Data Glacier
Your Deep Learning Partner

Final Project Report

Bank Marketing Campaign

Name: Samuel Alejandro Cueva Lozano
Email: samuelcl7@gmail.com
Country: Peru
Specialization: Data Science

Agenda

Business problem
Eda recommendation
Model building
Model selection
Performance metrics
Final recommendation

Business problem

Client: ABC bank: Portuguese banking institution

Problem Description: ABC Bank wants to sell it's term deposit product to customers and before launching the product they want to know whether a particular customer will buy their product or not (based on customer's past interaction with bank or other Financial Institution).

Business goal: Shortlist which customers have more chances to subscribe to the term deposit.

Dataset: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls.

There are three files:

- bank-additional-full.csv with all examples (41188) and 20 inputs, ordered by date (from May 2008 to November 2010).
- bank-additional.csv with 10% of the examples (4119), randomly selected from bank-additional-full.csv, and 20 inputs.
- bank-additional-names.txt with information about the attributes.

EDA Recommendations

After Exploratory Data Analysis and Feature Selection, the features that should be fed to the model are:

Numerical	Categorical	Target
• age • duration • campaign • previous • cons.price.idx • cons.conf.idx	• marital • default • job • contact • education • month • poutcome	• y : Imbalance of categorical target, This problem will be addressed when building the model using **SMOTE** method.

Model Building and Model Selection

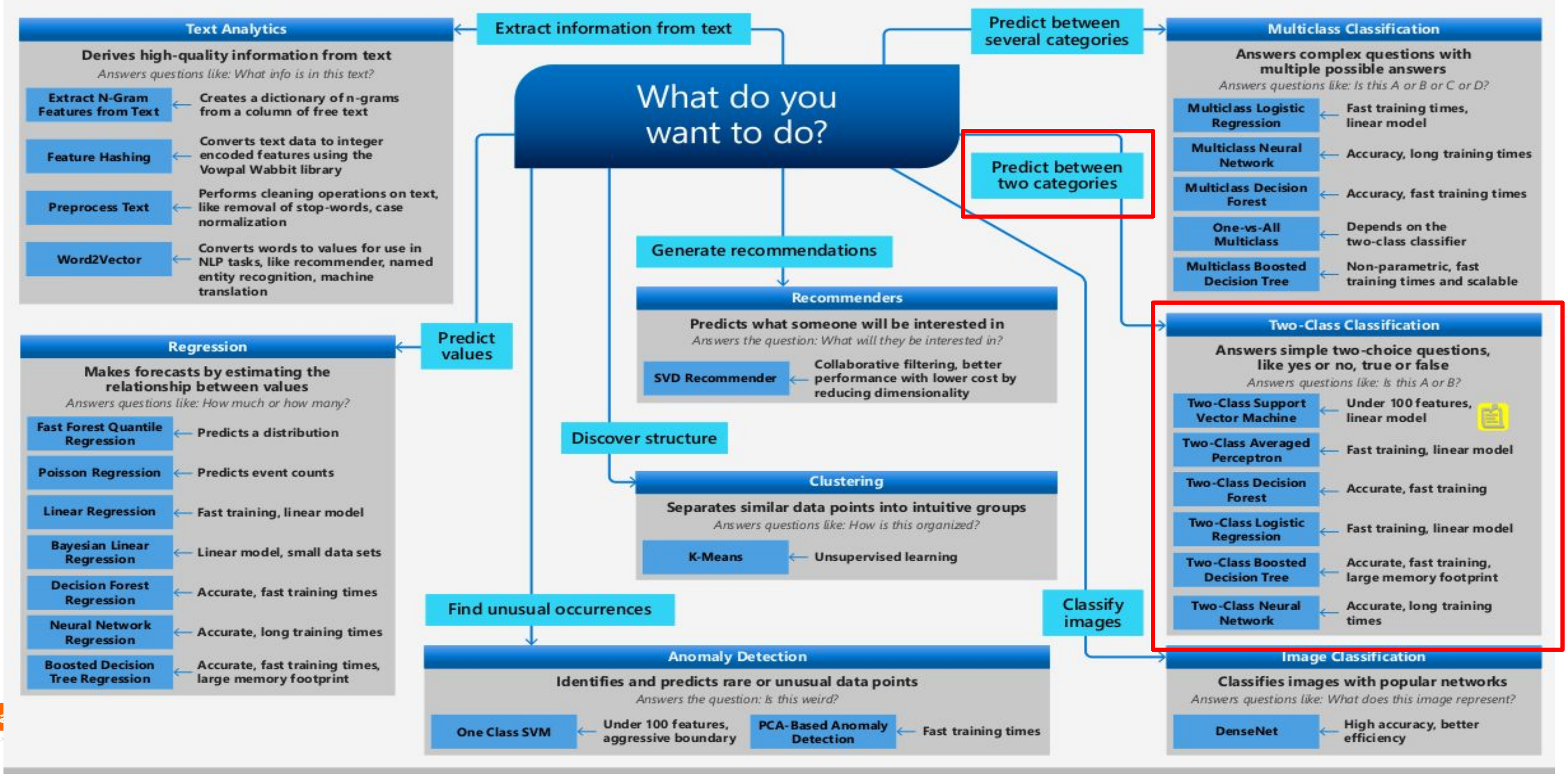
- Oversampling
- Base Model
- Linear Model
- Ensemble Model
- Boosting Model

Oversampling: SMOTE

Deal with data imbalance : SMOTE (Synthetic Minority Oversampling Technique) was used as an oversampling method with a sampling strategy of 0.5, this means that the minority class is oversampled until reaching 50% of the majority class.

	Fraction of positive examples in training dataset	shapes of the attribute matrix and target vector	shapes of the attribute matrix and target vector after splitting
before SMOTE	10.72%	((38625, 44), (38625,))	None
after SMOTE	33.33%	((51724, 44), (51724,))	((46551, 44), (5173, 44), (46551,), (5173,))

Model Selection (Microsoft)



Model Building

Base Model: Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

Sklearn library is used to build the model

```
from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(min_samples_split=100)

# Training
clf.fit(X_train, y_train)
```

Model Building

Linear Model : Logistic Regression

Logistic regression, despite its name, is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function.

Sklearn library is used to build the model

```
from sklearn.linear_model import LogisticRegression

clf = LogisticRegression(solver ='lbfgs',max_iter=5000 )

# Training
clf.fit(X_train, y_train)
```

Model Building

Ensemble model : Random Forest

In random forests each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set.

Sklearn library is used to build the model

```
from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(n_estimators=2000,min_samples_split=200)
# Training
clf.fit(X_train, y_train )
```

Model Building

Boosting model : XGBoost

XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

Sklearn library is used to build the model

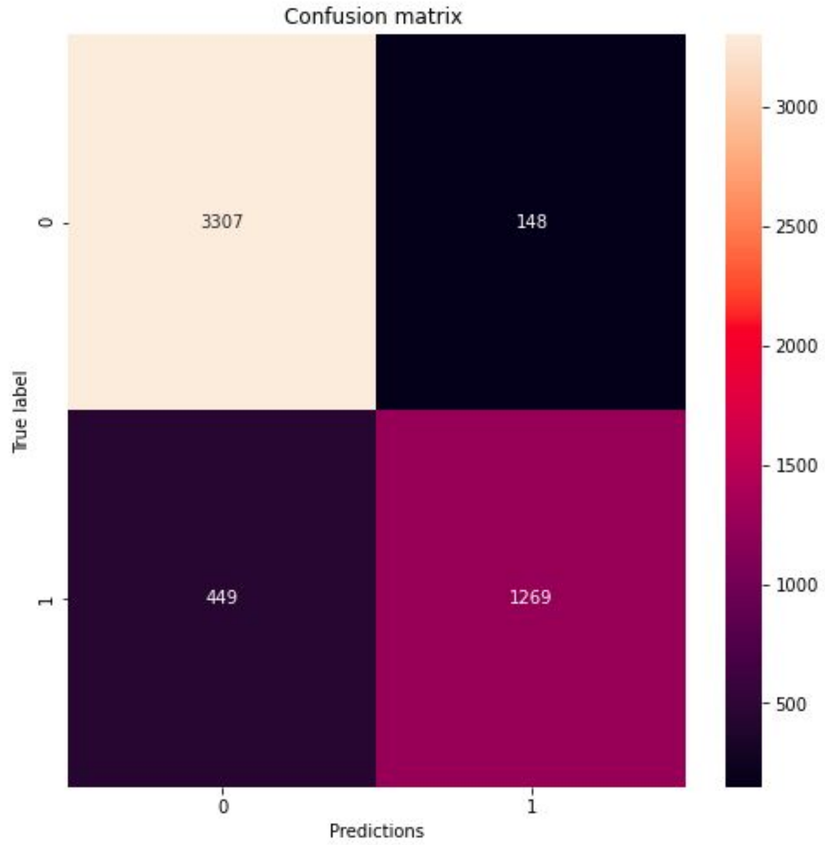
```
import xgboost

clf = xgboost.XGBClassifier(n_estimators=1000,learning_rate=0.01,use_label_encoder=False)

# Training
clf.fit(X_train,y_train)
```

Performance metrics

Base Model
Linear Model
Ensemble Model
Boosting Model

Base Model: Decision Tree

Confusion matrix in Test set



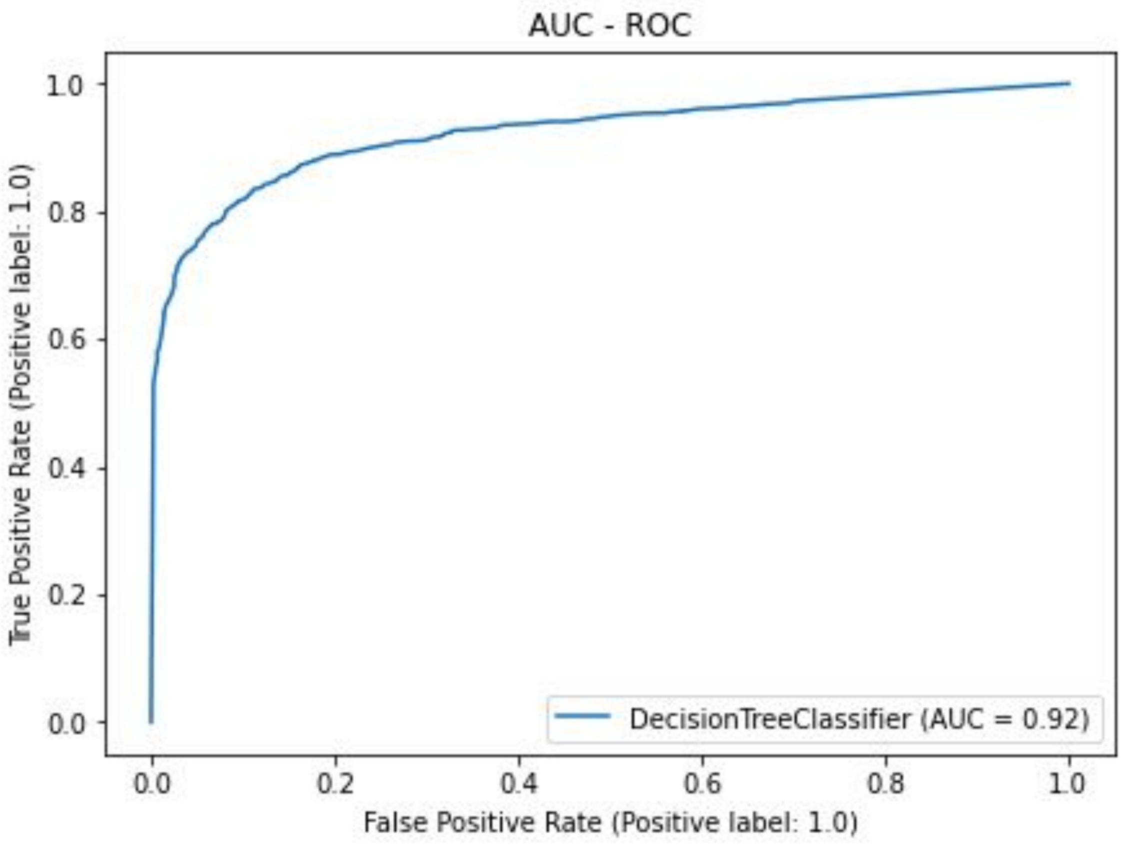
Classification report

	precision	recall	f1-score	support
0.0	0.88	0.96	0.92	3455
1.0	0.90	0.74	0.81	1718
accuracy			0.88	5173
macro avg	0.89	0.85	0.86	5173
weighted avg	0.89	0.88	0.88	5173

Data Glacier

Base Model: Decision Tree

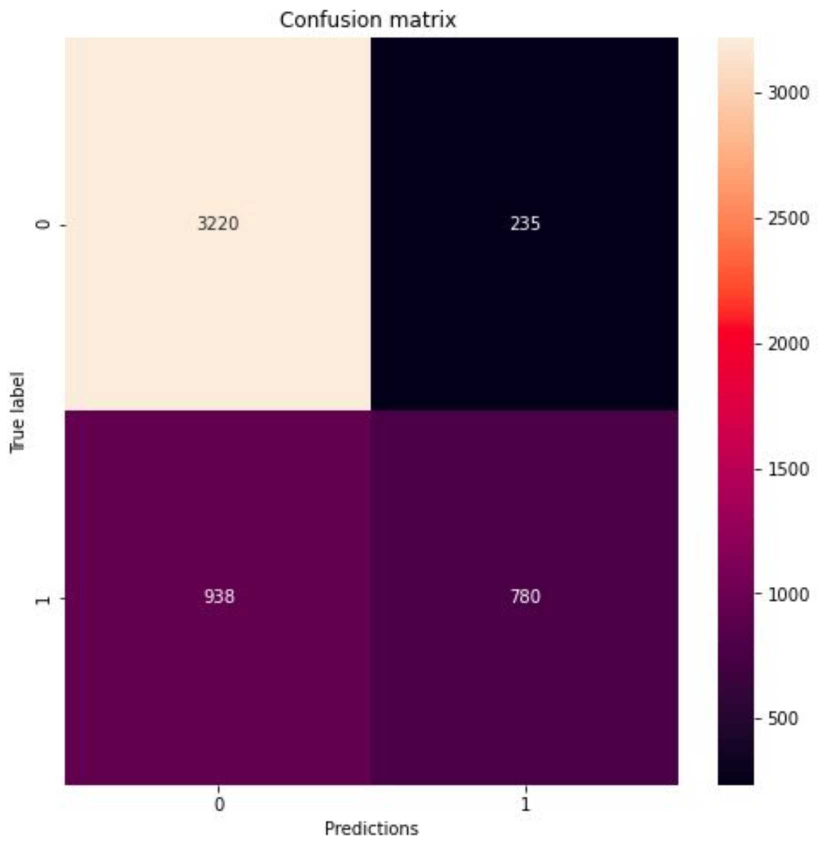
AUC - ROC (Area under the ROC Curve)



Data Glacier

Linear Model: Logistic Regression

Confusion matrix in Test set



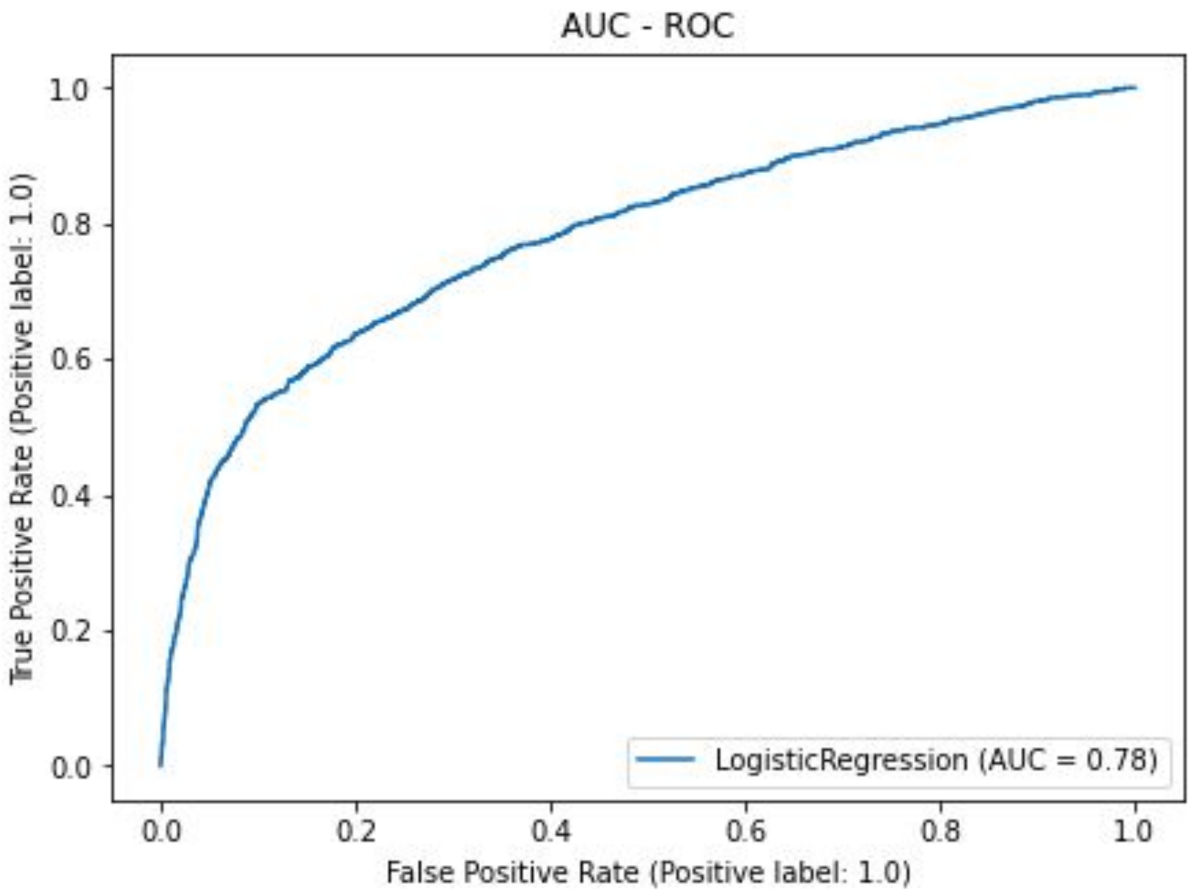
Classification report

	precision	recall	f1-score	support
0.0	0.77	0.93	0.85	3455
1.0	0.77	0.45	0.57	1718
accuracy			0.77	5173
macro avg	0.77	0.69	0.71	5173
weighted avg	0.77	0.77	0.75	5173

Data Glacier

Linear Model:Logistic Regression

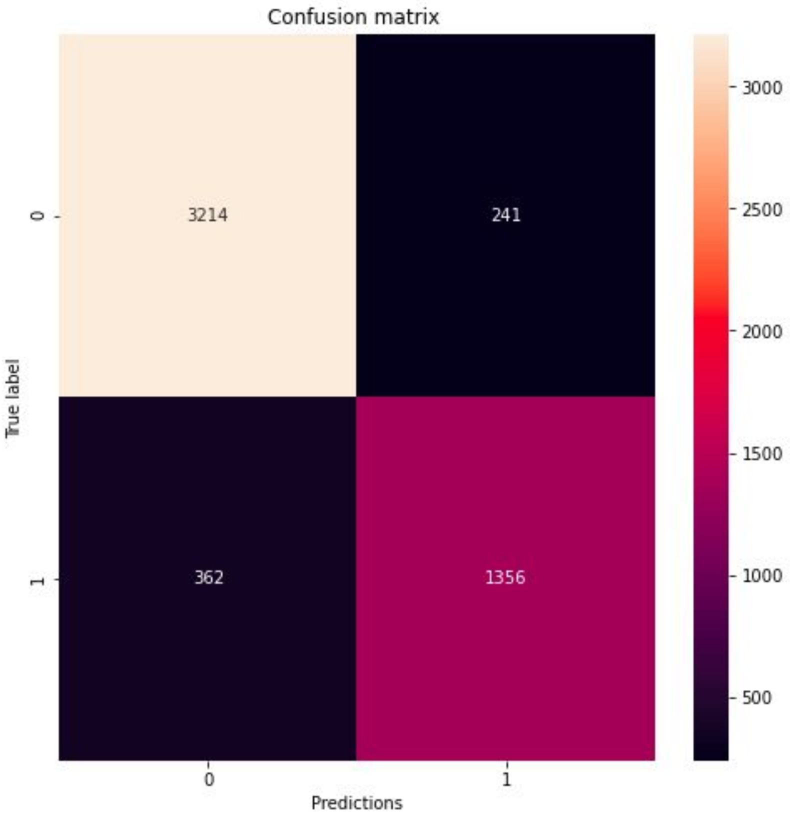
AUC - ROC (Area under the ROC Curve)



Data Glacier

Ensemble model: Random Forest

Confusion matrix in Test set



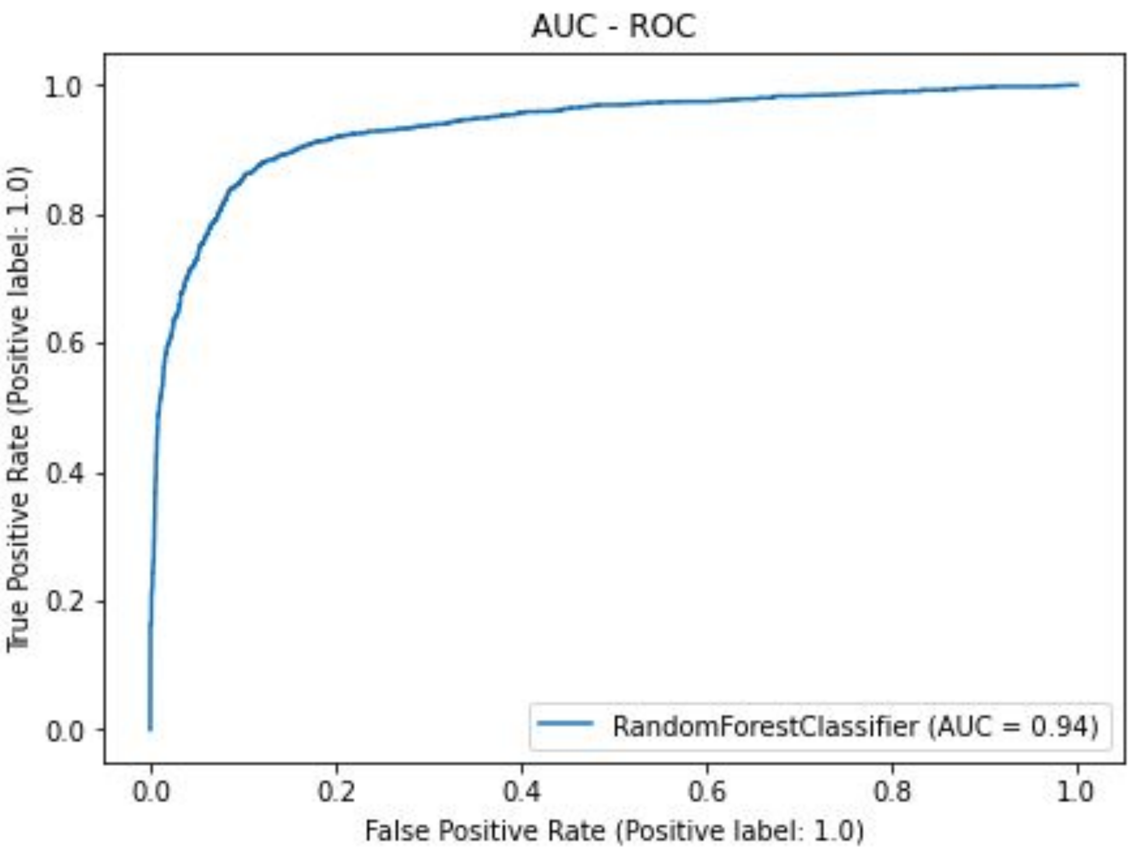
Classification report

	precision	recall	f1-score	support
0.0	0.90	0.93	0.91	3455
1.0	0.85	0.79	0.82	1718
accuracy			0.88	5173
macro avg	0.87	0.86	0.87	5173
weighted avg	0.88	0.88	0.88	5173

Data Glacier

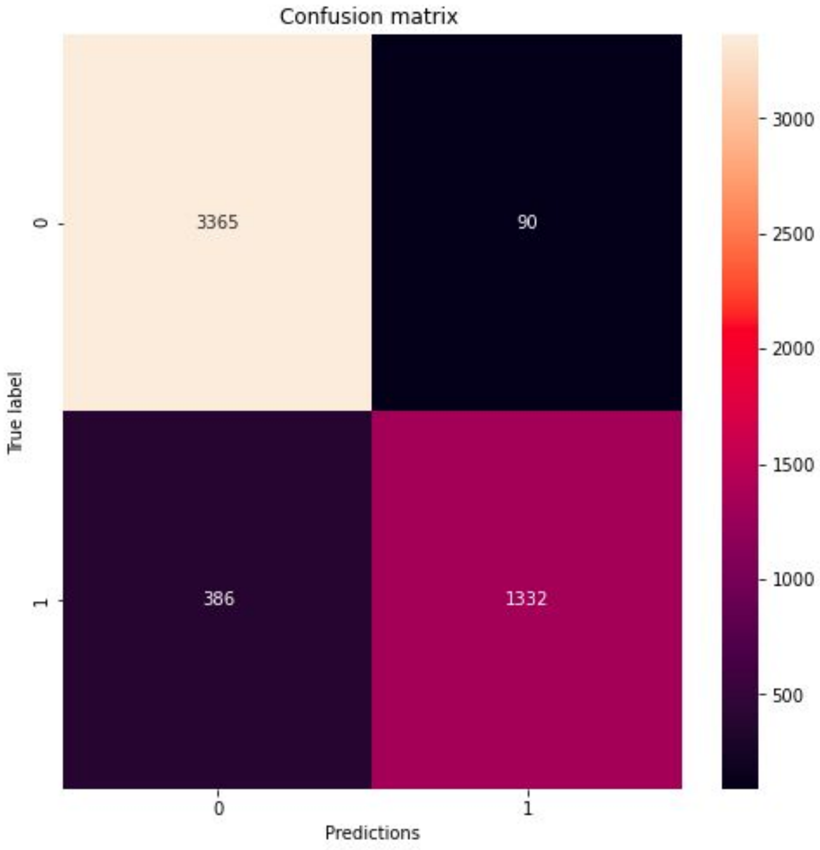
Ensemble model: Random Forest

AUC - ROC (Area under the ROC Curve)



Boosting model: XGBoost

Confusion matrix in Test set



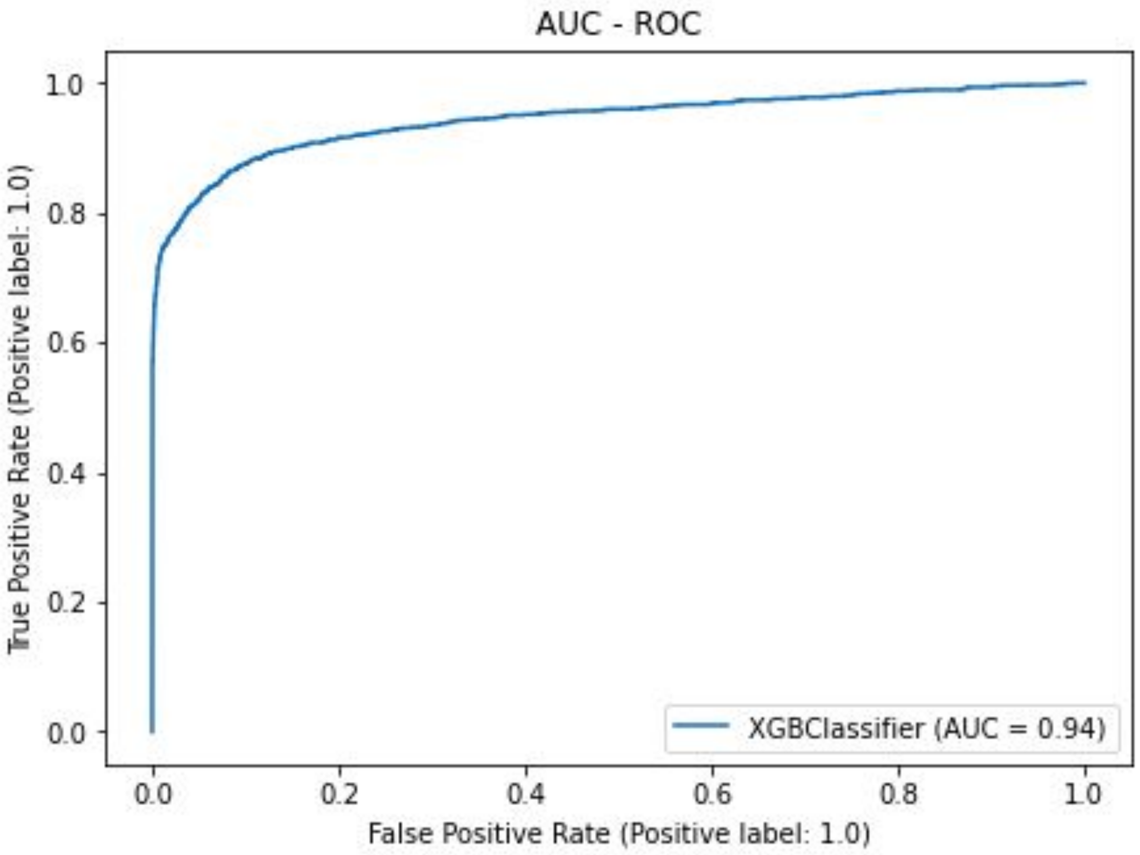
Classification report

	precision	recall	f1-score	support
0.0	0.90	0.97	0.93	3455
1.0	0.94	0.78	0.85	1718
accuracy			0.91	5173
macro avg	0.92	0.87	0.89	5173
weighted avg	0.91	0.91	0.91	5173

Data Glacier

Boosting model: XGBoost

AUC - ROC (Area under the ROC Curve)



Final Recommendations

The better model will be evaluated with the f1-score and AUC-ROC metrics in the test set:

Model	f1-score	AUC-ROC
Decision Tree	0.81	0.92
Logistic Regression	0.57	0.78
Random Forest	0.82	0.94
XGBoost	0.85	0.94

- The best performing model is XGBoost being a bit better than Random Forest
- The model with the lowest performance is the Logistic Regression

Thank You

Data Glacier
Your Deep Learning Partner