Domain Background

Recommender systems have dramatically increased in prevalence over the last 2 decades. From e-commerce where items we are likely to purchase are presented to personalized music and video suggestions, recommender systems play an increasingly important role in providing personalized experiences to customers.

Customer expectations that their data will be used in ways that provides them benefits are higher than ever before, increasing the importance of having recommender systems that provide accurate and timely personalized content.

Providing accurate, personalized recommendations has been a very active area of research over the last 10 years. The "Netflix Prize" of 2009[1] led to innovative algorithms being developed based on Collaborative Filtering and Matrix Factorization techniques. In recent years, advances in Deep Learning techniques are increasingly being applied to recommender systems, for example, Amazon has recently launched a personalization product that offers automated algorithms to create personalized recommendations based on Hierarchical Neural Networks[2]. Google have also created successful Deep Learning models for recommendation engines based on a 'Wide and Deep' Neural Network architecture[3].

The objective of this project is to develop a recommender system for a grocery store. Recommender systems have several practical applications in this sector including:

1.) Offering personalized prices, rewards and discounts
2.) Automated list building based on the predicted contents of future trips
3.) Providing 'healthy' suggested substitutes for commonly purchased items to support customer health goals
4.) Reminding customers if they may have forgotten an item when shopping on-line or highlighting substitutes that may be on sale

I have worked in the grocery industry as a Data Scientist for 11 years and am personally motivated to explore cutting edge recommender algorithms for personalization.

Problem Statement

The goal of this project is to create a recommender system based on a combination of Collaborative Filtering and Classification algorithms that can predict the probability that a customer will purchase an item within a specified timeframe by utilizing data on their past purchasing behavior. The output of the algorithm will be a ranked list of items that the customer is predicted to purchase with an associated probability.

Datasets and Inputs

In order to build the algorithms open source data provided by Dunnhumby utilized. The data can be located here:

https://www.dunnhumby.com/careers/engineering/sourcefiles

Specifically the “let’s get sort-of-real” dataset will be used. This dataset contains all transactions and all items purchased for a randomly selected 5,000 customers over 117 weeks. One dataset is provided for each of the 117 weeks with each week containing the columns:

SHOP_WEEK: The week of the transaction
SHOP_DATE: The date of the transaction
SHOP_WEEKDAY: The day of the week of the transaction
SHOP_HOUR: The hour of the day of the transaction
QUANTITY: The number of units purchased of an item
SPEND: The total spend on the item
PROD_CODE: The unique product code for the item
PROD_CODE_10: Product sub-class
PROD_CODE_20: Product class
PROD_CODE_30: Product category
PROD_CODE_40: Product department
CUST_CODE: The unique identifier for a customer
CUST_PRICE_SENSITIVITY: The Price Sensitivity segment assigned to the customer
CUST_LIFESTAGE: The lifestage the customer belongs to
BASKET_ID: The unique identifier for an individual transaction
BASKET_PRICE_SENSITIVITY: The Price Sensitivity segment assigned to the basket
BASKET_TYPE: Designates a transaction as ‘Full Shop’, ‘Top Up’, ‘Small Shop’ or unknown
BASKET_DOMAIN_MISSION: Designates the transaction as ‘Fresh’, ‘Grocery’, ‘Mixed’ or ‘Non-Food’
STORE_CODE: The unique identifier for a store
STORE_FORMAT: The format of the store (‘LS’, ‘MS’, ‘SS’ or ‘XLS’)
STORE_REGION: The region of the store

Using the data several features will be created as inputs into the model, these will include:

- The number of times the customer has purchased the item and their total spend on the item
- The time since the customer last purchased the item
- The average time between purchases of the item where the customer has purchased the item more than once
- Total spend and visits by sub-class, class, category and department
- Total spend and visits by the basket type and basket dominant mission segments

Solution Statement

In order to develop predictive algorithms and test their performance the data will be split into ‘build’ and ‘predict’ datasets (illustrated in figure 1).

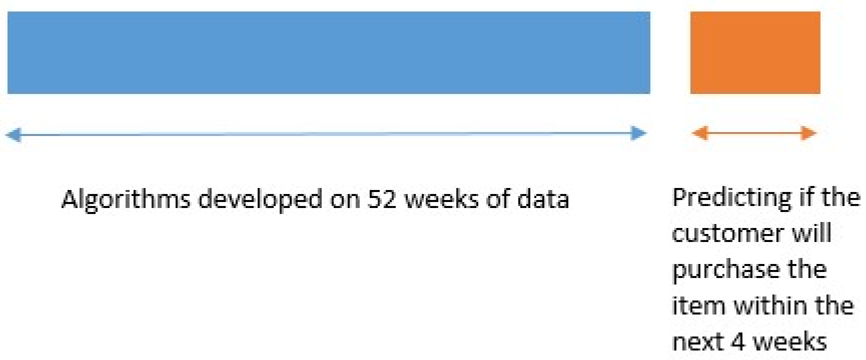
Algorithms developed on 52 weeks of data
Predicting if the customer will purchase the item within the next 4 weeks

Figure 1: illustration of time periods used for prediction

The following algorithms will be utilized in order to create the prediction:

- First, an ALS Matrix Factorization will be created using the 52 week history of customer and item purchases. This will generate 'customer' and 'item' factors that will be used as features, combined with other features and used in the following classification algorithms:
    - Simple Logistic Regression
    - Tree based models (Random Forest and Gradient Boosting classifiers)
    - A Deep Learning model constructed with TensorFlow or PyTorch

Benchmark Model

The closest benchmark for this problem within the grocery domain is the Instacart market basket Kaggle competition (https://www.kaggle.com/c/instacart-market-basket-analysis/leaderboard). The objective of the competition was to predict if a customer would re-purchase an item. This is a very similar problem in that it is in the grocery domain and is predicting whether a customer will purchase a given item in a defined timeframe. The leading F1 score for systems developed to solve this problem was typically ~0.4.

Research into top solutions for this problem showed that the following types of features were found to be highly predictive:

- Counts of the number of items, aisles, departments shopped in different intervals e.g. last 15, 30 days etc.
- Timing features such as when the item was last purchased
- User-based features such as the tendency of a customer to buy items they'd never purchased before

Boosting algorithms and simple DNN's with a few layers were typically found to perform the best.

Evaluation Metrics

In order to assess the accuracy of the model a number of different evaluation metrics will be utilized from Machine Learning and Information Retrieval. The following metrics are proposed:

- Mean Average Precision at “top k”
- AUC
- Precision / Recall and F1 score

Project Design

The overall pipeline for the algorithm development is illustrated in figure 2:

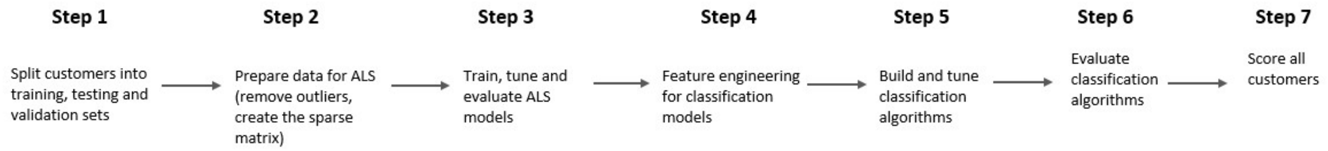
Step 1
Split customers into training, testing and validation sets
Step 2
Prepare data for ALS (remove outliers, create the sparse matrix)
Step 3
Train, tune and evaluate ALS models
Step 4
Feature engineering for classification models
Step 5
Build and tune classification algorithms
Step 6
Evaluate classification algorithms
Step 7
Score all customers

Figure 2: Algorithm development pipeline

The following steps will be followed in the development of the algorithm:

1.) Split customers into training, testing and validation datasets
2.) Prepare data for ALS. This involve treating outliers and creating the ‘customer’ and ‘item’ matrix required by Matrix Factorization algorithms
3.) Train, tune and evaluate the ALS models. Note that hyperparameters will be tuned utilizing the Hyperopt package for Bayesian Hyperparameter tuning
4.) Create additional features for classification models
5.) Build and tune classification algorithms with Logistic Regression, Random Forest and Gradient Boosting
6.) Evaluate the performance of the classifiers and select the ‘winning’ algorithm
7.) Score all customers using the ‘best’ classifier and hyperparameters

References:

[1] https://www.netflixprize.com/

[2] https://docs.aws.amazon.com/personalize/latest/dg/native-recipe-hrnn-metadata.html

[3] https://ai.googleblog.com/2016/06/wide-deep-learning-better-together-with.html

[4] https://cseweb.ucsd.edu/classes/fa17/cse291-b/reading/A%20Dynamic%20Recurrent%20Model%20for%20Next%20Basket%20Recommendation.pdf