile Range], is a statistical method to find outliers.
  - IQR = Q3 – Q1. The data points which fall below Q1 – 1.5 IQR or above Q3 + 1.5 IQR are outliers
    - Outliers are fixed with median values
- **Correlation between Columns – kind of duplication?**

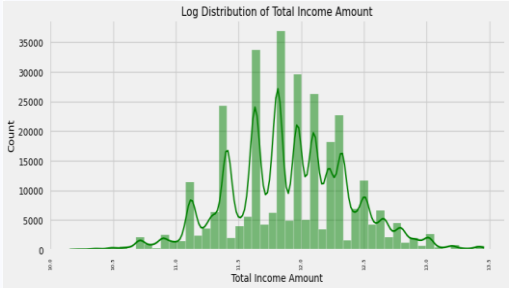


- Features has to be unique/distinct and has to correlate only with class labels [output].
- Using correlation technique, around 40 features are removed.

EDA – Continued.

- **Univariate Analysis**
  - Univariate analysis is done using the histogram : Ex: AMT_INCOME_TOTAL

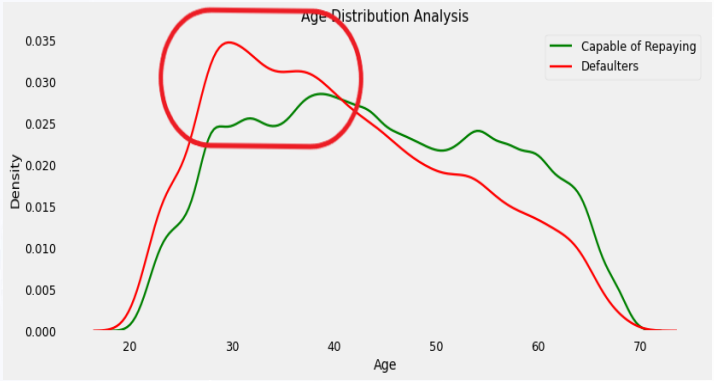




- ✓ Application Income total of less than 7million was plotted.
- ✓ Above 7million there are about 738 applications and the ratio of that application is less than 0.25%.
- ✓ This could worse the model performance, as the model can tend to predict always, on repaying capability side [positive]. Hence above 7m records are ignored for analysis and set to max of 7 million.
- ✓ Still highly skewed data.
- ✓ Highly skewed features are transformed to log transformation to represent data in symmetric

EDA – Continued.

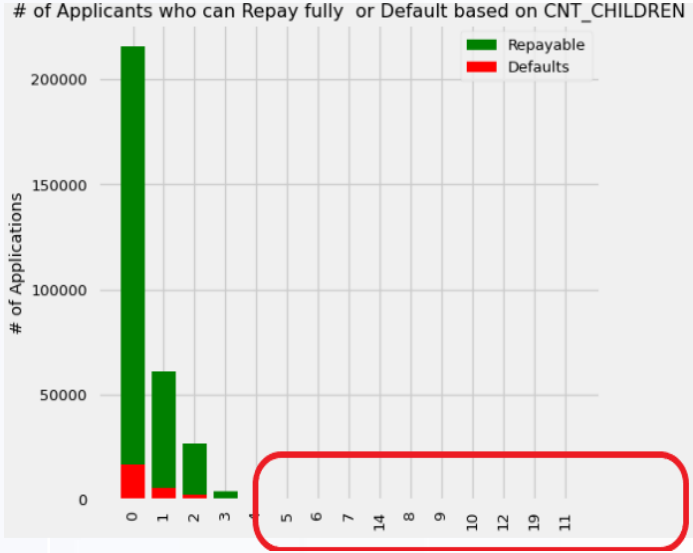
- **Age distribution analysis based on classes**
  - Univariate analysis is done using the histogram : Ex: Age



- ✓ Age between 25 and 35, the defaulters are high compared to the Age above 50
- ✓ Between Age 50-65, repay capability is higher. Also above 70, very less / no applications are approved / rejected.

EDA – Continued.

- CNT_CHILDREN has multiple categorical values from 0 to 14

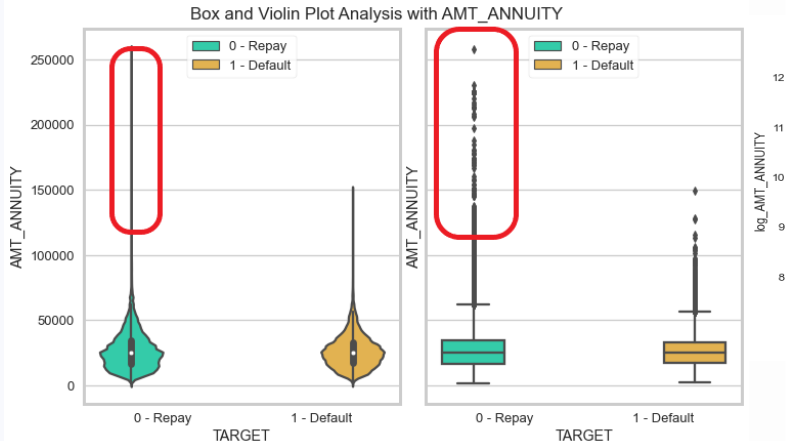


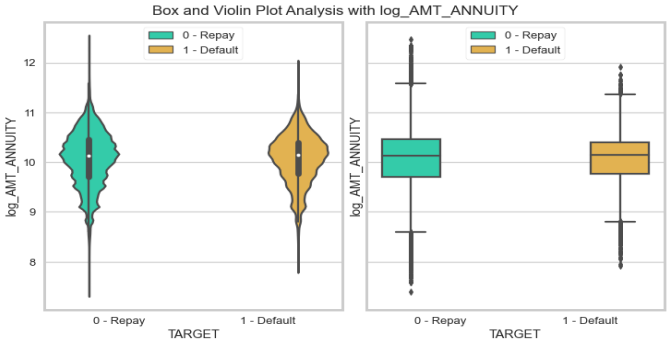
- ✓ A High number of applications shows 0 children.
- ✓ More children could be an Error. [19,14, 12 and 11]
- ✓ We can safely set it to 6 if the number of children is more than 6
- ✓ Till 99.9 percentile, the average children count is 4. hence it is safer to remove.

```
99.0 percentile value is 3
99.1 percentile value is 3
99.2 percentile value is 3
99.3 percentile value is 3
99.4 percentile value is 3
99.5 percentile value is 3
99.6 percentile value is 3
99.7 percentile value is 3
99.8 percentile value is 3
99.9 percentile value is 4
100 percentile value is  19
```

EDA – Continued.

- AMT_ANNUITY feature is a numerical one and will try to understand visually with Boxplot and violin plot.
- Box plot helps to visualize statistical view to find outliers [IQR].
- Violin plot combines density plot and IQR. Density helps to understand the data distribution like Gaussian, Pareto.
- There is no outliers, however the data is rightly skewed, hence, it is transformed to log distribution. Log distribution is a monotonic function.



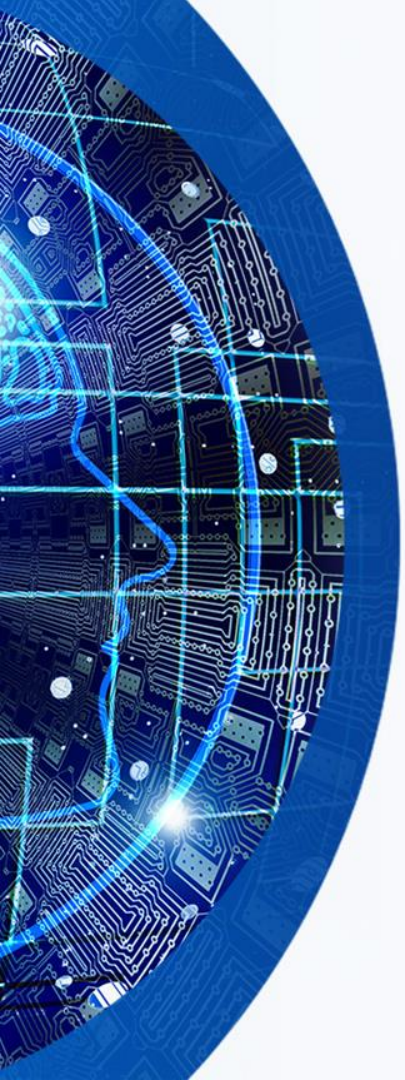


Feature Engineering

- In order to have better Machine Learning Models, more unique features are important. Feature Engineering, is one of the key step in ML to generate additional features.
- Additional features are created scientifically using polynomial-based and domain-based features.
- Merge Credit Bureau, Credit Card, installment details info and other details into Application.

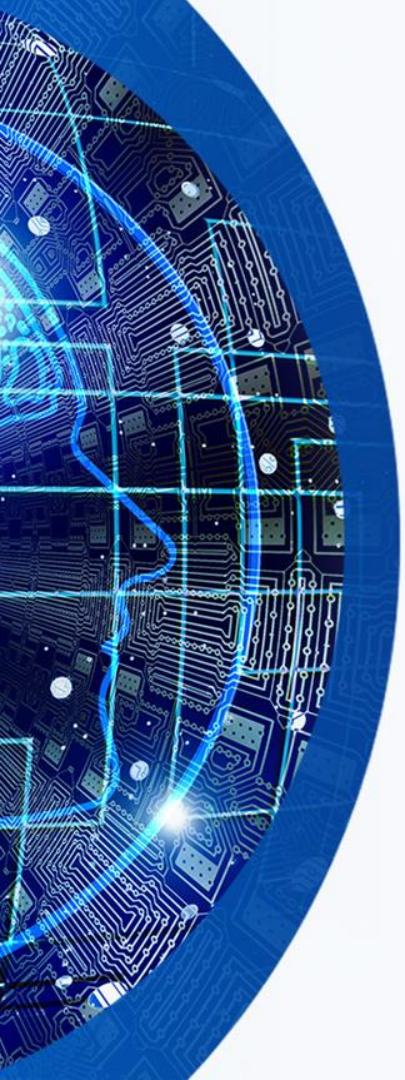
```
#Try to add some more features domain based
# Credit :- https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction#Feature-Engineering
epsilon=0.001 # to avoid div/0 error
def Feature_Engineering(df):
    df1 = pd.DataFrame()
    df1['SK_ID_CURR']=df['SK_ID_CURR']
    df1['pc_Credit_Income'] = np.round((df['AMT_CREDIT'] / (df['AMT_INCOME_TOTAL']+epsilon)),4)
    df1['pc_Annuity_Income'] = np.round((df['AMT_ANNUITY'] / (df['AMT_INCOME_TOTAL']+epsilon)),4)
    df1['pc_Credit_Annuity'] = np.round((df['AMT_CREDIT'] / (df['AMT_ANNUITY']+epsilon)),4)
    df1['Credit_Goods_Diff'] = df['AMT_CREDIT'] - df['AMT_GOODS_PRICE']
    df1['pc_Loan_Value']=np.round((df['AMT_CREDIT'] / (df['AMT_GOODS_PRICE']+epsilon)),4)
    df1['CREDIT_TERM'] = np.round((df['AMT_ANNUITY'] / (df['AMT_CREDIT']+epsilon)),4)
    df1['pc_Employment_Age'] = np.round((df['Employment'] / (df['Age']+epsilon)),4)
    return df1

df2=Feature_Engineering(df_train)
df_train = df_train.merge(df2, on = 'SK_ID_CURR', how = 'left')
print('Train data after Feature Engineering ~>{}'.format(df_train.shape))
```

- ✓ Polynomial features are like featureX, can be transformed to featureX^2, featureX^3, etc.
- ✓ Ex: EXT_SOURCE transformed to EXT_SOURCE^2, EXT_SOURCE^3
- ✓ Domain features are created based on functional.
- ✓ Ex: Loan Percentage field is created by dividing AMT_Credit/AMT_GOODS_PRICE. Ex: Percentage loan value is AMT_Credit / AMT_GOODS_PRICE
- ✓ Credit_Goods_Diff = AMT_CREDIT-AMT_GOODS_PRICE
- ✓ The more additional, non correlated features would improve model performance.



Vectorization – ML understandable format

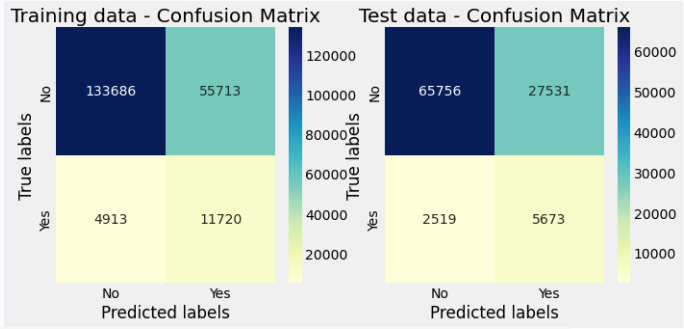
- Data preprocessing, Data Analysis and Feature Engineering steps helps identifying the issues with data and how to fix it. And also helps generating new features.
- Next is to convert the data into ML understandable format.
- Vectorization [matrix format] is next process and format the data.
- Home credit shared numerical and categorical features only.
- In order to feed these information to Model, the data to be vectorised.
- Categorical data can be vectorised using one-hot/response encoding.
- Numerical vectorization done using standard scalar method.
- There are many vectorization methods are available, depending upon the data, the correct vectorization technique to be chosen.

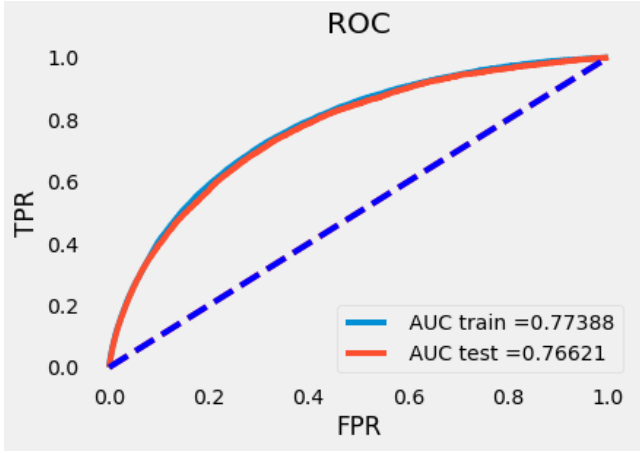


Modeling

- The pre-processed data fed into various models to predict and validate based on the metrics identified.
- Modeling is an art, and based on the statistical info few of the models can be narrowed down for training.
- Training time and Model interpretation, feature extraction / importance are also key aspects in selecting the right model.
- Hyper tuning the parameters to improve the accuracy
- Models should not over fit and under fit.
- Training data should be split into 70/30 for testing. K-Fold technique would help to shuffle the data for better results.
- Given dataset is highly imbalanced, Tree based models like Random Forest, XGBoost, LightGBM would perform good results.
- Linear Models requires, the data to be Normal / Gaussian distributed [well balanced / symmetric]. However, class weights [more weightage to minor classification data -defaulters] can improve the model prediction. Logistic Regression can be a base model to bench mark with other models.
- Deep learning models are agnostic and Feature Engineering is done automatically. Hence ANN(artificial neural network) is a good choice. However, Deep Network architecture is tedious.
- TabNet is state of art in deep learning for tabular data. Hence, this model also considered for training.

Modeling – Linear Regression results



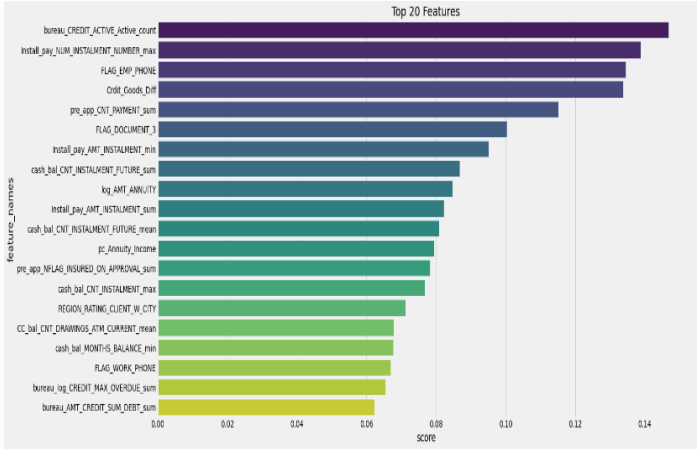


Train Results

	precision	recall	f1-score	support
0	0.96	0.71	0.82	189399
1	0.17	0.70	0.28	16633
accuracy			0.71	206032
macro avg	0.57	0.71	0.55	206032
weighted avg	0.90	0.71	0.77	206032

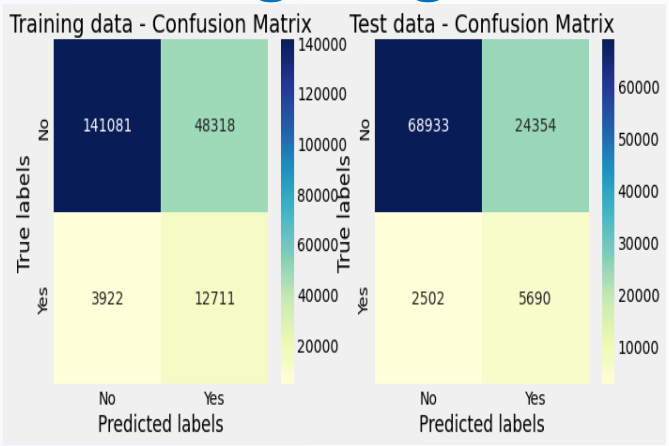
Test Results

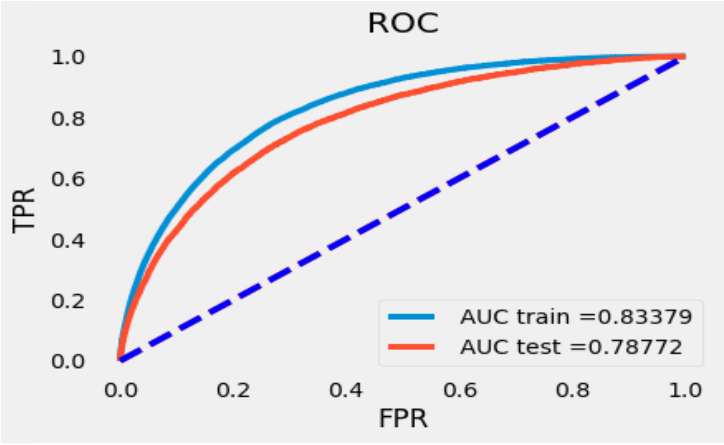
	precision	recall	f1-score	support
0	0.96	0.70	0.81	93287
1	0.17	0.69	0.27	8192
accuracy			0.70	101479
macro avg	0.57	0.70	0.54	101479
weighted avg	0.90	0.70	0.77	101479



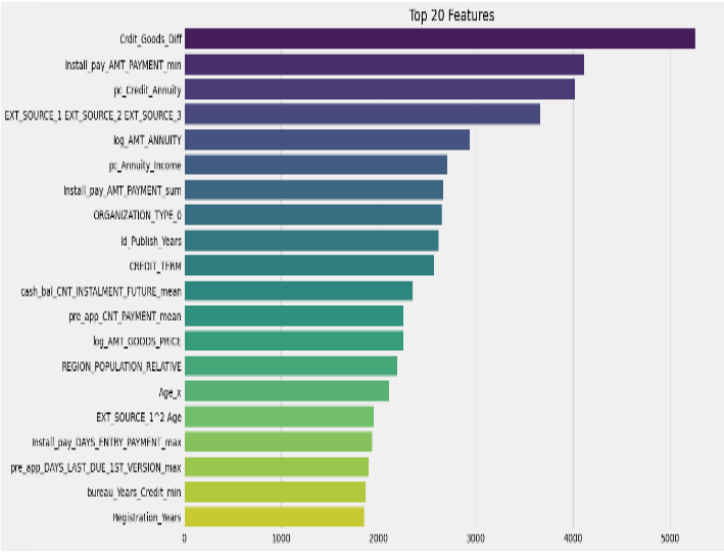
Both Train and Test results are closer. Which means, model would perform well for unseen data. If Overfits, will perform well in training, and for unseen data, will perform worst.

Modeling – LightGBM results

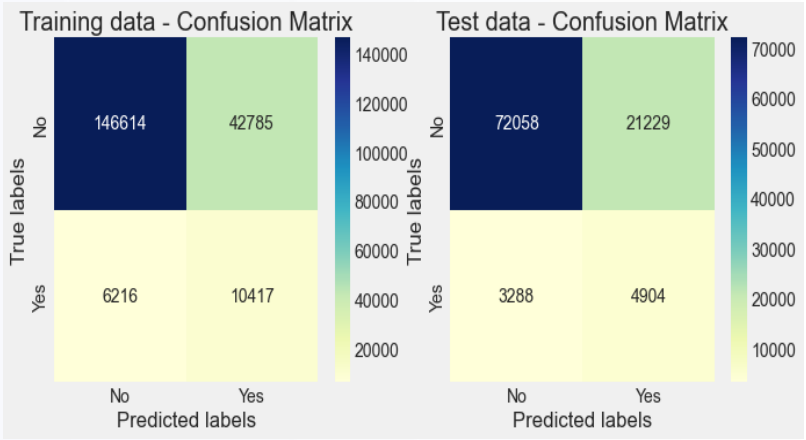


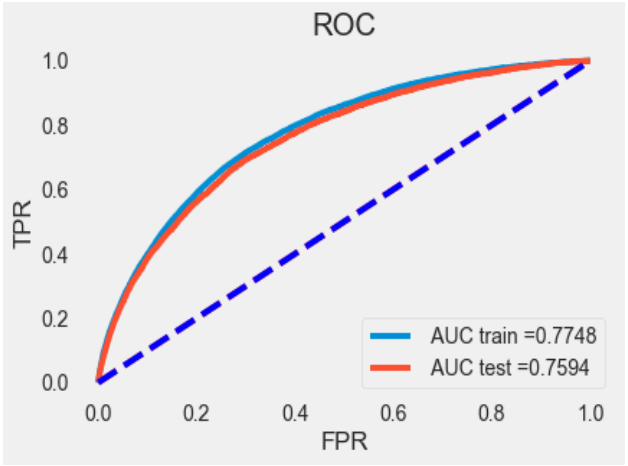


- Model is slightly overfitting. Further tuning, we can reduce the overfitting.
- False Negative misclassification is less compare to False Positive which is a good sign of better model for this kind of dataset.



Modeling – Deep Learning ANN results

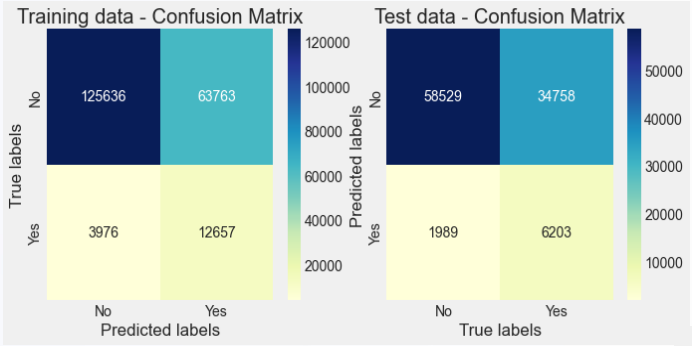


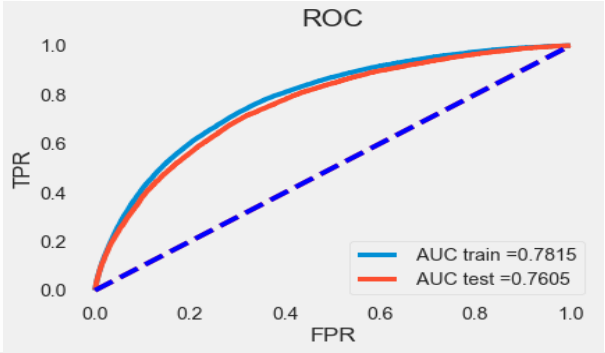


- Feature Importance/Feature Extraction can't be done directly from Deep Learning which is a drawback to this model. However works very well with higher dimensions.
- Model is not overfitting. Compare to LightGBM, FN misclass is high. Hence, LightGBM is better sofar.

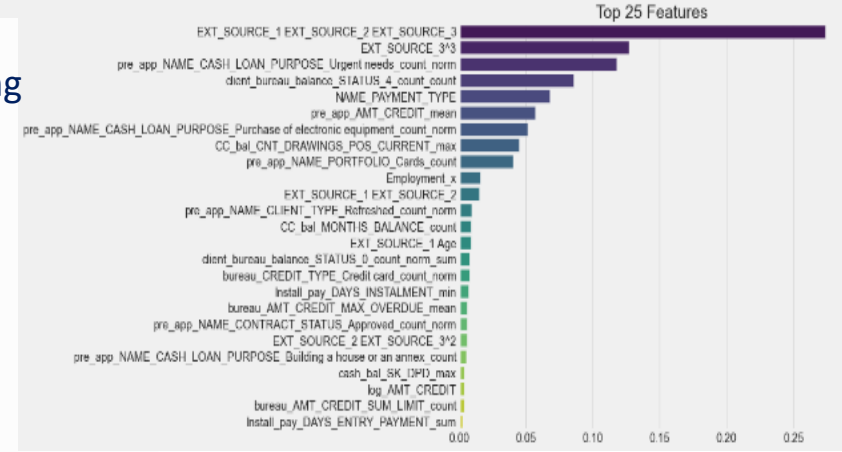
Modeling – TabNet

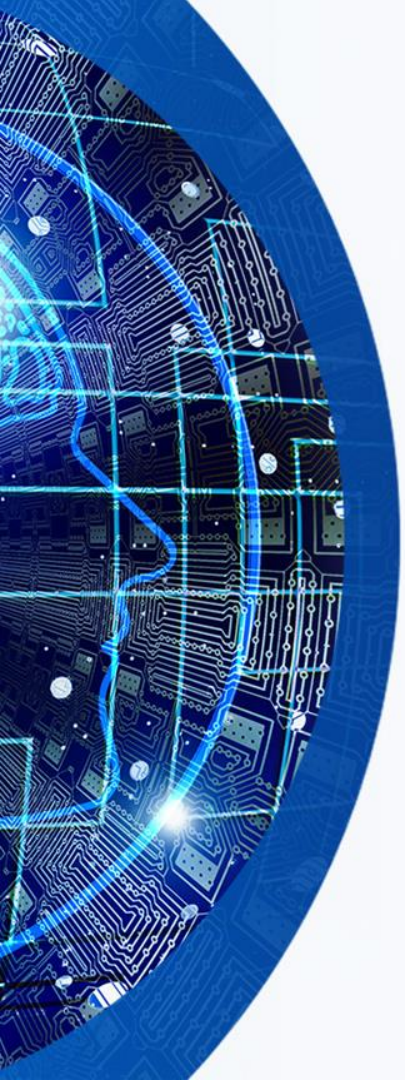
- TabNET is an encoder-decoder, tree-based neural network model suited for tabular data. And is trained using gradient descent-based optimization. Unlike ANN, Feature Importance can be extracted from TabNet.





- TabNet is state of art in deep learning and for Tabular data which perfectly fits for this use case.
- More tuning is required.
- Not overfitting, FN misclass is lesser than LightGBM model.
- However, LightGBM performance is better than this.





Why should I trust You (AI) ?

- **Explainable AI Framework (XAI):**
  - Model feature importance are global explanation.
  - It doesn't say which features are influenced on this particular data.
  - Industry experts/ SME how to trust the model ?. It can't be black box.
  - Explainable AI (XAI) is model agnostic and can be used with any Model/algorithm to get information about particular data.
  - LIME and SHAP are good XAI Framework.

    - LIME – it is black-box, interpretable and gives explanations that are locally faithful within the surroundings or vicinity of the observation/sample being explained.

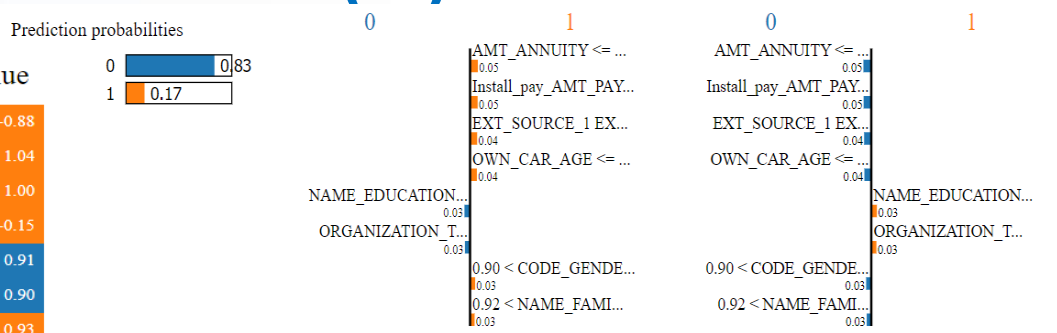
    - SHAP – Game theory based, black-box model to calculate the marginal contribution to the prediction for each feature and then uses Shapley values to calculate feature importance.

    - LIME is easier to understand than SHAP.

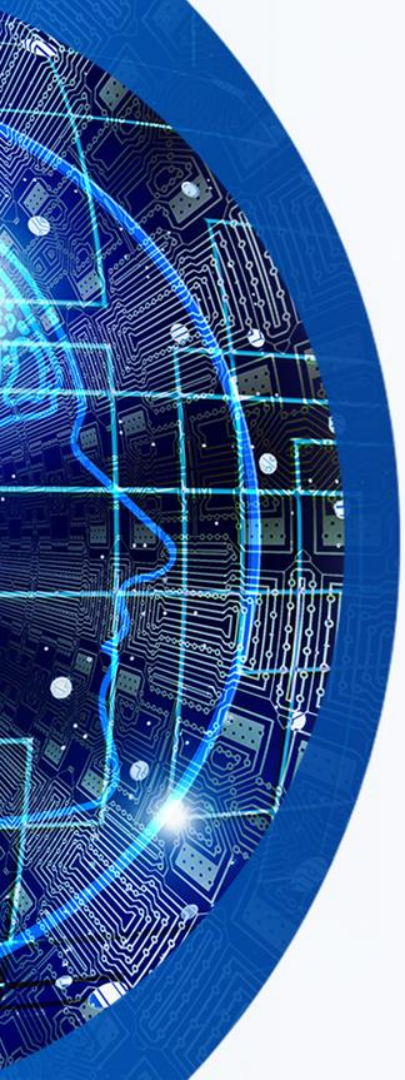
Why should I trust You (AI) ?

- LIME Sample

Feature	Value
AMT_ANNUITY	-0.88
Install_pay_AMT_PAYMENT_min	1.04
EXT_SOURCE_1 EXT_SOURCE_3	1.00
OWN_CAR_AGE	-0.15
NAME_EDUCATION_TYPE_0	0.91
ORGANIZATION_TYPE_0	0.90
CODE_GENDER_0	0.93
NAME_FAMILY_STATUS_0	0.92



- In this sample, the model is 83% is confident that the customer would repay and only 17% is confident that the customer would default. It gives confidence level (%).
- Among top 8 features, External sources, Amount Annuity, intallment payment, gender and family status are contributing towards defaulting.
- Education and organization type are contributes towards repay.
- AMT_Annuity is negative which means either customer loan amount is less or his income is more. Hence his repay capability is high.
- Hence LIME is model agnostic and can interpret locally with respect to particular data.



Summary

Model	Train AUC Score %	Test AUC Score %
Logistic Regression	77.39	76.62
XG Boost	85.86	78.53
LightGBM	83.38	78.77
Deep Learning – ANN	77.48	75.94
Deep Learning – TabNet	78.15	76.05

- **Kaggle submission – Test data**

Submission and Description	Private Score	Public Score
submission_lgbm.csv 4 months ago by SureshBabu T add submission details	0.77923	0.78160

- Kaggle submission with test data yielded almost same with training dataset, which means the model is doing good.
- Based on the AUC score, confusion matrix, F1Score and less misclassification on False Negative, LightGBM model performance is better. Also, this model, provides feature importance.
- Feature Importance, helps the functional team to justify the model.

Summary contd.

- Fine tuning Feature Engineering step, to extract more features can improve the performance further. Personally I feel this is key to ML success. Domain Expertise / business knowledge people can add value to this.
- Domain Experience /SME can help in identifying outliers, and can provide imputation technique to improve the performance.
- Separate Model can be built for data quality check alone.
- Since the data provided by Home Credit is Static. Hence, Model is talso trained on static data. But in production, the data is temporal in nature and newer data may deteriorate the performance over a period of time.
- Retraining the model at regular interval, would keep the production healthier.
- Giving more weightage to recent data /current trend would improve model performance. We need imply time-series analysis for this.
- Distributed Technologies using py-spark /spark ML, cloud services like AWS Sagemaker or Azure ML could improve overall development time and training.
- More data, always would yield better results.
- Adding more features like customer feedback, call details, social media details like twitter, FB details can improvise the prediction.
- It is always better to provide probability (%) than simple Yes/No.
- LIME/SHAP models could help the SME to justify the model prediction.

Questions and Feedback

Questions?

Feedback

Thank you