# Telco Churn Rate Analysis

Nathan Murstein, Hisham Salem, Dany Stefan, Jeewon Kim, Diwei Zhu, Oleg Kartavtsev, Uzair Ahmad

## Team Composition

1. **Nathan** Murstein (NathanMurstein) - Data Scientist
2. **Hisham** Salem (HishamSalem) - Data Analyst
3. **Dany** Stefan (dany-stefan) - Data Scientist
4. **Jeewon** Kim (jeewonk) - Data Scientist
5. **Diwei** Zhu (JuniperZhuDiwei) - Business Analyst
6. **Oleg** Kartavtsev (oleg19989) - Business Intelligence
7. **Uzair** Ahmad (uzairahmadxy) - Project Manager

Github Repository - MMA4_ChurnPrediction

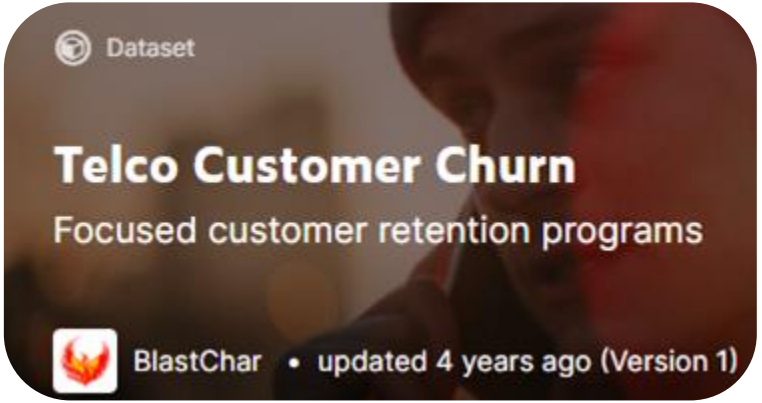
# Context and Data Source

**Objective** – **reduce churn** among Telco's clients

**Why?**

1. Churn leads to higher Customer Acquisition Costs & reduced revenue – acquiring new customers is more costly than keeping the existing ones
2. High churn rates are more likely to compound over time

**Ideal scenario for the company** – increased revenue, higher customer satisfaction and loyalty, higher market share




*https://www.kaggle.com/blastchar/telco-customer-churn*

**Churn** – the annual percentage rate at which customers stop subscribing to a service

**Possible reasons** – cancelling the service altogether, switching to a competitor, etc.

# Methodology Overview


**Expected outcome**
Prediction model - optimal model with high F1
Optimization - optimal monetary value of coupon promotion

**Primary performance measure** - **F1**

**Why?** Based on our tests a balanced approach is the the most monetary worthwhile approach for our business strategy hence why we optimize for f1.
In the scope of the project, identifying churn is more important than eliminating false positives and missing actual positive cases

# Exploratory data analysis

## Dataset overview:

- **Target**: Churn (binary)
- **Customer demographic/subscription variables:** gender, senior, monthly charge, contract type, tenure,
- **Telco service variables:** phone/internet service type, security/backup service, tech support…

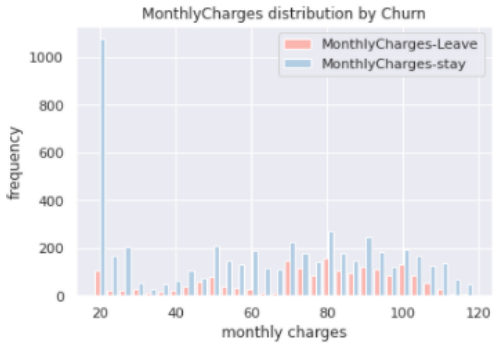
## Data preparation:

- Drop NA, cleaned empty values (shown as " " (a space string), not null)
- Encoded categorical and binary variables into integers

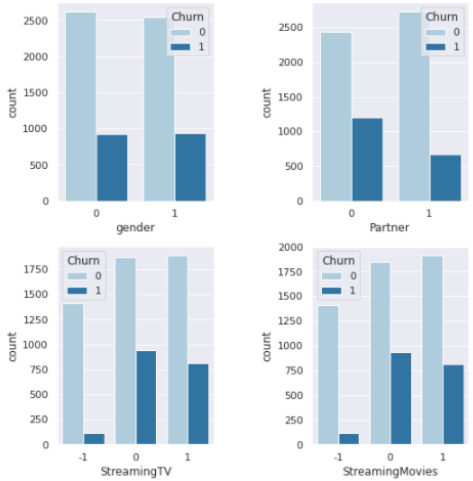
## EDA:

(full version on GitHub)

*Distribution of numerical data*




*Distribution of categorical data*




*Correlation matrix*

| | Churn | gender | SeniorCitizen | Partner | Dependents |
|---|---|---|---|---|---|
| Churn | 1.000000 | 0.008545 | 0.150541 | -0.149982 | -0.163128 |
| gender | 0.008545 | 1.000000 | 0.001819 | 0.001379 | -0.010349 |
| SeniorCitizen | 0.150541 | 0.001819 | 1.000000 | 0.016957 | -0.210550 |
| Partner | -0.149982 | 0.001379 | 0.016957 | 1.000000 | 0.452269 |
| Dependents | -0.163128 | -0.010349 | -0.210550 | 0.452269 | 1.000000 |
| tenure | -0.354049 | -0.005285 | 0.015683 | 0.381912 | 0.163386 |
| PhoneService | 0.011691 | 0.007515 | 0.008392 | 0.018397 | -0.001078 |
| MultipleLines | 0.036148 | 0.010284 | 0.113769 | 0.118037 | -0.019178 |
| InternetService | 0.316350 | 0.009643 | 0.259030 | 0.000938 | -0.177789 |
| OnlineSecurity | 0.023014 | 0.013233 | 0.081766 | 0.092034 | -0.028964 |
| OnlineBackup | 0.073934 | 0.011081 | 0.144762 | 0.091536 | -0.061970 |

# Causality check

Check causality between every binary variable and the Churn by treating binary variables as treatments in turn.

ATE result of all tests:

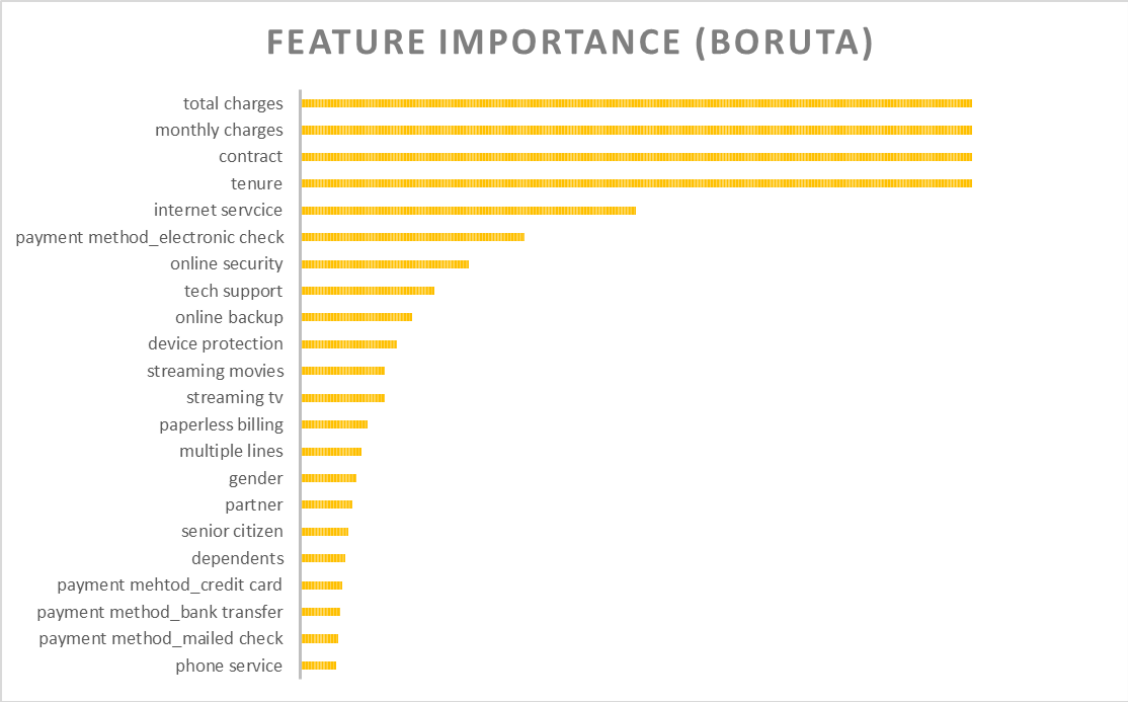
| | Treatment | LR | XGBoost_T | Neural Network | XGBoost_BaseX | XGBoost_BaseR |
|---|---|---|---|---|---|---|
| 0 | PaperlessBilling | 0.046118 | 0.052412 | 0.126943 | 0.048250 | 0.045284 |
| 1 | gender | 0.003772 | 0.003416 | 0.006964 | 0.003699 | 0.004890 |
| 2 | SeniorCitizen | 0.045521 | 0.058722 | -0.248546 | 0.058871 | 0.052779 |
| 3 | Partner | -0.000546 | 0.005794 | -0.088505 | 0.004461 | 0.002543 |
| 4 | Dependents | -0.021127 | -0.019804 | -0.094128 | -0.017211 | -0.017628 |
| 5 | PhoneService | -0.091797 | -0.076322 | 0.236776 | -0.133274 | 0.019827 |
| 6 | PaymentMethod_Bank transfer (automatic) | 0.033476 | -0.002217 | -0.049749 | 0.024361 | 0.020069 |
| 7 | PaymentMethod_Credit card (automatic) | 0.027120 | -0.030238 | -0.327045 | -0.012907 | -0.001930 |
| 8 | PaymentMethod_Electronic check | 0.102619 | 0.050909 | 0.238761 | 0.049111 | 0.017346 |
| 9 | PaymentMethod_Mailed check | 0.026782 | -0.042788 | -0.096065 | -0.021721 | -0.004676 |

- Target: either 0 or 1.
- The largest ATE is 0.23, but not consistent across tests.
- Cannot say there is significant differences in ATEs for Churn for the binary predictors

# Modelling

- Train-Validation-Test Approach
- Feature Selection - Boruta
- Imbalanced data - Oversampling, Undersampling
- Models: Logistic Regression, Naive Bayes, SVM, Random Forest, LGBM, XGB,
- Model Tuning - Optuna
- Custom Prediction Threshold - Precision, Recall Curves

# Feature Selection



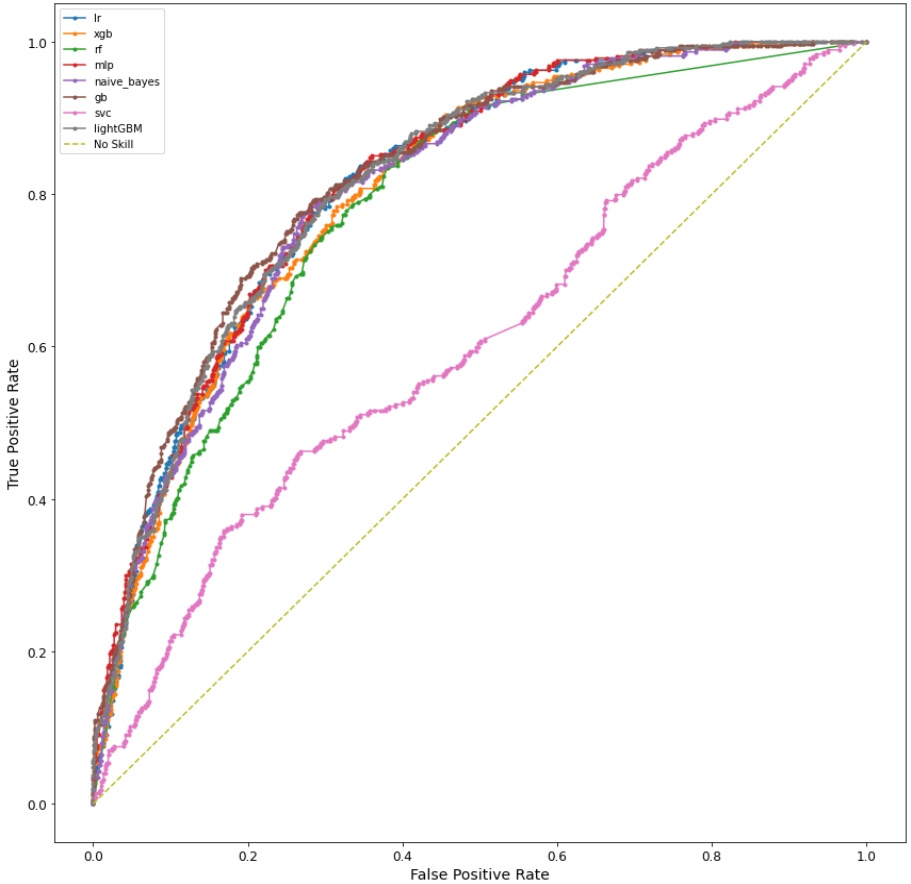
# Train 5 fold Average Oversample Results

| | Accuracy | Precision | Recall | F1Score |
|---|---|---|---|---|
| naive_bayes | 0.773608 | 0.700948 | 0.954722 | 0.808359 |
| mlp | 0.784746 | 0.763662 | 0.838015 | 0.791452 |
| gb | 0.807869 | 0.765488 | 0.887651 | 0.821981 |
| lr | 0.797579 | 0.746702 | 0.900969 | 0.816447 |
| svc | 0.575303 | 0.576032 | 0.571913 | 0.573936 |
| rf | 0.822760 | 0.810742 | 0.842373 | 0.826166 |
| xgb | 0.791162 | 0.710759 | 0.982324 | 0.824724 |
| lightGBM | 0.782082 | 0.699186 | 0.990799 | 0.819784 |

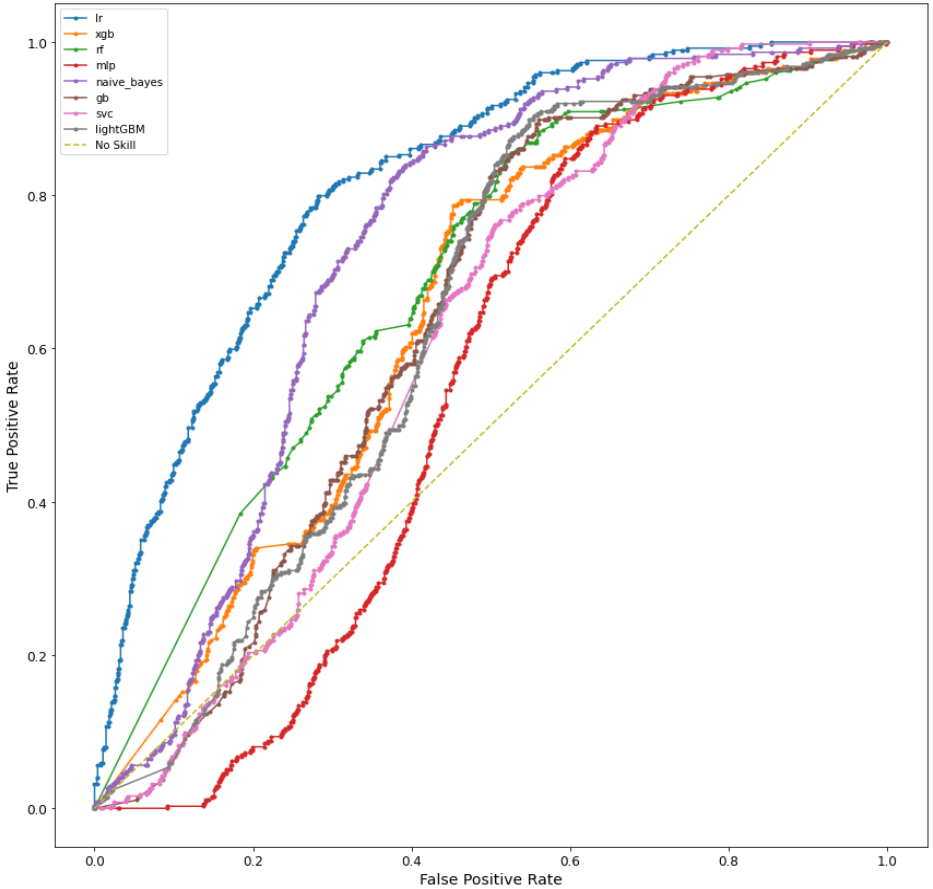
# Train 5 fold Average Undersample Results

| | Accuracy | Precision | Recall | F1Score |
|---|---|---|---|---|
| naive_bayes | 0.688853 | 0.623999 | 0.950787 | 0.753445 |
| mlp | 0.768445 | 0.831456 | 0.680014 | 0.742383 |
| gb | 0.816566 | 0.837837 | 0.787278 | 0.811205 |
| lr | 0.679086 | 0.638884 | 0.823440 | 0.719389 |
| svc | 0.612152 | 0.613959 | 0.606341 | 0.609531 |
| rf | 0.786895 | 0.799839 | 0.766285 | 0.782379 |
| xgb | 0.740957 | 0.672893 | 0.939225 | 0.783792 |
| lightGBM | 0.734437 | 0.661048 | 0.963828 | 0.784081 |

## OverSample




## UnderSample



# OverSample (No Tuning)

| | Accuracy | Precision | Recall | F1Score | True Negative | False Negative | True Positive | False Positive |
|---|---|---|---|---|---|---|---|---|
| **lightGBM** | 0.634684 | 0.414634 | 0.909091 | 0.569514 | 553 | 34 | 340 | 480 |
| **xgb** | 0.665245 | 0.435591 | 0.877005 | 0.582076 | 608 | 46 | 328 | 425 |
| **naive_bayes** | 0.648188 | 0.421326 | 0.866310 | 0.566929 | 588 | 50 | 324 | 445 |
| **lr** | 0.712864 | 0.476636 | 0.818182 | 0.602362 | 697 | 68 | 306 | 336 |
| **mlp** | 0.712864 | 0.476038 | 0.796791 | 0.596000 | 705 | 76 | 298 | 328 |
| **gb** | 0.742715 | 0.510490 | 0.780749 | 0.617336 | 753 | 82 | 292 | 280 |
| **rf** | 0.734186 | 0.500000 | 0.577540 | 0.535980 | 817 | 158 | 216 | 216 |
| **svc** | 0.589197 | 0.337061 | 0.564171 | 0.422000 | 618 | 163 | 211 | 415 |

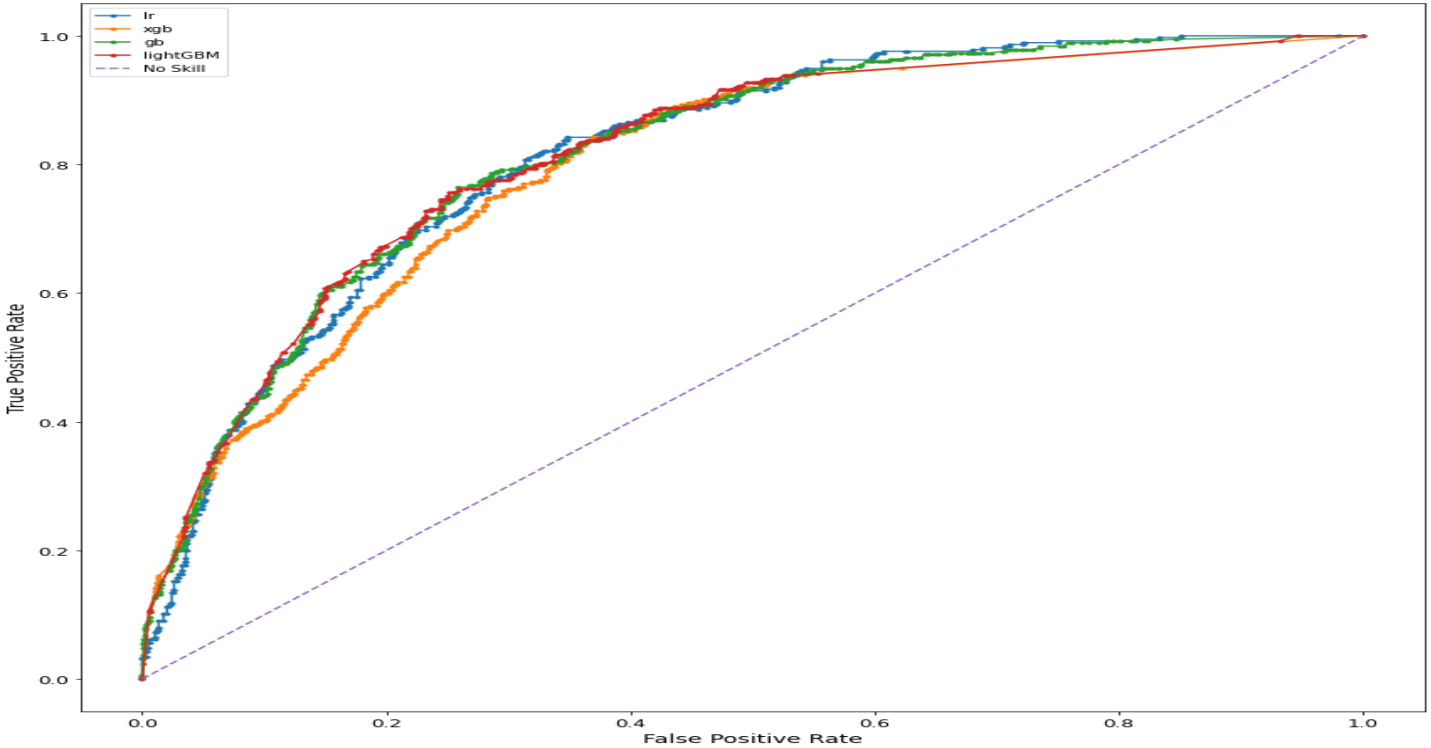
# UnderSample (No Tuning)

| | Accuracy | Precision | Recall | F1Score | True Negative | False Negative | True Positive | False Positive |
|---|---|---|---|---|---|---|---|---|
| **lightGBM** | 0.466951 | 0.323640 | 0.922460 | 0.479167 | 312 | 29 | 345 | 721 |
| **xgb** | 0.475480 | 0.326996 | 0.919786 | 0.482468 | 325 | 30 | 344 | 708 |
| **naive_bayes** | 0.647477 | 0.420779 | 0.866310 | 0.566434 | 587 | 50 | 324 | 446 |
| **lr** | 0.739161 | 0.505983 | 0.791444 | 0.617310 | 744 | 78 | 296 | 289 |
| **mlp** | 0.534471 | 0.333333 | 0.751337 | 0.461791 | 471 | 93 | 281 | 562 |
| **rf** | 0.604833 | 0.373961 | 0.721925 | 0.492701 | 581 | 104 | 270 | 452 |
| **gb** | 0.589197 | 0.362162 | 0.716578 | 0.481149 | 561 | 106 | 268 | 472 |
| **svc** | 0.586354 | 0.339506 | 0.588235 | 0.430528 | 605 | 154 | 220 | 428 |

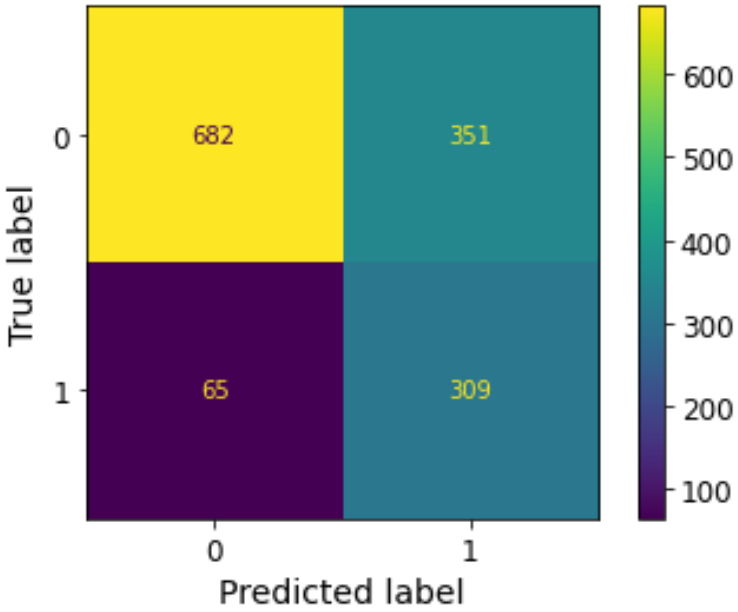
# OptunaTuned Models (F1 Emphasis)

| | Accuracy | Precision | Recall | F1Score | True Negative | False Negative | True Positive | False Positive |
|---|---|---|---|---|---|---|---|---|
| lr | 0.704335 | 0.468182 | 0.826203 | 0.597679 | 682 | 65 | 309 | 351 |
| xgb | 0.700782 | 0.462400 | 0.772727 | 0.578579 | 697 | 85 | 289 | 336 |
| gb | 0.736318 | 0.502582 | 0.780749 | 0.611518 | 744 | 82 | 292 | 289 |
| lightGBM | 0.621180 | 0.406139 | 0.919786 | 0.563473 | 530 | 30 | 344 | 503 |

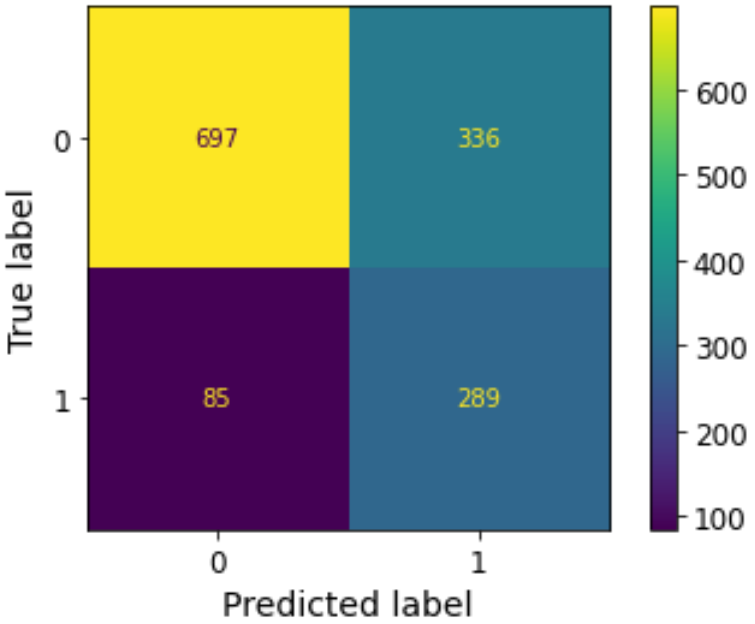
# ROC of Champion Models



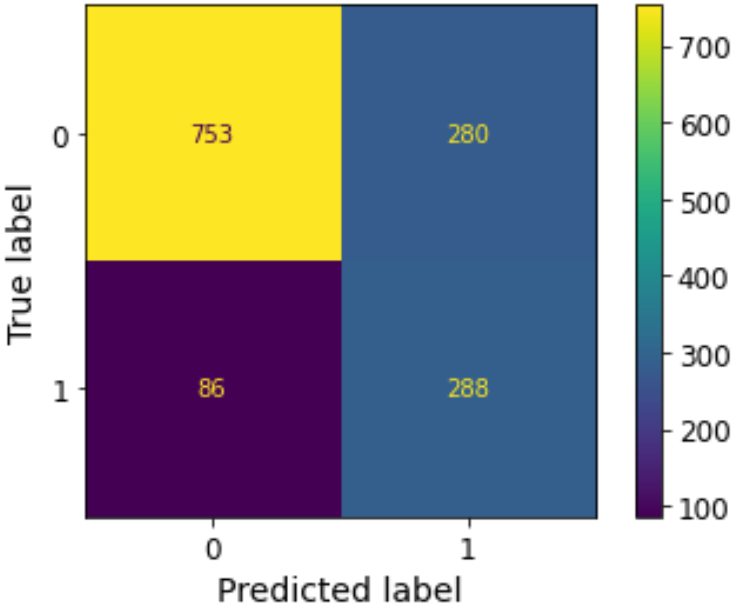
## Logistic Regression



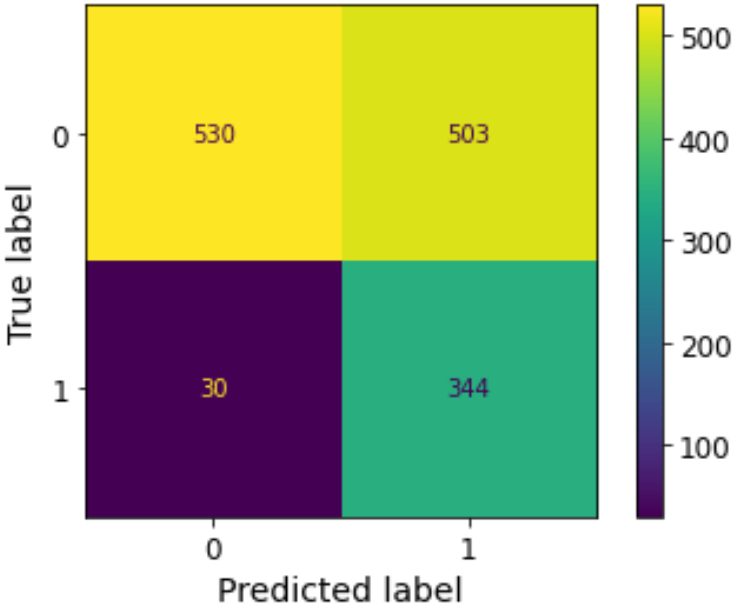

## XGBoost



## Gradient Boosting



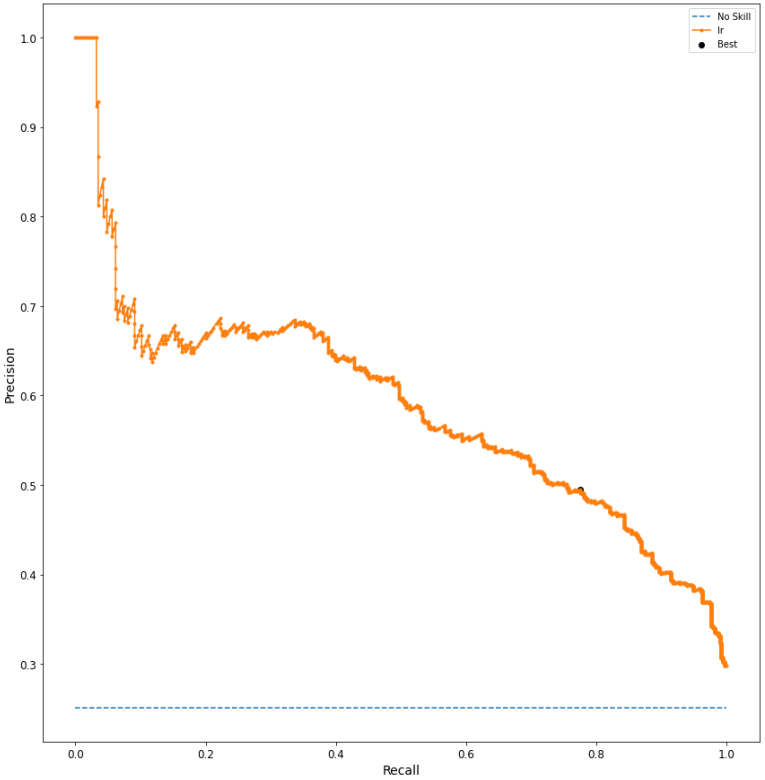

## Light GBM



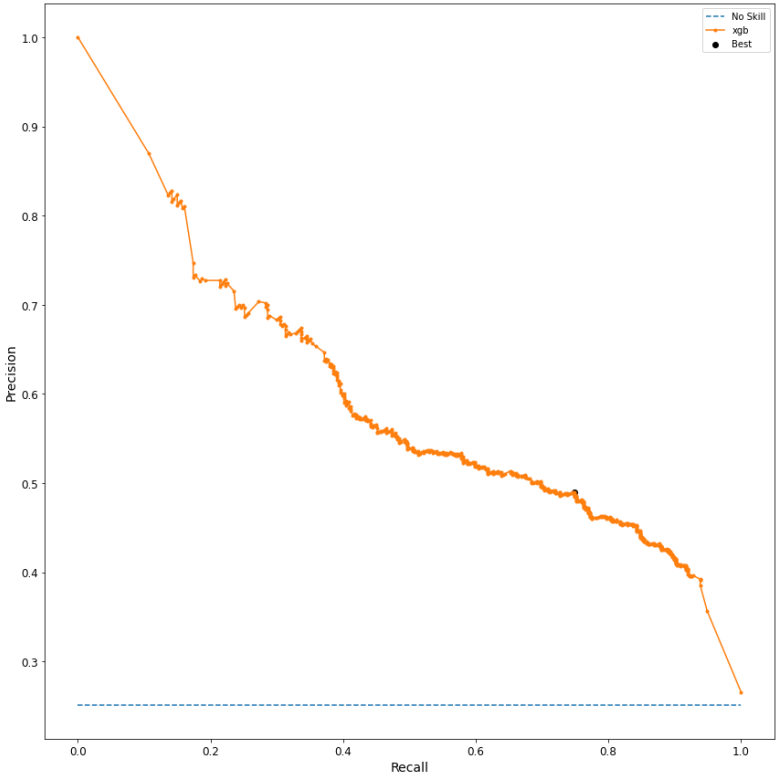
# Numerical Thresholds

| | Recall | Precision | Fscore | Threshold |
|---|---|---|---|---|
| **lr** | 0.775401 | 0.494881 | 0.604167 | 0.542832 |
| **xgb** | 0.748663 | 0.489510 | 0.591966 | 0.659565 |
| **gb** | 0.770053 | 0.514286 | 0.616702 | 0.500000 |
| **lightGBM** | 0.756684 | 0.521179 | 0.617230 | 0.870174 |

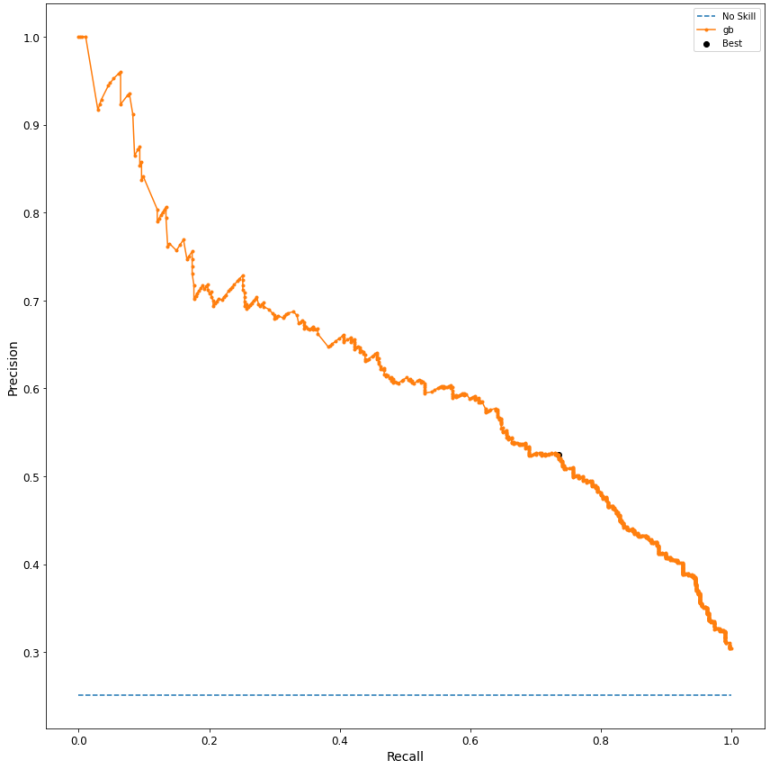
# Logistic Regression



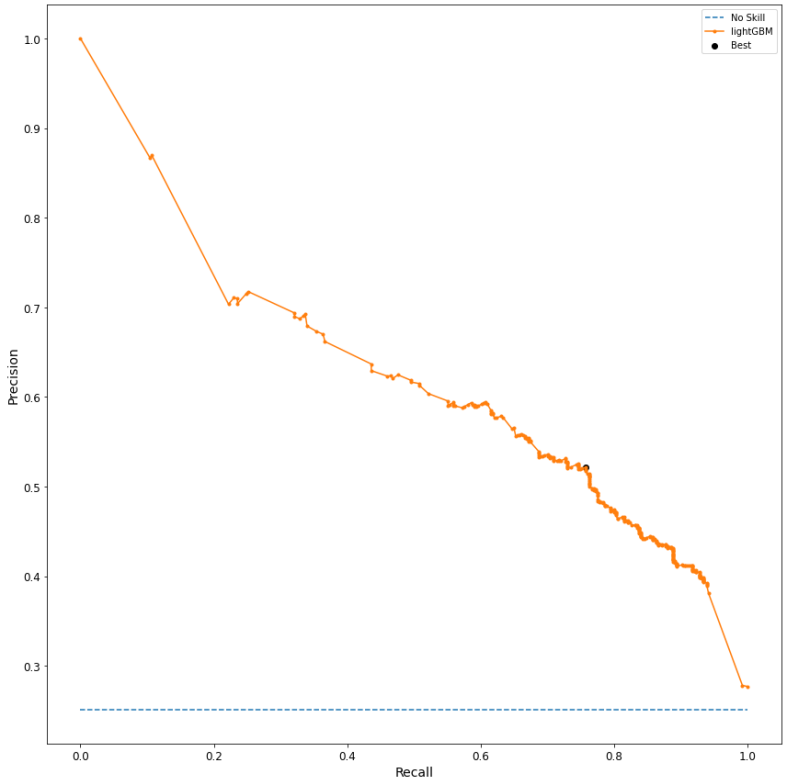

# XGBoost



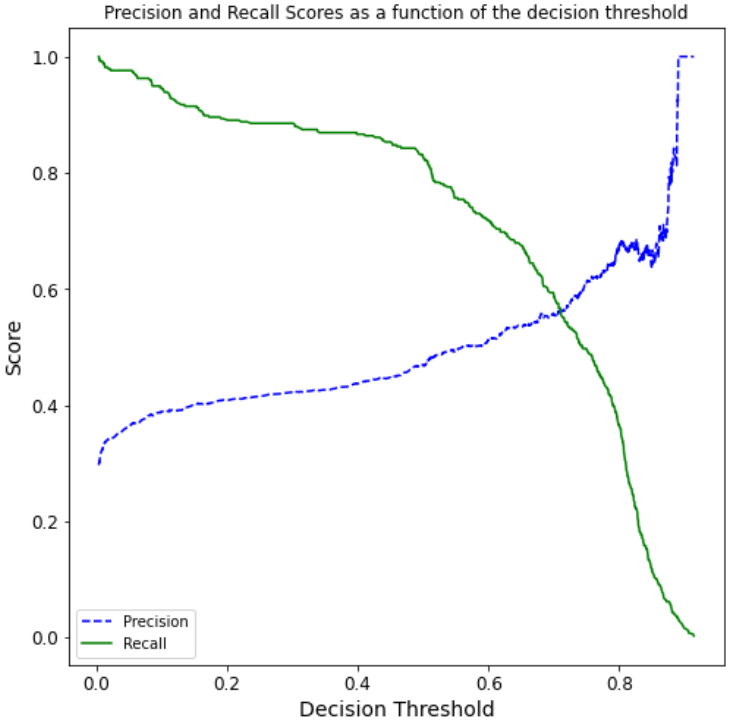
# Gradient Boosting



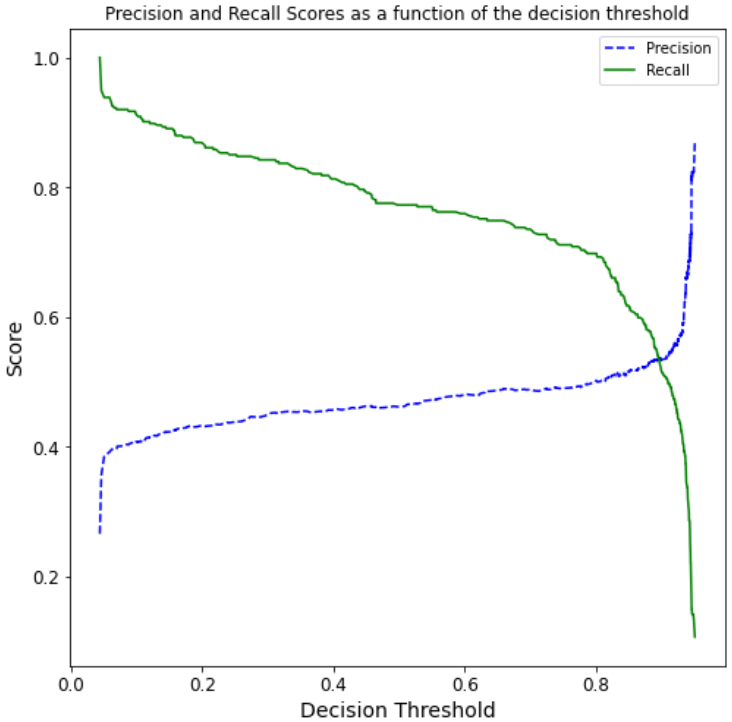

# Light GBM



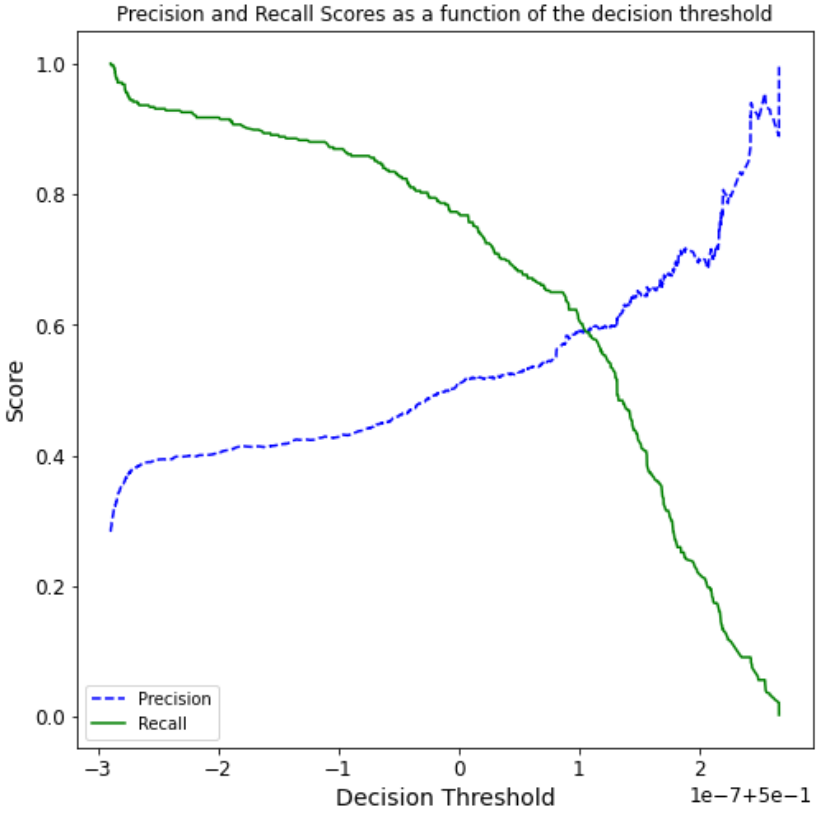
# Logistic Regression



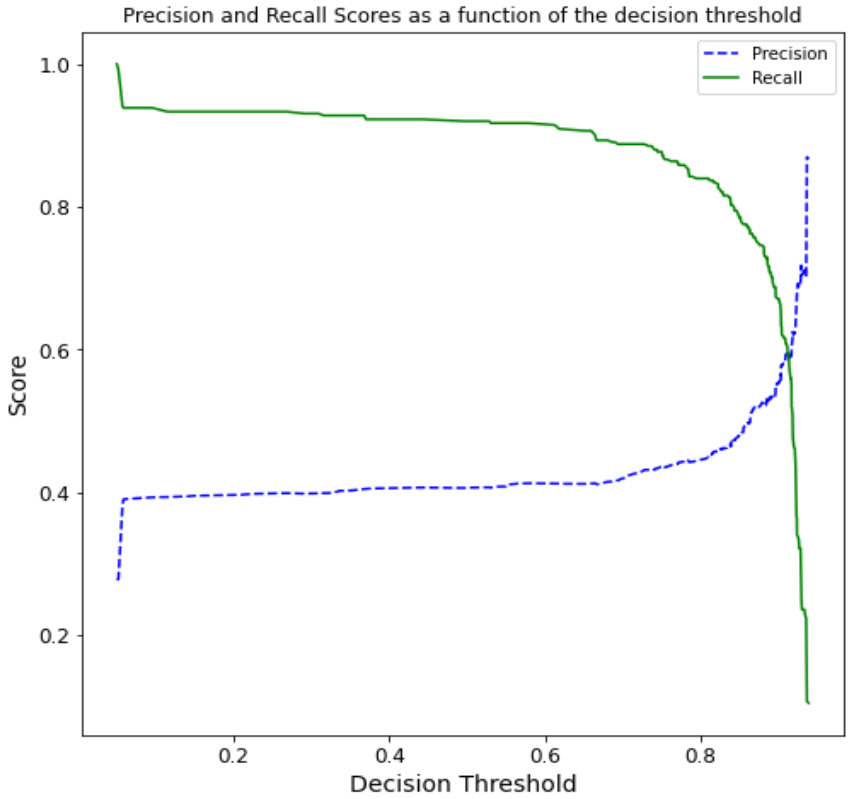

# XGBoost



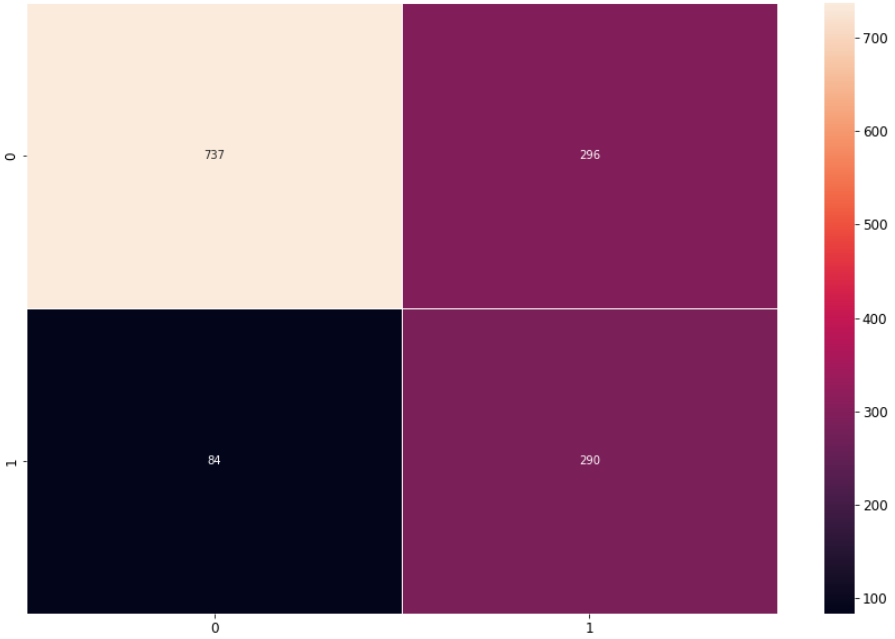
# Gradient Boosting



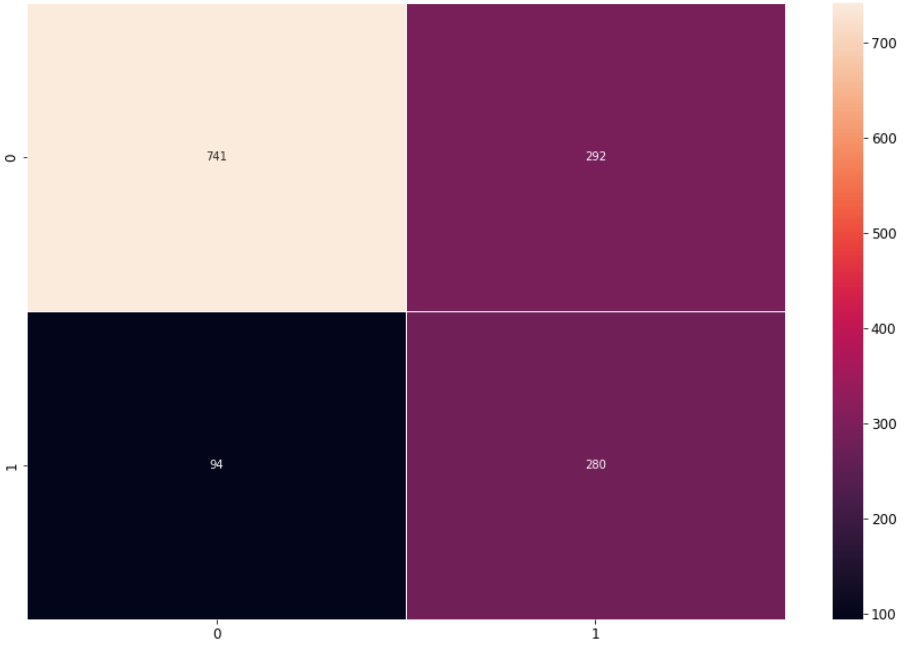

# Light GBM



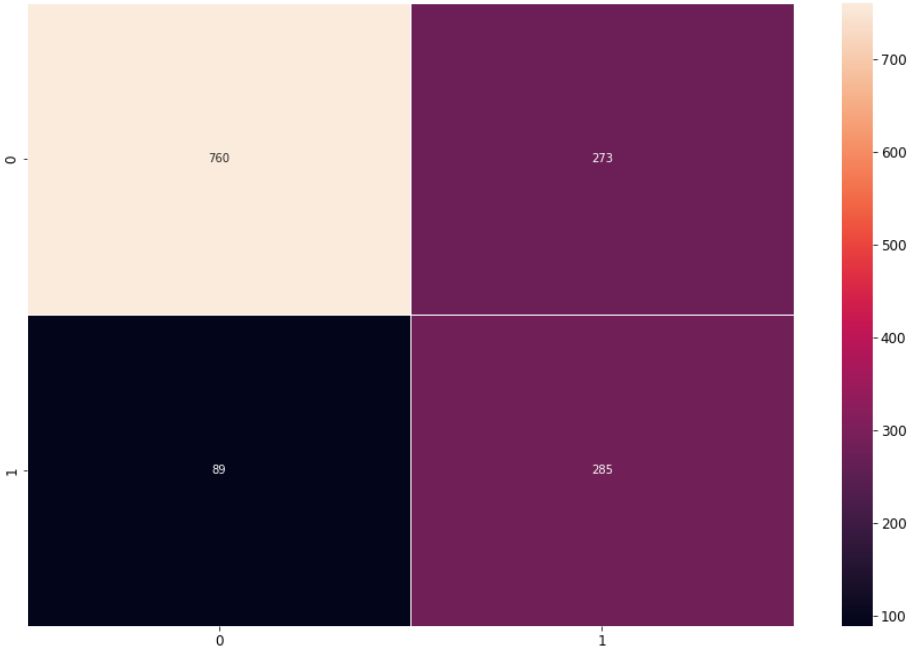
# Logistic Regression



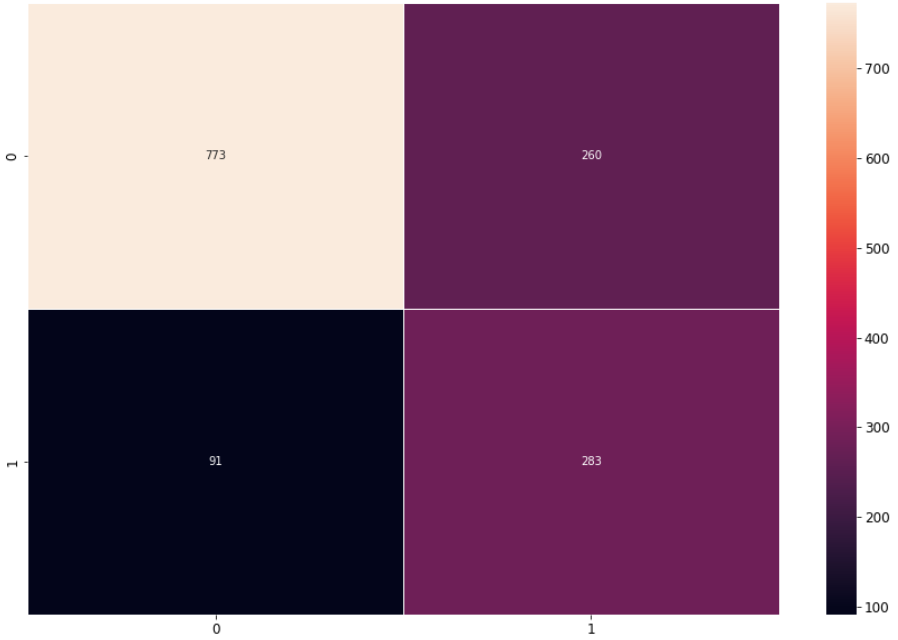

# XGBoost



## Gradient Boosting




## Light GBM



# False Positive Rate (False Alarm)

| Models | Base | Custom |
|---|---|---|
| Logistic Regression | 34% | 29% |
| XGB | 33% | 28% |
| Gradient Boosting | 27% | 26% |
| Light GBM | 49% | 25% |

# Optimization (Appendix)

## Assumptions

- If a customer is predicted to churn, it's monthly value will be expected as zero. (Otherwise it will be equal to the previous month's value.)
- If we send a 20% coupon to a customer predicted to churn, the customer will not churn.

## Decision Variables

$$Coupon_{i\,(i=1,\ldots,7043)}$$

Whether to send customer *i* a coupon or not.
Coupon *i* = 1 if Telco decides to send coupon to customer *i*, otherwise 0.

## Objective Function

$$Minimize \sum_{i=1}^{n} [Coupon_i * (MonthlyCharges_i - Mailingcost_i - Couponamount_i)] + \sum_{i=1}^{n} [(1 - Coupon_i) * (PredChurn_i - MonthlyCharges_i)]$$

## Result

- We can get the **target list of customers to send coupon** (customer *i* whose coupon *i* = 1)
- Based on the ground truth ('Churn' value), we can calculate the monetary value of coupon promotion, calculating **'Expected loss w/o promotion - Realized loss w/ promotion'**

# Optimization

**Key motivation:**

If Telco can predict the customer's churn before they leave, it can send coupons to prevent churn, eventually **achieving minimized loss**.

**Technical tool:**

With Gurobi, based on predicted churn, **we identified target customers** to send coupons and **calculate the monetary upside of the customer-retained condition** with each predictive model.

```
[('custom_XGB', 34355.45181236676),
 ('base_XGB', 33576.85181236675),
 ('custom_Light GBM', 33565.20181236675),
 ('custom_Gradient Boosting', 33033.151812366756),
 ('base_Gradient Boosting', 32844.80181236676),
 ('custom_Logistic Regression', 31798.401812366745),
 ('base_Logistic Regression', 31433.101812366745),
 ('base_Light GBM', 24032.70181236674)]
```

With prediction generated by 'Custom_XGB', we expect to see **the most monetary upside.**

The strategy brings financial upside that is more than 24K with every predictive model.

# Risk and Next step

## Threats to Validity

- **Uncertainties remain;** Change in environmental factors or Data integrity issues may exist
- **The assumption for the optimization model may become a threat to validity** –
  We can't guarantee that customers will be retained only by coupons.

## Next step for Predictive model:

- To further optimize our models, we can use **custom loss functions**.
- Potentially **apply the SPO+ loss** by Adam Elmachtoub and Paul Grigas.
- We can utilize **Optuna to optimize multiple scores** concurrently.

## Next step for Optimization model:

- We used constant coupon rate here, but we can **incorporate 'optimal coupon value' as a decision variable** as the next step.
- To guarantee coupon promotion's efficiency, we can conduct an **additional A/B test to calculate the causal effect of coupons on retention rate.**

# Thanks!

Questions?