Thomas Huelsnitz
ISYE 6501
Homework 10

Question 20.1

Describe analytics models that could be used to help the company monetize their data: How could the company use these data sets to generate value, and what analytics models might they need to do it?

There are lots of good answers, and I want you to think about two types – at least one of your answers should be based on just one data set, the one they've collected internally on customer browsing patterns on the web site; and at least one of your other answers should be based on combining more than one of the data sets.

Think about the problem and your approach. Then talk about it with other learners, and share and combine your ideas. And then, put your approaches up on the discussion forum, and give feedback and suggestions to each other.

Here are the three data sets to consider:

DATA SET #1 (purchased from an alumni magazine publisher)

- first name
- last name
- college or university attended
- year of graduation
- major or majors
- marital status
- number of children
- current city
- email domain
- financial net worth
- binary variables (one for each interest in the publisher's long list of various sports, activities, hobbies, games, etc.) showing whether each one was or wasn't listed by each person

DATA SET #2 (purchased from a credit bureau)

- first name
- middle name
- last name
- marital status
- sex
- year of birth
- current city
- whether they ever owned real estate
- email domain
- list of monthly payment status over the last five years for credit cards, mortgages, rent, utility bills, etc. – for each month and each payment:

- o what type of payment it was – for credit cards, it would say “Visa”, “American express”, etc., not just “credit card”
- o how much was owed
- o how much was paid
- o whether the person was considered to be in default

DATA SET #3 (collected by the company using web site tracking code)

- • title
- • first name
- • middle initial
- • last name
- • credit card type
- • credit card number
- • list of products purchased in the past, with date of purchase and ship-to address
- • which web pages the person looked at
- • how long the person spent on each page
- • what the person clicked on each page
- • estimate of how long the user’s eyes spent on each page viewed (for customers where the software was able to take over the device’s camera)

Internal Data Set

I will start by discussing how the company would monetize the single data set collected by their web site traffic code. Take for example someone who recently purchased protein powder. We could have a model which leverages all the available historical data for purchases of that specific protein powder (or even all protein powder brands with the same volume of product) to predict the typical time between subsequent orders. Our model may determine that the container of protein powder that the customer just ordered is typically ordered again about 20 days later. With this determination we can target the customer with protein powder ads around the time they are looking for a refill (or looking to try a new brand).

The typical time between orders will likely depend on other demographic information. For example, a 20 year-old male weightlifter would likely order protein powder more often than a 45 year-old female mother of 3. A good model for predicting an expected inter-order time based on appropriate historical information would be k-means clustering. Clustering with the information available to us would allow us to distinguish between our two example customers. Customers could be clustered based on their title, age (if available), total amount of money spent in the last year, and time spent viewing each of a representative set of web pages – maybe one each for a sports section, clothing section, cookware section, etc. This analysis would be restricted to all customers with >1 orders of protein powder. This model could take the average of all the inter-order times across the k-nearest neighbors for some suitably chosen k.

As a side note, I would prefer to avoid even capturing the eye-tracking data from customers. This could be very powerful information, but it seems to me that there are only two routes two capturing this data from someone shopping via a device and neither sound good for business to me:

1. Ask the shopper to allow eye tracking. I imagine the vast majority of people would not opt-in for this. This could lead to bias in the small sample of customers for whom we have eye-tracking data
2. Silently take control of user's camera. This sounds like it is probably illegal. Even if it is not, it would likely lead to distrust of the company once it inevitably became public knowledge.

I imagine that the time spent on a web page would give similar information as the eye-tracking data but without the risks described above.

Combined Data Set

Combining the company's data set with the external datasets improves our predictive capabilities. I would start by trying to join the three data sets for the customers that show up in all three. Determining overlap between the 3 datasets is an analytics exercise in itself. I would use the following fields from the datasets to do the matching:

Data Set 1: first name, last name, current city, email domain

Data Set 2: first name, middle name, last name, current city, email domain, monthly payment history (includes credit card type)

Data Set 3: first name, middle initial, last name, credit card type

I would use two different models for joining the data sets together. One each for the relationships:

DS1 <-> DS2

DS2 <-> DS3

where DS1 is the first data set, etc. We could use a fuzzy string matching metric like the Levenshtein distance (https://en.wikipedia.org/wiki/Levenshtein_distance) to calculate the similarity between corresponding fields. If we denote the Levenshtein distance between stringX and stringY as Lev(stringX, stringY) then we could do the following comparisons for each of the models:

DS1 <-> DS2

Lev(DS1."first name", DS2."first name")

min(Lev(DS1."last name", DS2."last name"), Lev(DS1."last name", DS2."middle name"))

Lev(DS1."current city", DS2."current city")

Lev(DS1."email domain", DS2."email domain")

DS2 <-> DS3

Lev(DS2."first name", DS3."first name")

Lev(DS2."middle name"[0], DS3."middle initial")

Lev(DS2."last name", DS3."last name")

Lev(DS2."credit card type", DS3."credit card type")

These are the features that we would feed into each model to assign a score to each possible relationship between two tables (NxM relationships for length N table compared to length M). Next, I would go through and manually create some labeled data from the starting data set by choosing some entries which appear to be the same person and labelling them as such. Once I created enough labeled data points (not sure how many would be necessary), I would then use the labeled data along with their corresponding features as calculated above as training data for a classification model. This could be an SVM or random forest for instance.

Once we are satisfied with our resulting combined data set, we can build logistic regression models for various products or classes of products to predict the probability of a customer purchasing the corresponding item. We could build a model which uses financial net worth, binary interest variables, year of birth, current city, monthly payment history, list of products purchased in the past and which web pages the customer looked at to predict if that customer will buy athletic shoes (or blenders, or a backpack, etc.). The training data for a given model would be easy to prepare. If we want a model for predicting athletic shoe purchase, then we would just engineer a feature for each customer which is a binary indicator for whether the customer has purchased any shoes on our defined list of athletic shoes. We could break up the data set into training and testing sets and build the model in the usual way.

The insights gained from the various models could be leveraged by tying advertisements to their corresponding models and targeting advertisements to customers whose probability for purchasing something is above some threshold.