Data Glacier

Your Deep Learning Partner

Final Presentation

Bank Marketing Campaign

30.08.2022

Efe KARASIL-Sefa SÖZER

Agenda

Executive Summary

Data Understanding

Data Transformation

Model Building

Applying Models

Final Model Selection

Executive Summary

- **Problem Description:** ABC Bank wants to sell its term deposit product to customers and before launching the product they want to develop a model which helps them in understanding whether a particular customer will buy their product or not (based on customer's past interaction with bank or other Financial Institution).

- **Problem Statement:** ABC Bank wants to sell it's term deposit product to customers and before launching the product they want to develop a model which help them in understanding whether a particular customer will buy their product or not (based on customer's past interaction with bank or other Financial Institution).

Data Understanding

Data Set Information:

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

There are four datasets:

1) **bank-additional-full.csv** with all examples (41188) and 20 inputs, ordered by date (from May 2008 to November 2010), very close to the data analyzed in [Moro et al., 2014]

2) bank-additional.csv with 10% of the examples (4119), randomly selected from 1), and 20 inputs.

3) bank-full.csv with all examples and 17 inputs, ordered by date (older version of this dataset with less inputs).

4) bank.csv with 10% of the examples and 17 inputs, randomly selected from 3 (older version of this dataset with less inputs).

The smallest datasets are provided to test more computationally demanding machine learning algorithms (e.g., SVM).

The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).

Data Understanding

- **Attribute Information:**
- Input variables:
- bank client data:
- 1 - age (numeric)
- 2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
- 3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
- 4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
- 5 - default: has credit in default? (categorical: 'no','yes','unknown')
- 6 - housing: has housing loan? (categorical: 'no','yes','unknown')
- 7 - loan: has personal loan? (categorical: 'no','yes','unknown')
- # related with the last contact of the current campaign:
- 8 - contact: contact communication type (categorical: 'cellular','telephone')
- 9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
- 10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
- 11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

Data Understanding

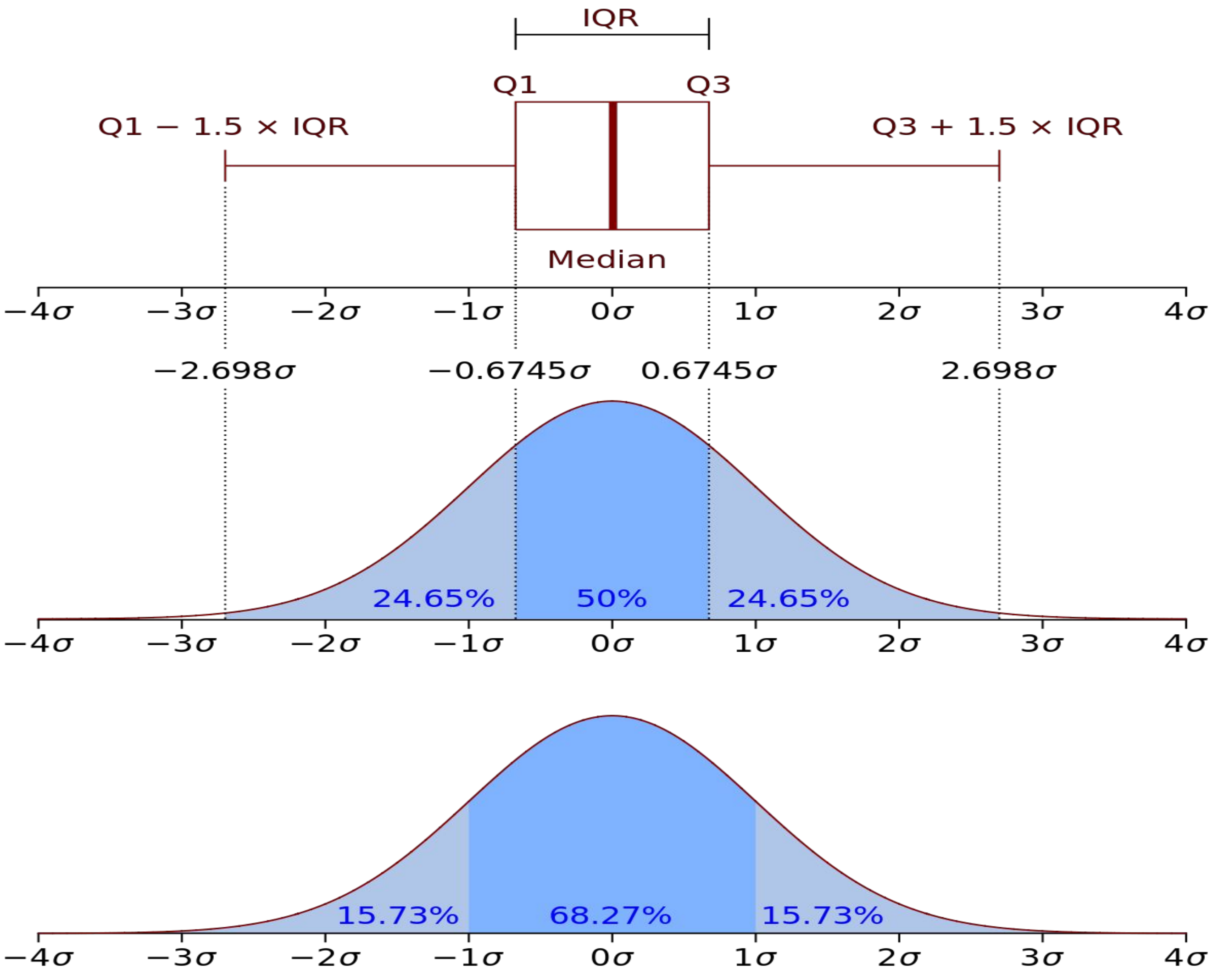
other attributes:

- 12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
- 13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
- 14 - previous: number of contacts performed before this campaign and for this client (numeric)
- 15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
- # social and economic context attributes
- 16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
- 17 - cons.price.idx: consumer price index - monthly indicator (numeric)
- 18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
- 19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
- 20 - nr.employed: number of employees - quarterly indicator (numeric)
-
- Output variable (desired target):
- 21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

Data Understanding

N°	Feature name	Description	Type
1	age	age	numeric
2	job	type of job	categorical
3	marital	marital status	categorical
4	education	level of education	categorical
5	default	has credit in default?	categorical
6	housing	has housing loan?	categorical
7	loan	has personal loan?	categorical
8	contact	contact communication type	categorical
9	month	last contact month of year	categorical
10	day of week	last contact day of the week	categorical
11	duration	last contact duration, in seconds	numeric
12	campaign	number of contacts performed in this campaign	numeric
13	pdays	number of days passed by after the last contact	numeric
14	previous	number of contacts performed for this client	numeric
15	poutcome	outcome of the previous marketing campaign	categorical
16	emp.var.rate	employment variation rate	numeric
17	cons.price.idx	consumer price index - monthly indicator	numeric
18	cons.conf.idx	consumer confidence index - monthly indicator	numeric
19	euribor3m	euribor 3 month rate - daily indicator	numeric
20	nr.employed	number of employees - quarterly indicator	numeric
21	y	has the client subscribed a term deposit?	binary

Data Transformation



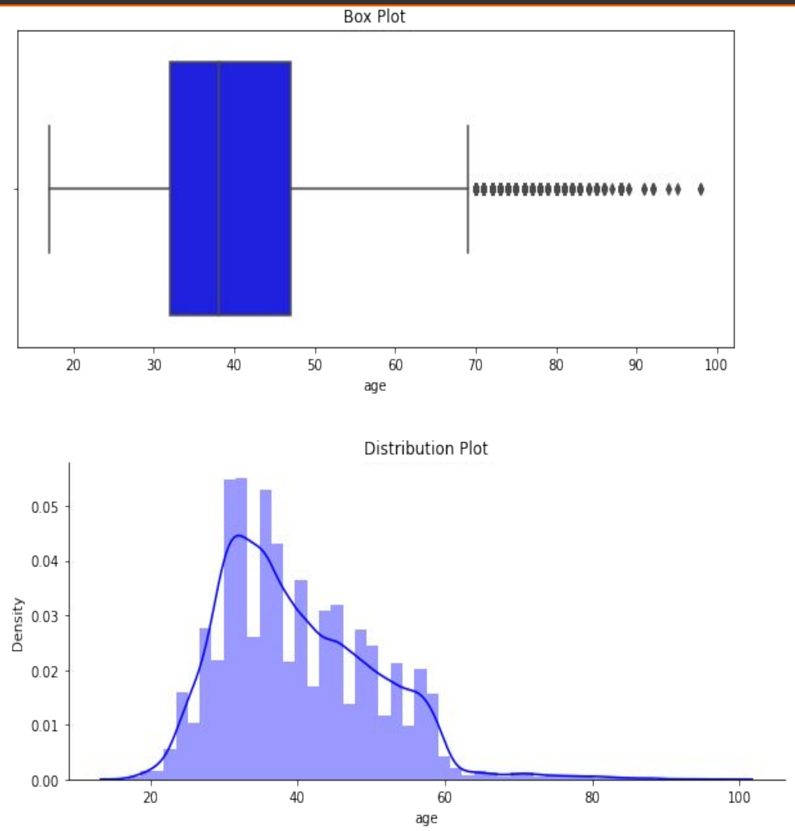
Outlier Detection and Handling

IQR(The interquartile range) method was used for outlier detection and handling.

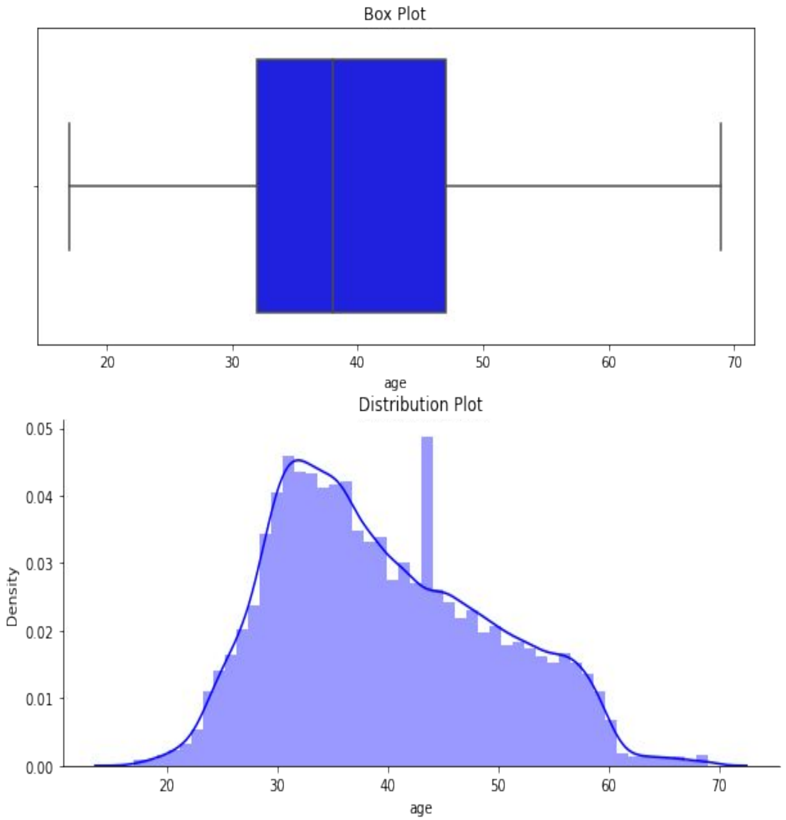
The interquartile range method defines outliers as values larger than Q3 + 1.5 * IQR or the values smaller than Q1 – 1.5 * IQR.

Outlier detection and handling was used for numerical values.

Data Transformation



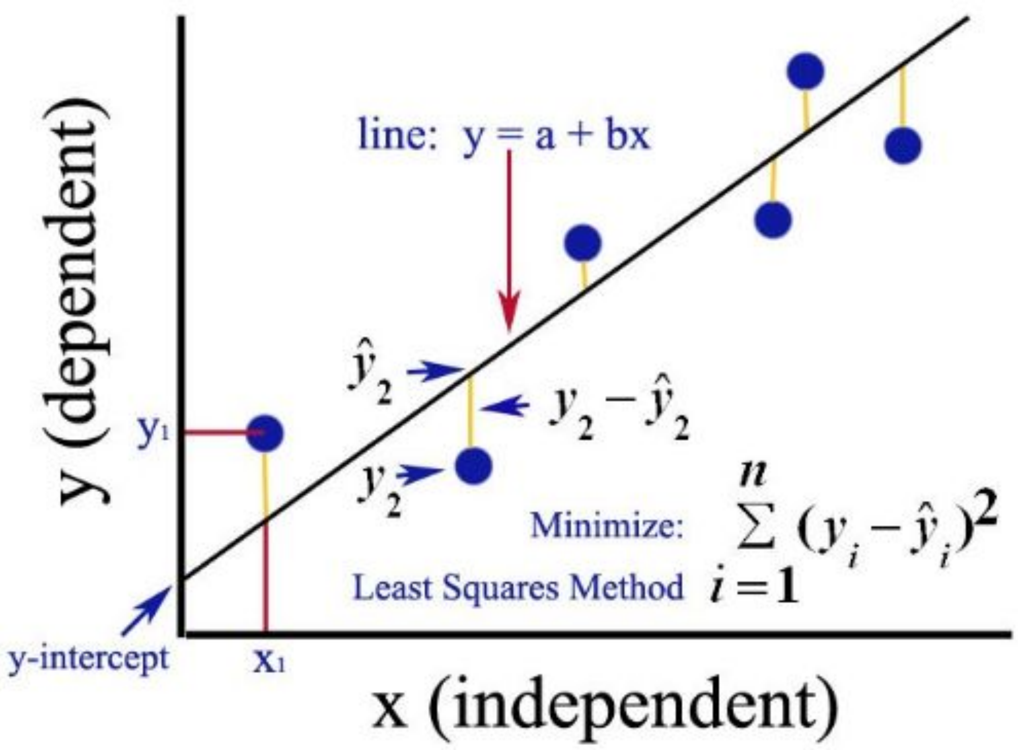
Original- Before deleting outliers



After deleting outliers

One of the examples can be seen in this slide, “Age” column-feature.

Data Transformation



The ordinary least squares (OLS) method is a linear regression technique that is used to estimate the unknown parameters in a model. The method relies on minimizing the sum of squared residuals between the actual and predicted values.

This method is used for feature importance and deleting the columns-features from the dataset for better results in ML models.

At the end of this method, 'Age', 'Job', 'Default', 'Housing' and 'Campaign' columns-features have been dropped.

Data Transformation

- **Getting rid of outliers with IQR method from numerical features**
- **Leaving ‘unknown’ values as they are in the categorical features to be processed as another category for their specific column-feature**
- **Using encoder to make categorical values numerical to prepare for machine learning models**
- **Using the ordinary least squares (OLS) method for feature selection, elimination**
- **Dropping “Duration” feature which should not be known beforehand to predict desired target “y” feature**

Model Building

- It was thought to be that classification models provide better results, since our desired output column for the machine learning models is a categorical value.

- Therefore, we used classification models.

- First, we made the learning data 75% the test data 25%. Then we tried the following models on these data.

The following algorithms selected for this classification problem include:

- **Linear Algorithms:**

Logistic Regression (LR) (Base Model)
Linear Discriminant Analysis (LDA).

- **Nonlinear Algorithms:**

Classification and Regression Trees (CART)
Gaussian Naive Bayes (NB)
k-Nearest Neighbors (KNN)

- **Ensemble Methods:**

*Boosting Methods:
AdaBoost (AB)
Gradient Boosting (GBM)
XGBClassifier(XGB)

*Bagging Methods:
Random Forests (RF)
Extra Trees (ET)

Model Building

- After a nice preprocessing of the data, we embed it into machine learning models. We have tried all branches of machine learning and observed that ensemble models give better results than Linear and Non-Linear models. Among the Ensemble models, the XGBoost, Adaboost and Gradient Boosting models are very close in accuracy metrics.

Here are the results of the machine learning models we used:

These results are obtained with k-fold(10) validation of **training data, the average and standard deviation of scores are given.

Ensemble Methods:

AdaBoost (AB): 0.860 (0.021)
Gradient Boosting (GBM): 0.863 (0.020)
Random Forests (RF): 0.813 (0.025)
Extra Trees (ET): 0.753 (0.023)
XGBClassifier(XGB): 0.864 (0.021)
CatBoost(CB): 0.846 (0.014)

Linear and Non-Linear Models:

Logistic Regression(LR): 0.825(0.023)
Linear Discriminant Analysis (LDA): 0.830 (0.020)
k-Nearest Neighbors (KNN): 0.773 (0.024)
Classification and Regression Trees (CART): 0.683(0.026)
Gaussian Naive Bayes (NB): 0.835 (0.021)

Applying Models

Although Linear and Non-Linear models were much less successful in training data when compared to Ensemble models, predictions(with machine learning) for these models were also made. Then, compared with the test data.

	Accuracy score	Mean squared error
Logistic Regression(LR):	0.944	(0.055)
Linear Discriminant Analysis (LDA):	0.930	(0.069)
k-Nearest Neighbors (KNN):	0.941	(0.058)
Classification and Regression Trees (CART):	0.939	(0.060)
Gaussian Naive Bayes (NB):	0.816	(0.184)

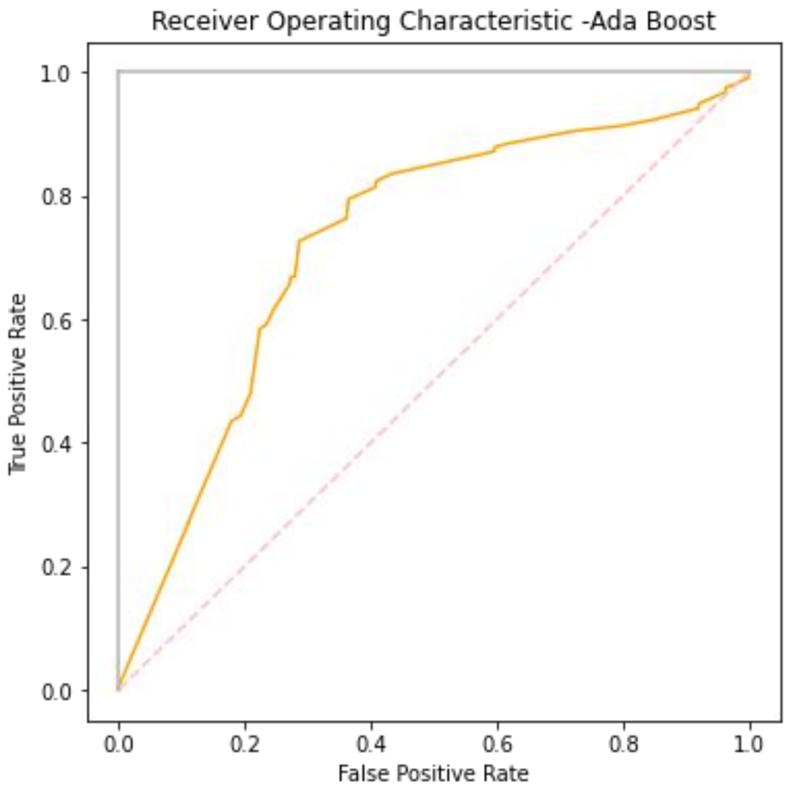
Applying Models

Ensemble Methods - model predictions and their score after evaluated with test data

	Accuracy score	Mean squared error
AdaBoost (AB):	0.946	(0.053)
Gradient Boosting (GBM):	0.947	(0.053)
Random Forests (RF):	0.943	(0.057)
Extra Trees (ET):	0.941	(0.058)
XGBClassifier(XGB):	0.947	(0.053)
CatBoost(CB):	0.944	(0.055)

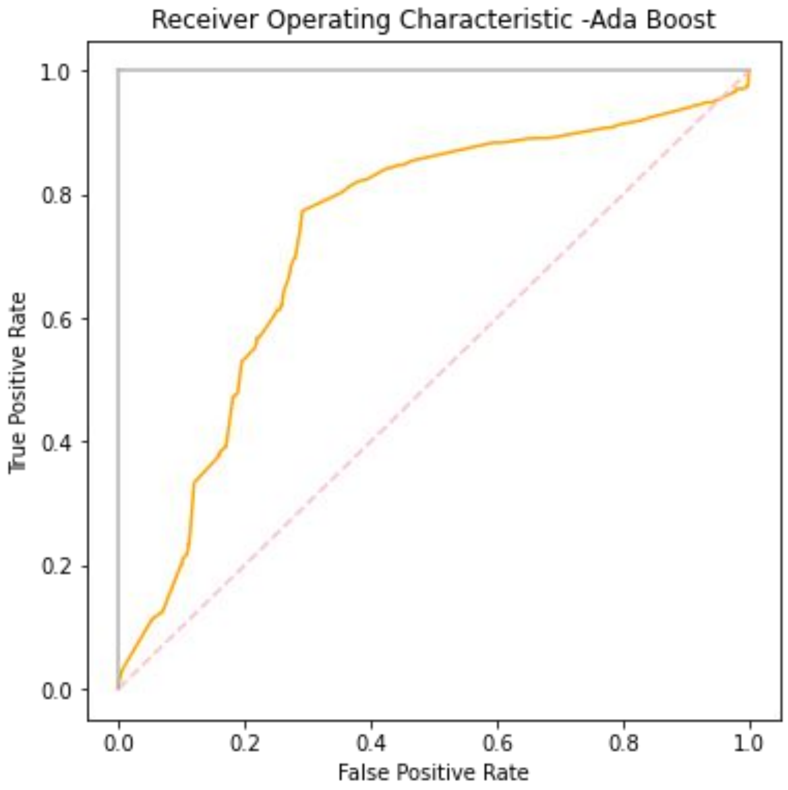
The same 3 models has given the best scores in prediction and test data as well, as it can be seen above

Applying Models



Ada Boost - Default

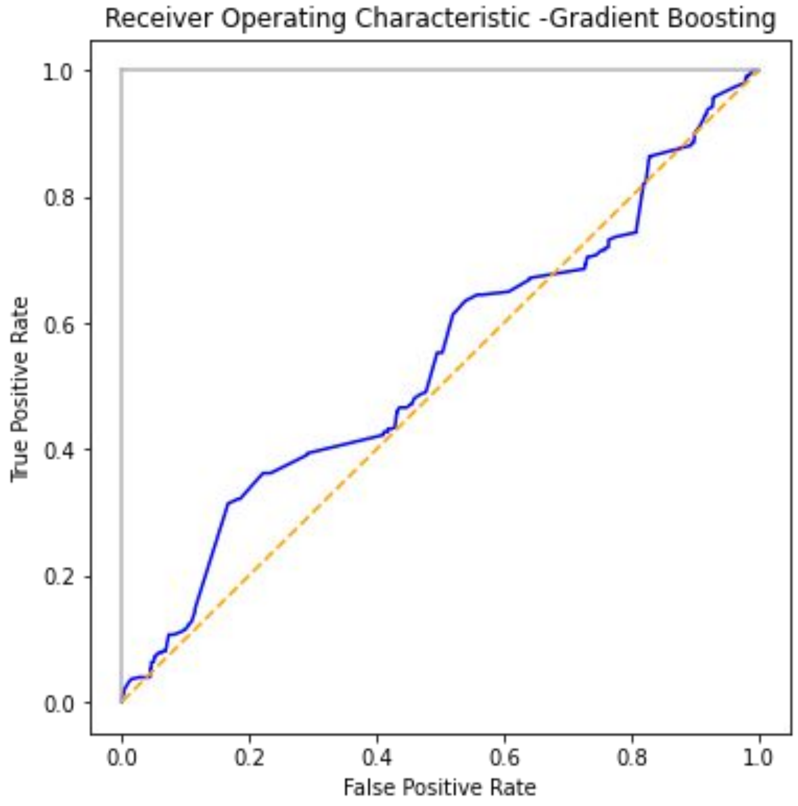
Area under the ROC curve: 0.726

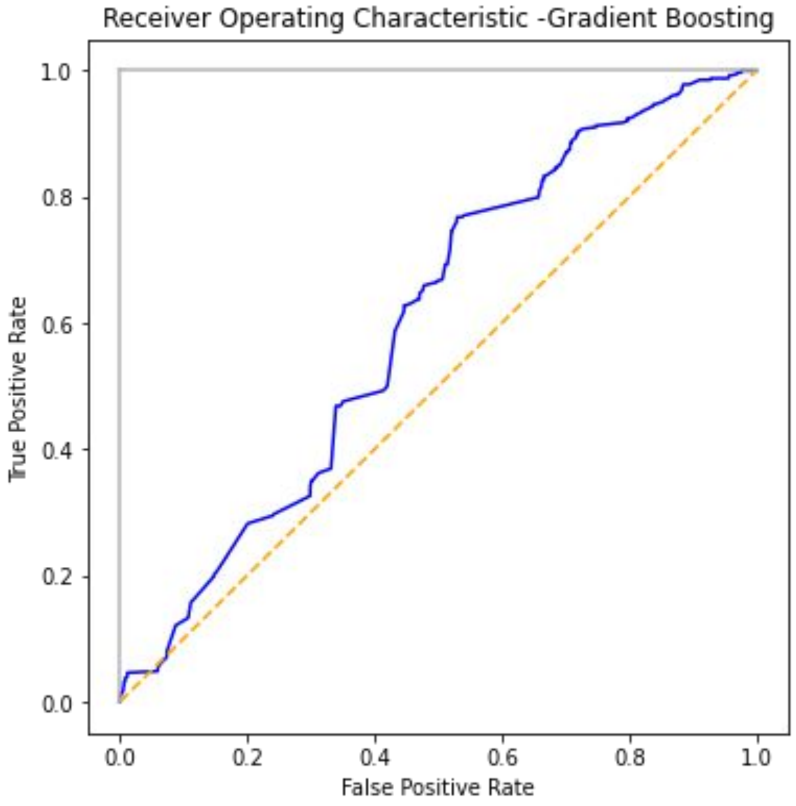


Ada Boost - After hyperparameter tuning

Area under the ROC curve: 0.731

Final Model Selection





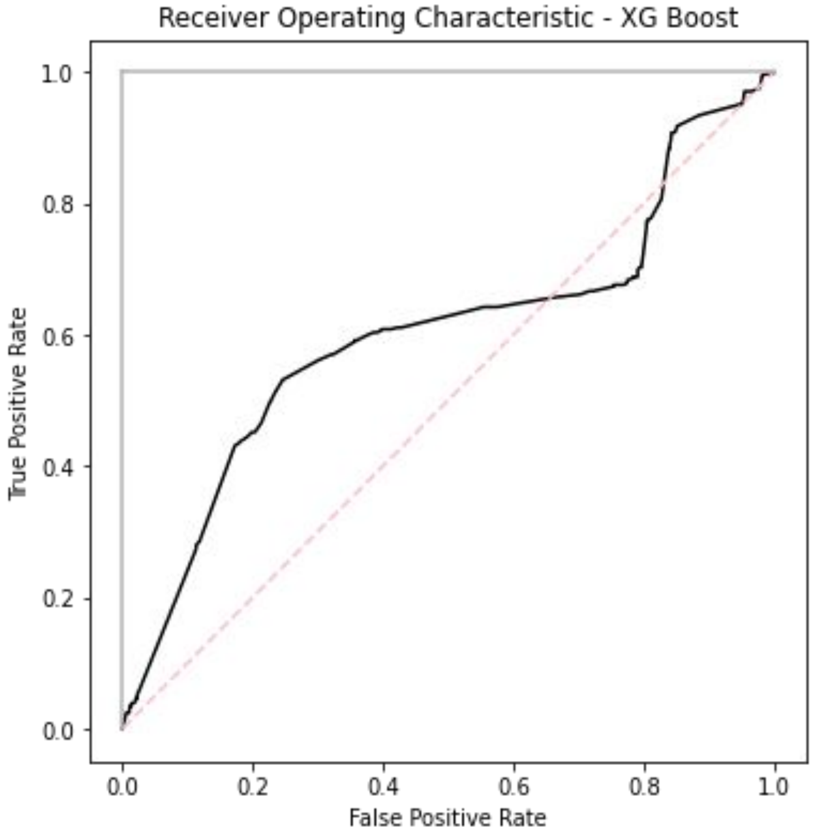
Gradient Boosting - Default

Area under the ROC curve: 0.538

Gradient Boosting- After hyperparameter tuning

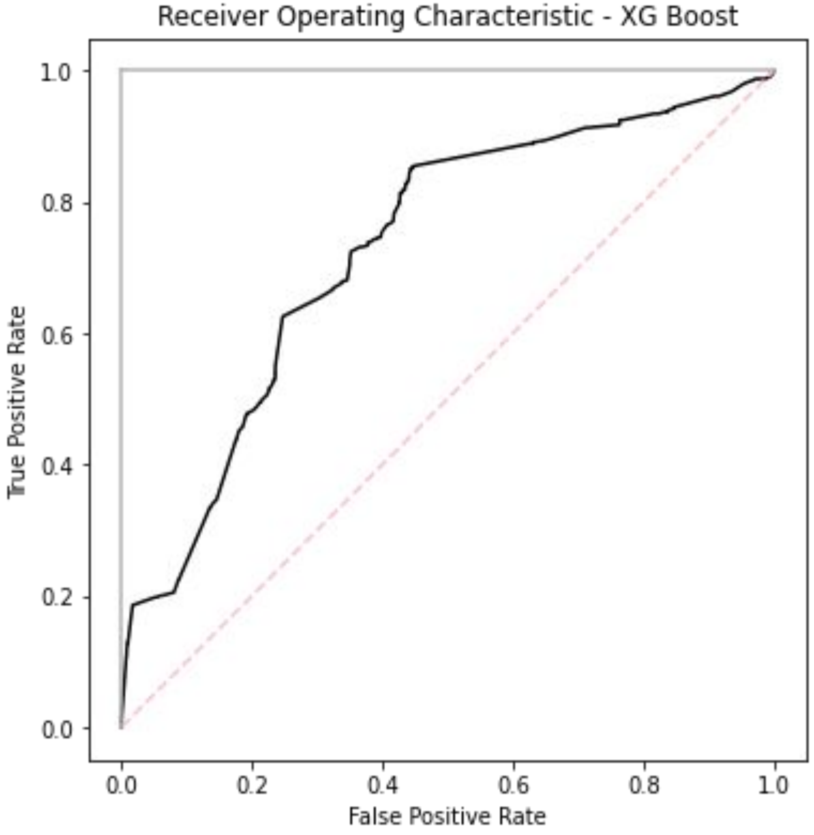
Area under the ROC curve: 0.603

Final Model Selection



XG Boost - Default

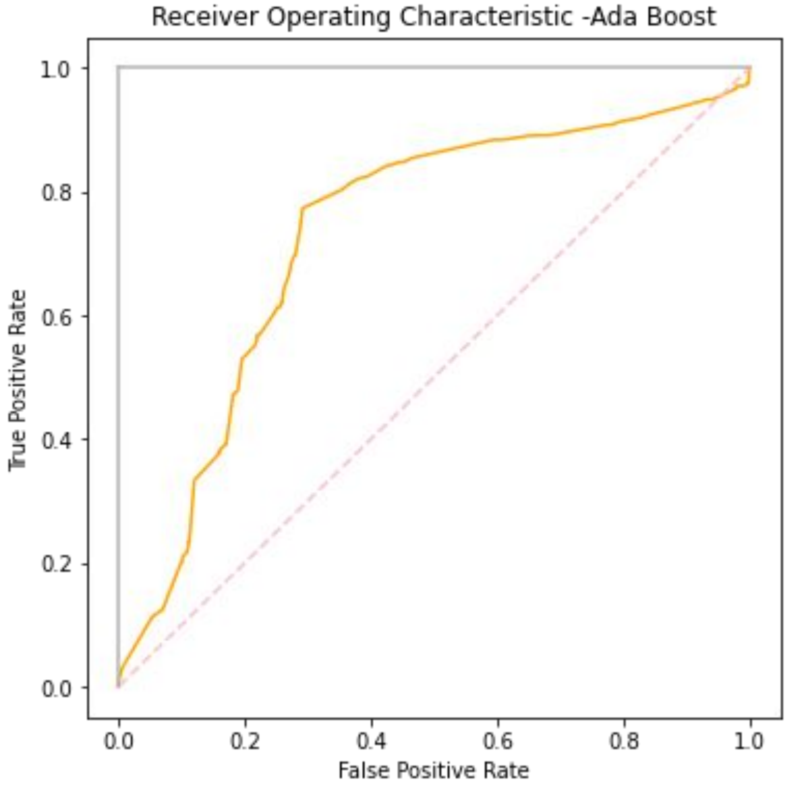
Area under the ROC curve: 0.601



XG Boost - After hyperparameter tuning

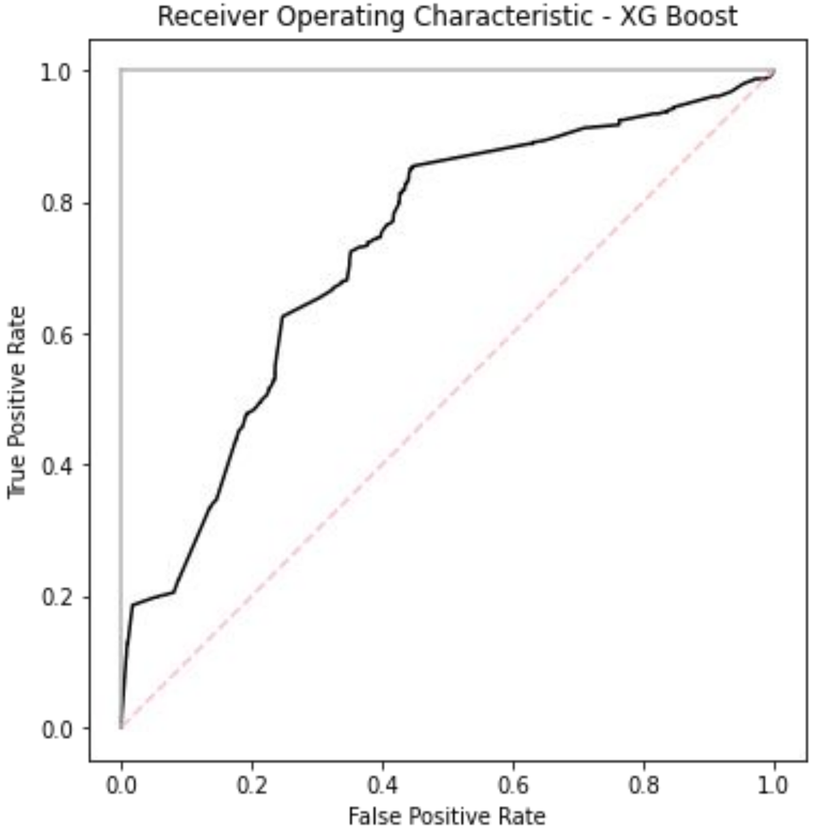
Area under the ROC curve: 0.728

Final Model Selection



Ada Boost - After hyperparameter tuning

Area under the ROC curve: 0.731



XG Boost - After hyperparameter tuning

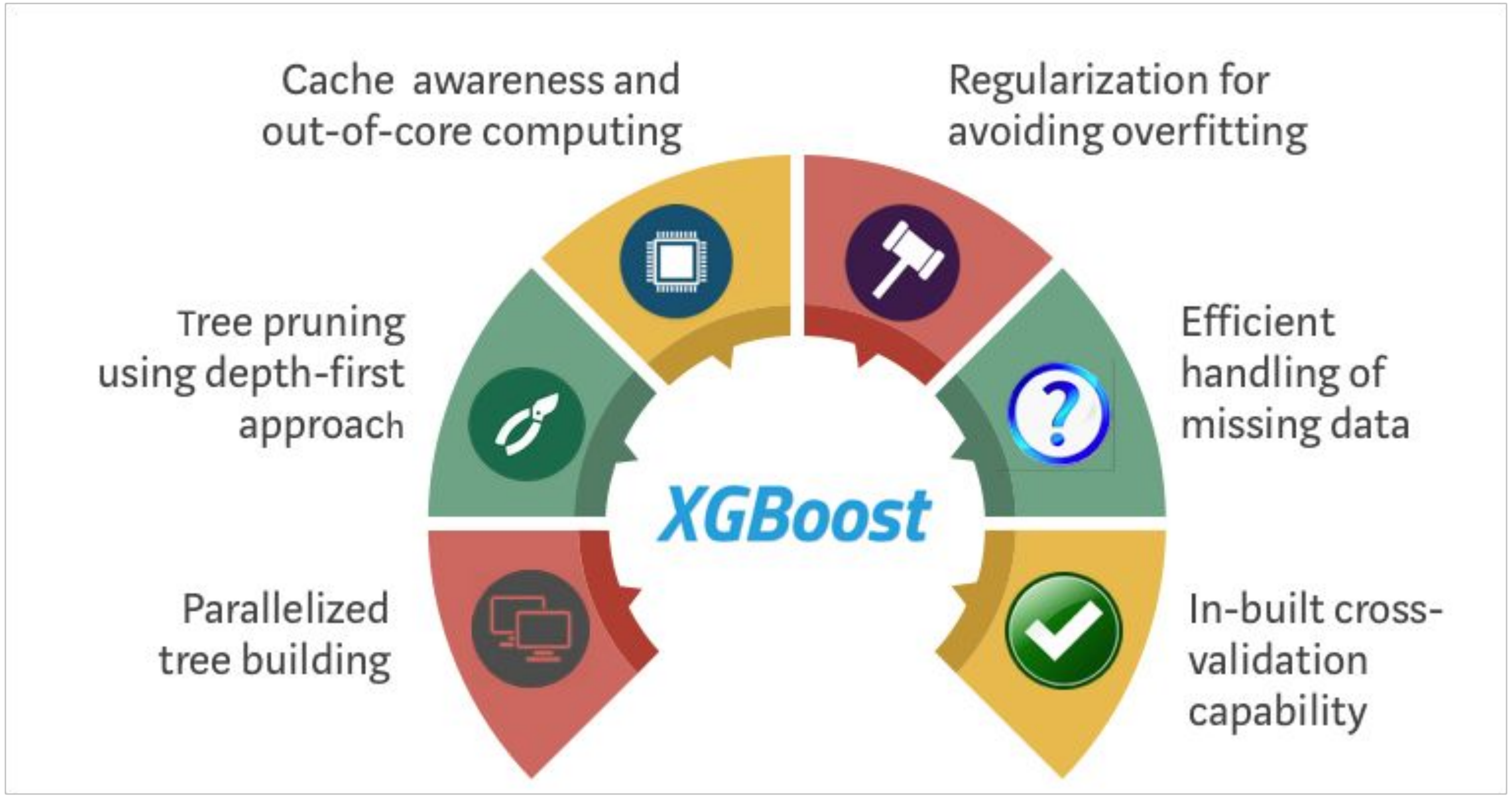
Area under the ROC curve: 0.728

Final Model Selection

After eliminating Gradient Boosting from last 3. we almost got the same results from both ADA Boost and XG Boost. Both can be used for this business to reach their goal efficiently. However, time performance also measured to see the fastest method.

	AdaBoost (AB)	XGBClassifier(XGB)
The spent time for measuring the accuracies for training data process(with k-fold validation learning) :	47 sec	31 sec
The spent time for measuring the accuracies, fitting the models, prediction and comparing with test data:	5 sec	3 sec

Final Model Selection



After all these steps, XGBoost(with hyperparameters) was the model that gave the best prediction results in a fair time.

In most evaluations throughout the whole process this model was the one of the top ones or the one that is the best.

Thank You

Data Glacier

Your Deep Learning Partner