**Machine Learning Engineer Nanodegree**

**Capstone Project:**

**Predicting Customer Segment for New Customers**

Leander Quiring  
April 27, 2017

**I. Definition**

*(approx. 1-2 pages)*

**Project Overview**

Customer segmentation is considered to be a cornerstone of personalized marketing. Much work has been done in taking historical purchases and building recommender systems; indeed, the success of companies such as Amazon has been credited to such systems. Recommenders are only as strong as the data they’re built on, and customer segmentation information may be a key input.

For brick and mortar retailers, the available data is neither as rich nor as complete as it may be for purely online retailers: full customer profiles are only available if the retailer has a solid loyalty program, and rarely include information that could be derived from things like referring websites and site searches. Even with a strong loyalty program, a retailer will find that there are many cash purchases, or purchases outside of the loyalty program where little is known about the customer.

These non-loyalty transactions can represent a significant portion of any dataset and, therefore, any trends found in the rich-data subset may be entirely wiped out by noise from the sparse. However, if a customer for which we have no data upon which to segment continuously uses the same form of payment (e.g. their primary credit card), a purchase history can be built up.

The goal of this project is to create a system with which we can predict a new customer’s segment knowing only their purchase history.

This project differs slightly from my initial proposal: originally, I had planned to predict the correct customer segment for baskets without customer segment data. However upon further reflection, the fundamental question “What segment does this customer belong to?” is of more interest than “what type of customer most likely purchased this basket.” This requires a change to my benchmark model, as below. Other proposal elements, such as methodology and evaluation metrics have not changed.

In this section, look to provide a high-level overview of the project in layman’s terms. Questions to ask yourself when writing this section:

* *Has an overview of the project been provided, such as the problem domain, project origin, and related datasets or input data?*
* *Has enough background information been given so that an uninformed reader would understand the problem domain and following problem statement?*

**Problem Statement**

I have taken a set of transactional data from fictitious retailer and use it to simulate the above situation, where segmentation data is not known for new customers, namely their two variables “customer life-stage” and “customer price sensitivity.” I have removed these two fields from a testing subset of the data and built a predictive model using other factors in the dataset such as product hierarchy, shop time/day, etc.

The dataset is the “50,000 customers” subset of DunnHumby’s “Let’s Get Sort-of-Real: Dummy Data to Test Techniques and Algorithms” dataset, available at https://www.dunnhumby.com/sourcefiles

There are two tables provided: a time table with calendar information (which is not of interest for this project), and the main table whose columns can be divided into five sections:

* Time (week, date, weekday, and hour of the transaction)
* Basket (basket ID, quantity and dollars spent of each item, two basket size classifications, basket price-sensitivity classification, and a rough basket driver)
* Product (product code and hour levels of product hierarchy)
* Customer (unique customer code, customer’s price sensitivity, life stage)
* Store (store code, size and location)

The first step of this project will be to aggregate transactions into customer profiles. I have chosen to follow the lead of Apeh, Gabrys and Schierz[[1]](#footnote-1) and use spend by a low-level category (Prod\_Code\_20) as the basis for my customer profiles. I also add the sum of spend by weekday and hour of day, as well as the count of baskets by size (small, medium, large) and type (small shop, top up, full shop, unclassified). The two segmentation target variables that I have predicted are price sensitivity (less affluent, mid-market, up market, unclassified) and life stage (young adult, older adult, young families, older families, pensioners, other, unclassified)

After doing appropriate pre-processing, I tested a variety of classification algorithms on the data, because I had no pre-existing preference about which ones to use. Each of the two segmentation targets was be handled separately, so as not to limit my options to algorithms that can handle multiple targets. I placed emphasis on ensemble methods and artificial neural networks, because I suspect this will be a complex problem (e.g. I did not expect to see linear separability or to be able to come up with a kernel function that would make Support Vector Machines a viable choice).

I then selected a small number of top performers for each target, based on classification accuracy and perform hyperparameter optimization to further refine them. Lastly, I created an ensemble of the top performers for each target and compared the voted results to the individual models.

In this section, you will want to clearly define the problem that you are trying to solve, including the strategy (outline of tasks) you will use to achieve the desired solution. You should also thoroughly discuss what the intended solution will be for this problem. Questions to ask yourself when writing this section:

* *Is the problem statement clearly defined? Will the reader understand what you are expecting to solve?*
* *Have you thoroughly discussed how you will attempt to solve the problem?*
* *Is an anticipated solution clearly defined? Will the reader understand what results you are looking for?*

**Metrics**

I will be considering two key metrics in evaluating this project:

* Classification accuracy:
* Precision/Recall:
  1. Precision:
  2. Recall:
  3. To aggregate to a single value for the entire dataset, a weighted average based on the # of each instance of the label (using the average=’weighted’ parameter for the Recall/Precision functions in scikitlearn.metrics)

An algorithm must outperform two benchmark models to be considered “successful:”

* the Naïve Classifier: classifies all customers as being members of the largest class in the training data.
* the Business Logic Classifier, as below.

Price Sensitivity:

Life Stage:

*In this section, you will need to clearly define the metrics or calculations you will use to measure performance of a model or result in your project. These calculations and metrics should be justified based on the characteristics of the problem and problem domain. Questions to ask yourself when writing this section:*

* *Are the metrics you’ve chosen to measure the performance of your models clearly discussed and defined?*
* *Have you provided reasonable justification for the metrics chosen based on the problem and solution?*

**II. Analysis**

*(approx. 2-4 pages)*

**Data Exploration**

As mentioned above, the dataset used is the “50,000 customers” subset of DunnHumby’s “Let’s Get Sort-of-Real: Dummy Data to Test Techniques and Algorithms” dataset, available at <https://www.dunnhumby.com/sourcefiles>. The transactions table, split into 117 CSV files, is structured as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Factor | Description | Sample | Used |
| shop\_week | Calendar week for the transaction | 200801 | N |
| shop\_date | Calendar date for the transaction | 20080101 | N |
| shop\_weekday | Day identifier for the transaction (1=Sunday) | 1-7 | Y |
| shop\_hour | Hour the transaction occurred | 8-22 | Y |
| quantity | # of the product purchased | 1 | N |
| spend | $ spent on the products purchased | 2.99 | Y |
| prod\_code | Unique product identifier | PRD0900049 | N |
| prod\_code\_10 | Hierarchy level 1 | CL00160 | N |
| prod\_code\_20 | Hierarchy level 2 | DEP00054 | Y |
| prod\_code\_30 | Hierarchy level 3 | G00016 | N |
| prod\_code\_40 | Hierarchy level 4 | D00003 | N |
| cust\_code | Customer unique identifier | CUST0000000031 | Y |
| cust\_price\_sensitivity | General price sensitivity of the customer | Less Affluent (LA), Mid-Market (MM), Up-Market (UM), Unclassified (XX) | Y |
| cust\_lifestage | High-level demographic classification | Young Adults (YA), Older Adults (OA), Young Families (YF), Older Families (OF), Pensioners (PE), Other (OT) | Y |
| basket\_id | Basket unique identifier | 994111400529101 | N |
| basket\_size | Relative basket size | S, M, L | Y |
| basket\_price\_sensitivity | Estimated price sensitivity of the basket | Less Affluent (LA), Mid-Market (MM), Up-Market (UM), Unclassified (XX) | N |
| basket\_type | Basket size classification | Small Shop, Top Up, Full Shop, XX | Y |
| basket\_dominant\_mission | Main group of products purchased | Fresh, Grocery, Mixed, Non Food, XX | N |
| store\_code | Store Identifier | STORE00001 | N |
| store\_format | Store Size | LS, MS, SS, XLS | N |
| store\_region | Store geographic area | E02 | N |

The above reflects, but does not copy, the dataset description in the DunnHumby PDF that accompanies the dataset.

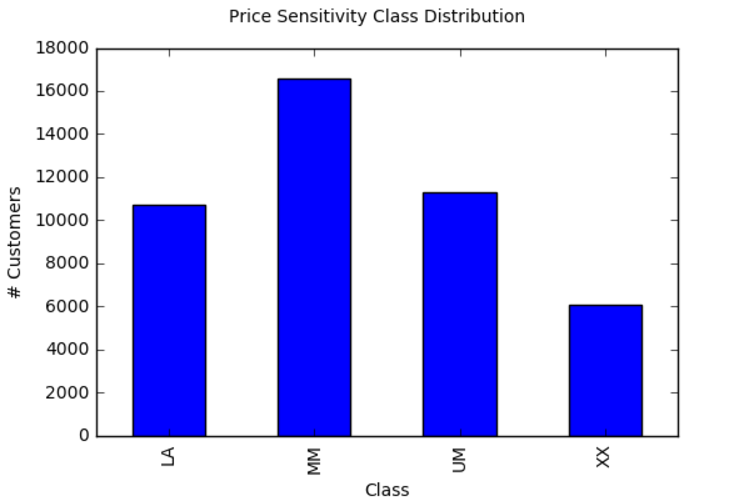
There are 31,057,875 transactions recorded in this dataset, for 50,000 customers. In the pre-processing stage, we turn this dataset into a “wide” table by pivoting on one of the hierarchy levels, and dropping certain columns that we do not use or would be considered “cheating” (e.g. basket\_price\_sensitivity)

One potential problem with this dataset is the relatively large class imbalance for customers who have an ‘OT’ life stage. I try to techniques to handle this imbalance, described in the pre-processing section, but this presents a significant barrier to being able to make use of any predictions.

In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section:

* *If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?*
* *If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?*
* *If a dataset is****not****present for this problem, has discussion been made about the input space or input data for your problem?*
* *Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)*

**Exploratory Visualization**





In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

* *Have you visualized a relevant characteristic or feature about the dataset or input data?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

**Algorithms and Techniques**

When deciding upon algorithms to test, my main criterion was the ability to model highly complex interactions. This decision was based on the fact that, in my research and experience, this problem is neither simple nor easy to solve.

Therefore, the methods I will attempt to use are:

* Random Forests
  + Default Parameters: 10,000 estimators, 10 features, entropy
* Gradient-Boosted Trees
  + Default Parameters: 0.05 as the learning rate, 1,000 estimators, 10 features
  + Good for large amts of data?
* k-Neighbours
  + Default parameters: 10 neighbours
* Multi-Layer Perceptrons
  + Default Parameters: lbfgs as the solver, 1\*10-5 as the learning rate, three hidden layers, each with 100 perceptrons
* Adaboost
  + Default parameters: 1000 estimators
* Grid search parameter optimization of the above
* Ensembles of the above

In deciding upon the above default parameters, some came from massively increasing the sklearn default parameter (e.g. I tried 1000 estimators instead of the default of 50 in sklearn), and others, such as the MLP parameters came from previous iterations of the project. This may hurt any improvement I get when performing the grid search parameter optimization, but I see no reason to not take advantage of what I learned during testing.

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

* *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?*
* *Are the techniques to be used thoroughly discussed and justified?*
* *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

**Benchmark**

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

* *Has some result or value been provided that acts as a benchmark for measuring performance?*
* *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

**III. Methodology**

*(approx. 3-5 pages)*

**Data Preprocessing**

The first preprocessing step is to combine all 117 CSV files into a single dataframe. There are 31,057,875 transactions recorded in this dataframe, for 50,000 customers. After dropping rows with NA values, we have 21,967,768 transactions.

Next, we turn the ‘long’ dataframe into a ‘wide’ dataframe. We pivot and sum SPEND data at the 2nd hierarchy level, PROD\_CODE\_20, grouped by CUSTOMER\_ID. This level was picked to balance the number of variables the pivot introduces (how many are there?). Any nulls, i.e. when a customer has never purchased from a category, are replaced with zeroes. This establishes the customer’s spend profile by product hierarchy.

We then join the summarized SPEND with a number of other factors:

* Sum of SPEND by SHOP\_WEEKDAY to create a spend pattern by day (e.g. does a customer spend most of his or her money on Tuesday or Saturday),
* Sum of SPEND by SHOP\_HOUR to create a spend pattern within a day (e.g. does a customer spend most of her money during typical work hours?),
* Count of item-level transactions per BASKET\_SIZE (e.g indicates how many unique items a customer buys in Small baskets),
* Count of item-level transactions per BASKET\_TYPE (e.g indicates how many unique items a customer buys in Full Shop baskets), and
* The two target variables, Life Stage and Price Sensitivity. These are unique by customer, rather than by transaction.

We split the data into training (70%) and testing (30%) sets. I chose not to explicitly stratify the split because of the reasonably large number of customers.

The final step is to consider two ways to handle the class imbalance, which is particularly noticeable in the Life Stage set. I create four additional training sets, two per target variable:

* Downsampled (denoted ‘Balanced’ in the code), were we find the class with the lowest number of data points and randomly select that number from other class. Unfortunately, this involves dropping a lot of valuable data, particularly in the Life Stage set, where the small class has only 1,255 members, resulting in 7,530 of 31,254 data points. The price sensitivity set keeps 16,892 customers.
* Upsampled (denoted ‘Upsampled’ in the code), where we find the class with the largest number of data points and randomly resample all the other classes such that they all have the same number. We ensure that at least one copy of each data point makes it into the training set. We now have training datasets of 46,512 and 92,976 data points for training classifiers on Price Sensitivity and Life Stage, respectively.

These two attempts to handle class imbalance are based on recommendations made by:

* <https://www.datarobot.com/blog/classification-with-scikit-learn/>
* <http://www.site.uottawa.ca/~nat/Courses/csi5388/Class-Imbalances.ppt>

In particular, the decision to test the impact of both comes from recommendations in the latter presentation.

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

* *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?*
* *Based on the****Data Exploration****section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?*
* *If no preprocessing is needed, has it been made clear why?*

**Implementation**

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
* *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
* *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

**Refinement**

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* *Has an initial solution been found and clearly reported?*
* *Is the process of improvement clearly documented, such as what techniques were used?*
* *Are intermediate and final solutions clearly reported as the process is improved?*

**IV. Results**

*(approx. 2-3 pages)*

**Model Evaluation and Validation**

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

* *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
* *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
* *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
* *Can results found from the model be trusted?*

**Justification**

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* *Are the final results found stronger than the benchmark result reported earlier?*
* *Have you thoroughly analyzed and discussed the final solution?*
* *Is the final solution significant enough to have solved the problem?*

**V. Conclusion**

*(approx. 1-2 pages)*

**Free-Form Visualization**

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

* *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

**Reflection**

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

* *Have you thoroughly summarized the entire process you used for this project?*
* *Were there any interesting aspects of the project?*
* *Were there any difficult aspects of the project?*
* *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

**Improvement**

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

* *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
* *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
* *If you used your final solution as the new benchmark, do you think an even better solution exists?*

**Before submitting, ask yourself. . .**

* Does the project report you’ve written follow a well-organized structure similar to that of the project template?
* Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
* Would the intended audience of your project be able to understand your analysis, methods, and results?
* Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
* Are all the resources used for this project correctly cited and referenced?
* Is the code that implements your solution easily readable and properly commented?
* Does the code execute without error and produce results similar to those reported?

1. Apeh, E.T., Gabrys, B. & Schierz, A. (2011) Customer profile classification using transactional data. *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, 21 Oct. 2011 [↑](#footnote-ref-1)