CAPSTONE PROJECT

DATA SCIENTIST NANODEGREE

KWAME ADU MANU

APPLICATION OF SUPERVISED MACHINE LEARNING ALGORITHMS IN PREDICTING CUSTOMER CHURN IN THE BANKING INDUSTRY IN GHANA

Project Overview

The banking industry exists in every economy and is one of the most important sectors in every country. The banking sector is identified as a unit of the economy that is responsible for managing financial assets for their clients. Additionally, banks invest the financial assets as a means of wealth creation (Hall, 2020). “The sector also includes the regulation of banking activities by government agencies, insurance, mortgages, investor services, and credit cards (Hall, 2020)”.

Customer churn or attrition is one major problem for banks and as Verbeke et al (2011) asserts, putting in efforts to retain customers eventually leads to profitability. This is also affirmed by Tsai and Chen (2010) and they identified churn management as an important strategy for businesses. Just like most competitive industries, customers are more likely to switch between banks and this can be as a result of several factors such new products and services, customer experience and others. Nie et al. identified that banks can increase profits up to 85% by focusing on and improving their rate of retention by 5%. Banks that are able to predict customer churn can send targeted marketing campaigns to retain customers (Ganesh, et al., 2000).

This project leverages supervised machine learning techniques in predicting customer churn in the banking industry and uses dataset from Kaggle.

Problem Statement

Customer churn is when a customer decides to and also actions on their decision to end their engagement with a company by ceasing to use any product or service (Colgate et al, 1996).

New technology in the finance sector has led to changes in customer demand and this has resulted in competition for banks (The Economist, 2019). One of the biggest challengers to the banking industry has the mobile money fintech innovation.

Banking in Africa is highly competitive with the proliferation of many banks and the ability of banks to leverage on existing data to correctly predict customers most likely to churn is an important activity.

Metrics

There are several metrics for evaluating the performance of Machine Learning algorithms. For this project, the confusion matrix was used.

The accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation:

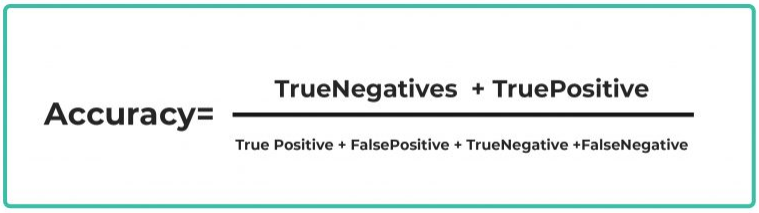


Fig 1 Source: https://uruit.com/blog/churn-prediction-machine-learning/

The recall or true positive rate (TP) is the proportion of positive cases that were correctly identified, as calculated using the equation:

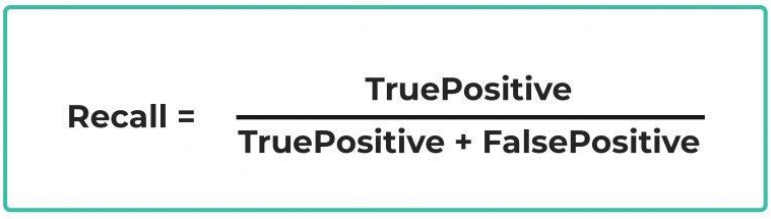


Fig 2 Source: https://uruit.com/blog/churn-prediction-machine-learning/

Precision metric measures how precise a model is in predicting positive cases

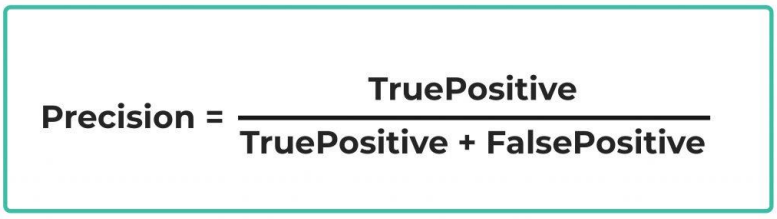


Fig 3 Source: https://uruit.com/blog/churn-prediction-machine-learning/

Data Exploration

This section explores the processes that were undertaken to load the data and perform some exploratory analysis.

The data was obtained from Kaggle (https://www.kaggle.com/janiobachmann/bank-marketing-datasetz)

Reading the data:

```
#read data
df = pd.read_csv('C:/Users/kwame.adu/Desktop/Kwame/Learning/Data Science/Capstone Project/bank_data.csv')
```

Fig 4

Reviewing data:

	customer_id	age	job	marital	education	default	balance	housing	loan	duration	products	churn	location
0	1001	59	admin.	married	secondary	0	2343	1	0	1042	1	1	Achimota
1	1002	56	admin.	married	secondary	0	45	0	0	1467	1	1	Legon
2	1003	41	technician	married	secondary	0	1270	1	0	1389	1	1	Kisseman
3	1004	55	services	married	secondary	0	2476	1	0	579	1	1	Kwabenya
4	1005	54	admin.	married	tertiary	0	184	0	0	673	2	1	Abokobi
5	1006	42	management	single	tertiary	0	0	1	1	562	2	1	Adjen Kotoku
6	1007	56	management	married	tertiary	0	830	1	1	1201	1	1	Accra New Town
7	1008	60	retired	divorced	secondary	0	545	1	0	1030	1	1	Adabraka
8	1009	37	technician	married	secondary	0	1	1	0	608	1	1	Abossey Okai
9	1010	28	services	single	secondary	0	5090	1	0	1297	3	1	Kaneshie

Fig 5

Reviewing data type:

```
#list of columns and data type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11162 entries, 0 to 11161
Data columns (total 13 columns):
customer_id    11162 non-null int64
age            11162 non-null int64
job            11162 non-null object
marital        11162 non-null object
education      11162 non-null object
default        11162 non-null int64
balance        11162 non-null int64
housing        11162 non-null int64
loan           11162 non-null int64
duration       11162 non-null int64
products       11162 non-null int64
churn          11162 non-null int64
location       11162 non-null object
dtypes: int64(9), object(4)
memory usage: 1.1+ MB
```

Fig 6

Describing the data:

The describe function provides some key statistics about the dataset such as the mean age which is identified as 41 years.

```
df.describe()
```

	customer_id	age	default	balance	housing	loan	duration	products	churn
count	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000	11162.000000
mean	6581.500000	41.231948	0.015051	1528.538524	0.473123	0.130801	371.993818	2.508421	0.793585
std	3222.336187	11.913369	0.121761	3225.413326	0.499299	0.337198	347.128386	2.722077	0.404750
min	1001.000000	18.000000	0.000000	-6847.000000	0.000000	0.000000	2.000000	1.000000	0.000000
25%	3791.250000	32.000000	0.000000	122.000000	0.000000	0.000000	138.000000	1.000000	1.000000
50%	6581.500000	39.000000	0.000000	550.000000	0.000000	0.000000	255.000000	2.000000	1.000000
75%	9371.750000	49.000000	0.000000	1708.000000	1.000000	0.000000	496.000000	3.000000	1.000000
max	12162.000000	95.000000	1.000000	81204.000000	1.000000	1.000000	3881.000000	63.000000	1.000000

Fig 7

Checking for null cells:

The next activity was to identify null values and the ‘isnull’ function identified all cells had data.

```
#checking cells without data
df.isnull().sum()
```

```
customer_id    0
age            0
job            0
marital        0
education      0
default        0
balance        0
housing        0
loan           0
duration       0
products       0
churn          0
location       0
dtype: int64
```

Fig 8

Job Distribution in Customer Attrition

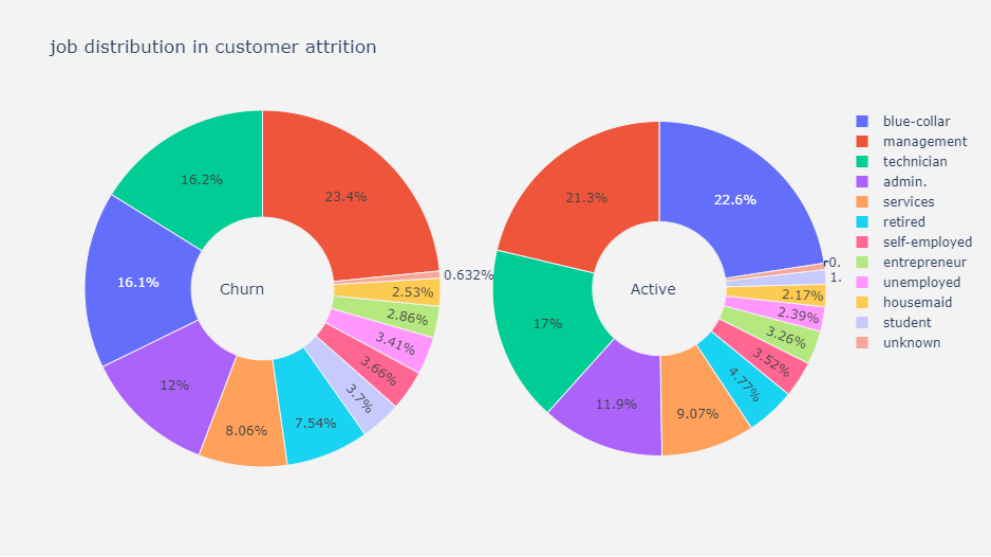


Fig 9

This chart shows the distribution of the job types and the distribution across their status (churn or active customers). The chart shows that a significant portion of churn occurs within the

management job sector. There is also significant attrition from technicians and blue-collar jobs with 16.2% and 16.1% from these sectors respectively.

Education Distribution in Customer Attrition

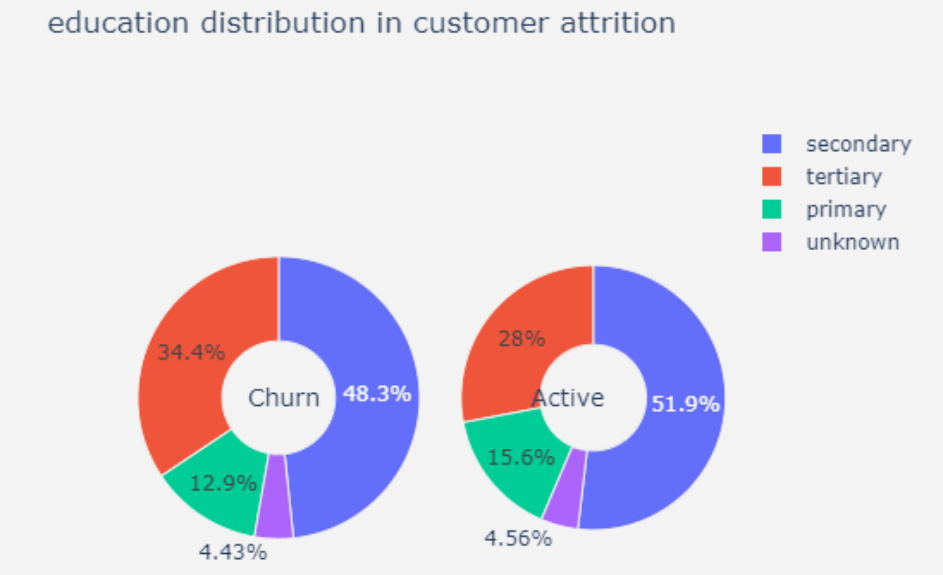


Fig 10

Age Distribution

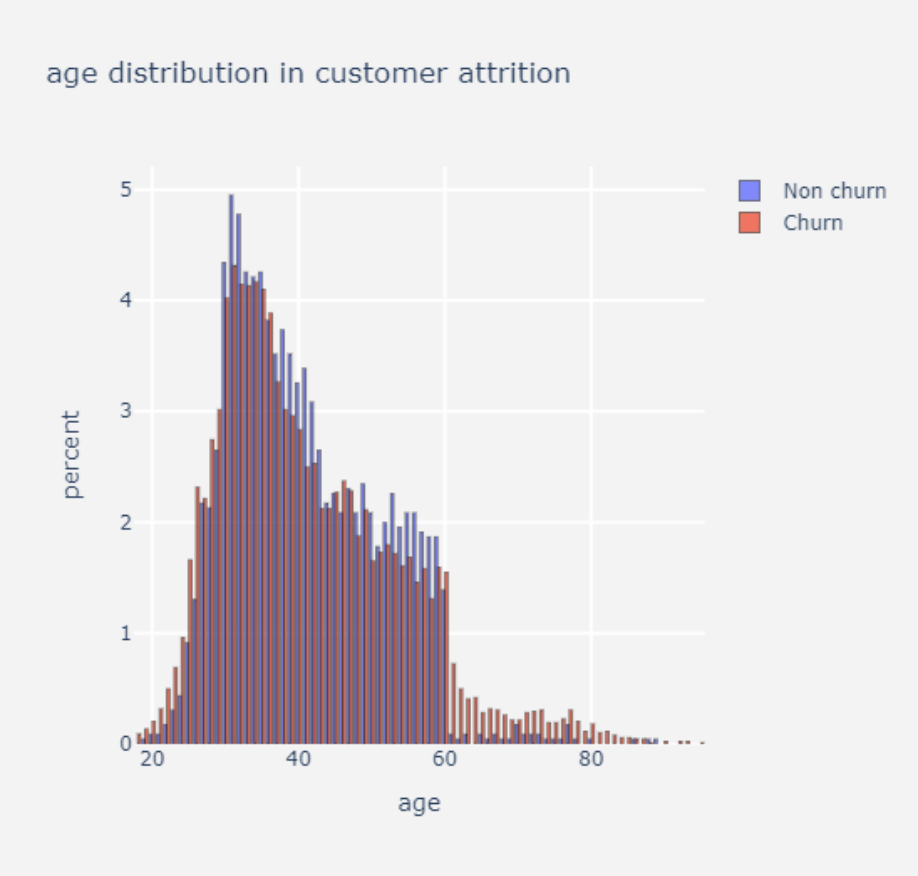


Fig 11

Housing Distribution in Customer Attrition

To understand the relationship between customers with or without housing and churn, the values yes or no was converted to 0 and 1. A basic interpretation of the fig 12 shows more customers without housing churned than those with housing.

Fig 12

Loan distribution

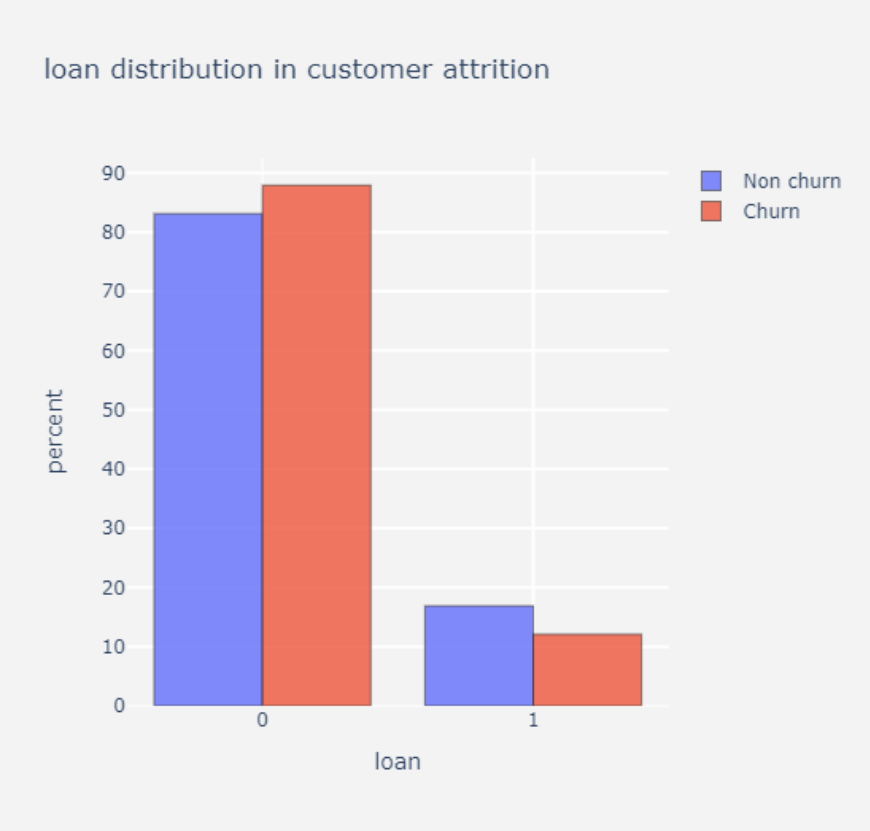


Fig 13

Customer balance distribution

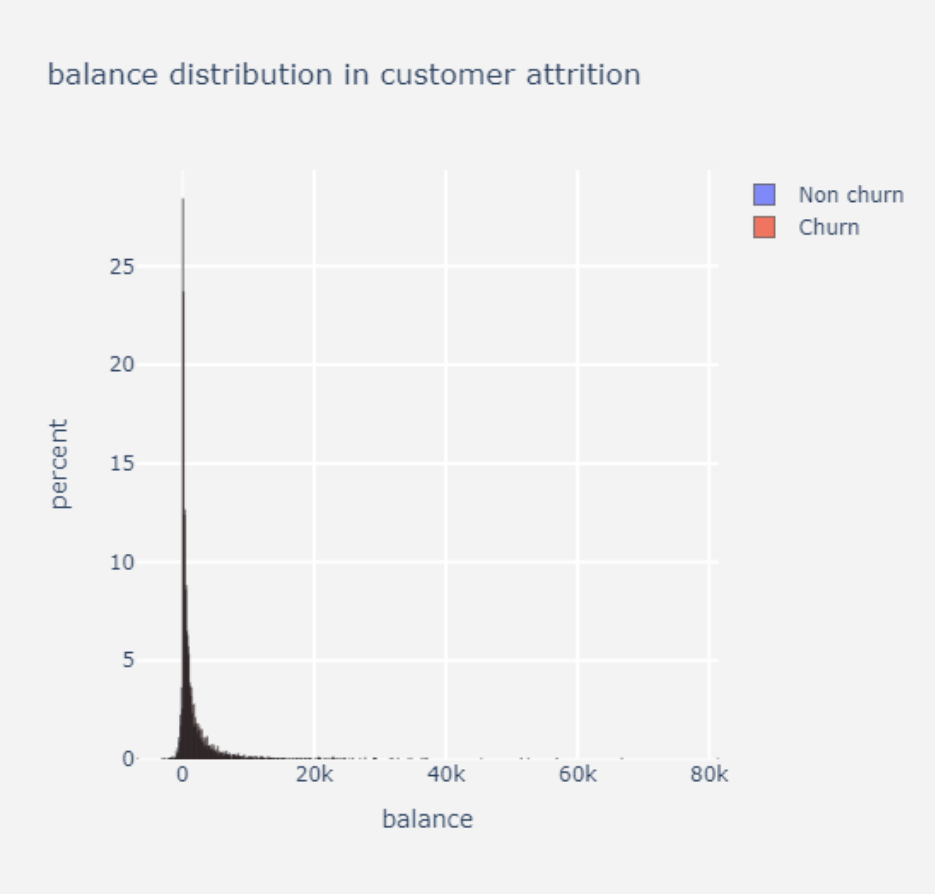


Fig 14

Correlation Matrix

The correlation matrix is key in multivariate analysis because it captures the pairwise degrees of relationship between different components of a random vector (Pham-Gia and Choulakian, 2014).

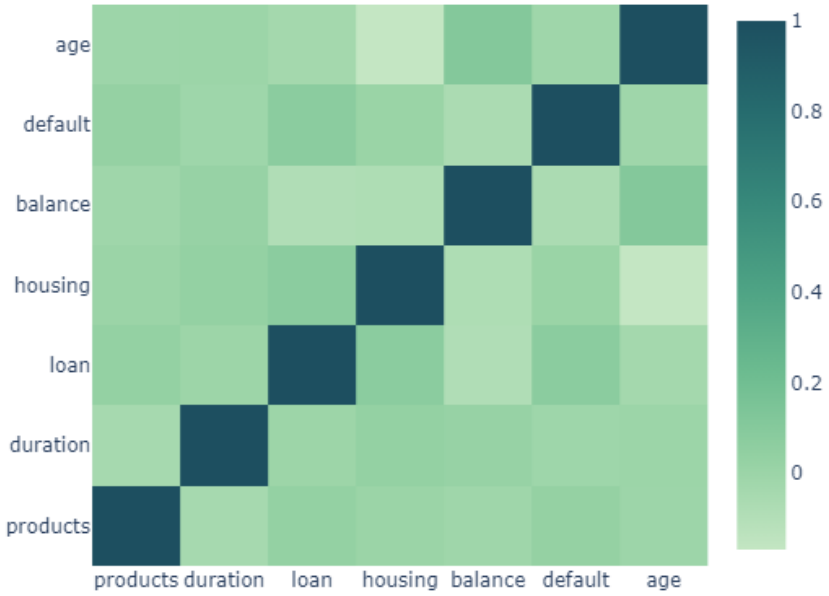


Fig 15

The scatter plot shows the degree of relationship between the numerical values and churn. For example, age has a positive relationship with churn.

Preparing Dataset for Model

Data for machine learning algorithms are often cleaned, formatted and restructured, known as pre-processing. Non-numeric data would have been converted to numeric with one-hot encoding. This a method for converting categorical variables to numerical values.

Implementation – Creating a Training and Predicting Pipeline

A training and predicting pipeline were created to properly evaluate the performance of each model. This allows for quick and effective training of models using various sizes of training data and perform predictions on the testing data. 80% of the data was used for training and 20% for testing.

```
X = df[['age', 'default', 'balance', 'housing', 'loan']]
y = df["churn"]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state=42)

print("Training set has {} samples.".format(X_train.shape[0]))
print("Testing set has {} samples.".format(X_test.shape[0]))
```

```
Training set has 8929 samples.
Testing set has 2233 samples.
```

Evaluating Model Performance

The XGBoost stands for eXtreme Gradient Boosting, which is a boosting algorithm based on gradient boosted decision trees algorithm. XGBoost applies a better regularization technique to reduce overfitting, and it is one of the differences from the gradient boosting.

Checking accuracy on test set:

```
from xgboost import XGBClassifier

model = XGBClassifier()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
predictions = [round(value) for value in y_pred]

accuracy = accuracy_score(y_test, predictions)
print("Accuracy: %2f%%" % (accuracy * 100.0))
```

```
C:\Users\kwame.adu\AppData\Local\Continuum\anaconda3\lib\site-packages\xgboost\sklearn.py:1146: UserWarning:

The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do
the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y)
as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].

[20:57:38] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.
3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicit
ly set eval_metric if you'd like to restore the old behavior.
Accuracy: 79.131214%
```

Checking accuracy on both test and train set:

```
#train and test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05, random_state=56)
#building the model & printing the score
xgb_model = xgb.XGBClassifier(max_depth=5, learning_rate=0.08, objective= 'binary:logistic',n_jobs=-1).fit(X_train, y_train)
print('Accuracy of XGB classifier on training set: {:.2f}'
       .format(xgb_model.score(X_train, y_train)))
print('Accuracy of XGB classifier on test set: {:.2f}'
       .format(xgb_model.score(X_test[X_train.columns], y_test)))
```

```
C:\Users\kwame.adu\AppData\Local\Continuum\anaconda3\lib\site-packages\xgboost\sklearn.py:1146: UserWarning:

The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].

[20:57:48] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
Accuracy of XGB classifier on training set: 0.80
Accuracy of XGB classifier on test set: 0.79
```

Accuracy of XGB classifier on training set was identified as 80% and 79% on test set.

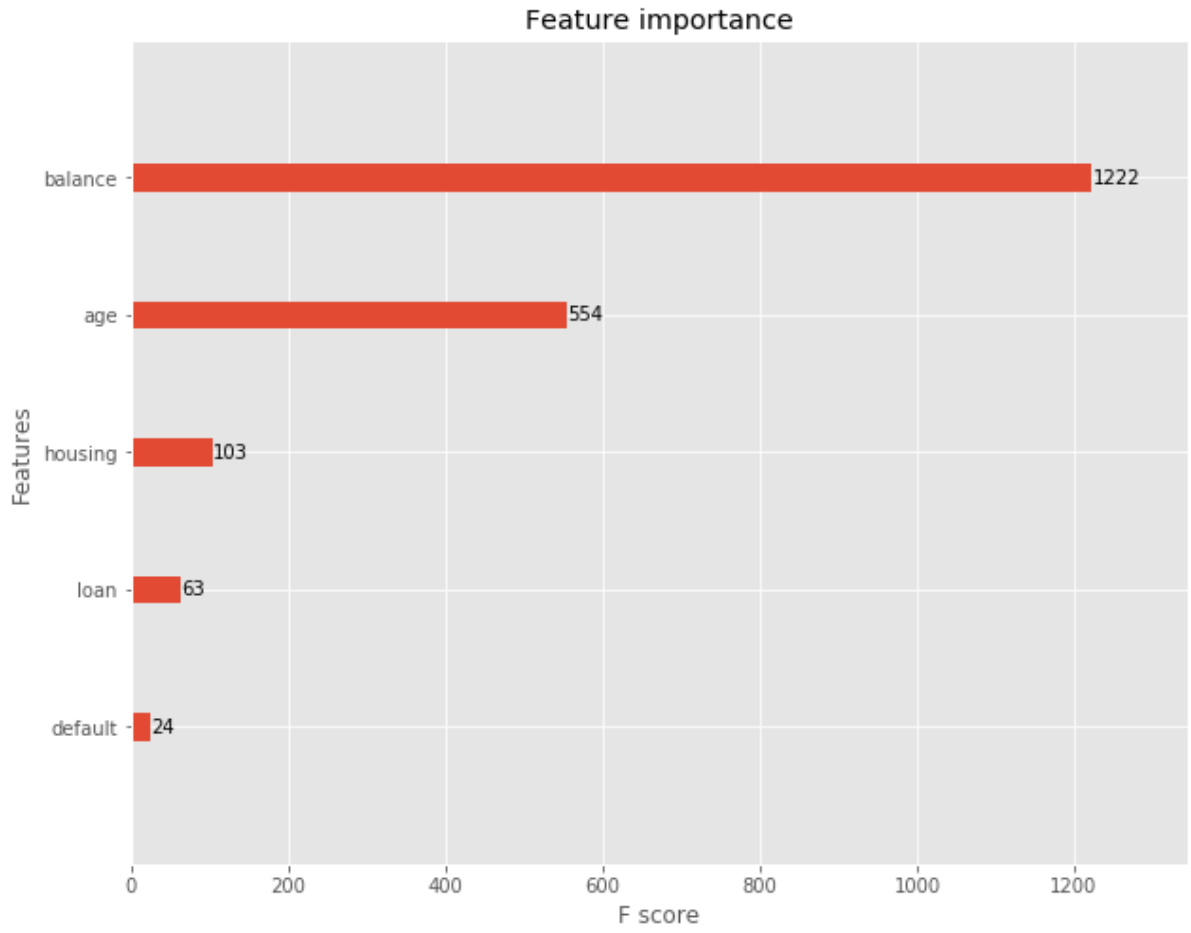
Predicting Recall and Precision

```
#Calculate precision and recall

dtrain = xgb.DMatrix(X_train, label=y_train)
dtest = xgb.DMatrix(X_test, label=y_test)

param = {'max_depth':3, 'eta':1, 'objective':'multi:softprob', 'num_class':5 }
num_round = 2
bst = xgb.train(param, dtrain, num_round)

pred = bst.predict(dtest)
improv_pred = np.asarray([np.argmax(line) for line in pred])

print("Precision = {}".format(precision_score(y_test, improv_pred, average = 'macro')))
print("Recall = {}".format(recall_score(y_test, improv_pred, average = 'macro')))
#print("Accuracy = {}".format(accuracy_score(y_test, improv_pred, average = 'macro')))
```

```
[20:59:49] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
Precision = 0.3962432915921288
Recall = 0.5
```

Printing out feature importance

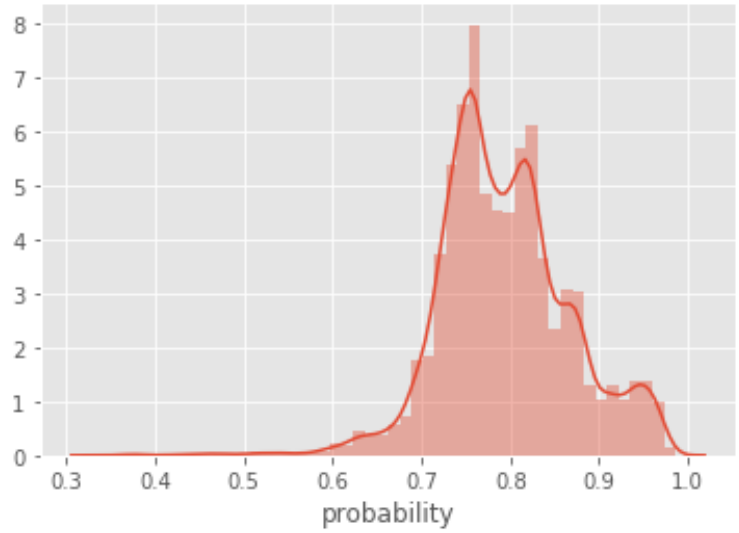
The XGB classifier has an important feature that allows to print out the most important features in the dataset. This aligns with the correlation performed earlier as balance and age had strong positive relationship with churn are ranked here in the important features chart.



Predicting Customers Likely to churn

After pre-processing the data, evaluating the model the script below was run to predict the probability of customers to churn:

```
df['probability'] = xgb_model.predict_proba(df[X_train.columns])[:,1]
df[['customer_id', 'probability']].head(10)
```



Prediction gives us the list of customers with their probability to churn. The table below represents the top ten customers with the highest probability to churn. First customers has a 98% probability of churning and this presents an opportunity for the marketing teams to target this customer with marketing campaigns.

customer_id	probability
8883	0.986498833
2237	0.983321071
7604	0.981517375
3933	0.979828537
4770	0.978136241
6053	0.978136241
4005	0.977509499
4597	0.977509499
4942	0.977509499
6015	0.977509499

Summary

The ability to predict customer churn is critical to businesses staying profitable. According to Harvard Business Review its between 5 to 25 times more expensive to acquire a new customer than retain an existing one. Churn prediction is therefore key for businesses like banks and telecom operators to stay profitable.

In addition to predicting the probability of customers churning, machine learning algorithms are able to predict important features or datasets that can inform marketing teams as to which areas of the business are causing customers to churn. As demonstrated in this project, Banks

can leverage on existing datasets to accurately predict customer churn and also leverage on feature highlight capabilities in classification algorithms like XGBoost to identify features contributing to churn and to what degree.

In addition to predicting churn:

1. Customer complaints and other relevant datasets can be included to improve prediction.
2. This can helps banks and other industries be proactive instead only reacting to churn.
3. Machine learning algorithms can be developed to recommend products (NBA) either for active or customers with high probability to churn.

References

Colgate, M., Stewart, K. & Kinsella, R., 1996. Customer Defection: A study of the student market in Ireland. International Journal of Bank Marketing, 14(3), pp. 23-29.

Ganesh, J., Arnold, M. J. & Reynolds, K. E., 2000. Understanding the Customer Base of Service Providers: An Examination of the Differences between Switchers and Stayers. Journal of Marketing, 64(3), pp. 65-87.

Hall, M. 2020. How the Banking Sector Impacts Our Economy. Retrieved from: https://www.investopedia.com/ask/answers/032315/what-banking-sector.asp

Nie G, Rowe W, Zhang L, Tian Y, Shi Y (2011) Credit card churn forecasting by logistic regression and decision tree. Expert Syst Appl 38:15273–15285

Pham-Gia, T. and Choulakian, V. (2014) Distribution of the Sample Correlation Matrix and Applications. Open Journal of Statistics, 4, 330-344.
http://dx.doi.org/10.4236/ojs.2014.45033

The Economist, 2019. A Whole New World: How technology is driving the evolution of intelligent banking, London: The Economist Intelligence Unit (EIU).

Tsai C-F, Chen M-Y (2010) Variable selection by association rules for customer churn prediction of multimedia on demand. Expert Syst Appl 37:2006–2015

Verbeke W, Martens D, Mues C, Baesens B (2011) Building comprehensible customer churn prediction models with advanced rule induction techniques. Expert Syst Appl 38:2354–2364