Instacart Market Basket Analysis

PREDICT NEXT BASKET

Prajakta Gujarathi | Capstone Project | 10/5/2017

Definition

PROJECT OVERVIEW:

Instacart, a grocery ordering and delivery app, aims to make it easy to fill customer's refrigerator and pantry with their personal favorites and staples when they need them. After selecting products through the Instacart app, personal shoppers review order and do the in-store shopping and delivery for customers. But achieving this simplicity cost effectively at scale requires an enormous investment in engineering and data science.

Knowing customer's next order will be helpful for Instacart business in following way:

- Optimize algorithm which enroute Instacart shoppers, for timely and efficient delivery
- Balancing supply and demand of customers. This includes estimating Instacart's capacity to fulfill orders to create optimal customer experience.
- Provide customers with recommendation by analyzing their previous purchase history.

Below is the paper that I found on internet on similar lines, where next order is predicated given the historical data of customers:

http://www.bigdatalab.ac.cn/~junxu/publications/SIGIR2015_NextBasketRec.pdf

Data for Instacart market basket analysis problem can be found at:

https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2

PROBLEM STATMENT

In this project goal is to use previous transactional data of customer to develop models that predict which products a user will buy again. Task involve are following:

1. Download data from Recently, Instacart open source data.
2. Explore, visualize and analyze important dimensions
3. Preprocess data by adding or reducing dimension as required.
4. Train a different models using preprocessed data and identify best model
5. Predict next carts products for test users.
6. Use test data to analyze final model performance.

I consider this as classic classification problem. Where given order product is pair we predict if is present class 1 or if it is absent class 0.

Datasets and Inputs:

Input for this problem are going to be features such as User based Features, product based features and user product based feature which are either manually created or from given data set.

- Users Related features:
  1. User total orders
  2. Users total items
  3. Total distinct items
  4. Users average days between the orders
  5. Users average basket size
- Product Related features:
  6. Aisle ID
  7. Department ID
  8. Total order count of products
  9. Total reorder count of products
  10. Reorder rate
- UserXProduct Related features:
  11. User number of order per products
  12. Users order rate
  13. User product last order ID
  14. User product average position in cart
  15. Users product reorder rate
  16. User product day since last order
  17. User product order same day as last

Output will be ordered followed by list of product that we are going to predict that will be present in next order.

Solution Statement:

- PREPROCESSING AND DATA PREPARATION:

  In this step I merge data from Orders.csv, Orders_product_prior.csv, order_products_train.csv, aisle.csv and department.csv.

  Create few new features that I feel are important like reorder ratio, user order ratio, average numbers orders in a cart etc.

- MODEL CREATION:

  In this step I applied various supervised learning models on preprocessed data like Logistic regression, Adaboosting, Xgboost and light gradient boosting etc

  Using Grid Search tuned the model parameters. Select the best model with max F1 score of prediction.

Benchmark Model

Looking at the distribution of classes (Order product pair present in next order vs not present) it's clear most products are not present in next orders. This can greatly affect accuracy, since we could simply say "product not present in next order" and generally be right, without ever looking at the data! Making such a statement would be called naive, since we have not considered any information to substantiate the claim. So I use naïve predictor as my Baseline Model. I am considering ordered productid pair absent that is class 0 for all the data. Baseline model is implemented in baselinemodel.py as Naïve Predictor.

Code:

```
dtrain_predictions = [0] * df_train.shape[0]
TP = label.count() - np.sum(label)  # Counting the ones as this is the naive case. Note that
'income' is the 'income_raw' data
# encoded to numerical values done in the data preprocessing step.
FP = label.count() - TP  # Specific to the naive case

TN = 0  # No predicted negatives in the naive case
FN = 0  # No predicted False negatives in the naive case

accuracy = float(TP) / float(label.count())
recall = 1
precision = accuracy
beta = 0.5
```

```
fscore = (1 + 0.25) * (precision * recall) / ((0.25 * precision) + recall)

# Print the results
print "Naive Predictor: [Accuracy score: {:.4f}, F-score: {:.4f}]".format(accuracy, fscore)
```

>> **Naive Predictor: [Accuracy score: 0.9022, F-score: 0.9202]**

Evaluation Metrics:

F1 is a common metric for binary classifiers; it takes into account both true positives and true negatives with equal weight.

$$PRE = \frac{TP}{TP + FP}$$

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$$

$$F_1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC}$$

TP = will be product which are actually present in next order.

FP = will be product that incorrectly predicated to be in next order.

TN = will be product which are not present in next orders.

FN = will be products which are supposed to be in next order be predicated as not.

The problem here is to approximate P(u,p | user's prior purchase history) which stands for how much likely user u would repurchase product p given prior purchase history.

F1 score is optimal choice for this problem as F1 actually doesn't weight TP and TN equally. It disregards true negatives because they're likely to be far more common like since we could simply say "product not present in next order" and generally be right, without ever looking at the data, and thus not as critical to get right.

So my main model is a binary classifier. Features are created manually or automatically are fed to the classifier which will predict that given orderId ProductId pair will be purchased 1 or not 0.

Project Design:

- ### EXPLORATORY DATA ANALYST:

  Perform EDA on data in order to find out:

  Important features
  How the data distribution
  Handling missing data
  Calculated statistics relevant to the problem

- ### PREPROCESSING AND DATA PREPARATION:

  In this step I merge data from Orders.csv, Orders_product_prior.csv, order_products_train.csv, aisle.csv and department.csv.

  #### Handling missing data:

  Thankfully the data obtained it clean data with no missing values.

  Create few new features that I feel are important like reorder ratio, user order ratio, average numbers orders in a cart etc.

- ### MODEL CREATION:

  In this step I applied various model on preprocessed data like Logistic regression, Adaboosting, Xgboost and light gradient boosting etc

  Using Grid Search tuned the model parameters.

- ### MODEL EVALUATION:

  For model evaluation I am going to used f1 score used for binary classification.

- ### PREDICT TEST DATA VALUES:

  Final best model is then used to predict next orders of Test data. Prediction are saved in format order_id followed by Product_ids present in that order.