Predicting Commercial Bank Customer Retention (Churn) Rate

APSTA-GE 2401: Statistical Consulting Seminar, Final Presentation

Tong Jin, Andy Tan, Zixuan Zhou

New York University

12/07/2020

Topics

- Introduction
- Data Prerocessing
- Data Processing
- Models

Introduction

Introduction

This project applies machine learning techniques to predict whether or not customers of a commercial bank will stay with the bank (continue with their products and services). As we enter a data-driven era, banks need a better understanding of customer demands as they expanding their business models. Specifically, banks are interested in learning about customer's churn rate and the change of their financial interests. The purpose of predicting customer's retention probability is to accommodate customer demands and to achieve precision marketing in order to retain revenues.

This project has two affiliations:

It is the final project of the APSTA-GE 2401： Statistical Consulting Seminar course at New York University.

It is the team project of an active competition on Data Castle: 2020 Financial Modeling Competition by Xiamen International Bank.

The Competition

With the development of finance and data science, banks established many contact marketing strategies, both online and off-line, in order to satisfy various customer demands, covering both regular business products and special channel trading services. Facing numerous requests, banks need to better understand customer's preferences on service selection. For daily business, they need to detect potential triggers that cause customers to leave. They also need to predict whether or not a customer will stay, given his/her financial situations. Through prediction and detection, banks can proactively provide marketing incentives to targeted customers who are likely to not retain. In this way, banks can achieve precision marketing and maximize revenues.

This competition requires competitors to build a prediciton model based on real-world customer data. It also asks competitors to provide feasible business solutions based on their model results.

Tasks

As we embracing the era of digital finance, many financial institutions adapt themselves to a data-based business mode by applying various data mining and machine learning techniques to their services. The Xiamen International Bank ("The Bank") has been heavily invested in data-driven financial services in recent years. As the bank expanding its business, it requires a better understanding of customer demands, especially revenue-related preference. Coordinated with Data Castle, the bank launched this competition, aiming to invite data scientists and statisticians to provide solutions regarding a real-world service problem: predicting customer retentions.

The task of this competition is to design and implement a supervised machine learning algorithm to predict whether or not customers will continue their businesses with the bank in the near future. Specifically, the bank is interested in learning about customer's churn rate and the probability of changing their financial status. The purpose of predicting customer's retention probability is for precision marketing that prevents the bank from losing revenues.

To solve this problem, our team accomplished the following tasks:

1. Select and build a model combination that predicts retention probability based on the dataset.

2. Based on model results, present a professional business proposal for precision marketing.

Research Design

The strategy of supervised learning is to train models using the `X_train` data and validate model performance using the `y_train` data. After training, we fit the model to the `X_test` data. The model will then generate predictions, `y_test`, based on `X_test`.

To increase model performance, we splited the train set into two sets: 80% of the train data goes to the `X_train` set and 20% of the data goes to the `X_test` set. Then, we conducted a 5-fold cross validation and selected the best performed model output. We also find tuned hyperparameters using randomized search.

Raw Data

There are three main data packages:

- `x_train`: the train data package containing all features.
- `y_train`: the data package for feature test.
- `x_test`: the train data package for prediction. It contains the same features as `x_train`.

Details – Features

There are 5 feature categories marked with different initials:

- `X`: end-of-month balance, `ncol_X = 8`
  - structural, checking, savings, CDs, funds, loans, ...
- `B`: customer behavior, `ncol_B = 7`
  - transfer (mobile, branch), transaction data, frequency, ...
- `E`: big events, `ncol_E = 18`
  - new account opening, mobile app activation, large transfer, ...
- `C`: savings, `ncol_C = 2`
  - number of saving products and amounts
- `I`: customer information, `ncol_I = 20`
  - age, sex, level, occupation, income, indicators (marital, mobile app, mobile pay) ...

Total number of features: 55

Codebook, customer's asset

Variable Name	Description
cust_no	custumer's ID (primary key)
X1	structured deposit balance
X2	time deposit balance
X3	demand deposit balance
X4	financial products balance
X5	fund balance
X6	asset management balance
X7	loan balance
X8	large deposit certificate balance

Codebook, customers' behaviors

Variable Name	Description
cust_no	custumer's ID (primary key)
B1	mobile banking login times
B2	transfer-in times
B3	transfer-in money amount
B4	transfer-out times
B5	transfer-out money amount
B6	latest transfer time
B7	number of transfers in a season

Codebook, important behaviors

Variable Name	Description
cust_no	custumer's ID (primary key)
E1	account opening date
E2	online banking opening date
E3	mobile banking opening date
E4	first online banking login date
E5	first mobile banking login date
E6	first demand deposit date
E7	first time deposit date
E8	first loan date
E9	first overdue date
E10	first cash transaction date
E11	first bank-securities transfer date
E12	first transfer at counter date
E13	first transfer via online banking date

Codebook, savings

Variable Name	Description
cust_no	custumer's ID (primary key)
C1	deposit products value
C2	number of deposit products

Codebook, valid customer IDs

This set contains valid customer IDs in the season Z.

Variable Name	Description
cust_no	custumer's ID (primary key)

Codebook, customer information

Variable Name	Description
cust_no	custumer's ID (primary key)
l1	gender
l2	age
l3	class
l4	tag
l5	occupation
l6	deposit customer tag
l7	number of products owning
l8	constellation
l9	contribution
l10	education level
l11	family annual income
l12	field description
l13	marriage description

Data Prerocessing

Preprocessing

We started from examining the `y_train` because it contains labels that can validate our model predictions. `y_train` contains random sampled label data from two quarters: Q3_2020 and Q4_2020. Since the customer ID column, `cust_no`, only contains unique values, we determined our data processing strategies as follows:

y_train

1. Use **quarter** to separate data processing procedures. We created two training sets, `X_train_Q3` and `X_train_Q4`, and merged them before applying models. In this way, we bypassed duplicated customer IDs in both `y_train` sets caused by random sampling. This allowed us to maximize the number of labels that can be validated.

    - `y_Q3_3` contains 69126 rows, `y_Q3_3` contains 76170 rows.
    - `y_train` has 62397 duplicated customer IDs.
    - `y_train` has 40090 completely identical records (same customer ID, same label).
    - Two samples are heavily overlapped.
    - 22307 customers changed their churn preference from Q3 to Q4.

1. Based on quarterly-separated `y_train` set, we merged `X_train` raw data accordingly. For each quarter, we dropped duplicated customer IDs except for the last occurrance.

2. During data preprocessing, we examined records in the `cust_avli` column of the `X_train` sets. These sets contain the ID of all effective customers. We confirmed that these ID are the same as those in the `y_train` set. Therefore, we trimmed the dataset based on the `cust_no` column in the `cust_avli`, separated by quarters.

    - Confirmed that `cust_avli` is the key indexing column.

3. Merged and Trimmed datasets.

X_train

We trimmed features based on customer IDs in the validation set, `y_train`. For example:

After dropping duplicated customer IDs, aum_Q3 has 493441 rows and 9 columns.

After dropping duplicated customer IDs, aum_Q4 has 543823 rows and 9 columns.

After trimming, aum_Q3 has 69126 rows and 9 columns.

After trimming, aum_Q4 has 76170 rows and 9 columns.

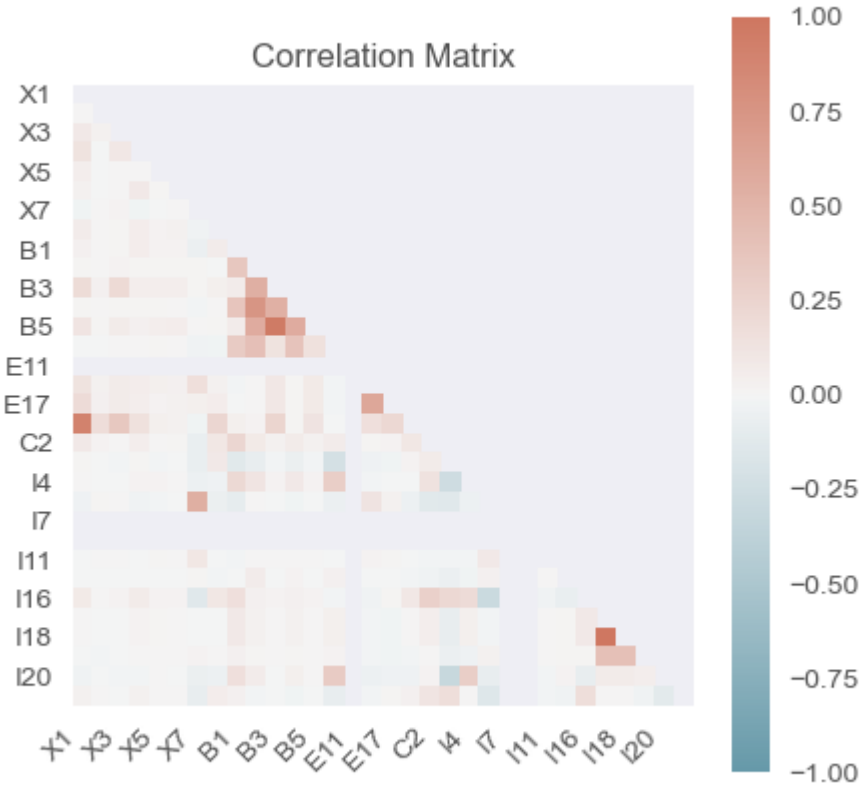
Trimmed Data

After preprocessing, we have three sets:

1. `X`: contains 145296 rows and 55 features.
2. `y`: contains 145296 rows and 1 label column.
3. `X_true`: hold-out set, contains 76722 rows and 55 features.

Data Processing

Correlation Matrix



We confirmed that the `label` column does not have strong correlations with other features.

Intersting to see customer's constellation is highly correlated with their contributions.

Customer's transfer times and amounts are also highly correlated.

Dealing with missing values

We first inspected the feature set.

1. There are 55 features in the feature set.

2. We checked the correlation among features and the label.

3. We checked if there are any missing values in the set. We found multiple columns that contain missing values, ranging from 0.005% to 100%. For columns containing a large portion of missing values, we dropped the column to reduce computational burden. For columns containing a small portion of missing values, we applied a deep learning library, Datawig, which learns machine learning models using deep neural networks to impute missing values in the data.

    - After dropping columns containing large portion of missing values, we reduced number of features to 45.
    - After imputing missing values using deep learning, we managed to keep all columns with small portion of missing values. This allows maximum information to be retained.

Models

Model Selections

We selected the following models:

- Logistic Regression with Elastic Net (as baseline)
- Random Forest (bagging)
- Gradient Boosting Machine (boosting)
- Multilayer Perceptron (neural network)

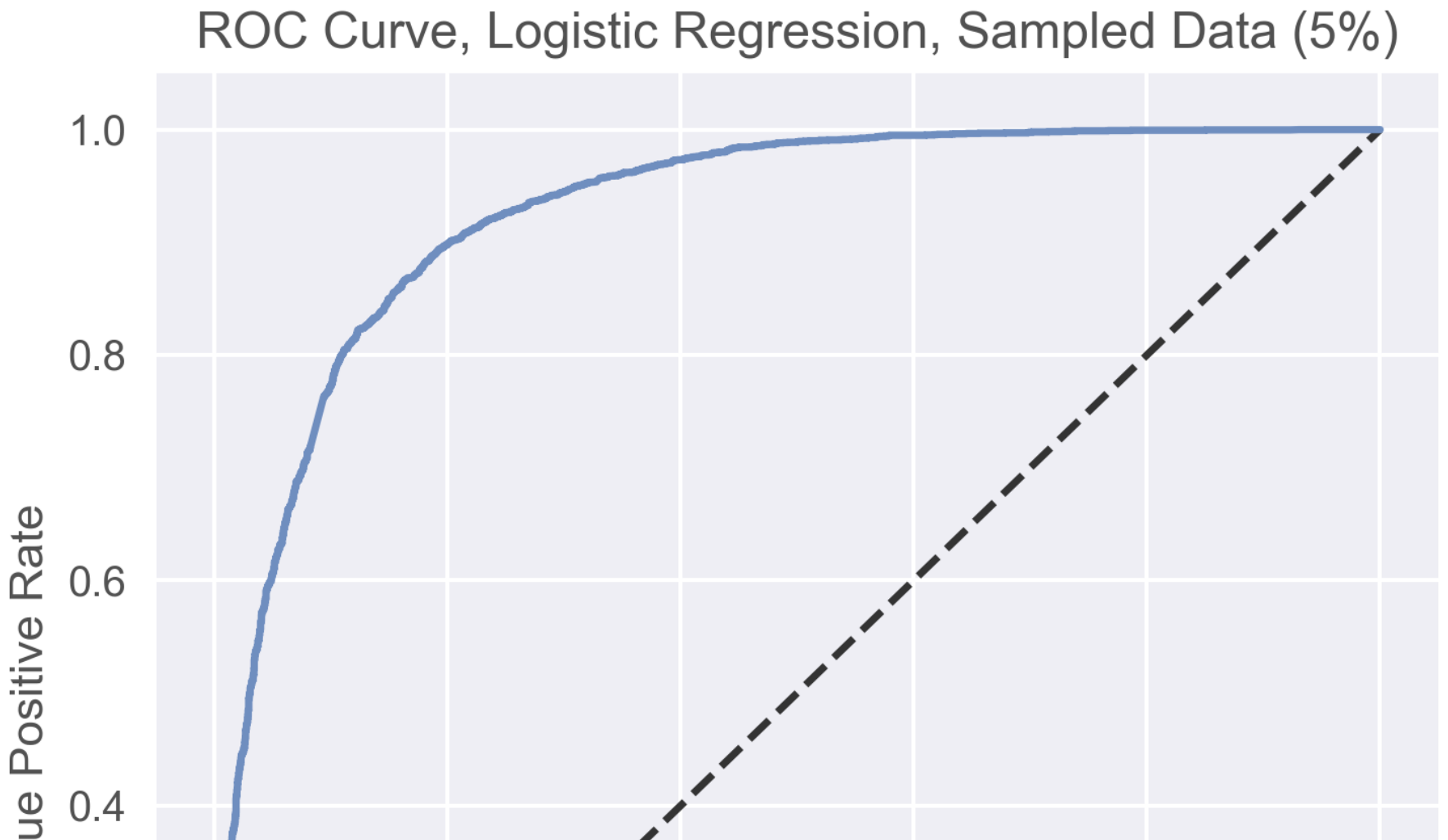
We conducted three feature reduction processes:

- Fit model with original data (completed)
- Fit model with feature selected data using SVD (in-progress)
- Fit model with the data containing features that contribute to 80% of information (in-progress)

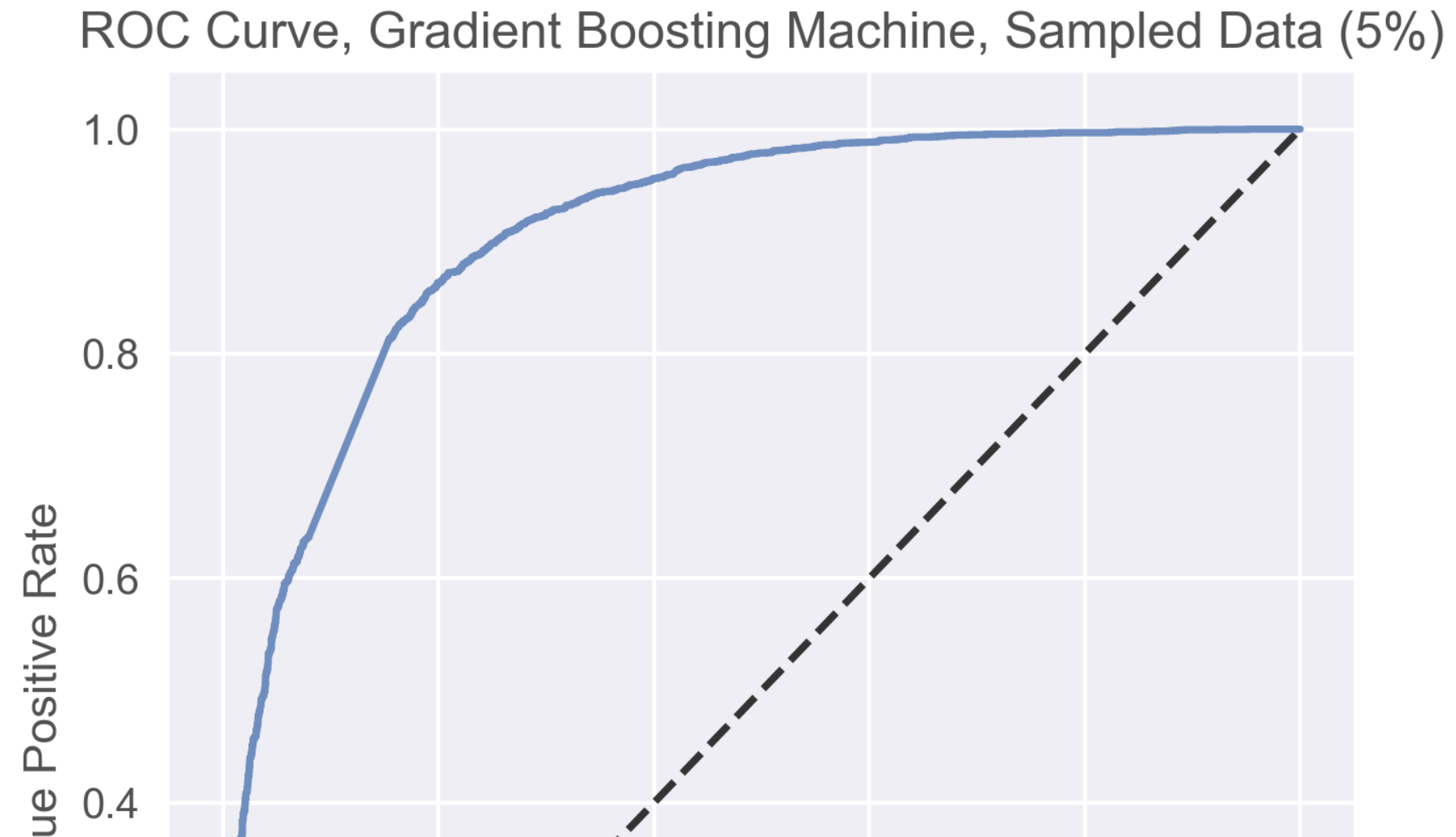
Strategies

- 2:8 train test split
- Grid search (randomized search) for hyperparamter tuning
- 5-fold cross validation
- SVD for feature reduction

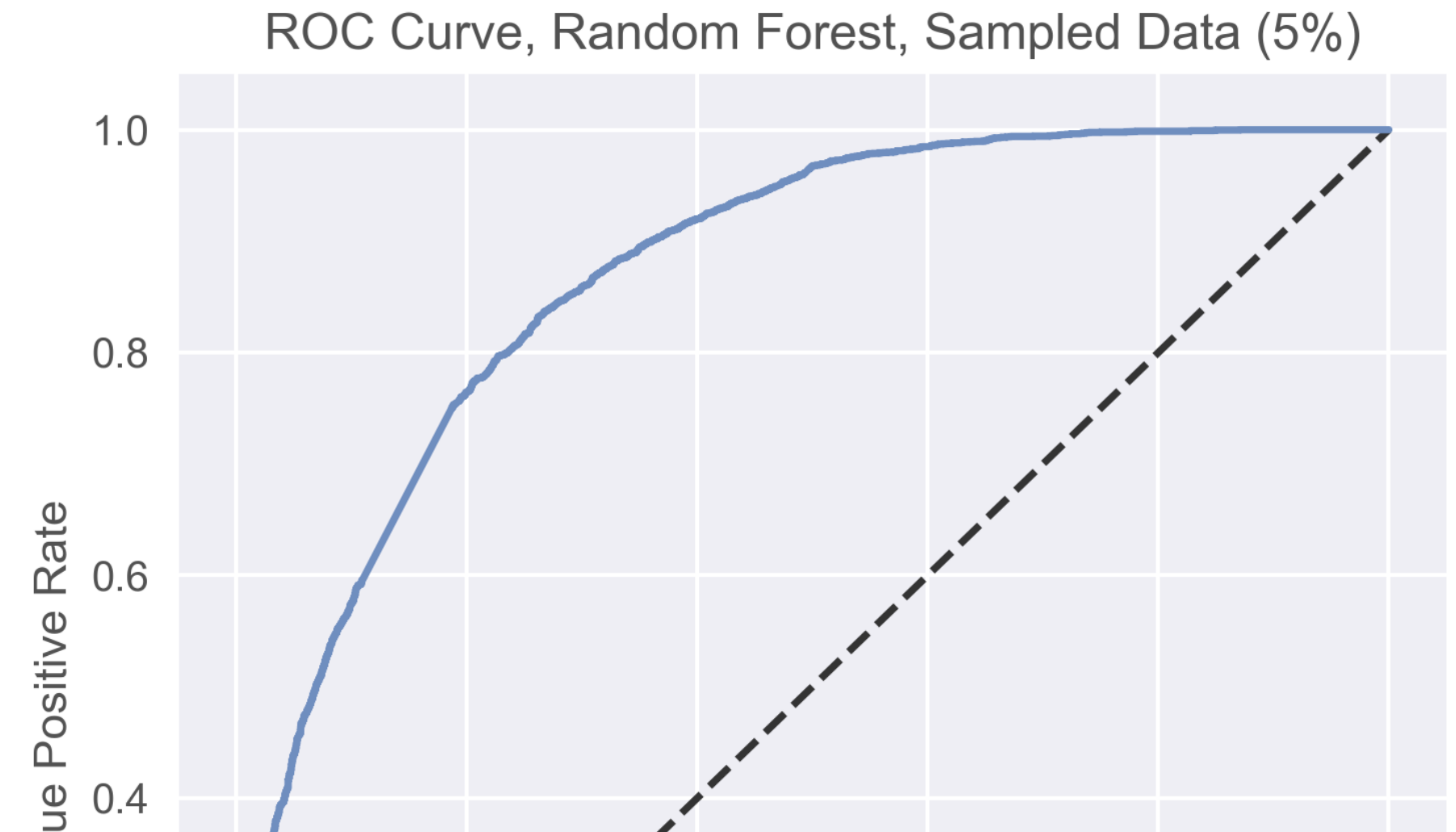
Log Reg


ROC Curve, Logistic Regression, Sampled Data (5%)
1.0
0.8
0.6
0.4
ue Positive Rate

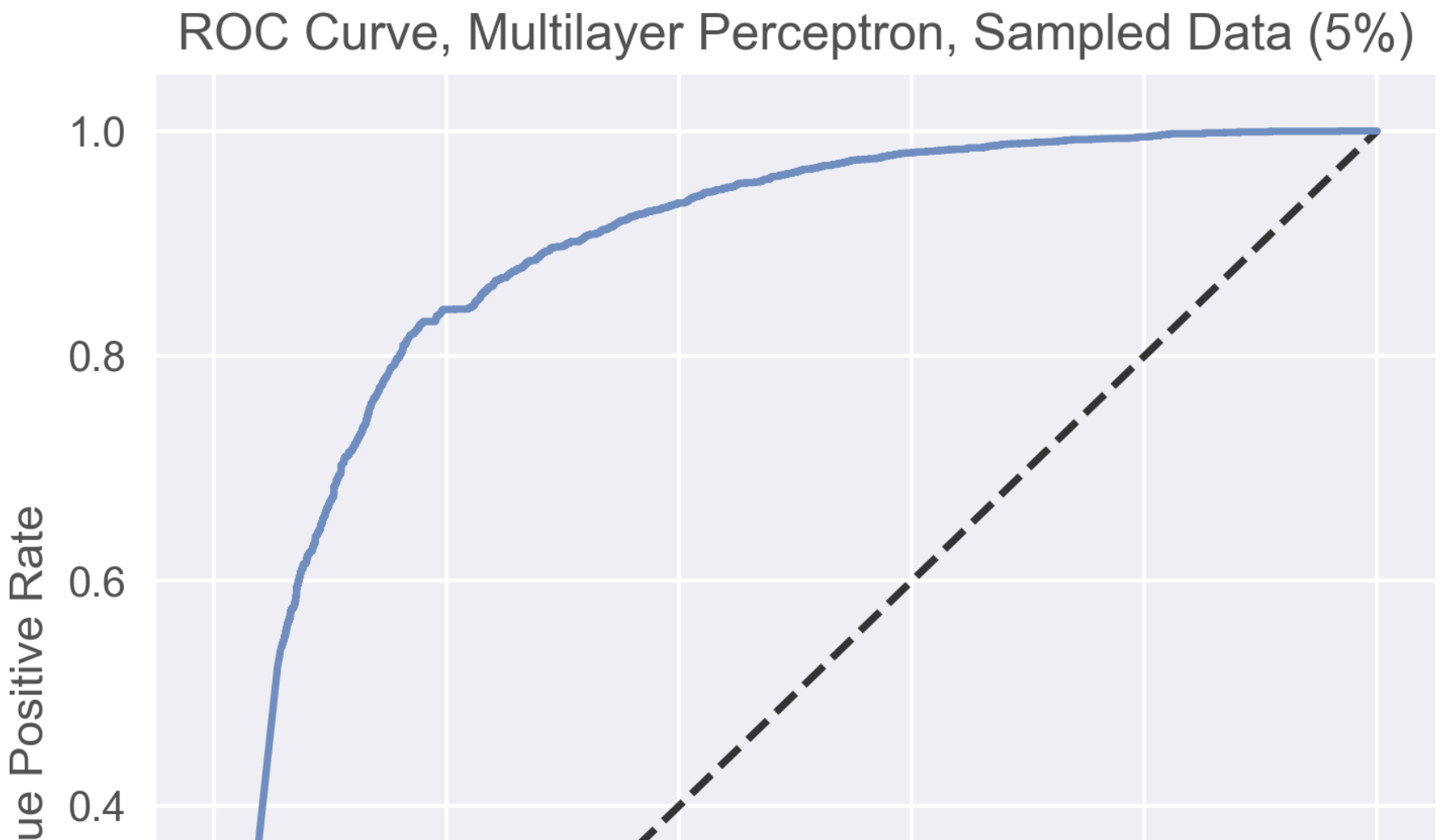
GBM


ROC Curve, Gradient Boosting Machine, Sampled Data (5%)
1.0
0.8
0.6
0.4
ue Positive Rate

RF


ROC Curve, Random Forest, Sampled Data (5%)
1.0
0.8
0.6
0.4
ue Positive Rate

MLP


ROC Curve, Multilayer Perceptron, Sampled Data (5%)
1.0
0.8
0.6
0.4
ue Positive Rate

To-Do

- Fit model with feature selected data using SVD (in-progress)
- Fit model with the data containing features that contribute to 80% of information (in-progress)
- Data mining (EDA) to get more insights.

Contact

Team

Team Name: A3SR

Team Members:

- Tong Jin, NYU
- Zheng Tan, NYU
- Zixuan Zhou, NYU