# Porto Seguro’s Safe Driver Prediction

Zexi Yuan, Jianqiao Liu, Wenlin Ou

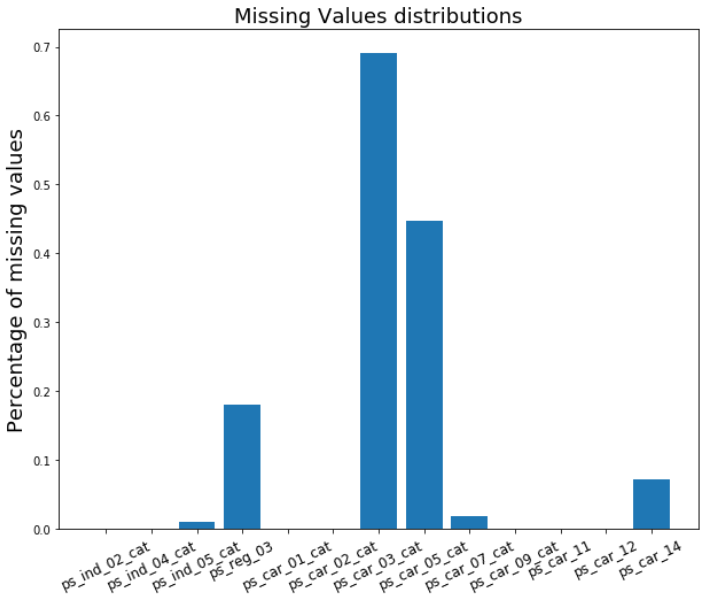
# Introduction

- Problem background
  - Dataset Description
- Approaches
  - Algorithms implemented
  - Evaluation Criteria
- Experiment Results
  - Data Preprocessing
  - Tuning Parameters
- Conclusions
- Q & A

# Part I: Problem Background

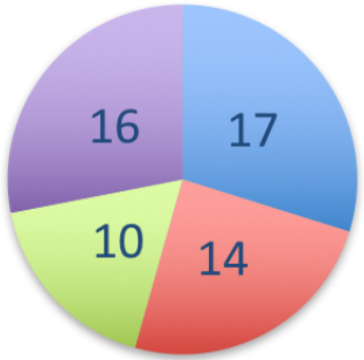
Porto Seguro is one of Brazil's largest auto and homeowner insurance companies, hence setting reasonable insurance prices is beneficial for both the company and customers. We are given a dataset that contains drivers' driving history and whether they have filed a claim in the past year. Our goal is to predict the probability that a driver will file a claim during the next year.

# Dataset




## Dataset:

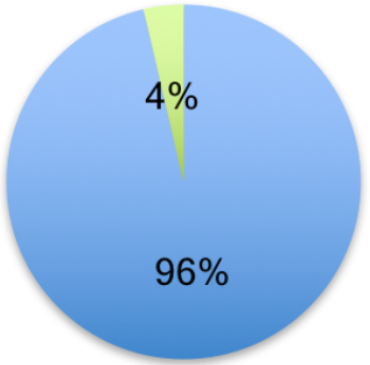
### 57 Features




Binary Categorical

Continuous Ordinal

### 595212 Instances




No Claim Claim

# Part II: Approaches

- Logistic Regression
- Random Forest
- XGB

# Model Evaluation: AUC & F-Score

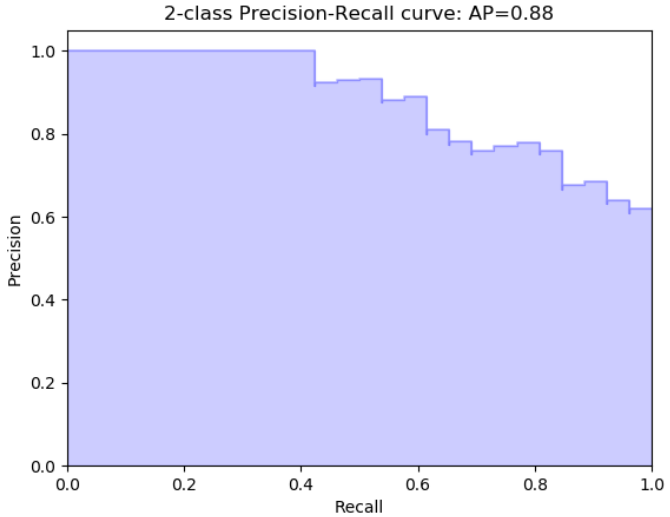
- AUC: area under ROC curve
  - ROC: TPR-FPR curve
- F-score

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}}.$$

- Recall (TPR) = TP/(TP+FN)
- Precision = TP/(TP+FP)
- FPR = FP/(FP+TN)

# Average Precision Score

- The precision-recall curve shows the tradeoff between precision and recall for different threshold.




- AP score is the area under the Precision-Recall Curve

$$\mathrm{AP} = \sum_{k=1}^{N} p(k)\Delta r(k)$$

# Part III: Results

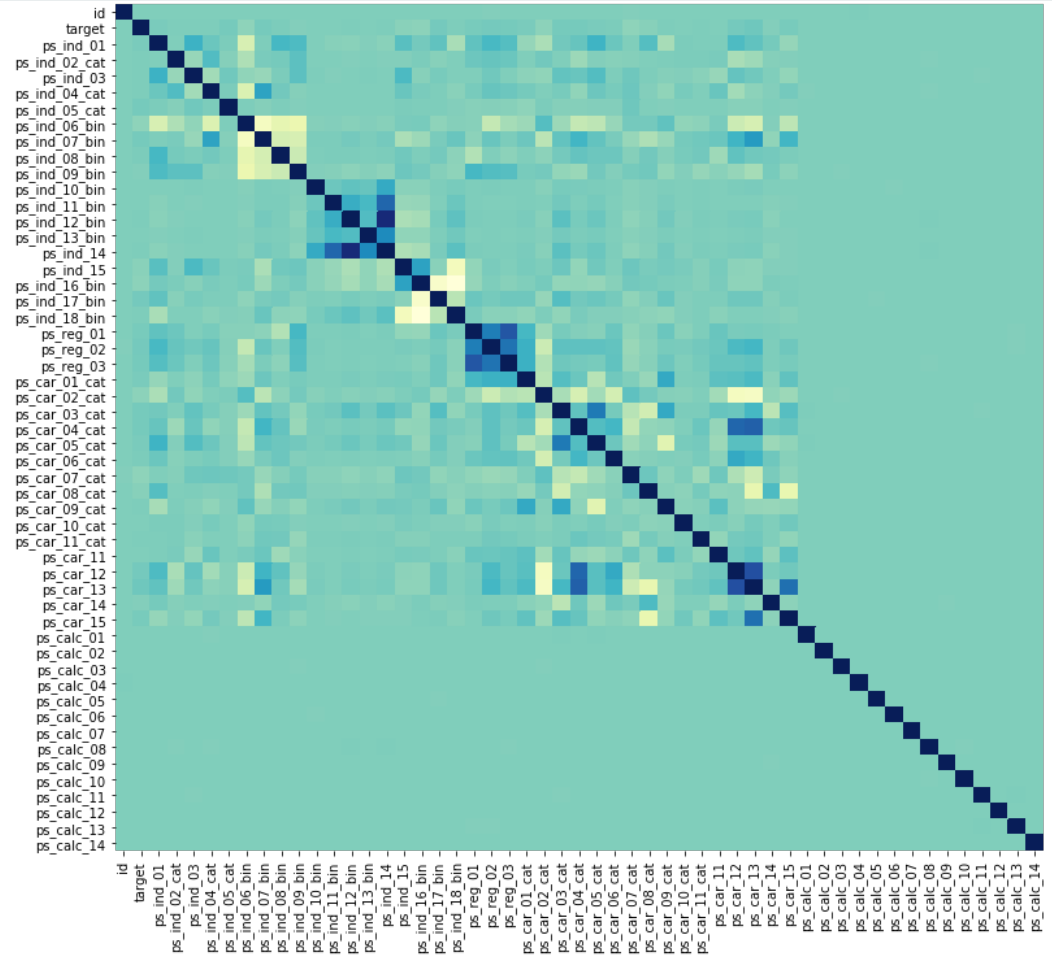
- Data Preprocessing
- Experiment on Algorithms

# Data Preprocessing

- Feature Selection
- One hot encoding
- Missing Values

# Pearson Correlation

$$\rho_{X,Y} = \frac{\mathrm{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$



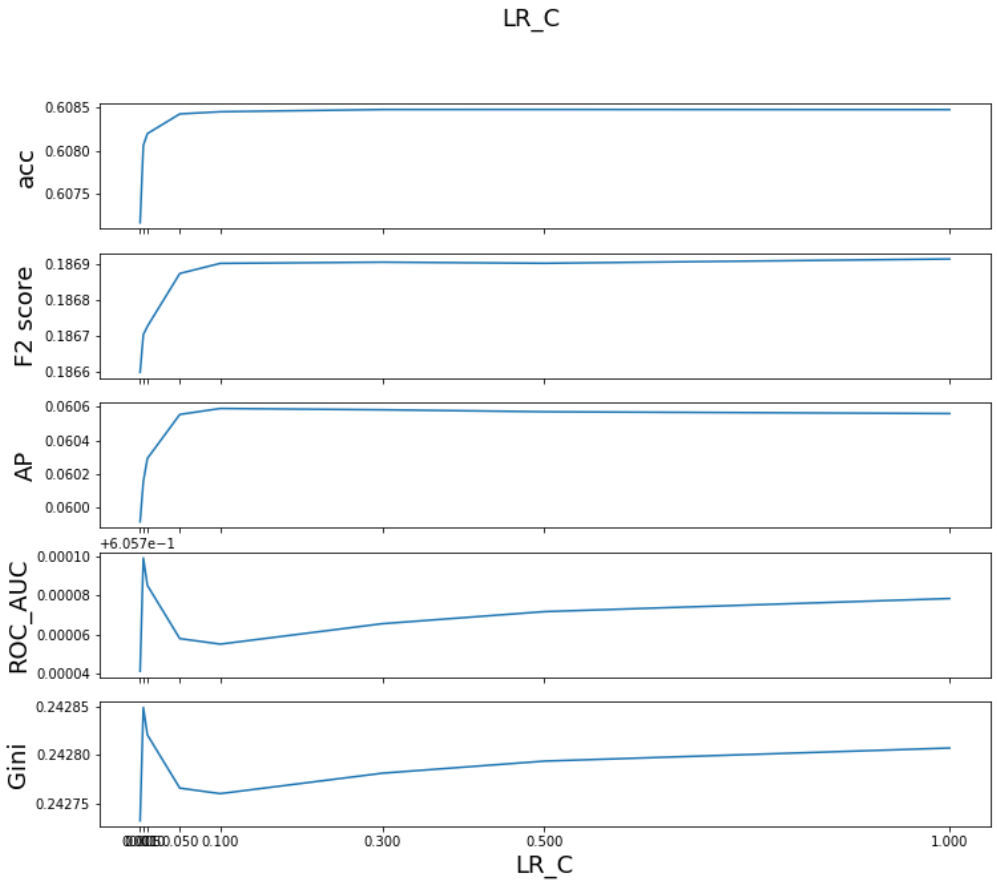
# What is “One Hot Encoding”?

| CompanyName | Categoricalvalue | Price |
|---|---|---|
| VW | 1 | 20000 |
| Acura | 2 | 10011 |
| Honda | 3 | 50000 |
| Honda | 3 | 10000 |

| VW | Acura | Honda | Price |
|---|---|---|---|
| 1 | 0 | 0 | 20000 |
| 0 | 1 | 0 | 10011 |
| 0 | 0 | 1 | 50000 |
| 0 | 0 | 1 | 10000 |

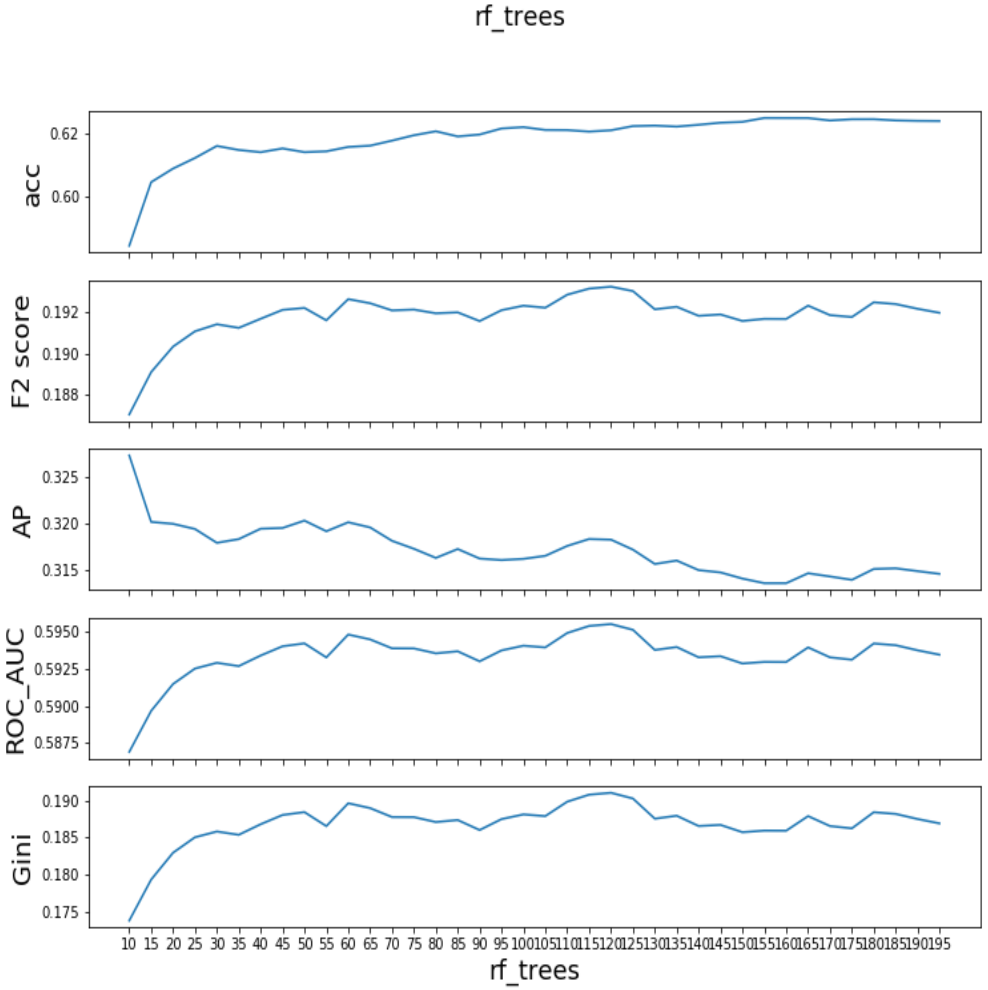
# Logistic Regression

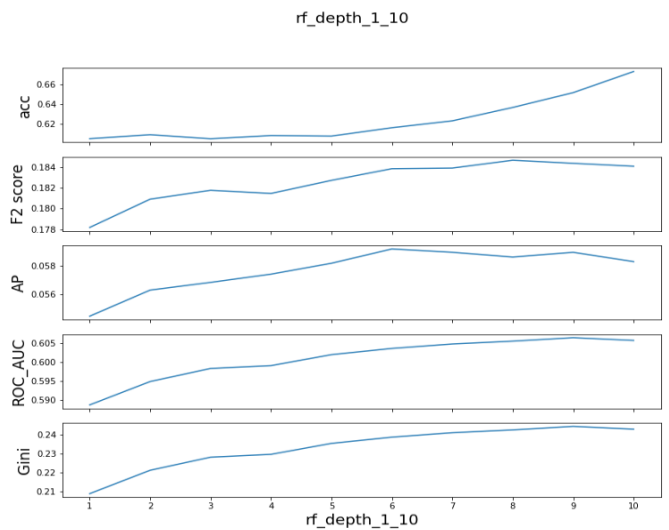
- Package: sklearn.linear_model.LogisticRegression
- Missing Value
  - Fill the missing values with mean for continuous data
- Data Standardization (0 mean and 1 std)
- Parameter: Inverse of regularization strength C = 0.1

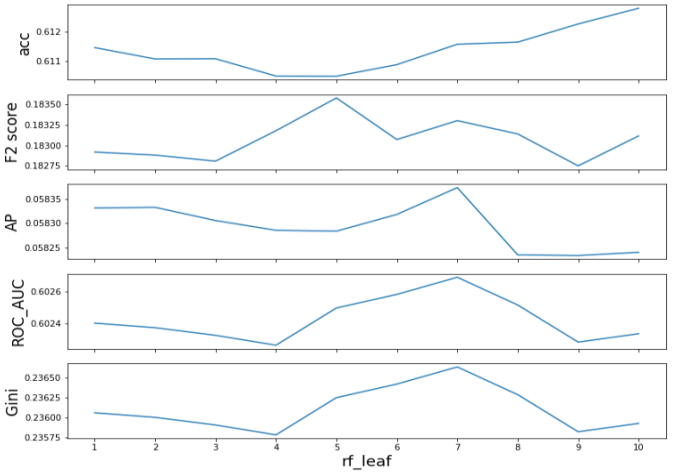

LR_C
acc
0.6085
0.6080
0.6075
F2 score
0.1869
0.1868
0.1867
0.1866
AP
0.0606
0.0604
0.0602
0.0600
+6.057e−1
ROC_AUC
0.00010
0.00008
0.00006
0.00004
Gini
0.24285
0.24280
0.24275
0.050
0.100
0.300
0.500
1.000
LR_C

# Random Forest

- Package: sklearn.ensemble.RandomForestClassifier
- Optimal Parameter
  - Number of trees: 120
  - Tree depth: 9
  - Leaf number: 7


rf_trees
acc
F2 score
AP
ROC_AUC
Gini
rf_trees


rf_depth_1_10
acc
F2 score
AP
ROC_AUC
Gini
rf_depth_1_10


acc
F2 score
AP
ROC_AUC
Gini
rf_leaf

# XGBoosting

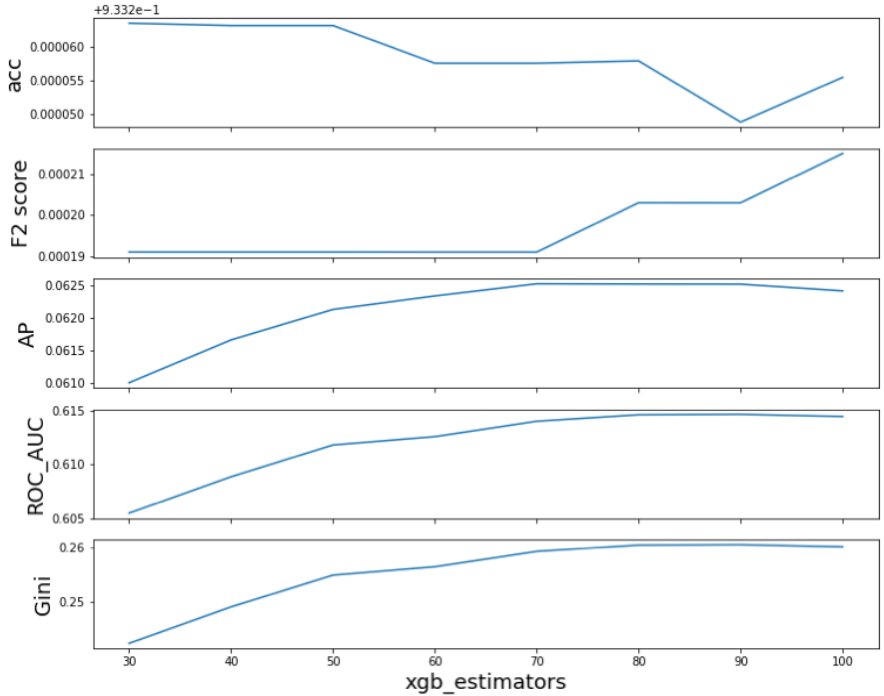
$$\text{obj}(\theta) = \sum_{i}^{n} l(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k)$$

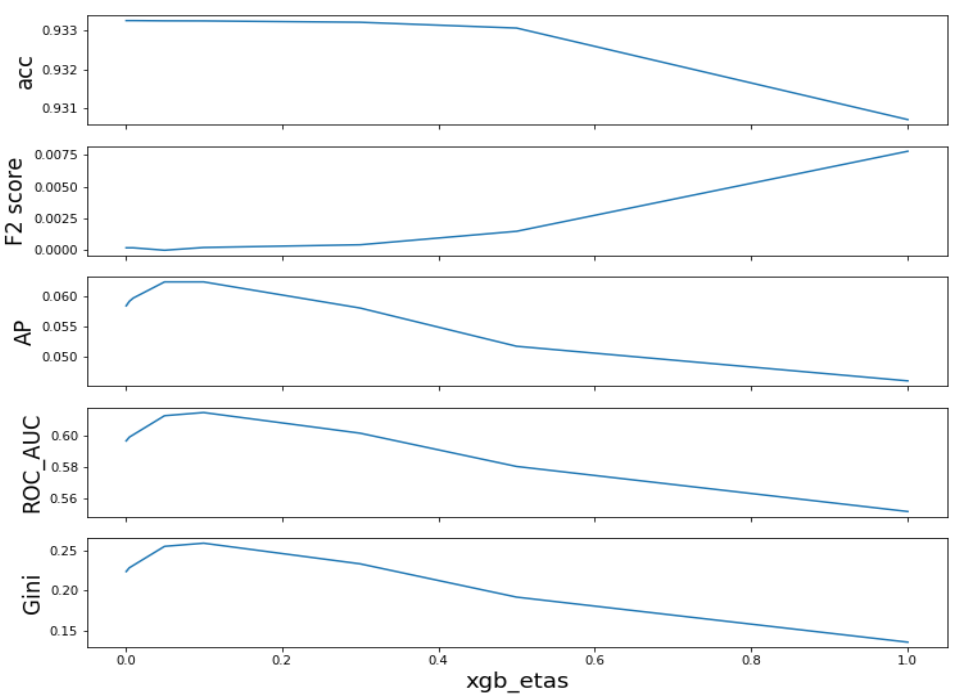
$$\text{obj}^{(t)} = \sum_{j=1}^{T} [G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2] + \gamma T$$

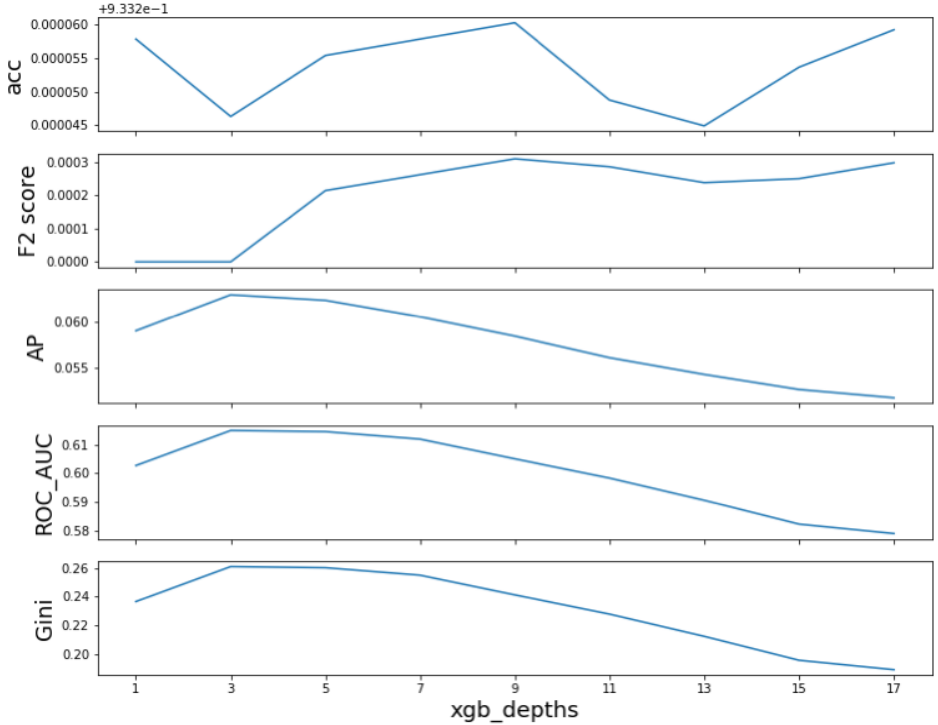
$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma$$

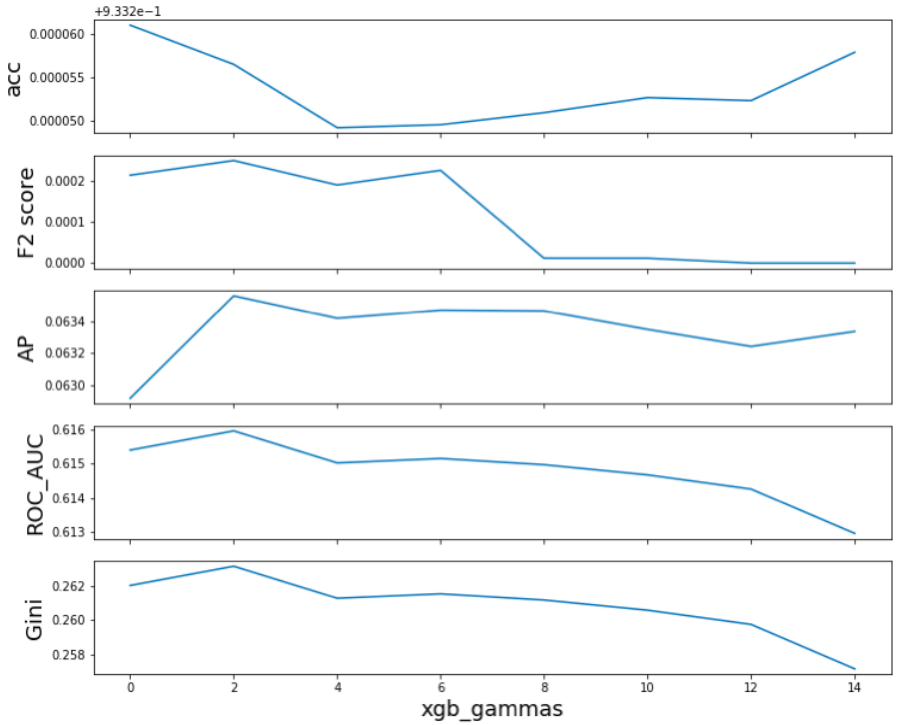
# XGBoosting

- Package from https://github.com/dmlc/xgboost/
- Optimal Parameters
  - number of estimators: 90
  - tree depth: 5
  - learning rate: 0.1
  - Gamma: 2


+9.332e−1
acc
0.000060
0.000055
0.000050
F2 score
0.00021
0.00020
0.00019
AP
0.0625
0.0620
0.0615
0.0610
ROC_AUC
0.615
0.610
0.605
Gini
0.26
0.25
30
40
50
60
70
80
90
100
xgb_estimators

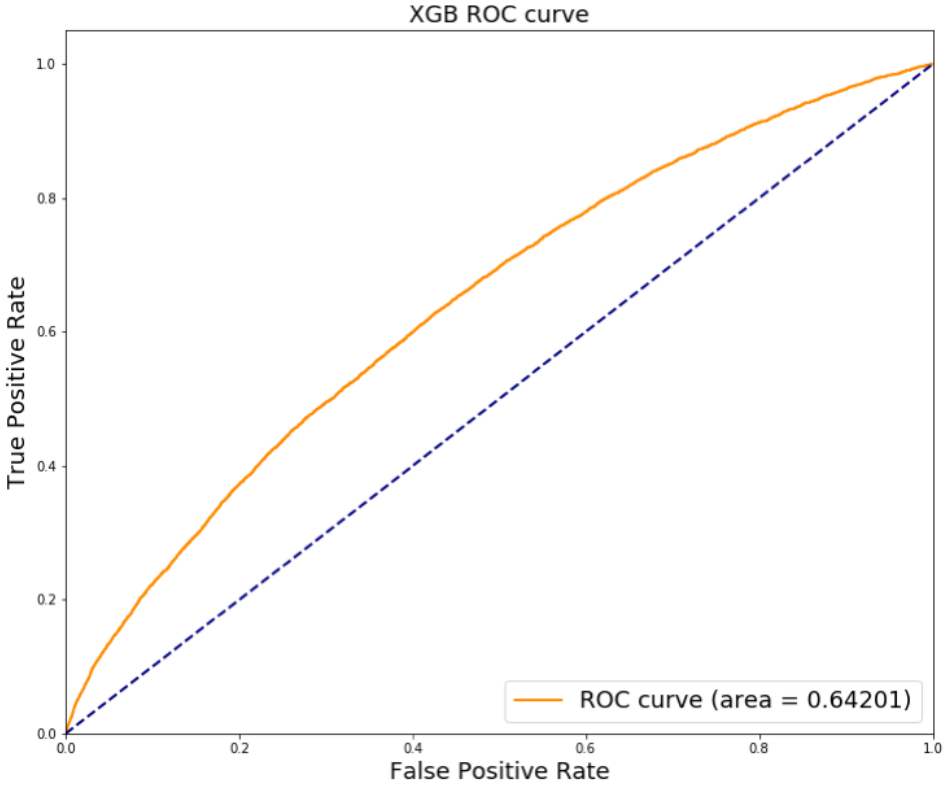

acc
0.933
0.932
0.931
F2 score
0.0075
0.0050
0.0025
0.0000
AP
0.060
0.055
0.050
ROC_AUC
0.60
0.58
0.56
Gini
0.25
0.20
0.15
0.0
0.2
0.4
0.6
0.8
1.0
xgb_etas

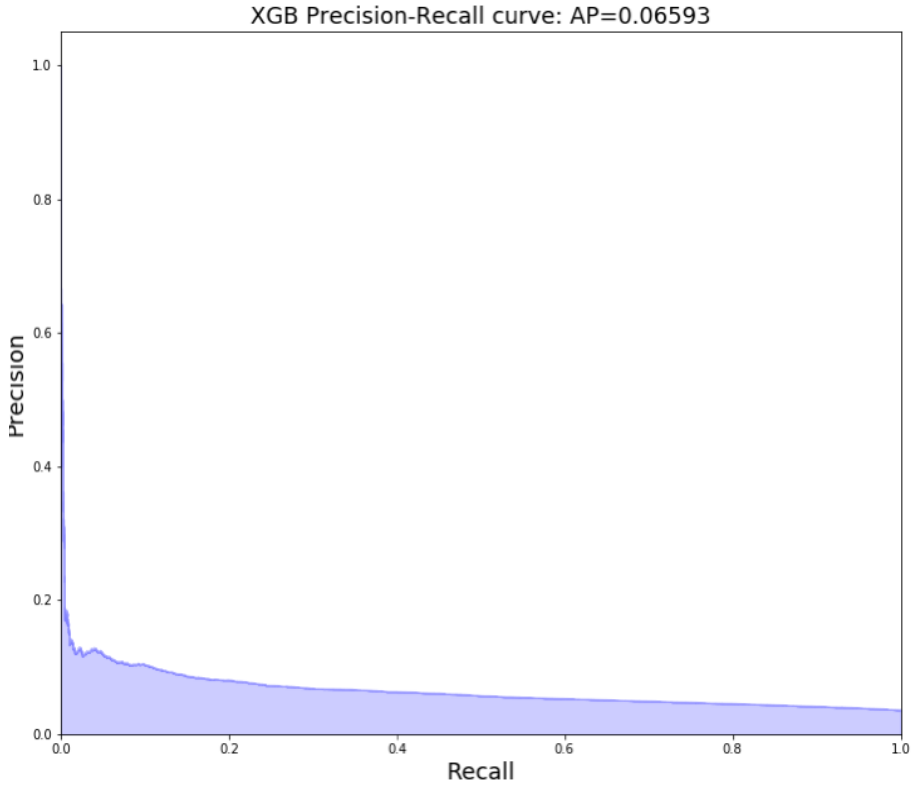

+9.332e−1
acc
F2 score
AP
ROC_AUC
Gini
xgb_depths


+9.332e−1
acc
F2 score
AP
ROC_AUC
Gini
xgb_gammas

# Comparison

| | Accuracy | F2 score | AP | ROC_AUC | Gini |
|---|---|---|---|---|---|
| **XGB** | 0.964127 | 0.000936 | 0.065808 | 0.642011 | 0.284023 |
| **RF** | 0.665444 | 0.192977 | 0.061271 | 0.633579 | 0.267158 |
| **LR** | 0.627770 | 0.192531 | 0.063220 | 0.632558 | 0.265116 |


XGB ROC curve
True Positive Rate
False Positive Rate
ROC curve (area = 0.64201)
0.0
0.2
0.4
0.6
0.8
1.0


XGB Precision-Recall curve: AP=0.06593
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

# Part IV: Conclusions

- Something to share:
    - AP Score as an extra evaluation criterion
- Challenges we encountered:
    - Feature selection
    - No time for grid search

# Q & A:

Thank You!