Springboard

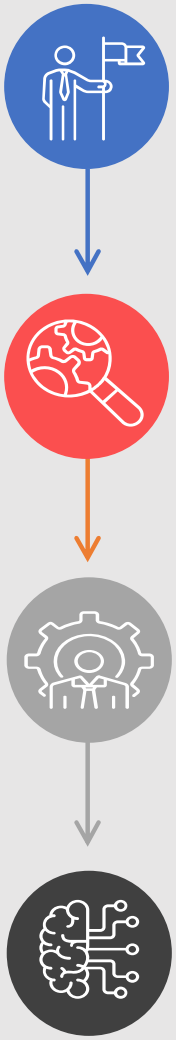
MARKETING CAMPAIGN

Insight and Prediction

Jonathan Lampkin

Special Thanks to:

Yuxuan Xin



Problem Statement

Exploratory Data Analysis

Data Pre-Processing

Modeling & Evaluation

Metric Definition

Recall

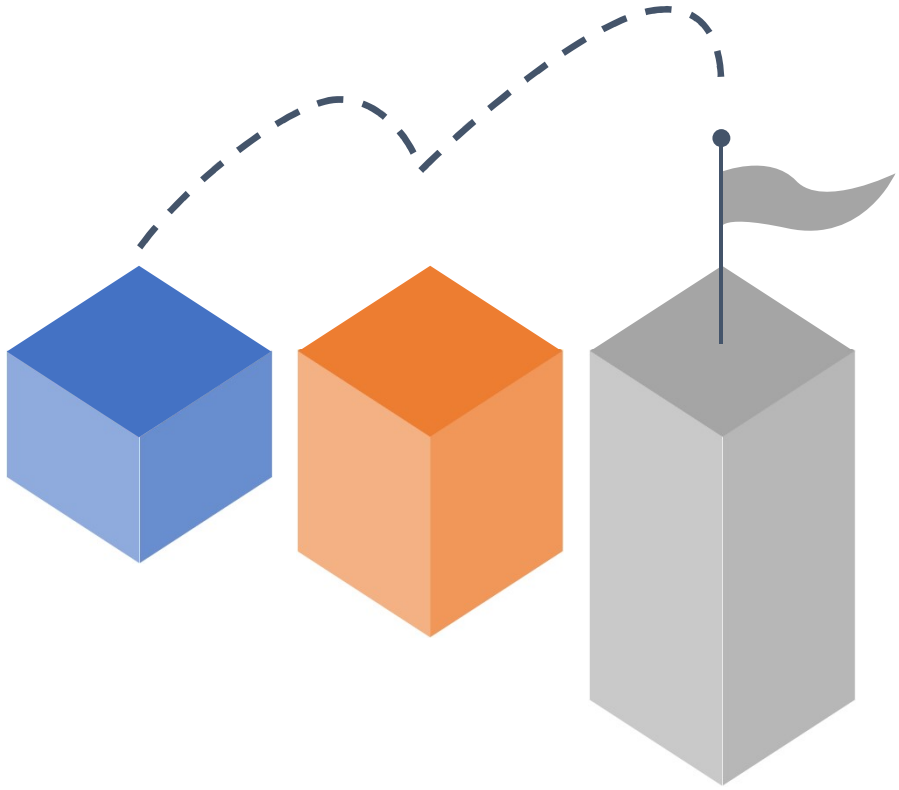
Out of those who would've made a deposit, how many did we correctly target?

Precision

How many of those predicted to subscribe to a term-deposit, did?

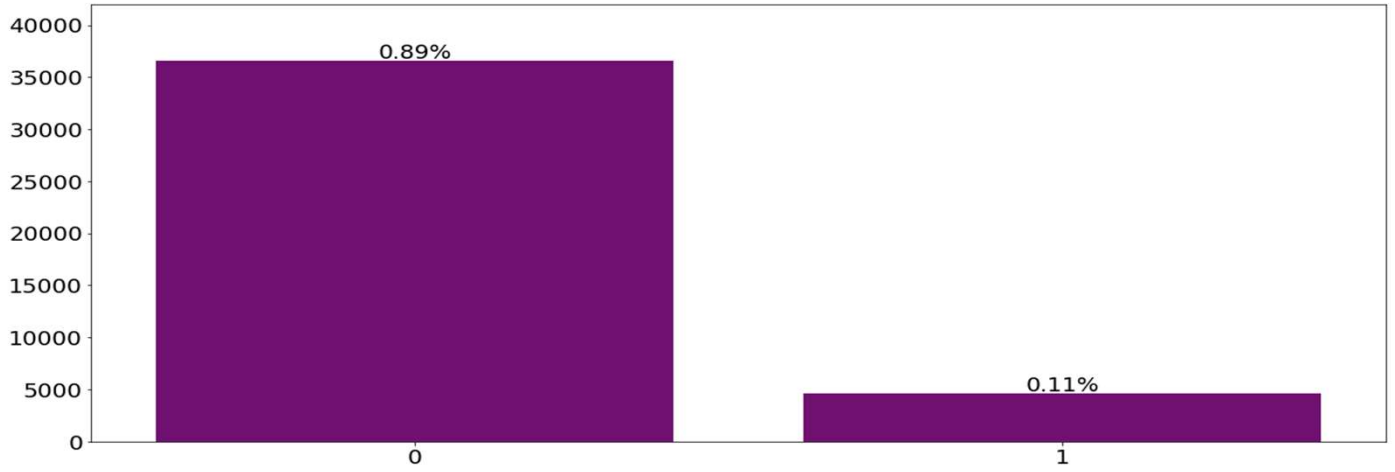
F1 – Score

Harmonic mean of Recall and Precision



Quick look at the data

Imbalance of classes

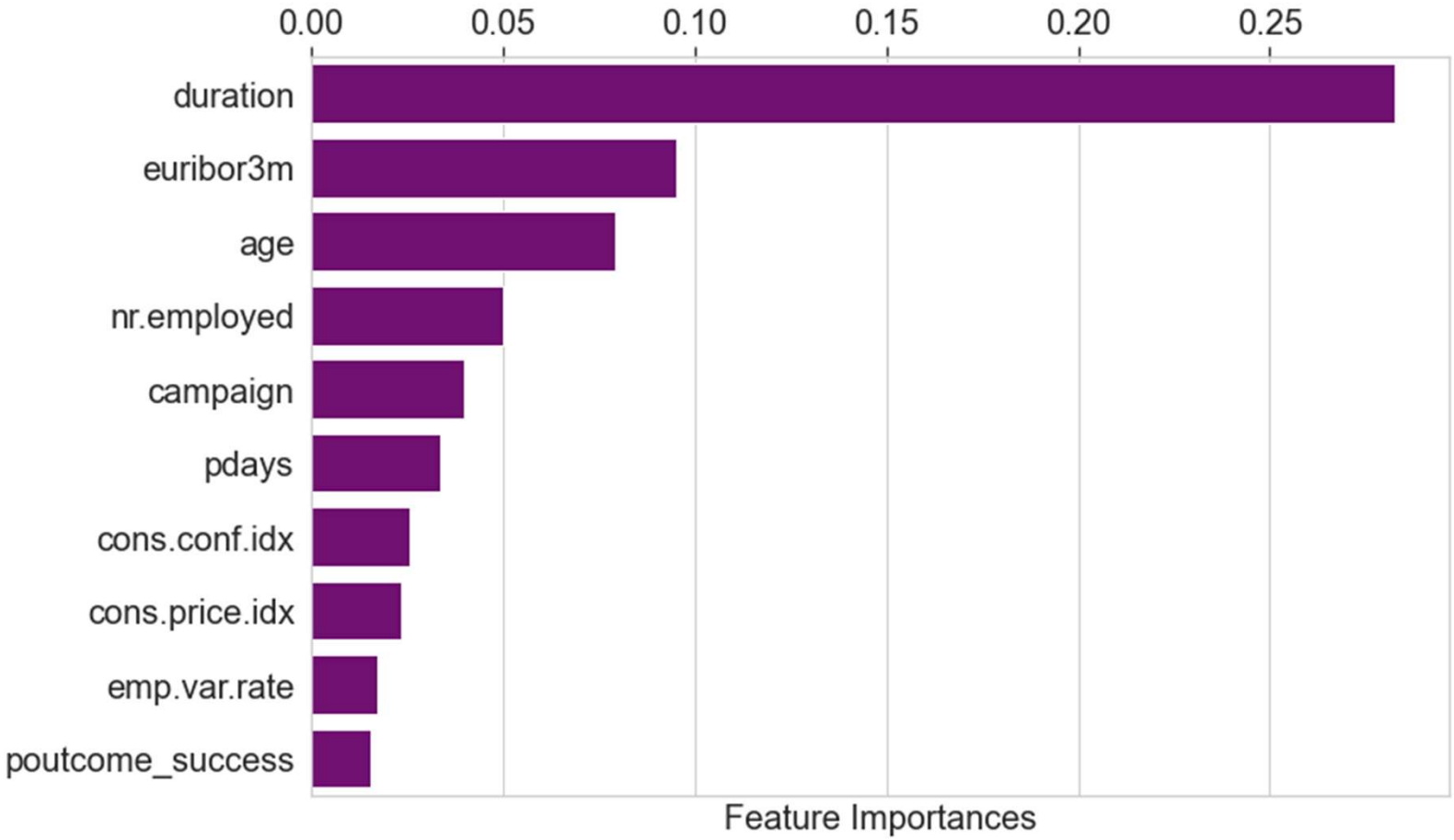



```
In [4]: df[['age','job','marital','education','duration','euribor3m','cons.conf.idx','y']].sample(5)
```

Out[4]:

	age	job	marital	education	duration	euribor3m	cons.conf.idx	y
11678	36	admin.	single	high.school	16	4.959	-41.8	no
7557	33	technician	single	professional.course	613	4.864	-36.4	no
20621	32	admin.	married	university.degree	1842	4.965	-36.1	yes
1734	25	technician	single	professional.course	156	4.855	-36.4	no
34043	29	admin.	single	high.school	230	1.281	-46.2	no

Feature Importance



Data Prepared for Machine Learning

```
# Normalizing feature matrix using Z-score
X = (X - X.mean()) / X.std()
```

	age	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	job_admin.	job_blue-collar	job_entrepreneur	j
13944	-0.194224	-0.204906	0.195412	-0.349490	0.839050	0.591417	-0.474785	0.773566	0.845160	-0.582016	1.857619	-0.191428	
27405	0.573438	-0.565915	0.195412	-0.349490	-0.115780	-0.648995	-0.323538	0.230453	0.398110	-0.582016	-0.538310	-0.191428	
21139	1.149185	-0.204906	0.195412	-0.349490	0.839050	-0.227462	0.951256	0.773566	0.845160	-0.582016	-0.538310	-0.191428	
29500	-0.865929	1.239130	0.195412	-0.349490	-1.197921	-0.864944	-1.425479	-1.277808	-0.940270	-0.582016	-0.538310	-0.191428	
4459	-0.769971	-0.565915	0.195412	-0.349490	0.648084	0.722714	0.886436	0.711875	0.331676	-0.582016	-0.538310	-0.191428	
3417	0.573438	1.600139	0.195412	-0.349490	0.648084	0.722714	0.886436	0.714181	0.331676	-0.582016	-0.538310	-0.191428	
9145	-0.769971	-0.204906	0.195412	-0.349490	0.839050	1.536410	-0.280325	0.775872	0.845160	1.718125	-0.538310	-0.191428	
17060	-0.769971	4.849221	0.195412	-0.349490	0.839050	0.591417	-0.474785	0.772989	0.845160	-0.582016	-0.538310	5.223772	
10602	1.341100	-0.565915	0.195412	-0.349490	0.839050	1.536410	-0.280325	0.772413	0.845160	-0.582016	-0.538310	-0.191428	
32994	1.341100	-0.204906	-5.095879	1.671116	-1.197921	-1.179366	-1.231019	-1.338923	-0.940270	-0.582016	1.857619	-0.191428	

Machine Learning Algorithms

Random Forest

- Tree Based Ensemble using Bagging

K Nearest Neighbor

- Distance Based Clustering Algorithm

Logistic Regression

- Probabilistic Linear Function

Extreme Gradient Boosting

- Gradient Boosting algorithm using parallel processing

Light Gradient Boosting

- Computationally conservative version of Gradient Boosting

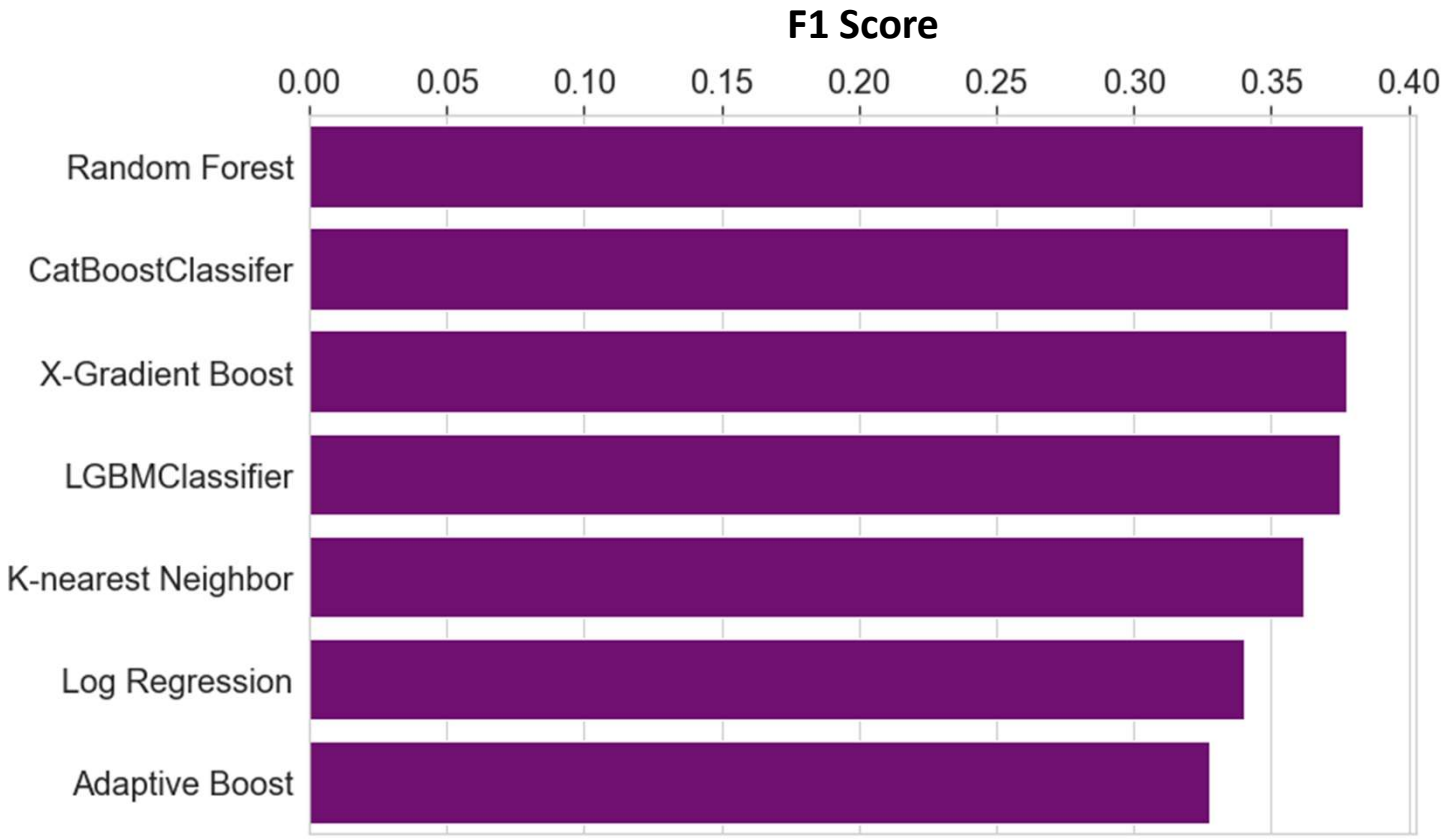
Categorical Boosting

- Gradient Boosting Algorithm designed for handling categorical features

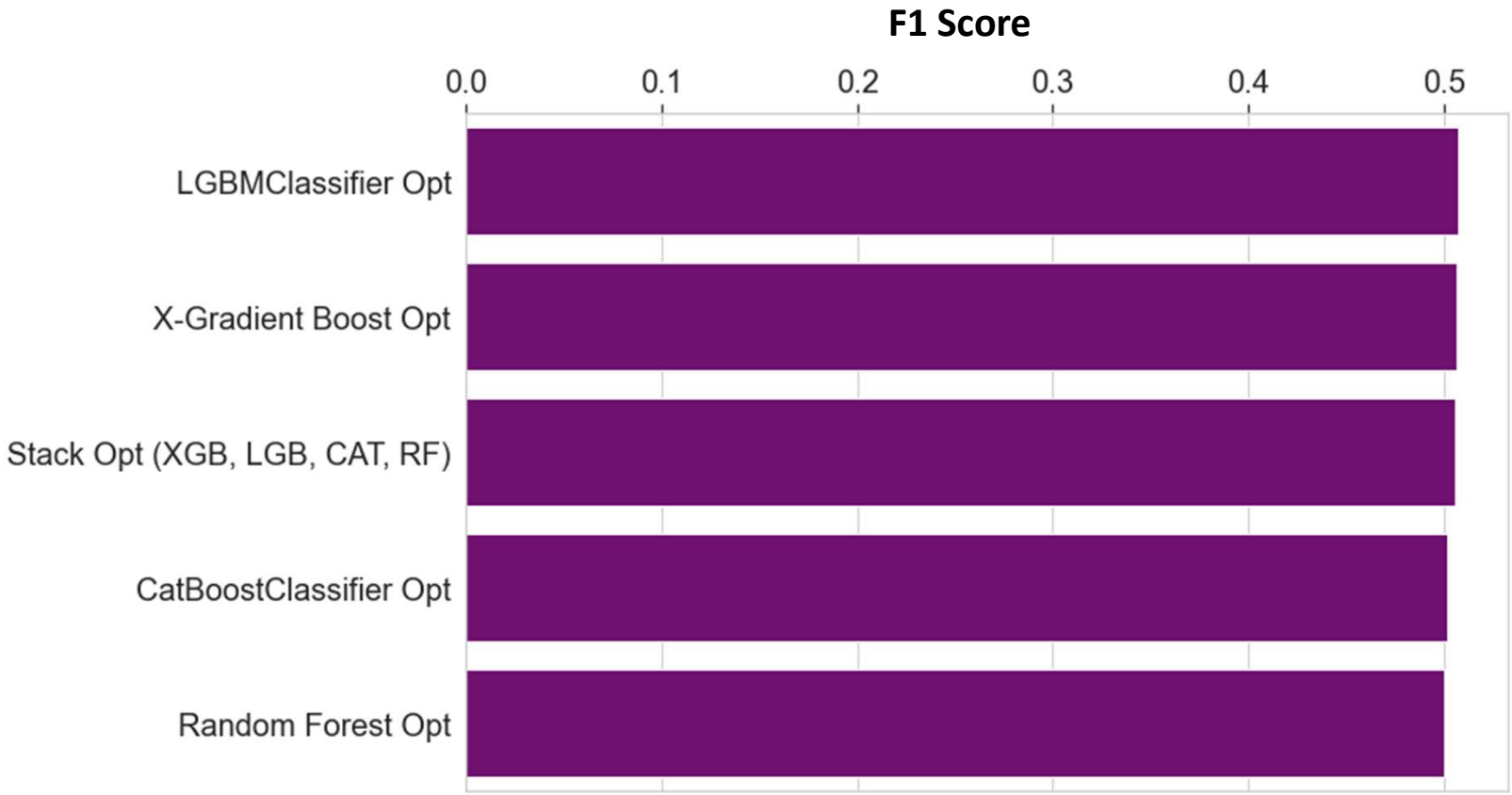
Adaptive Boosting

- Early version of Boosting that doesn’t use a loss function

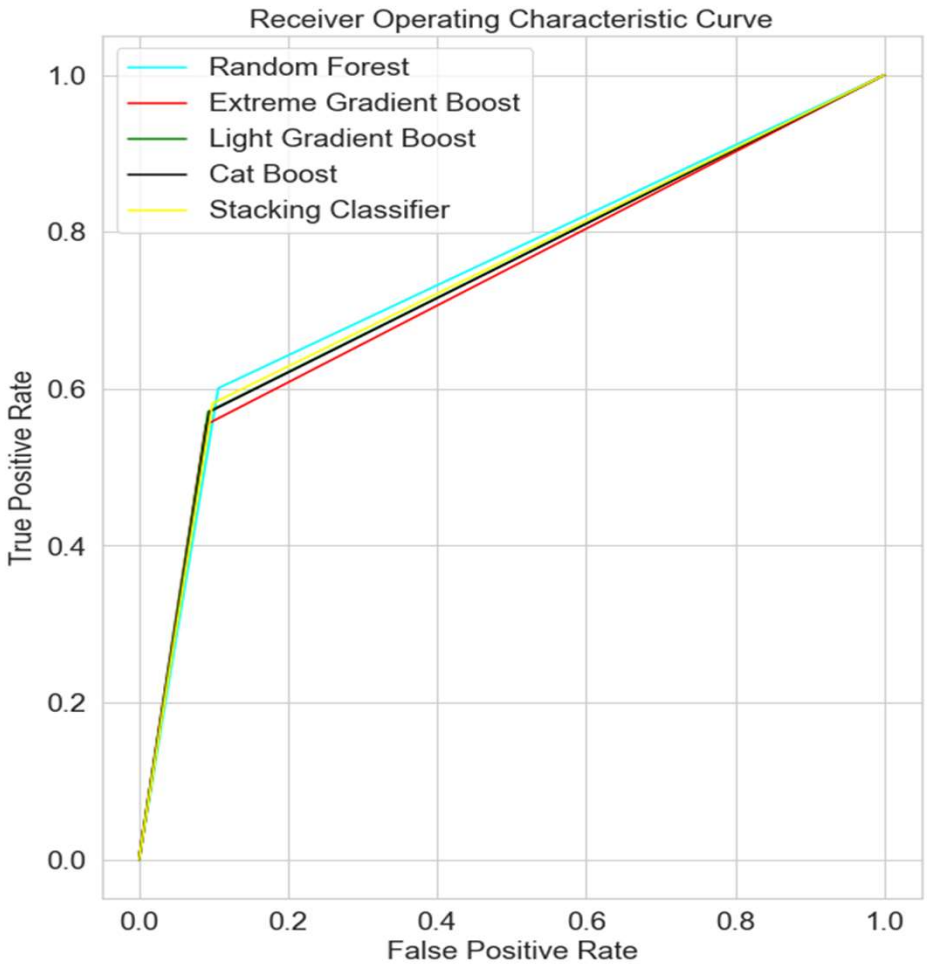
Default Performance

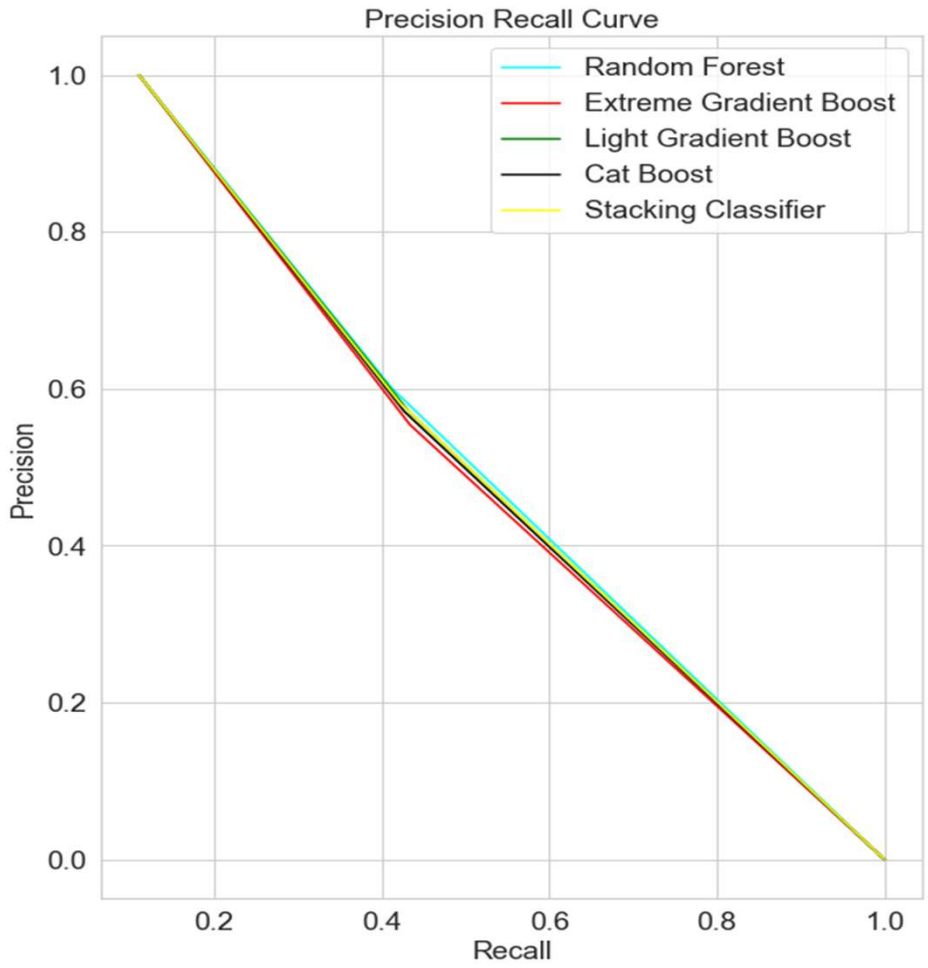


Optimized Performance



ROC and PR Curve





Performance Result