Term Deposit Marketing

Overview

Client is a small startup focusing mainly on providing machine learning solutions in the European banking market. They work on a variety of problems including fraud detection, sentiment classification and customer intention prediction and classification.

They are interested in developing a robust machine learning system that leverages information coming from call center data.

Ultimately, they are looking for ways to improve the success rate for calls made to customers for any product that they offer. Towards this goal we are working on designing an ever evolving machine learning product that offers high success outcomes while offering interpretability for our clients to make informed decisions.

Data and Modeling

Data

The data comes from direct marketing efforts of a European banking institution. The marketing campaign involves making a phone call to a customer, often multiple times to ensure a product subscription, in this case a term deposit. Term deposits are usually short-term deposits with maturities ranging from one month to a few years. The customer must understand when buying a term deposit that they can withdraw their funds only after the term ends. All customer information that might reveal personal information is removed due to privacy concerns. Below is data description:

Features:

age : age of customer (numeric)
job : type of job (categorical)
marital : marital status (categorical)
education (categorical)
default: has credit in default? (binary)
balance: average yearly balance, in euros (numeric)
housing: has a housing loan? (binary)
loan: has personal loan? (binary)
contact: contact communication type (categorical)
day: last contact day of the month (numeric)
month: last contact month of year (categorical)
duration: last contact duration, in seconds (numeric)

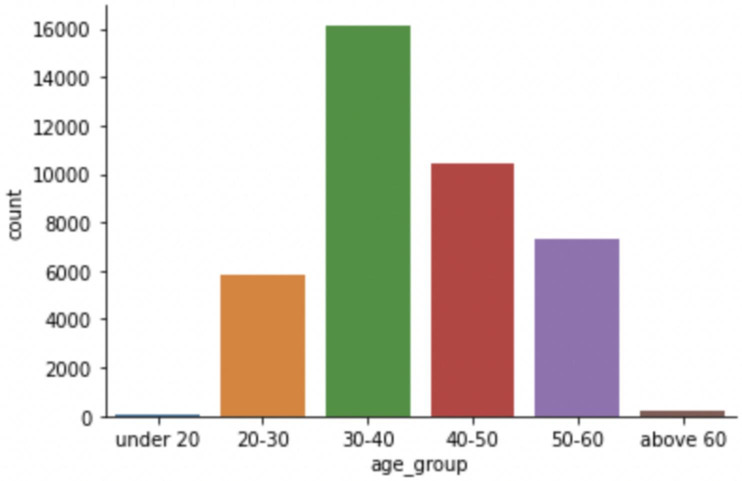
campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

Output (desired target):
y - has the client subscribed to a term deposit? (binary)

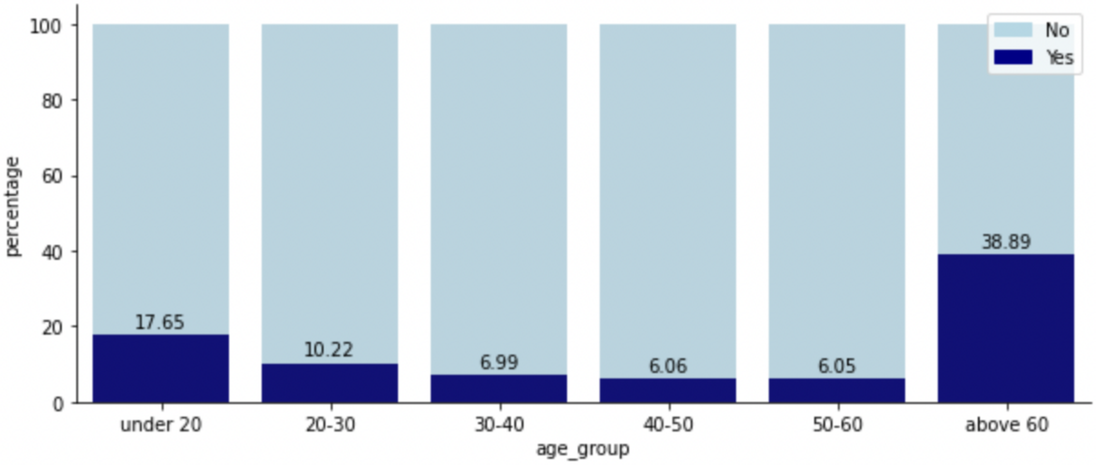
Attributes X1 to X6 indicate the responses for each question and have values from 1 to 5 where the smaller number indicates less and the higher number indicates more towards the answer.

Data exploration analysis

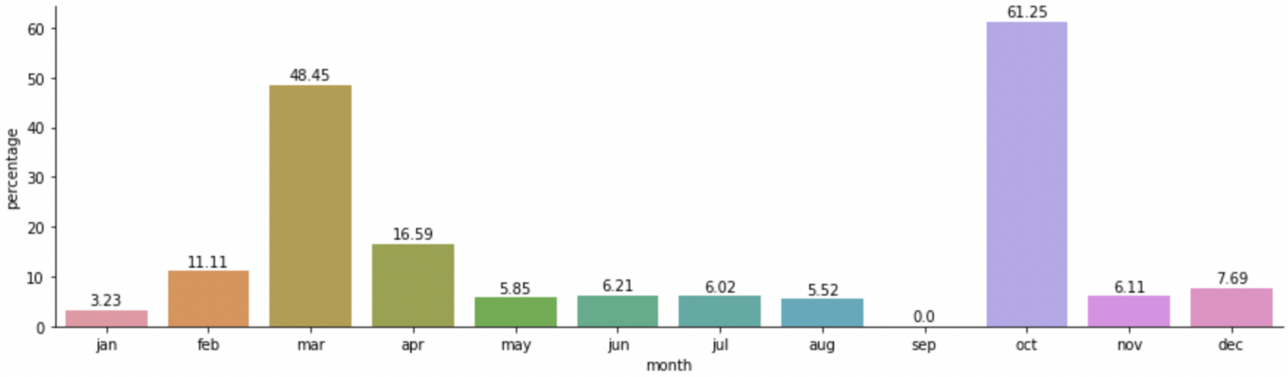
Let's take a look at the distribution of age in this dataset. As we can see, the majority of the people who were called are in the age group of 30-50 years old. People who are under-aged were not called because they can't make the decision on finance products yet. For other age groups, if the distribution matches with the distribution of our general population, then it means that we are not targeting a specific group of people based on their age.



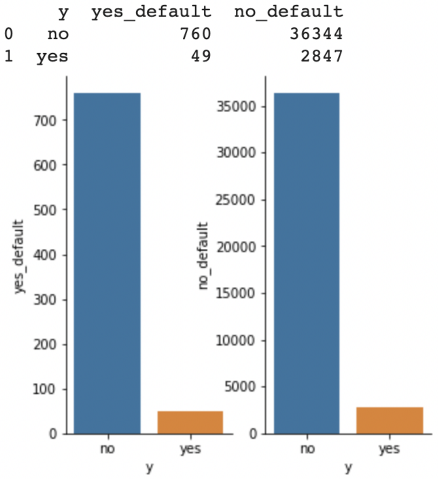
Let's look at how each age group makes decisions on our client's products. As seen below, people who are above 60 and people that are under 20 are most likely to be convinced.



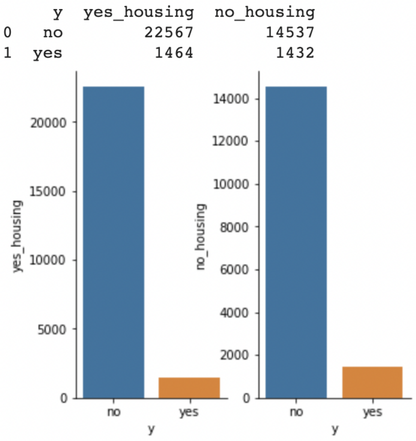
Let’s look at the distribution by month next. The graph below shows the percentage of people who subscribed to a term deposit. The months of March and October have unusually high subscribing rate while September’s is 0%. There is not enough information within this dataset to know if this is a cyclical effect or if this is an artificial external effect (such as marketing campaigns, political policy, etc.)



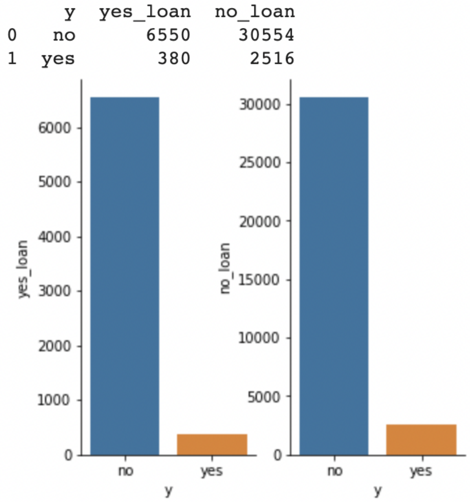
Other features such as (having credit in, having a housing loan, and having personal loan) don’t seem to have an obvious effect on whether a potential lead would convert or not



```
     y  yes_default  no_default
0   no          760       36344
1  yes           49        2847
```



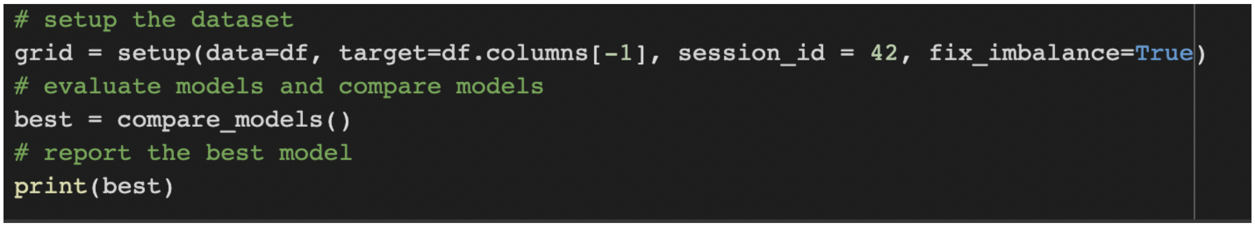
```
     y  yes_housing  no_housing
0   no        22567       14537
1  yes         1464        1432
```



```
     y  yes_loan  no_loan
0   no      6550    30554
1  yes       380     2516
```

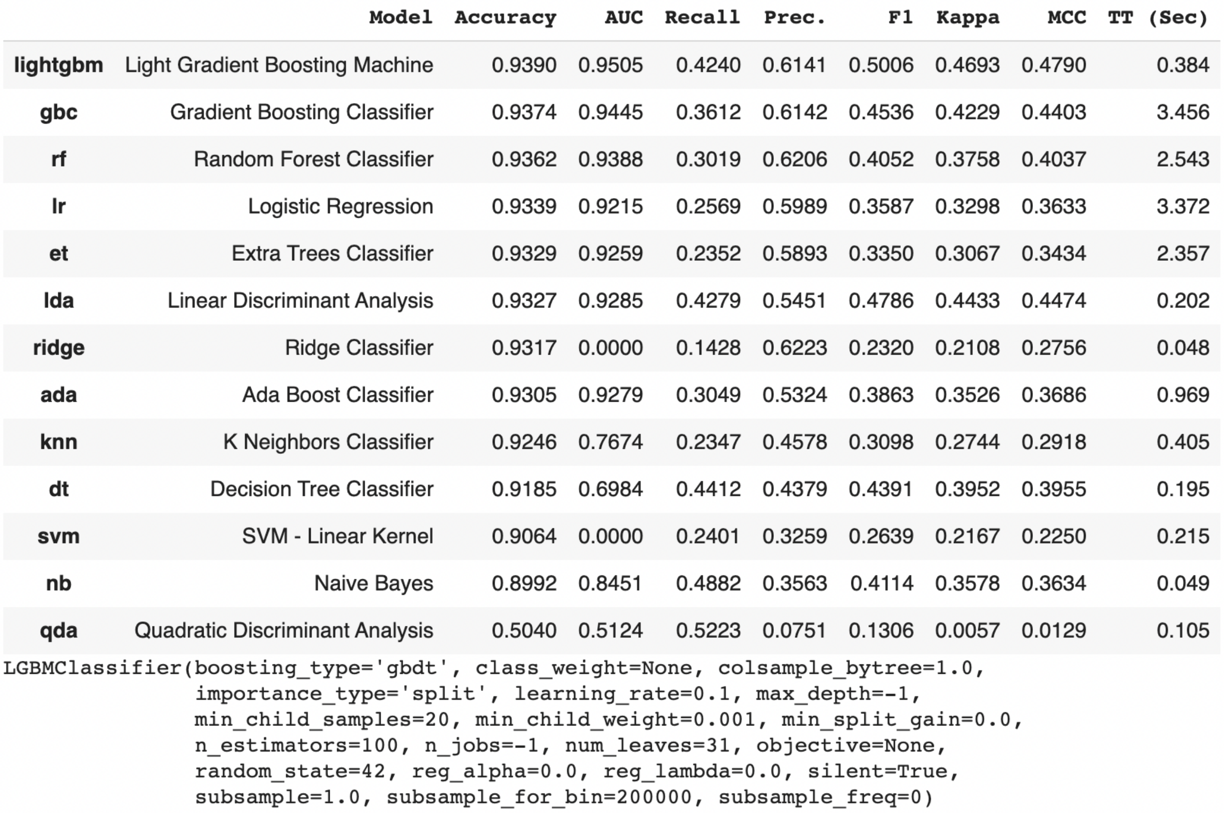
Features Exploration

This is a binary classification problem. In this case, Pycaret is used to establish basic performance for a few models



```
# setup the dataset
grid = setup(data=df, target=df.columns[-1], session_id = 42, fix_imbalance=True)
# evaluate models and compare models
best = compare_models()
# report the best model
print(best)
```

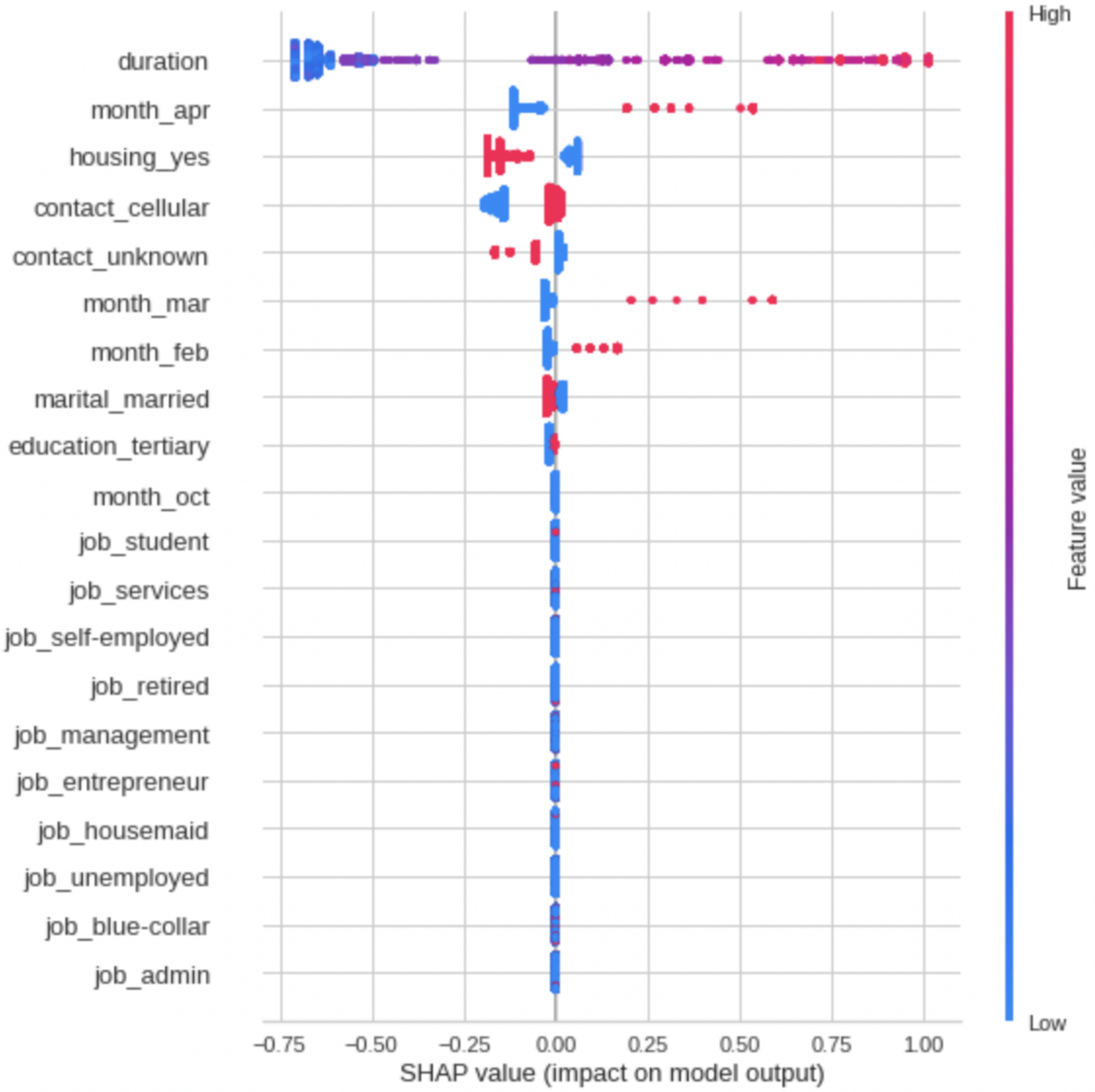
The result is as follows. As we can see, even though accuracy is high, the recall and precision score are not great. In this particular application, it can be argued that recall and precision is more important because our client would most likely want to capture as many potential subscribers as possible. If the rate of false negatives is too high, we are losing opportunities. If the rate of false positive is too high, we are wasting time on calling people who would not subscribe - a waste of company resources.

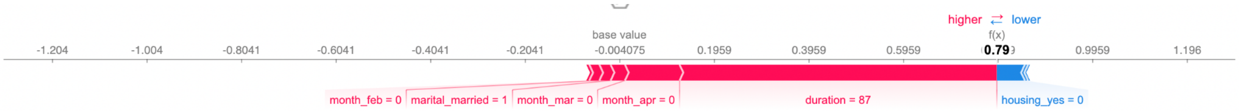


	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.9390	0.9505	0.4240	0.6141	0.5006	0.4693	0.4790	0.384
gbc	Gradient Boosting Classifier	0.9374	0.9445	0.3612	0.6142	0.4536	0.4229	0.4403	3.456
rf	Random Forest Classifier	0.9362	0.9388	0.3019	0.6206	0.4052	0.3758	0.4037	2.543
lr	Logistic Regression	0.9339	0.9215	0.2569	0.5989	0.3587	0.3298	0.3633	3.372
et	Extra Trees Classifier	0.9329	0.9259	0.2352	0.5893	0.3350	0.3067	0.3434	2.357
lda	Linear Discriminant Analysis	0.9327	0.9285	0.4279	0.5451	0.4786	0.4433	0.4474	0.202
ridge	Ridge Classifier	0.9317	0.0000	0.1428	0.6223	0.2320	0.2108	0.2756	0.048
ada	Ada Boost Classifier	0.9305	0.9279	0.3049	0.5324	0.3863	0.3526	0.3686	0.969
knn	K Neighbors Classifier	0.9246	0.7674	0.2347	0.4578	0.3098	0.2744	0.2918	0.405
dt	Decision Tree Classifier	0.9185	0.6984	0.4412	0.4379	0.4391	0.3952	0.3955	0.195
svm	SVM - Linear Kernel	0.9064	0.0000	0.2401	0.3259	0.2639	0.2167	0.2250	0.215
nb	Naive Bayes	0.8992	0.8451	0.4882	0.3563	0.4114	0.3578	0.3634	0.049
qda	Quadratic Discriminant Analysis	0.5040	0.5124	0.5223	0.0751	0.1306	0.0057	0.0129	0.105

```
LGBMClassifier(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
               importance_type='split', learning_rate=0.1, max_depth=-1,
               min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
               n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
               random_state=42, reg_alpha=0.0, reg_lambda=0.0, silent=True,
               subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
```

However, these models allows us to take a peek behind the curtain and see which feature has the biggest impact on our outcome.

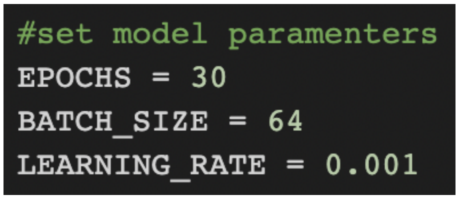




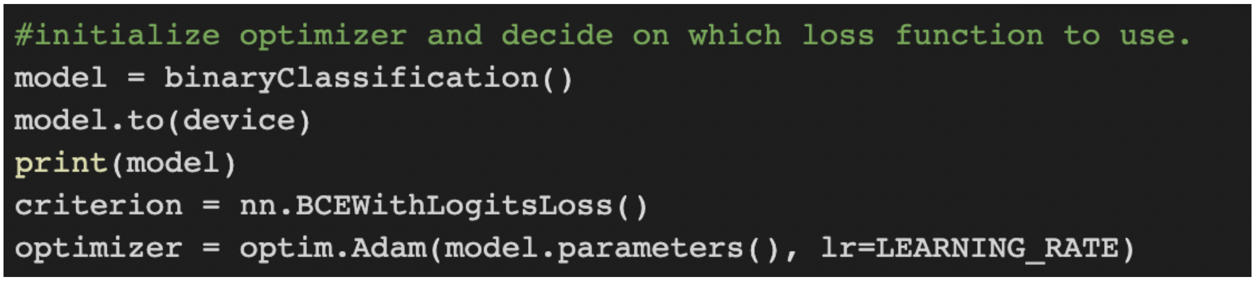
Beside the months which we already established above, duration of last contact, having a housing loan and having a cellphone makes the most impact. The longer the duration, the less likely that our potential leads would subscribe. This might not be a causal effect and would need to be studied more. One explanation could be that people do not like to be dragged and convinced so when being pressed, they respond by turning away. On the other hand, people who have housing loans have an increased chance to be our new subscribers.

Modeling and Result

For this project, we decided to go with deep learning neural networks using Pytorch.

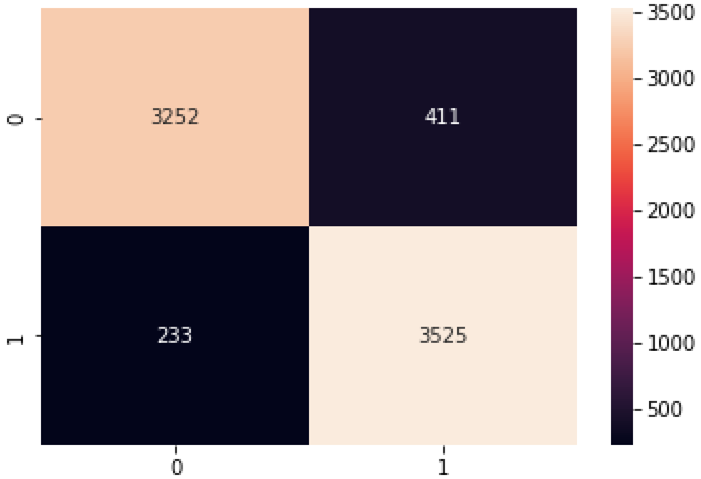


```
#set model paramenters
EPOCHS = 30
BATCH_SIZE = 64
LEARNING_RATE = 0.001
```

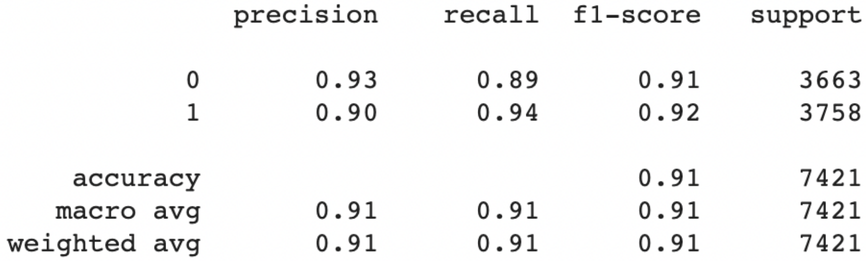


```
#initialize optimizer and decide on which loss function to use.
model = binaryClassification()
model.to(device)
print(model)
criterion = nn.BCEWithLogitsLoss()
optimizer = optim.Adam(model.parameters(), lr=LEARNING_RATE)
```

The result of this approach is much more promising compared to previous models. Confusion matrix shows that we have reduced the false positives and false negatives.



Numerically, we achieved above 90% precision, recall and f1-score.



```
              precision    recall  f1-score   support

           0       0.93      0.89      0.91      3663
           1       0.90      0.94      0.92      3758

    accuracy                           0.91      7421
   macro avg       0.91      0.91      0.91      7421
weighted avg       0.91      0.91      0.91      7421
```