MODULE 6: PREDICTIVE MODELING FOR TEMPORARY DATA

CASE STUDY ACTIVITY TUTORIAL

6.2 PREDICTION ENGINEERING USING UK RETAIL DATASET

UK Retail Dataset Case Study

October 16, 2017

1 Prediction engineering case study using UK Retail Dataset

In this case study, we will study prediction engineering. Prediction engineering is a step in predictive modeling, where we: * Define an outcome we are interested in predicting * Scan the data to find the past occurrences of the outcome * These past occurrences become training examples for ma-chine learning/modeling * We will then use featuretools to extract features and learn a predictive model.

In this particular case study, we are focusing a retail dataset openly available at

We will define the prediction problem as the one where the customer has more than `k` purchases

```
In [113]: import featuretools as ft
          from utils import (find_training_examples, load_uk_retail_data,
                             engineer_features_uk_retail, preview)
          import pandas as pd
          import numpy as np
          from sklearn.model_selection import train_test_split
          from sklearn.preprocessing import Imputer
          from sklearn.ensemble import RandomForestClassifier
          from sklearn.metrics import precision_recall_fscore_support, confusion_matrix
          ft.__version__
          %load_ext autoreload
          %autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

2 Step 1: Load and prepare data

```
In [24]: item_purchases, invoices, items,customers = load_uk_retail_data()
```

The dataset has the following tables: * `item_purchases` * `invoices` * `items` * `customers`

The following relations exist * A customer may have multiple invoices * An item may have been purchased multiple times * An invoice may have multiple item purchases

```
In [25]: entities = {
             "item_purchases": (item_purchases, "item_purchase_id", "InvoiceDate" ),
             "items": (items, "StockCode"),
             "customers": (customers,"CustomerID"),
             "invoices":(invoices,"InvoiceNo","first_item_purchases_time")
             }

         relationships = [("customers", "CustomerID","invoices", "CustomerID"),
                          ("invoices", "InvoiceNo","item_purchases", "InvoiceNo"),
                          ("items", "StockCode","item_purchases", "StockCode")]
```

3 Step 2 : Find training examples

In the code snippet below, we are trying to find training examples from the data. We set the following parameters: * `prediction_window`=14 days * `training_window`=21 days * `lead` = 7 days * `threshold`=2 --> specifies the number of purchases that the customer need to have in the future to be considered engaged

```
In [105]: label_times = find_training_examples(item_purchases, invoices,
                                               prediction_window=pd.Timedelta("14d"),
                                               training_window=pd.Timedelta("21d"),
                                               lead=pd.Timedelta("7d"),
                                               threshold=5)
```

```
In [123]: preview(label_times,5)
```

```
Out[123]:      CustomerID    t_start cutoff_time  purchases>threshold
          0       17505.0 2011-05-18  2011-06-08                False
          516     16444.0 2011-05-18  2011-06-08                False
          517     16889.0 2011-05-18  2011-06-08                False
          518     17613.0 2011-05-18  2011-06-08                 True
          519     17152.0 2011-05-18  2011-06-08                False
```

In the output above, we are showing the first 5 training examples. The first column is the CustomerID, the second column is the timestamp after which we can use the data for generating features. The third column is the last timestamp we can use the data from the customer. The fourth column is the label. It is `True` if the customer had more than 5 purchases in the period between (`cutoff_time+lead`, `cutoff_time+lead+prediction_window`)

4 Step 3: Now lets generate features.

Next we generate features for each of the training examples. We use featuretools to generate the features. Featuretools is an automated feature engineering software. We go into detail about this software package in the NYC-Taxi case study. Here we simply use the tool to generate features.

```
In [107]: feature_matrix=engineer_features_uk_retail(entities,relationships,
                                                     label_times,training_window='21d')
```

In [122]: preview(feature_matrix,10)

```
Out[122]:             WEEK(first_invoices_time)  HOUR(first_invoices_time)  \
          CustomerID
          12353.0                            20                         17
          12359.0                             2                         12
          12360.0                            21                          9
          12380.0                            23                          9
          12415.0                             1                         11
          12417.0                            50                         11
          12423.0                            51                         10
          12426.0                            21                         12
          12431.0                            48                         10
          12437.0                             2                         14

                      MAX(item_purchases.Quantity)  STD(item_purchases.UnitPrice)  \
          CustomerID
          12353.0                              NaN                            NaN
          12359.0                              NaN                            NaN
          12360.0                              NaN                            NaN
          12380.0                              NaN                            NaN
          12415.0                            600.0                       2.367284
          12417.0                             24.0                       5.565414
          12423.0                              NaN                            NaN
          12426.0                              NaN                            NaN
          12431.0                             24.0                       2.658325
          12437.0                             48.0                       6.323762

                      DAY(first_invoices_time)  IS_WEEKEND(first_invoices_time)  \
          CustomerID
          12353.0                           19                            False
          12359.0                           12                            False
          12360.0                           23                            False
          12380.0                            7                            False
          12415.0                            6                            False
          12417.0                           17                            False
          12423.0                           21                            False
          12426.0                           29                             True
          12431.0                            1                            False
          12437.0                           12                            False

                      MINUTE(first_invoices_time)  MONTH(first_invoices_time)  \
          CustomerID
          12353.0                              47                           5
          12359.0                              43                           1
          12360.0                              43                           5
          12380.0                              49                           6
          12415.0                              12                           1
```

```
12417.0                                 51                              12
12423.0                                 54                              12
12426.0                                 26                               5
12431.0                                  3                              12
12437.0                                 13                               1

            MAX(item_purchases.UnitPrice)  MEAN(item_purchases.Quantity)  \
CustomerID
12353.0                               NaN                            NaN
12359.0                               NaN                            NaN
12360.0                               NaN                            NaN
12380.0                               NaN                            NaN
12415.0                             12.50                     110.378378
12417.0                             28.00                      11.608696
12423.0                               NaN                            NaN
12426.0                               NaN                            NaN
12431.0                              7.95                       8.000000
12437.0                             18.00                      18.375000

                          ...                        \
CustomerID                ...
12353.0                   ...
12359.0                   ...
12360.0                   ...
12380.0                   ...
12415.0                   ...
12417.0                   ...
12423.0                   ...
12426.0                   ...
12431.0                   ...
12437.0                   ...

            MAX(invoices.STD(item_purchases.UnitPrice))  \
CustomerID
12353.0                                             NaN
12359.0                                             NaN
12360.0                                             NaN
12380.0                                             NaN
12415.0                                        2.376640
12417.0                                        5.565414
12423.0                                             NaN
12426.0                                             NaN
12431.0                                        2.658325
12437.0                                        6.323762

            STD(invoices.MAX(item_purchases.Quantity))  \
CustomerID
12353.0                                            NaN
```

```
12359.0                                                    NaN
12360.0                                                    NaN
12380.0                                                    NaN
12415.0                                                  350.0
12417.0                                                    0.0
12423.0                                                    NaN
12426.0                                                    NaN
12431.0                                                    0.0
12437.0                                                    0.0

            MEAN(invoices.STD(item_purchases.UnitPrice))  \
CustomerID
12353.0                                              NaN
12359.0                                              NaN
12360.0                                              NaN
12380.0                                              NaN
12415.0                                         1.188320
12417.0                                         5.565414
12423.0                                              NaN
12426.0                                              NaN
12431.0                                         2.658325
12437.0                                         6.323762

            MAX(invoices.MEAN(item_purchases.Quantity))  \
CustomerID
12353.0                                             NaN
12359.0                                             NaN
12360.0                                             NaN
12380.0                                             NaN
12415.0                                      113.260274
12417.0                                       11.608696
12423.0                                             NaN
12426.0                                             NaN
12431.0                                        8.000000
12437.0                                       18.375000

            MAX(invoices.STD(item_purchases.Quantity))  \
CustomerID
12353.0                                            NaN
12359.0                                            NaN
12360.0                                            NaN
12380.0                                            NaN
12415.0                                     131.485301
12417.0                                       6.761394
12423.0                                            NaN
12426.0                                            NaN
12431.0                                       6.947004
12437.0                                      14.247259
```

```
            MEAN(invoices.MAX(item_purchases.UnitPrice))  \
CustomerID
12353.0                                              NaN
12359.0                                              NaN
12360.0                                              NaN
12380.0                                              NaN
12415.0                                            8.375
12417.0                                           28.000
12423.0                                              NaN
12426.0                                              NaN
12431.0                                            7.950
12437.0                                           18.000

            MEAN(invoices.MAX(item_purchases.Quantity))  \
CustomerID
12353.0                                             NaN
12359.0                                             NaN
12360.0                                             NaN
12380.0                                             NaN
12415.0                                           250.0
12417.0                                            24.0
12423.0                                             NaN
12426.0                                             NaN
12431.0                                            24.0
12437.0                                            48.0

            MEAN(invoices.MEAN(item_purchases.Quantity))  \
CustomerID
12353.0                                              NaN
12359.0                                              NaN
12360.0                                              NaN
12380.0                                              NaN
12415.0                                         6.630137
12417.0                                        11.608696
12423.0                                              NaN
12426.0                                              NaN
12431.0                                         8.000000
12437.0                                        18.375000

            STD(invoices.MAX(item_purchases.UnitPrice))  \
CustomerID
12353.0                                             NaN
12359.0                                             NaN
12360.0                                             NaN
12380.0                                             NaN
12415.0                                           4.125
12417.0                                           0.000
```

```
12423.0                                                      NaN
12426.0                                                      NaN
12431.0                                                    0.000
12437.0                                                    0.000

            STD(invoices.MEAN(item_purchases.UnitPrice))
CustomerID
12353.0                                                      NaN
12359.0                                                      NaN
12360.0                                                      NaN
12380.0                                                      NaN
12415.0                                                 0.773973
12417.0                                                 0.000000
12423.0                                                      NaN
12426.0                                                      NaN
12431.0                                                 0.000000
12437.0                                                 0.000000

[10 rows x 27 columns]
```

5 Step 4: Let's train a model using Random Forests

Now we are ready to train a model and evaluate it. To do this, we: * First split our training examples in train_test_split * Impute missing values * Train a model using training data * Test on the data set aside for testing

We can split the data using the function `train_test_split` and specifying the proportion we want for testing. In this case we specified that as 35%

```
In [109]: y=label_times['purchases>threshold']
          X_train, X_test, y_train, y_test = train_test_split(feature_matrix,
                                                             y, test_size=0.35)
```

We can impute the missing values or `NaN` values in the feature_matrix using the `Imputer` in scikit-learn. It replaces the `NaN` values in a feature column with the `mean` of the rest of the entries in that column. This is a simple imputation startegy

```
In [116]: imp = Imputer(missing_values='NaN', strategy='mean', axis=0)
          imp = imp.fit(X_train)
          X_train_imp = imp.transform(X_train)
```

We can train a RandomForest classifier (a type of ensemble classifier). We make use of scikit-learn package for this as well.

```
In [117]: clf = RandomForestClassifier(random_state=0,n_estimators=10,
                                       class_weight="balanced",verbose=True)
          clf.fit(X_train_imp, y_train)

[Parallel(n_jobs=1)]: Done  10 out of  10 | elapsed:    0.0s finished
```

```
Out[117]: RandomForestClassifier(bootstrap=True, class_weight='balanced',
                      criterion='gini', max_depth=None, max_features='auto',
                      max_leaf_nodes=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=10, n_jobs=1, oob_score=False, random_state=0,
                      verbose=True, warm_start=False)
```

6 Step 5: Test the model

To test a model, we: * First impute the missing values * Use the trained classifier to predict the labels

```
In [118]: X_test_imp = imp.transform(X_test)
          predicted_labels = clf.predict(X_test_imp)
```

```
[Parallel(n_jobs=1)]: Done  10 out of  10 | elapsed:    0.0s finished
```

We evaluate by calculatin

```
In [120]: tn, fp, fn, tp = confusion_matrix(y_test, predicted_labels).ravel()
```

```
In [121]: tp,fp
```

```
Out[121]: (0, 10)
```