Santander Bank Customer Transaction Prediction

Data Mining – CMPE 255

Team 2
Harshitha Yerraguntla
Nithya kuchadi
Shreya hagallahalli srinivas

Agenda

- Objective
- Data Overview
- Implementation
- Data Preprocessing
- Feature Engineering
- Cross Validation
- Modelling & Results
- Ensemble Modelling impact
- Challenges
- Tools Used
- Questions

Objective

Identifying who among the current customers will make certain transaction!!!

- Is the customer satisfied?
- Will a customer buy this product?
- Will customer avail this service?

Data Overview

- Train Data: 200,000
- Test Data : 200,000
- Target Variable: Binary (0/1)
- Features:
  - 200 anonymous features
  - All Continuous Variables

```
data_train.head()
```

	ID_code	target	var_0	var_1	var_2	var_3	var_4	var_5	var_6	var_7	...
0	train_0	0	8.9255	-6.7863	11.9081	5.0930	11.4607	-9.2834	5.1187	18.6266	...
1	train_1	0	11.5006	-4.1473	13.8588	5.3890	12.3622	7.0433	5.6208	16.5338	...
2	train_2	0	8.6093	-2.7457	12.0805	7.8928	10.5825	-9.0837	6.9427	14.6155	...
3	train_3	0	11.0604	-2.1518	8.9522	7.1957	12.5846	-1.8361	5.8428	14.9250	...
4	train_4	0	9.8369	-1.4834	12.8746	6.6375	12.2772	2.4486	5.9405	19.2514	...

5 rows × 202 columns

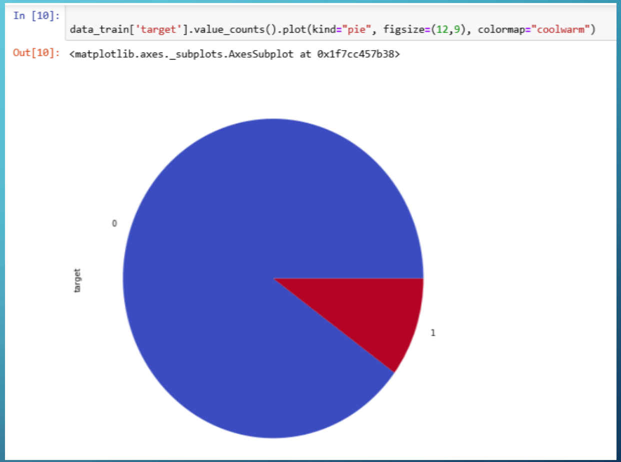
```
data_train.shape
```

(200000, 202)

```
data_train.describe()
```

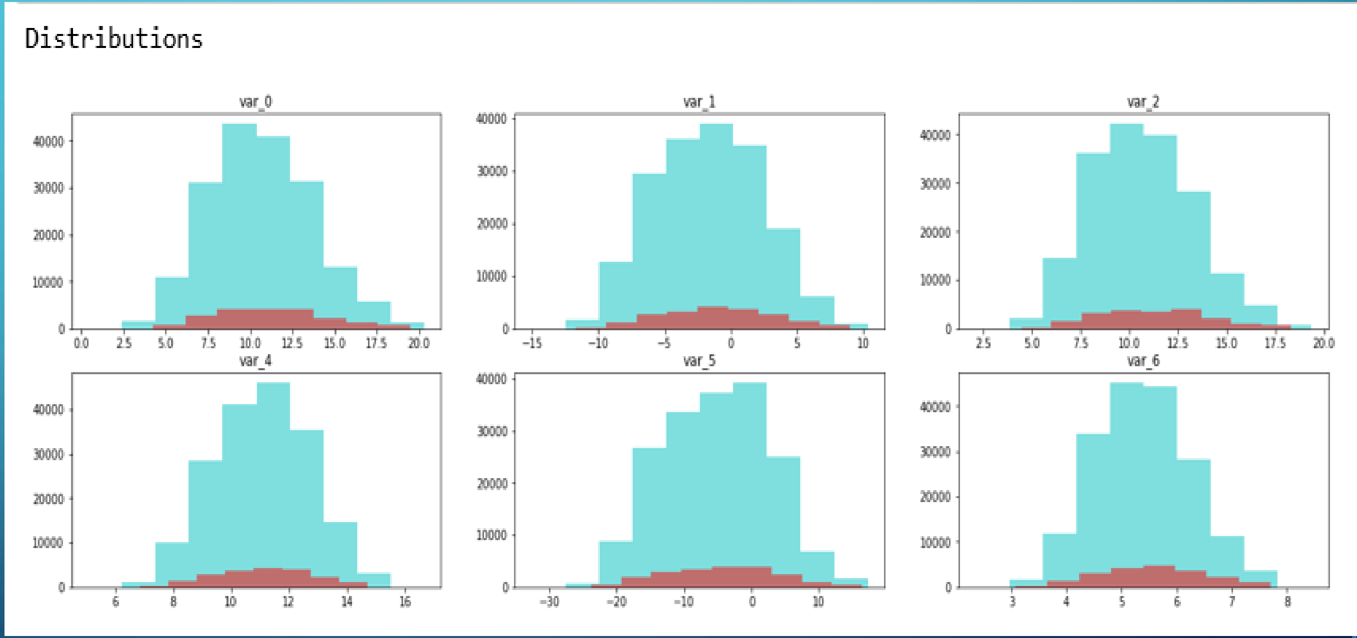
Data Visualizations

- 90% of the training data is labeled as 0, and the other 10% is labeled as 1
- Evaluation criteria:
    - ROC –AUC score
- Target 0 -179902
- Target 1 -20098



FEATURE DISTRIBUTION

The distribution of the features in in terms of class 0 and class 1



Implementation

1
- Missing Data, Correlation, Outlier Removal

2
- Model Fitting

3
- Feature Engineering

4
- Hyper parameter tuning and CV

5
- Model Fitting

6
- Model Evaluation/selection

Data Preprocessing

- No missing values
- No duplicate values
- No much correlation between the features for dimension reduction

```
#Least correlated features
corr.head()
```

```
var_26   var_139    -0.009844
var_53   var_148    -0.009788
var_6    var_80     -0.008958
var_1    var_80     -0.008855
var_2    var_13     -0.008795
dtype: float64
```

```
# Most correlated Features
corr.tail()
```

```
var_146   var_169    0.009071
var_183   var_189    0.009359
var_81    var_174    0.009490
          var_165    0.009714
var_0     var_0      1.000000
dtype: float64
```

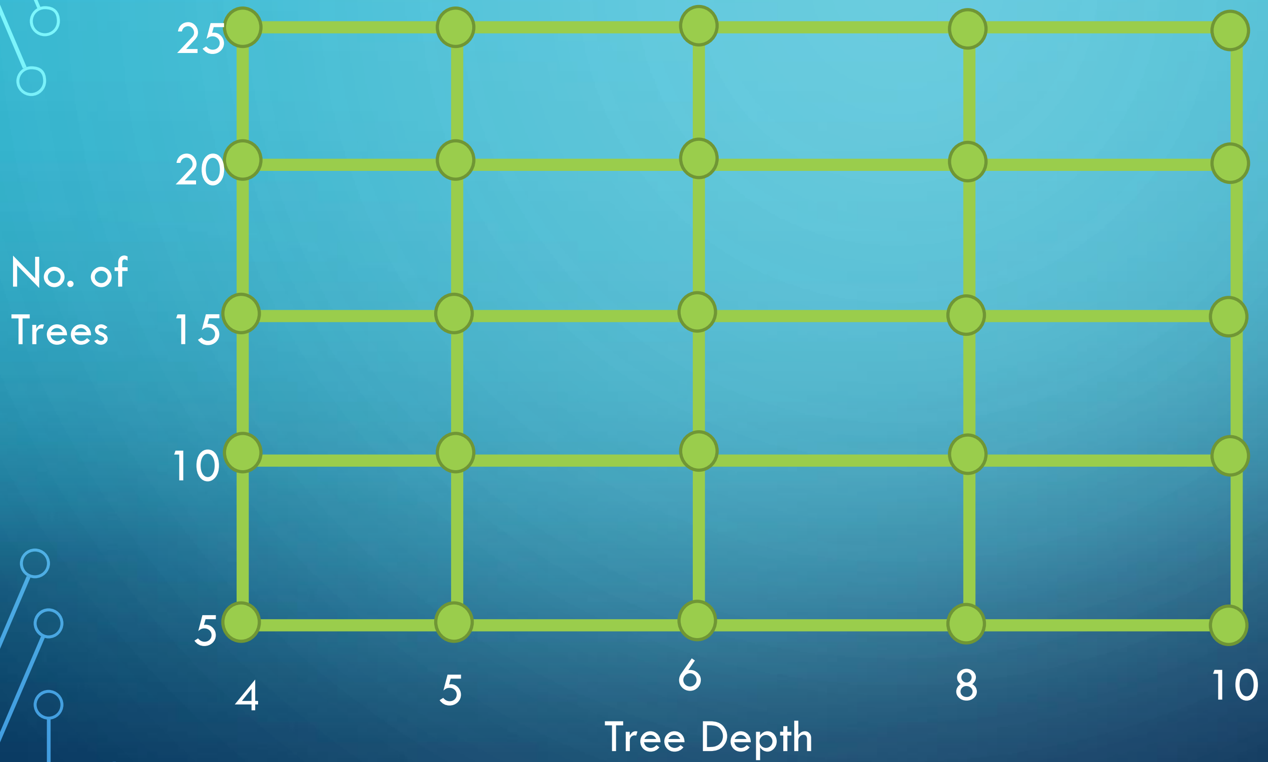
Correlation

Feature Engineering

- New Features
  - Sum
  - Median
  - Mean
  - Min
  - Max
  - Standard deviation

sum	min	max	mean	std	skew	kurt	med
1456.3182	-21.4494	43.1127	7.281591	9.331540	0.101580	1.331023	6.77040
1415.3636	-47.3797	40.5632	7.076818	10.336130	-0.351734	4.110215	7.22315
1240.8966	-22.4038	33.8820	6.204483	8.753387	-0.056957	0.546438	5.89940
1288.2319	-35.1659	38.1015	6.441159	9.594064	-0.480116	2.630499	6.70260
1354.2310	-65.4863	41.1037	6.771155	11.287122	-1.463426	9.787399	6.94735

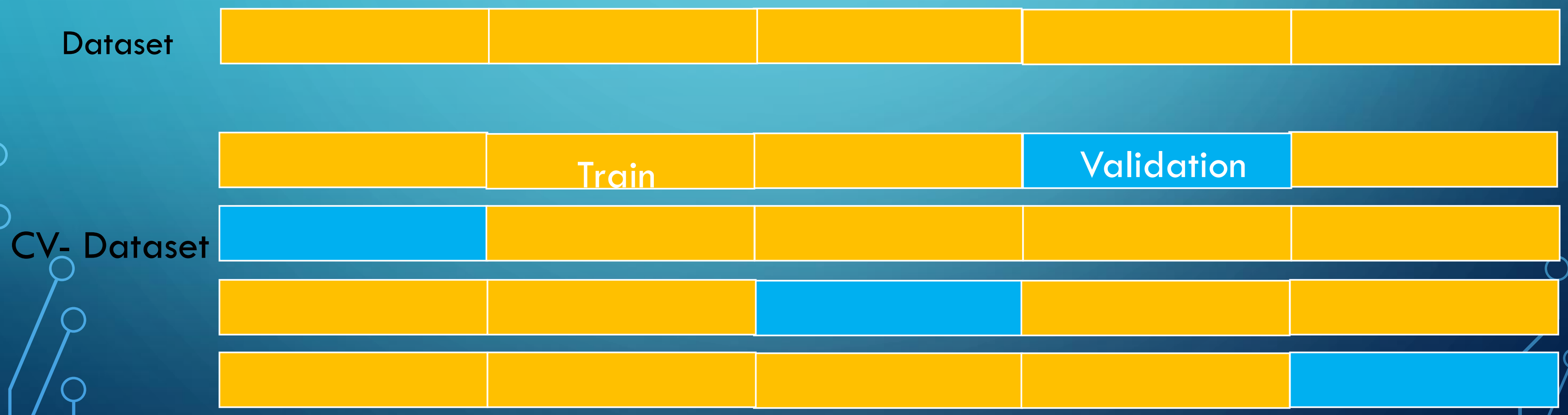
Grid search



How to choose Best Combination?

Cross Validation

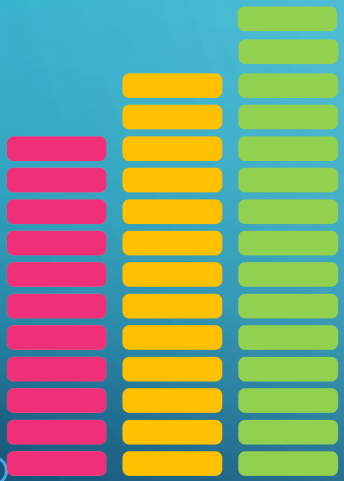
- K-Fold Cross Validation with stratified sampling
- Splits data into non overlapping and mutually exclusive samples
- Use (k-1) for Training, 1 for Validation



Models

- Logistic Regression
- Random Forest
- Cat Boost
- XGBoost
- Light Gradient Boosting

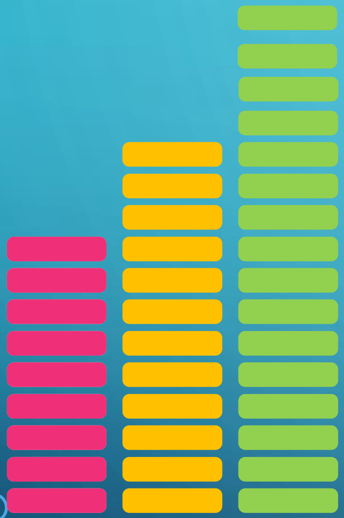
Logistic Regression



55	59	63
Model 1	**Model 2**	**Model 3**

- Model 1 – Logistic Regression with PCA
- Model 2 – Logistic Regression with CV (7-fold) and Regularization
- Model 3 – Logistic Regression with CV (10-fold) and Regularization

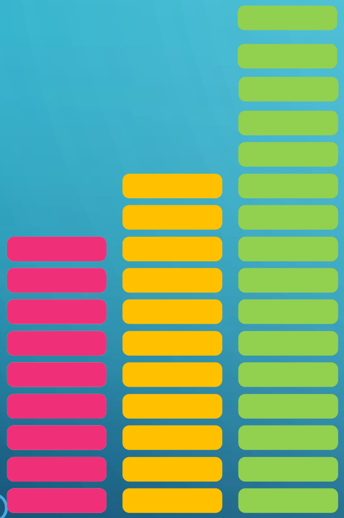
Random Forest



51	65	84
Model 1	**Model 2**	**Model 3**

- Model 1 – Vanilla Random Forest
- Model 2 – Random Forest with CV (5-fold) and grid search
- Model 3 – Random Forest with CV (5-fold) and grid search & AUC as objective function

CatBoost



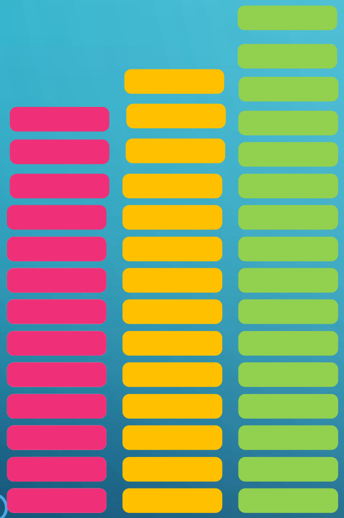
69 Model 1

73 Model 2

90 Model 3

- Model 1 – Random parameters in CatBoost
- Model 2 – CatBoost with CV (10-fold) and grid search
- Model 3 – CatBoost with CV (10-fold) and grid search and wider hyper-parameter tuning

XG Boost



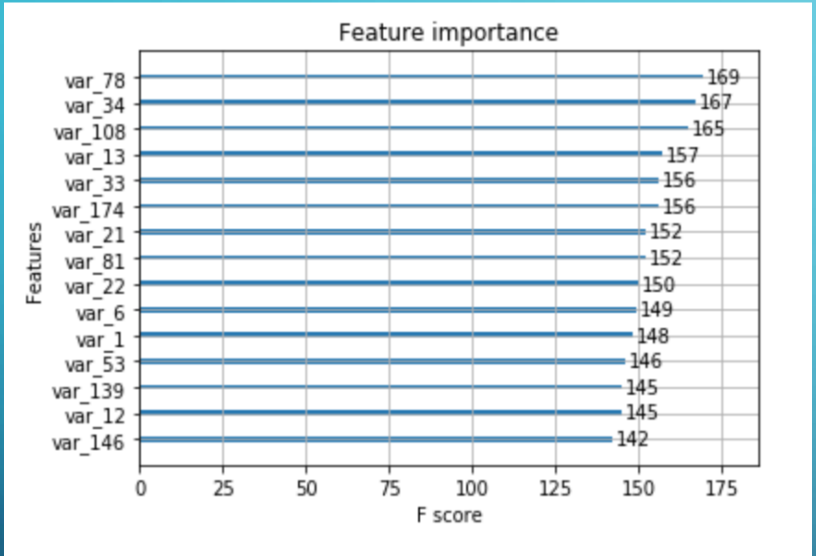
86
Model 1

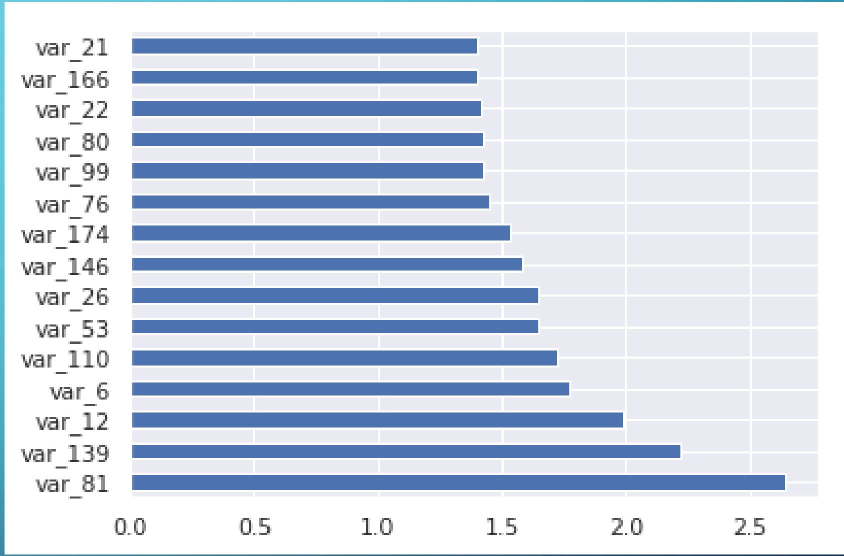
89
Model 2

90
Model 3

- Model 1 – XgBoost with grid search
- Model 2 – XgBoost with CV (10-fold), L1, L2 regularization
- Model 3 – XgBoost with CV (10-fold), Extra Features and wider hyper-parameter tuning

Feature Importance





XG Boost

Cat Boost

Light Gradient Boosting

89
Model 1

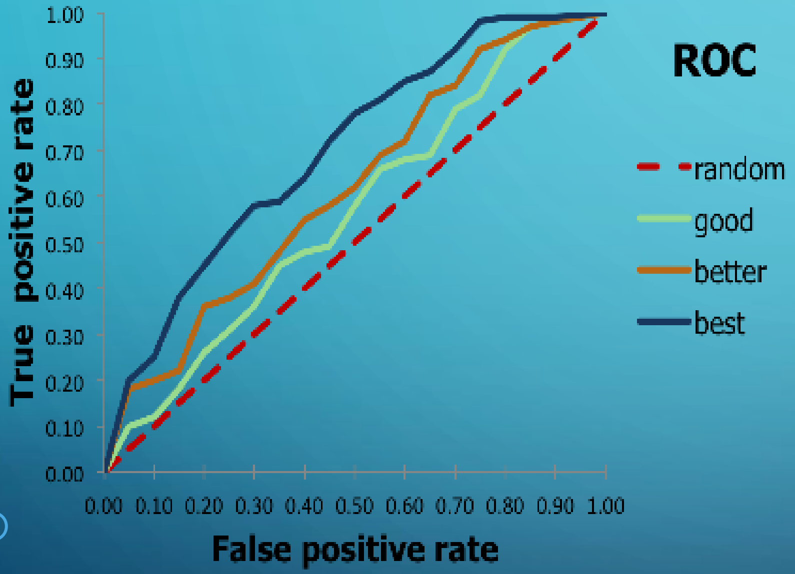
90
Model 2

- Model 1 – LGB with grid search
- Model 2 – LGB with more hyper parameter tuning, extra features, regularization

Ensemble Impact

- Better Performance
- AUC improvement
- GBM training take more time compared to Random Forest Bagging
- Overfitting in Boosting , hence L1 and L2 regularization parameters are also tuned
- Less likely to overfit in Random forest.
- RF is easier tune than GBM.
- Prediction time is more in RF (trees Count)

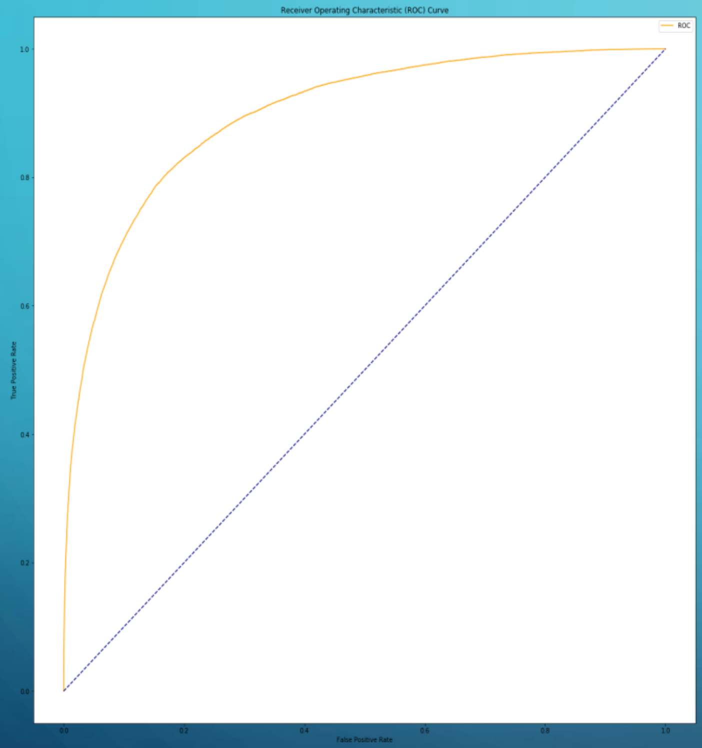
ROC-AUC Score



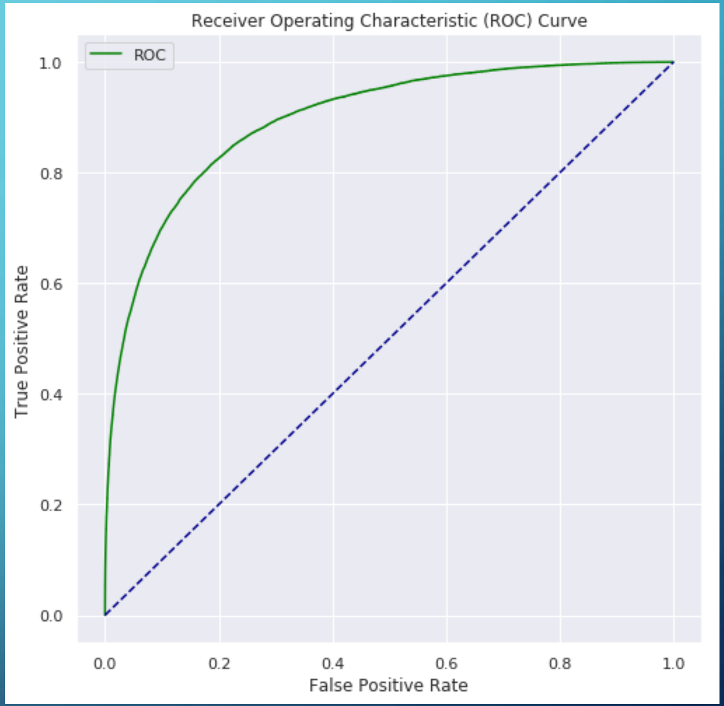
AUC : Area under ROC curve.
How well model performs.
How well model predicts the customer going to perform the particular transaction.

Higher the AUC, better the model

ROC Curve

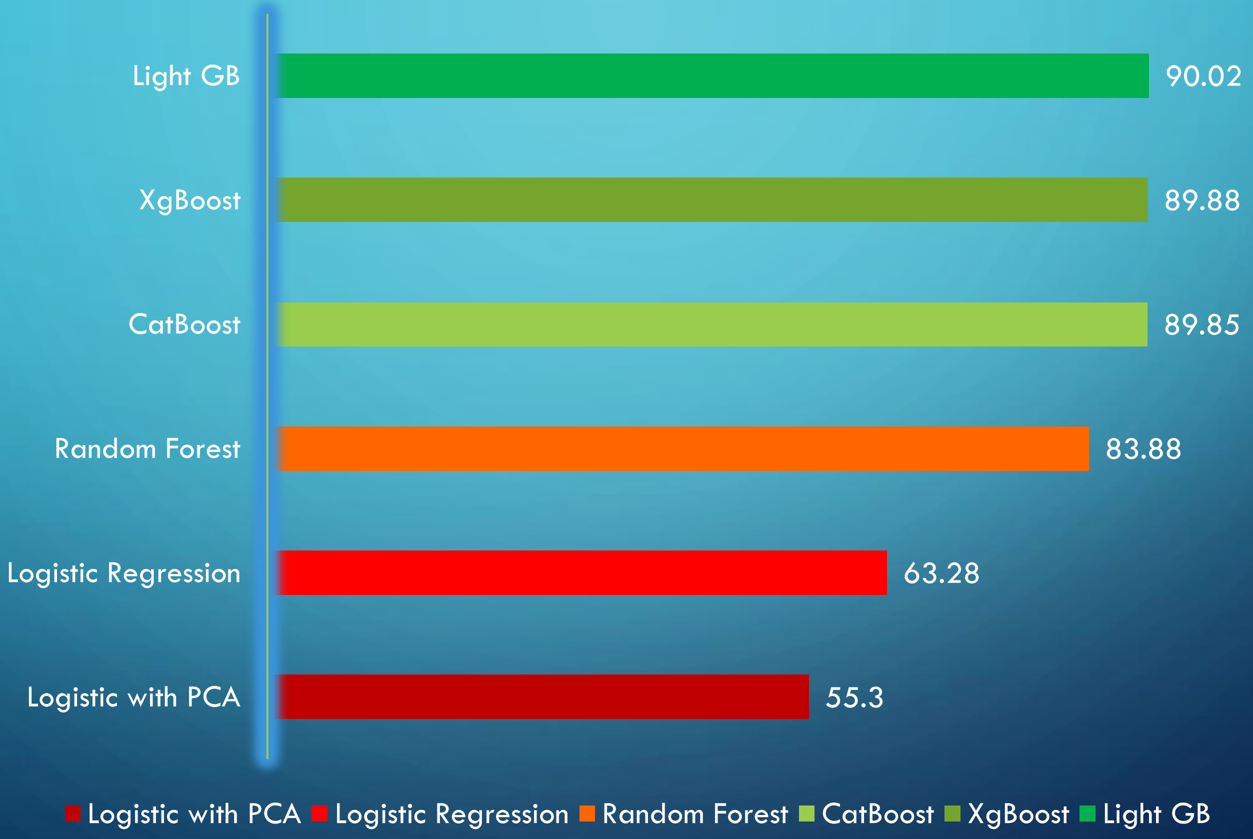


XG Boost AUC-ROC



Cat Boost

Evaluation


Light GB
90.02
XgBoost
89.88
CatBoost
89.85
Random Forest
83.88
Logistic Regression
63.28
Logistic with PCA
55.3
Logistic with PCA
Logistic Regression
Random Forest
CatBoost
XgBoost
Light GB

CHALLENGES

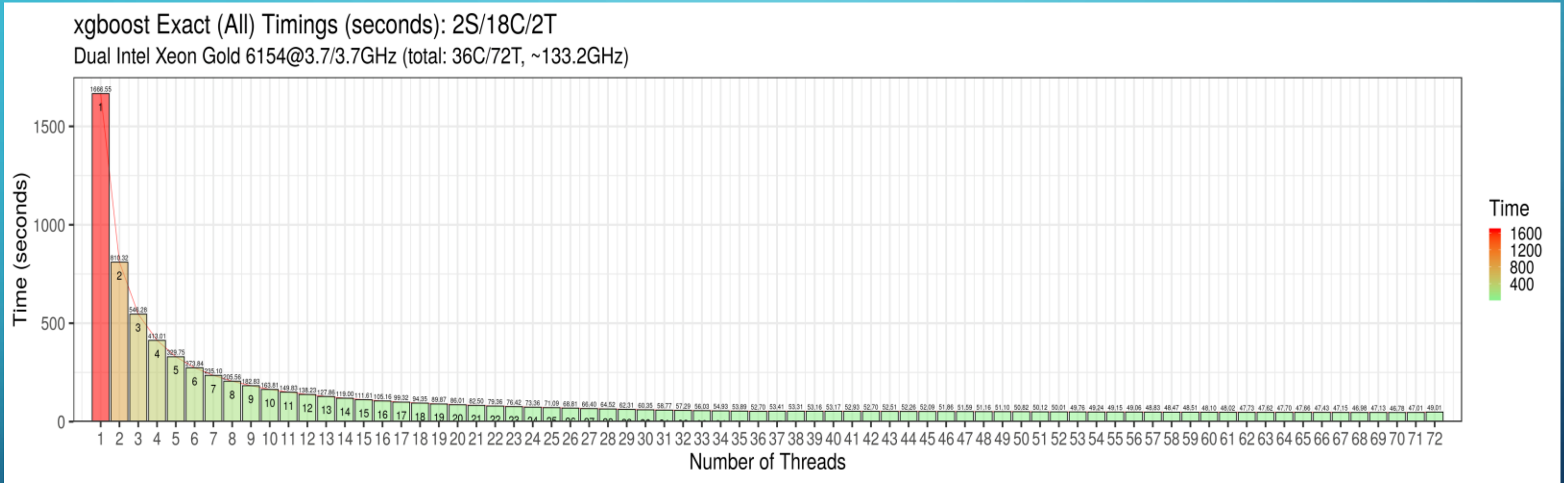
Training Time

Anonymous Features

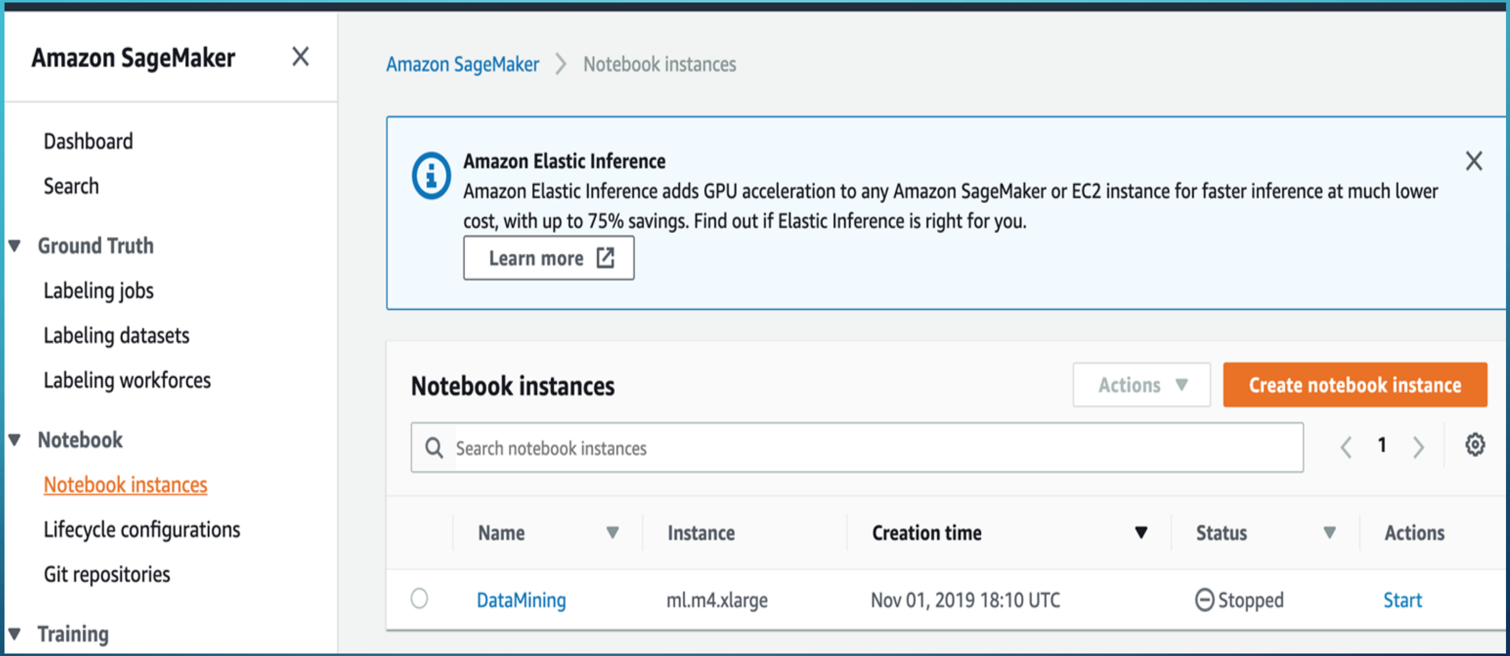
Hyper parameters

Computational Power

Xgboost Speed



AWS SAGE MAKER



TOOLS

scikit learn

jupyter

Pandas

Amazon SageMaker

CONDA

REFERENCES

- https://www.kaggle.com/c/santander-customer-transaction-prediction
- https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/
- https://medium.com/data-design/getting-the-most-of-xgboost-and-lightgbm-speed-compiler-cpu-pinning-374c38d82b86
- https://catboost.ai/docs/concepts/python-reference_catboostclassifier.html

THANK yOU

QUESTIONs ?