Why use ML for marketing? Strategies and use cases

MACHINE LEARNING FOR MARKETING IN PYTHON

Karolis Urbonas
Head of Analytics & Science, Amazon

datacamp

Types of machine learning

- Supervised learning
  - Given X, can we predict Y?
  - **Classification** - when Y is categorical (e.g. Churned/Not-churned, Yes/No, Fish/Dog/Cat).
  - **Regression** - when Y is continuous (e.g. Purchases, Clicks, Time Spent on Website).
- Unsupervised learning
  - Given X, can we detect patterns and clusters that are homogenous?
- Reinforcement learning
  - Given a current state and a number potential actions, which path maximizes the reward?

Supervised learning data parts and steps

1. Define the target (dependent variable or **Y**) - what do we want to predict?
   - **Example 1** - which customers will churn? [**CLASSIFICATION**]
   - **Example 2** - which customers will buy again? [**CLASSIFICATION**]
   - **Example 3** - how much will customers spend in the next 30 days? [**REGRESSION**]
2. Collect features (independent variables or **X**) which could have predictive power:
   - **Example 1** - Purchase patterns prior churning.
   - **Example 2** - Number of missed loan payments prior defaulting on a loan.

Supervised learning data format

- X by N+1 matrix:
  - X number of observations (customer, vendor, product)
  - N + 1 number of columns (N features + 1 target variable)

Feature 1	Feature 2	...	Feature N	Target Y
11	21		N1	Y1
12	22		N2	Y2

Unsupervised learning

Collect usage or purchase data and run model to identify homogenous groups i.e. clusters or segments of data:

- Example 1 - customer segmentation by their product purchases.
- Example 2 - product segmentation for bundling

Unsupervised learning data format

- X by N matrix:
  - X number of observations (customer, vendor, product)
  - N number of columns (N features)
- A list of independent variables (features) as separate columns for each observation

Feature 1	Feature 2	...	Feature N
11	21		N1
12	22		N2

Let's practice!

MACHINE LEARNING FOR MARKETING IN PYTHON

Preparation for modeling

MACHINE LEARNING FOR MARKETING IN PYTHON

Karolis Urbonas
Head of Analytics & Science, Amazon

Data sample

```
telco_raw.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSup
0	7590-VHVEG	Female	No	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	No	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	No	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	No	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	No	No	No	2	Yes	No	Fiber optic	No	...	No	

5 rows × 21 columns

Data types

```
telco_raw.dtypes
```

```
customerID          object
gender              object
SeniorCitizen       object
Partner             object
Dependents          object
tenure               int64
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
TotalCharges       float64
Churn               object
```

Separate categorical and numerical columns

Separate the identifier and target variable names as lists

```
custid = ['customerID']
target = ['Churn']
```

Separate categorical and numeric column names as lists

```
categorical = telco_raw.nunique()[telcom.nunique()<10].keys().tolist()
categorical.remove(target[0])
numerical = [col for col in telco_raw.columns
                    if col not in custid+target+categorical]
```

One-hot encoding

This is a typical categorical data type column

Color
Red
White
Blue
Red

One-hot encoding result

And this is how it looks when we transform it with one-hot encoding.

Color		Red	White	Blue
Red	---------->	1	0	0
White	---------->	0	1	0
Blue	---------->	0	0	1
Red	---------->	1	0	0

One-hot encoding categorical variables

One-hot encoding categorical variables

```
telco_raw = pd.get_dummies(data=telco_raw, columns=categorical, drop_first=True)
```

Scaling numerical features

```
# Import StandardScaler library
from sklearn.preprocessing import StandardScaler
# Initialize StandardScaler instance
scaler = StandardScaler()
# Fit the scaler to numerical columns
scaled_numerical = scaler.fit_transform(telco_raw[numerical])
# Build a DataFrame
scaled_numerical = pd.DataFrame(scaled_numerical, columns=numerical)
```

Bringing it all together

```
# Drop non-scaled numerical columns
telco_raw = telco_raw.drop(columns=numerical, axis=1)
# Merge the non-numerical with the scaled numerical data
telco = telco_raw.merge(right=scaled_numerical,
                        how='left',
                        left_index=True,
                        right_index=True
                        )
```

Let's practice pre-processing data!

MACHINE LEARNING FOR MARKETING IN PYTHON

ML modeling steps

MACHINE LEARNING FOR MARKETING IN PYTHON

Karolis Urbonas
Head of Analytics & Science, Amazon

Supervised learning steps

1. **Split** data to training and testing
2. **Initialize** the model
3. **Fit** the model on the training data
4. **Predict** values on the testing data
5. **Measure** model performance on testing data

Supervised learning with code

First, let's load the libraries.

```
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

Supervised learning steps with code

```
# 1. Split data to training and testing
train_X, test_X, train_Y, test_Y = train_test_split(X, Y, test_size=0.25)
# 2. Initialize the model
mytree = tree.DecisionTreeClassifier()
# 3. Fit the model on the training data
treemodel = mytree.fit(train_X, train_Y)
# 4. Predict values on the testing data
pred_Y = treemodel.predict(test_X)
# 5. Measure model performance on testing data
accuracy_score(test_Y, pred_Y)
```

Unsupervised learning steps

1. **Initialize** the model
2. **Fit** the model
3. **Assign** cluster values
4. **Explore** results

Unsupervised learning with code

First, let's load the libraries.

```
from sklearn.cluster import KMeans
import pandas as pd
```

Unsupervised learning with code

```
1. Initialize the model
kmeans = KMeans(n_clusters=3)
# Fit the model
kmeans.fit(data)
# 3. Asign cluster values
data.assign(Cluster=kmeans.labels_)
# 4. Explore results
data.groupby('Cluster').mean()
```

Let's go build some models!

MACHINE LEARNING FOR MARKETING IN PYTHON