EBOOK USE CASE

SHAPELETS

How to identify your **customer churn and predict its probability**

Get ahead by solving Big Data challenges

Contents

Y=sections

The case:
Predict your churn

1. INTRODUCTION

What is included in this eBook and why

Shapelets is a powerful platform that makes data science and data analysis easy and collaborative. Instead of describing our data analytics and visualization processes to you, we'll actually show you.

In this eBook, we provide a simple data analysis solution for behaviour analysis using Shapelets. In particular, we are interested in the factors that influence a bank's customer's decision to churn. Shapelets has helped data scientists solve specific problems many times, and the results have been incredibly successful. Now it is your turn to check how it works.

Who is this eBook for

This eBook was written specifically for data scientists and will be helpful for data engineers, data analysts and business users who are interested in building, deploying and visualizing data models.

Getting started

With this use case, we present a toy example of the construction of a simple data analysis solution aimed towards behaviour analysis using Shapelets. In particular, we are interested in the factors that influence the decision of churning in bank customers.

See the full dataset

Customer churn is one of the most relevant metrics to businesses, informing about how good the company is at retaining customers. In this case study, we aim to obtain insights about the factors involved in the decision of a bank customer to churn and to build accurate, explainable models in order to make predictions about churn that allow us to anticipate churn before it actually happens.

While **this case study focuses on bank customers,** the following approach is **applicable to any other sector** involving customer retention, as long as large customer databases are available. Furthermore, while churning reduction is the objective of this use case, other objectives could be achieved, such as live marketing strategies, obtaining an understanding of customer habits to improve service quality based on demand or reducing product failures based on user profiles.

The use case is based on a dataset containing customer information from **10k anonymized bank customers** which contains 20 features commonly available, involving demographic, customer relationship and transactional information. Some of the customers in this dataset have already churned and this information is used as ground truth to try to figure out what differentiates churning from non-churning customers and build a model that can execute this classification task minimising the classification errors.

Furthermore, the models obtained in this study can provide the churn probability for a given customer. This is great as it allows to prioritise actions on highly probable churning customers, for example providing them with special discounts or promotions.

The use case is organised as follows. First, a **high-level dataset review** is performed to understand the data available and its quality. Then, **an exploratory data analysis (EDA)** is performed in order to quickly discover relevant features or engineer them. Next comes the **data modelling stage**, in which predictive models are built and the performance of the model on new, unseen data is estimated. Finally, we will study the most relevant conclusions from the analysis.

2. THE CHALLENGE

Several challenges arise in this case study, some of which are quite common to many data science studies:

Working with datasets when limited background information is available.
However, this should not be the case in real applications.

Dealing with missing data or in general with datasets that have been produced
without consideration to posterior data analysis processes.

Identify biases in the data that allow distinguishing churning from non-churning customers,
which can be used to filter the relevant features to be used in predictive models.

Understand and choose the right metrics for the specific problem being solved.

Come up with useful insights for the business and help prioritise customer-related activities and their targets.

Understand, select, train and use predictive models efficiently,
maximising their expected performance on the chosen metrics for unseen data.

3. METHODOLOGY

The methodology **to predict churn** is based on three main steps commonly followed in Data Science studies:

A high-level dataset review **to understand the data available and its quality.** Here, three tasks are performed: learning which features and labels are available, discovering missing features and learning about the characteristics of the features to see if they are binary or categorical.

A data modelling stage, in which predictive models are built and the performance of the model on new, unseen data is estimated. In this example, we simply split the data-set into train/test sets to train and evaluate three state-of-the-art models with an arbitrary choice of hyperparameters. A more elaborated approach to guarantee correct model generalisation and to obtain reliable classification metrics would involve considering a validation dataset or using some cross-validation procedure in order to select the best model and its hyperparameters.

An **exploratory data analysis (EDA)** in order to quickly discover important features or engineer them. In this case, we are simply visualising each of the relevant features using the right plot according to their nature in order to learn if there is a bias in that feature when the customer churns.

4. METRICS

Since the problem is posed in the form of a classification problem, the chosen metrics are precision and recall, which are defined next:

Recall or True Positive Rate (TPR) – Number of predicted positives that are actual positives, divided by the number of actual positives.

Precision or Positive Predictive Value (PPV) – Number of predicted positives that are actual positives, divided by the number of predicted positives.

Another relevant metric is the **probability of false alarm or False Positive Rate (FPR) –** The number of false positives divided by the number of negatives.

The recall is more relevant in this case as it penalises the wrong classification of actual positives. A model may consider many or even all the samples as positives and thus obtain a precision as high as desired, but in order to make sure the right customers are addressed, the number of correct guesses should be compared against the number of actual positives. This is exactly what recall does.

The receiver operating characteristics (ROC) curve is another common way of visualising the performance of classification models. It helps visualise the different ways in which a model can be used to provide a more or less conservative behaviour in the predictions, helping to define the right trade-off between the number of predicted positives and the probability of false alarms. This allows you to choose the right model threshold.

Finally, the confusion matrix is a very straightforward way of visualising classification performance once the model threshold has been chosen. It basically summarises the classification responses against the ground truth data.

5. SYNTHESISED RESOLUTION

See here the full resolution in Python

Add the imports for the Shapelets models:

Include the data app basic usage information (login and registration information) and import the data using Pandas and Numpy.

```
# Copyright (c) 2021 Grumpy Cat Software S.L.
#
# This Source Code is licensed under the MIT 2.0 license.
# the terms can be found in LICENSE.md at the root of
# this project, or at http://mozilla.org/MPL/2.0/.

#Shapelets
from shapelets import init_session
from shapelets.dsl.data_app import DataApp

import pandas as pd
import numpy as np
```

```
# Initialize a session and create the DataApp
client = init_session("admin","admin")
app = DataApp(name="bank_customer_churn_prediction",
description="In this dataapp, a study of churn in bank customers is performed.")

Once we have Pandas available, we can read the data from the csv document.

# Read the data
df = pd.read_csv('bank_customer_churn.csv')
```

Once we have Pandas available, we can read the data from the csv document.

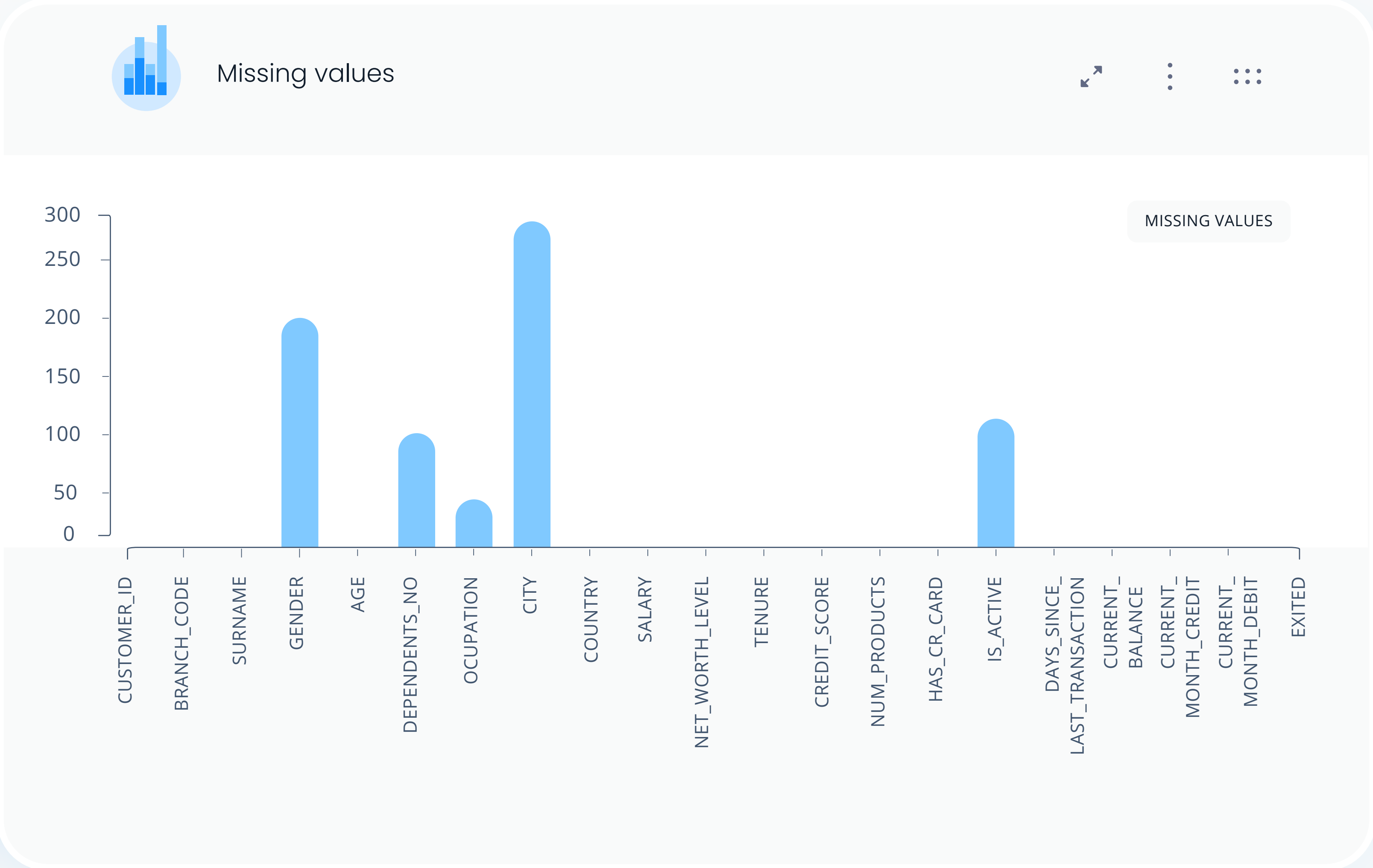
```
# Read the data
df = pd.read_csv('bank_customer_churn.csv')
```

Add the markdown, fill it with the appropriate text and place it in the data app.

```
# Add markdown text
md = app.markdown("""
  # Using Shapelets to prevent bank customer churn

## 1. Introduction
In this study, we aim to accomplish the following tasks:

- Identify and visualize factors contributing to customer churn
- Build a prediction model that will perform the following:

a) Classify whether a customer is going to churn or not\n
b) Obtain churn probability as part of the previous classification, to make it easier for customer service to
target
low-hanging fruits in their efforts to prevent churn\n

## 2. Data set review
In this section we seek to explore the structure of our data, in order to understand the input space the data set
and to prepare the sets for exploratory and prediction tasks.
```

Review the features and labels to identify what attributes will be necessary and what data manipulation needs to be carried out before exploratory analysis and prediction modelling. Include the features available in the data app.

Create a figure and place it in the data app in order to check the missing and unique values (like in the example below). Add markdown text for each figure.



SHAPELETS

Create a chart on the proportion of customers churned and retained, and add the markdown.

```
## 3. Exploratory data analysis (EDA)""")
app.place(md2)

# Plot the proportion of customers retained
labels = 'Exited', 'Retained'
sizes = [df.exited[df['exited']==1].count(), df.exited[df['exited']==0].count()]
explode = (0, 0.1)
fig1, ax1 = plt.subplots(figsize=(12, 6))
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')
plt.title("Proportion of customers churned and retained", size = 20)
img = app.image(fig1)
app.place(img)
```

Check the churning relationship with (1) categorical variables and (2) continuous variables:

```
# Check the churning relationship with categorical variables
fig2, axarr = plt.subplots(2, 2, figsize=(12, 12))
sns.countplot(x='country', hue='exited', data=df, ax=axarr[0][0])
sns.countplot(x='gender', hue='exited', data=df, ax=axarr[0][1])
sns.countplot(x='has_cr_card', hue='exited', data=df, ax=axarr[1][0])
sns.countplot(x='is_active', hue='exited', data=df, ax=axarr[1][1])
img2 = app.image(fig2)
app.place(img2)

# Check the churning relationship with continuous variables
fig3, axarr = plt.subplots(3, 2, figsize=(12, 12))
sns.boxplot(y='age',x = 'exited', hue = 'exited',data = df , ax=axarr[0][0])
sns.boxplot(y='salary',x = 'exited', hue = 'exited',data = df, ax=axarr[0][1])
sns.boxplot(y='tenure',x = 'exited', hue = 'exited',data = df, ax=axarr[1][0])
sns.boxplot(y='credit_score',x = 'exited', hue = 'exited',data = df, ax=axarr[1][1])
sns.boxplot(y='days_since_last_transaction',x = 'exited', hue = 'exited',data = df, ax=axarr[2][0])
sns.boxplot(y='current_balance',x = 'exited', hue = 'exited',data = df, ax=axarr[2][1])
img3 = app.image(fig3)
app.place(img3)
```

Once we have completed these steps, we can train the models. For this case, we will train 3 models and evaluate them. To start with the training, we need first to drop all the nans and data values in the dataframe (split them in train and test data).

```
# Drop infinite values and nans
df.replace([np.inf, -np.inf], np.nan, inplace=True)
df.dropna(inplace=True,how='any')

# Split into train and test data
df_train = df.sample(frac=0.8, random_state=200)
df_test = df.drop(df_train.index)

# Define continuous and categorical variables to be used in the study
continuous_vars = ['credit_score', 'age','tenure','current_balance','num_products','salary','cu-
rrent_month_debit',
                   'current_month_credit']
cat_vars = ['has_cr_card', 'is_active', 'country', 'gender']

# Build the dataframe holding the training data
df_train = df_train[['exited'] + continuous_vars + cat_vars]

# One hot encode the categorical variables
lst = ['country', 'gender', 'has_cr_card', 'is_active']
remove = list()
for i in lst:
   if (df_train[i].dtype == str or df_train[i].dtype == object):
        for j in df_train[i].unique():
            df_train[i+'_'+j] = np.where(df_train[i] == j,1,-1)
        remove.append(i)
df_train = df_train.drop(remove, axis=1)

# Perform MinMax scaling of the continuous variables
minVec = df_train[continuous_vars].min().copy()
maxVec = df_train[continuous_vars].max().copy()
df_train[continuous_vars] = (df_train[continuous_vars]-minVec)/(maxVec-minVec)
```

Place one markdown in the data app to explain the training of the three models. Add the code for the three models:

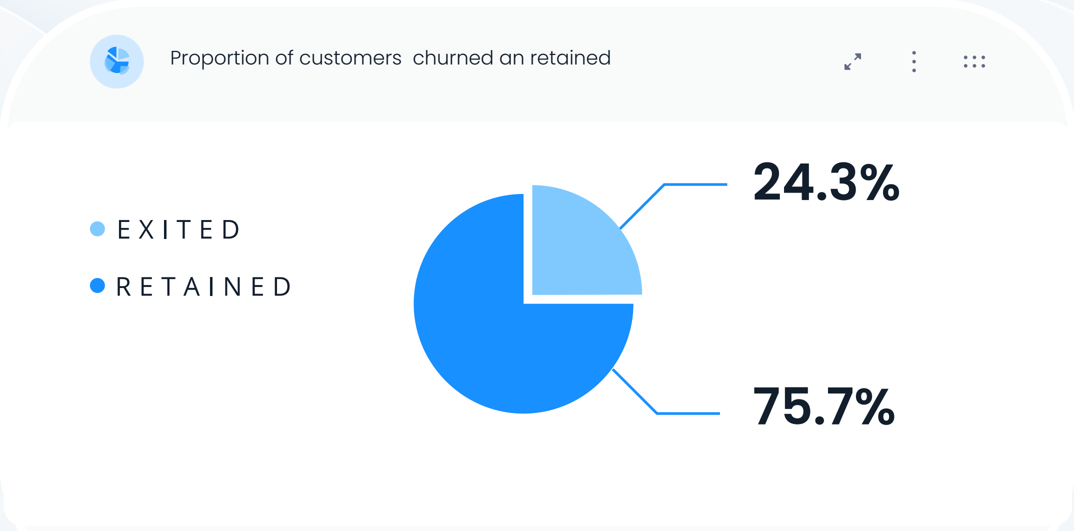
```
   # Add markdown text
md16 = app.markdown('''# 4. Model fitting, model selection and classification results
We will split our dataset into training and test sets (80% - 20% of the data) and train three classification
models:
 - Logistic regression
 - Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel
 - Random forest
The ROC curves corresponding to the training an test results of these three models are shown next. The perfor-
mance '''+
'''of a random model is added for reference.''')
app.place(md16
# Fit primal logistic regression
log_primal = LogisticRegression()
log_primal.fit(df_train.loc[:, df_train.columns ≠ 'exited'],df_train.exited)

# Fit SVM with RBF Kernel
SVM_RBF = SVC(kernel='rbf', probability=True)
SVM_RBF.fit(df_train.loc[:, df_train.columns ≠ 'exited'],df_train.exited)

# Fit Random Forest classifier
RF = RandomForestClassifier(max_depth=8)
RF.fit(df_train.loc[:, df_train.columns ≠ 'exited'],df_train.exited)
```

Define the function in order to get the appropriate scores and plot a ROC curve for the training data

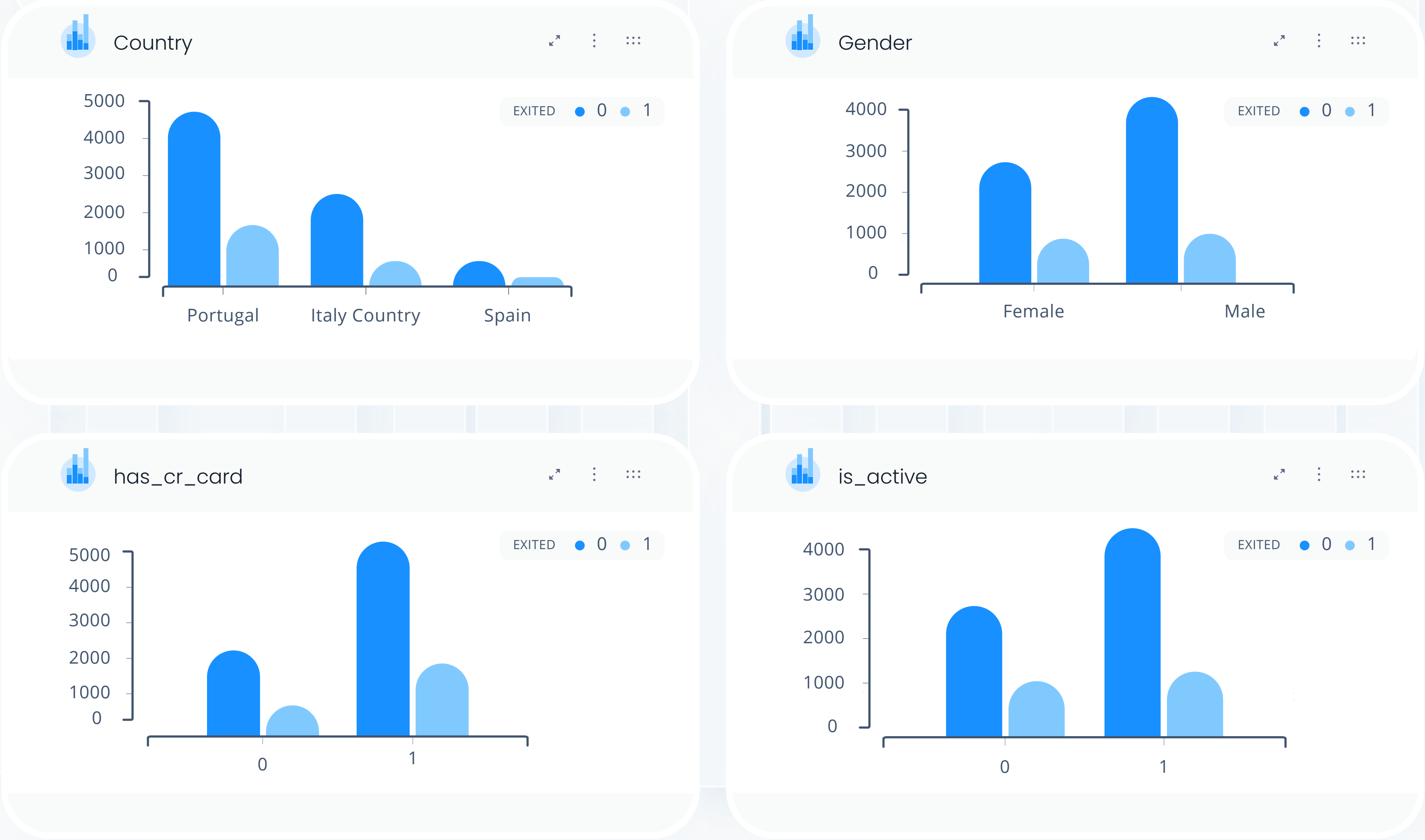
```
# Compute scores
def get_auc_scores(y_actual, method, method2):
    auc_score = roc_auc_score(y_actual, method)
    fpr_df, tpr_df, _ = roc_curve(y_actual, method2)
    return (auc_score, fpr_df, tpr_df)

# Define X and y
y = df_train.exited
X = df_train.loc[:, df_train.columns ≠ 'exited']
```

Once we have these scores for the training data, we can create a graph showing the results of the three models. I applied the whole process to the test data. Please make sure that all of the one hot encoded variables that appear in the train data also appear in the subsequent data. After we run the test data, we will have the final results.

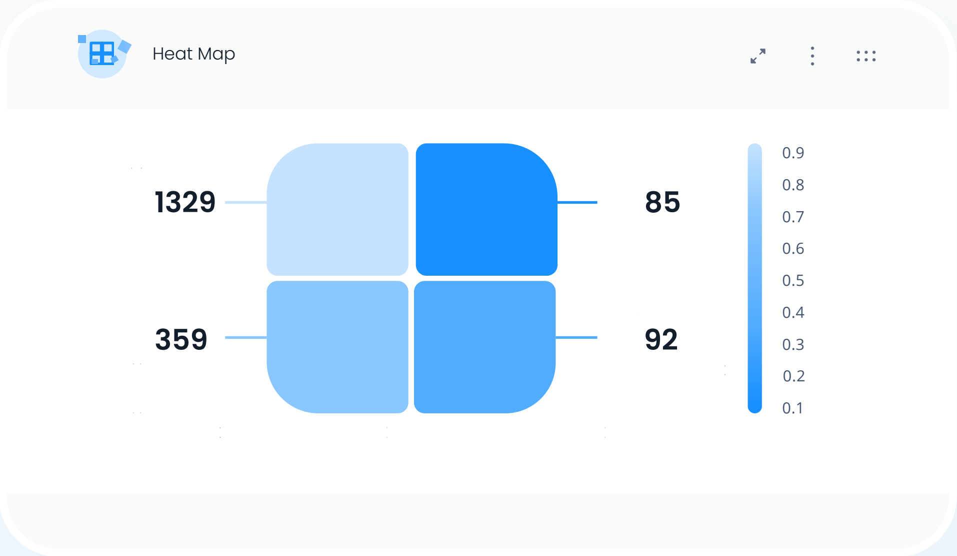
An immediate indicator to obtain from the dataset is that about 24% of the customers have churned. This is a static figure, which could and should be computed periodically to monitor how it is affected by the actions of the company:



Relevant information can be easily concluded just by drawing adequate plots of the available data. In the next figure, for example, it can be quickly deduced that most churning customers have credit cards and that the bank has a very large amount of inactive customers. Again, these metrics could be monitored frequently to try to learn more about their drivers.



As a conclusion of the use case, we can obtain a model capable of classifying any customer, old or new, and providing a probability of churn. With this probability, the customers most likely to churn can be immediately addressed with the right retention strategy in order to revert the possible churn. Of course, the model will make mistakes in its predictions, but overall it does pretty good, as can be observed in the following confusion matrix: when the model predicts that a customer will not churn it only gets it wrong in 7% of customers, and it is already able to remove more than 70% of the customers from the analysis, letting the company focus in those more likely to churn.



```
# Compute and plot confusion matrix
pred_val = RF.predict(df_test.loc[:, df_test.columns ≠ 'exited'])
cm = confusion_matrix(df_test.exited,pred_val)
fig8 = plt.figure(figsize=[8, 8])
norm_cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
sns.heatmap(norm_cm, annot=cm, fmt='g', xticklabels
=['Predicted: No','Predicted: Yes'],
            yticklabels=['Actual: No','Actual: Yes'], cmap='bone')
img8 = app.image(fig8)
app.place(img8)
```

With Shapelets relevant metrics/KPIs can be monitored frequently and insights like the aforementioned ones can be instantly and seamlessly shared from the data scientist to all relevant departments in the organisation.

6. RESULTS

Several interesting results arise from this study:

The first result that is obtained is probably already available since it is quite straightforward to obtain: the churning rate. **In this example, about 24% of the customers have churned.**

One can discover issues in international business branches, by comparing the churning ratios across countries. In this case, **the churning ratio remains constant across countries.**

Gender appears to be a relevant feature to churning. **The proportion of female customers churning is greater than that of male customers**, but overall, most churning customers are male.

The overall proportion of inactive members is quite high suggesting that the bank may need a program implemented to turn this group into active customers.

Customers with extreme salaries churn more.

With regard to tenure, churning is less common on customers that have been with the bank for several years. An effort in retention during the first 2-3 years could reduce churning.

The random forest appears to be a good model for this classification problem. However, the use of validation techniques is recommended in order to select the best type of model and its hyperparameters.

The best model obtained the following metrics: a precision of 50% (half of the predicted churning customers actually churn), a recall of around 21% (this fraction of the churning customers can be correctly classified) and a false positive rate of 7% is obtained (7% of the customers that the model believes will not churn actually do churn).

7. HOW DOES SHAPELETS HELP SOLVE THIS CHALLENGE?

Shapelets is great for solving data science and data analysis problems and for easily sharing across the organisation the solutions produced. The access to databases and distributed processing is immediate, seamless and fully scalable. No skills in web development or development operations are needed in order to come up with fully-featured data apps and to effortlessly share them across the organisation. For building use cases, the user does not need to learn new ways to solve data science problems, since Shapelets relies on several native tools commonly used by data scientists. In this particular use case, we rely on matplotlib and seaborn for visualisation and scikit learn for machine learning.

8. CONCLUSION

Find powerful insights in your big data.

Shapelets platform was designed to make your insights even more powerful and help you solve problems quickly. It was built for you and your data science team, giving you more collaboration, more analytics, and a faster way to solve problems unique to your business. We hope you found it useful and will try using Shapelets on your business projects.

If you need more info about this use case or Shapelets, please

CONTACT US

Adrián **Carrio**

LEAD DATA SCIENTIST

9. ABOUT THE AUTHOR

Adrián is the Lead Data Scientist at Shapelets. He received his degree in Industrial Engineering from the University of Oviedo and his PhD in Automation and Robotics (Cum Laude) from the Technical University of Madrid. Previously, he was a researcher in Arizona State University and the Massachusetts Institute of Technology (MIT) and has published more than 30 scientific publications and one patent.
He combines strong expertise in Data Analysis, Machine Learning and Pattern Recognition and deploying these technologies in a variety of industrial settings.
Furthermore, he has worked on numerous technology transfer projects using AI systems in various sectors for companies such as Arcelor Mittal and Airbus. He has also co-founded ThermoHuman (thermography for health and sports) and Dronomy (autonomous drones).

You can find more articles by Adrián in our Blog, and you can also follow him on LinkedIn.

10. ABOUT SHAPELETS

Shapelets is a data and software company. Our goal is to disrupt the Big Data analysis ecosystem. Starting with a groundbreaking analysis platform, with an open-source philosophy. We offer a comprehensive, flexible and open environment to manage the behaviour in data on the move with academically produced groundbreaking algorithms, state of the art software, and cutting-edge data apps. Our improved data visualization tools help empower data scientists to communicate with all organisations' stakeholders.

The innovation of Shapelets is based on being a solution with an open-source part, focused on the processing and analysis of platform data, integrable with any type of storage platform, which automates machine learning functions through machine learning algorithms and where all algorithms are optimized for the available resources (CPU-GPU-Multicore). To learn more, follow Shapelets on Twitter, LinkedIn, GitHub and YouTube.

EBOOK USE CASE

SHAPELETS

shapelets.io