

An aerial photograph of the CN Tower in Toronto, Canada, taken during the 'golden hour' of sunset. The sun is low on the horizon to the right, casting a warm, orange glow across the city and the sky. The CN Tower's distinctive white structure, including its spherical observation deck and the long antenna mast, is the central focus. The surrounding urban landscape, with various buildings and streets, is visible in the foreground and background, all bathed in the soft light of the setting sun.

Solutions for data science problems

performed by

Sergey Platonov

BERLIN, GERMANY

[GITHUB.COM/SERGEPLATONOV/](https://github.com/SERGEPLATONOV/)

No special license is provided.

March 2018

Contents

1	Introduction	5
1.1	Objective	6
2	Time series forecasting	7
2.1	Feature Engineering	7
2.2	ML Methods	8
2.3	Cross validation and comparing models	8
2.4	Examples	8
3	Customer e-mail categorization	9
3.1	Model description	9
3.2	Feature Engineering	10
3.3	ML methods	10
3.4	Examples	10

4	Repeat purchase probability	13
4.1	Data reduction	13
4.2	Feature Engineering	13
4.3	Methods	14
4.4	Examples	14
5	Inventory Management Optimization	15
5.1	Input of the optimization model	15
5.2	Linearizing the problem	16
5.3	Mixed Integer Problem	16
5.4	Control values	16
5.5	Methods	16
5.6	Examples	16

1. Introduction

The typical goals and deliverables associated with data science are :


- **Prediction (predict a value based on inputs)**
- **Classification (e.g., spam or not spam)**
- Recommendations (e.g., Amazon and Netflix recommendations)
- Pattern detection and grouping (e.g., classification without known classes)
- Anomaly detection (e.g., fraud detection)
- Recognition (image, text, audio, video, facial, ...)
- Actionable insights (via dashboards, reports, visualizations, ...)
- Automated processes and decision-making (e.g., credit card approval)
- Scoring and ranking (e.g., FICO score)
- Segmentation (e.g., demographic-based marketing)
- **Optimization (e.g., risk management)**
- **Forecasts (e.g., sales and revenue)**

Each of these is intended to address a specific goal and/or solve a specific problem. For example, a data scientist may have a goal to create a high performing prediction engine. On the other hand the business that plans to utilize the prediction engine, may have the goal of increasing revenue, which can be achieved by using this prediction engine. It can therefore not be emphasized enough that the ideal data scientist has a fairly comprehensive understanding about how businesses work in general, and how a company's data can be used to achieve top-level business goals. However with significant business domain expertise, a data scientist should be able to regularly discover and propose new data initiatives to help the business achieve its goals and maximize their KPIs.

1.1 Objective

In this report I explain my working methods, machine learning algorithms and visualization techniques. I hope this report can help to adequately estimate my skills. My discussion will be limited to four main problems:

- time series forecasting
- text categorization
- repeat purchase probability and customer lifetime value
- linear optimization in logistics

 For any questions do not hesitate to contact me, my e-mail address is: *platonov.serge@gmail.com*, also as part of my own documentation I created a GitHub page where you can download all the codes I programmed and find more information. The link to this page is: <https://github.com/sergeplatonov>, take your time to surf.

2. Time series forecasting

Time series, in general, are difficult to forecast. If they were easy to forecast then all data scientists would be wealthy, having accurately forecast the value of all of the stocks. The reality is that hedge funds, on average, do not outperform the market and that time series forecasting is typically very poor and applies only to very short durations. The main problems are that there is a lot of noise, there are many hidden influences, models are overly simplistic, influencers do not behave as we think they should, the interplay between linearity and nonlinearity is subtle and confusing, ... ad infinitum.

2.1 Feature Engineering

An important role of the data scientist is to correctly prepare the data to feed an operating algorithm. This preparation likely consists of cleaning, arranging the data and reshaping it in order to reach the intended goal. Here i want to highlight some hints of the data preparation or feature engineering:

- Modular arithmetic calculations: e.g. converting a timestamp into day of the week, or time of day. If your model needs to know that something happens on the fourth July of every month, it will be nearly impossible to determine this from timestamps.
- On a similar vein, creating new features from the data you have available can drastically improve your predictive power. This is where domain knowledge is extremely important - if you know of, or think you know of a relationship then you can include variables that describe that relationship.
- Dimension reduction is typically performed by either feature selection or feature transformation. Reducing the dimension through feature selection doesn't help in all ML algorithms, but an algorithm may or may not benefit from feature transformation (for example principal component analysis) depending on how much information is lost in the process. The only way to know for sure is to explore whether feature transformation provides better performance.

2.2 ML Methods

There are a number of approaches that can be applied to predict time series. Some are better than others depending on the characteristics of the time series (like the distribution dependence for each parameter over time etc.). Methods include:


- Regression - Using time-based features such as week, month, day, day of week, etc as predictors. You can also add in external predictors that may influence the target (e.g. weather and temperature may affect sales of umbrellas).
- Autoregression and especially ARIMA - Autoregressive Integrated Moving Average - Using autocorrelation (lags) as predictors.
- Deep learning and especially Recurrent Neural networks (RNN). In short, RNN models provide a way to not only examine the current input but the one that was provided one step back, as well. If we turn that around, we can say that the decision reached at time step $t-1$ directly affects the future at step t .
- LSTM - a special kind of RNN that learns long-term dependencies.
- Other methods.

2.3 Cross validation and comparing models

Time series are fun in that all training data can usually be turned into supervised learning training sets. Once can simply take a time series and roll back time. That is when we pick a point in time and pretend that you don't have any additional data, then produce a forecast and see how well you did. You can march through the time series doing this n times in order to get an assessment of the performance of your model and to compare models while taking the necessary precautions to prevent overfitting. This is the formulation of the cross validation process.

2.4 Examples

To be more specific I use a data project from Rossmann to illustrate the prediction of the time series that you can find at Github.

 Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality.

The task is to predict 6 weeks of daily sales for 1,115 stores located across Germany. Reliable sales forecasts enable store managers to create effective staff schedules that increase productivity and motivation.



<https://www.kaggle.com/c/rossmann-store-sales>

3. Customer e-mail categorization

Undoubtedly, categorizing e-mails based on the content poses many challenges. In reality, there is a constant stream of new information being passed through e-mails each day and what we learn from previous e-mails may not be able to tell us much about future e-mails. E-mail threads will branch off onto new topics, and each user organizes their e-mails in different ways.

Within this task I will attempt to accurately classify e-mails into folders using e-mail content, such as an e-mail's headers and its body. This is a Natural Language Processing task that aims to make sense of text documents, by converting text into numerical feature vectors.

3.1 Model description

For the computer to make inferences of the e-mails, it has to be able to interpret the text by making a numerical representation of it. One way to do this is by using something called a Bag-of-words model. It will take the e-mails as a string and convert it into a numerical vector to show the frequency that each unique word appears over the entire dataset.

Bag-of-word model is an orderless document representation — only the counts of words mattered. One way to store this spatial information within the text is the n-gram model (basically it will split the text into groups of n words with the right order). Term-frequency-inverse document frequency (TF-IDF) is another way to judge the topic of an article by the words it contains. With TF-IDF, words are given weight – TF-IDF measures relevance, not frequency. That is, wordcounts are replaced with TF-IDF scores across the whole dataset. A general alternative to the use of dictionaries is the hashing trick, where words are directly mapped to indices with a hashing function. By mapping words to indices directly with a hash function, no memory is required to store a dictionary.

3.2 Feature Engineering

Here are the list of things that needs to be performed on the data:

- Convert date column to datetime
- Remove non-topical folders
- Remove folders containing too few e-mails (less than 2)
- Select employees with over 1000 e-mails
- Drop rows with missing values
- Encode class labels
- Define the Bag-of-words model
- Tokenization i.e. breaking up sentences into words
- Remove unwanted characters from the message, Subject, X-To and X-From columns
- Assemble matrices
- Count tokens
- Remove stop-words

The regular expressions, which includes punctuation marks and non-word characters need to be removed. I will use Python's regular expression (regex) library to remove these characters. I will also focus on just one employee for the classification problem. After seeing the steps involved in classifying one employee's e-mails, we can apply the same approach for a few other employees. We also remove folders that do not contain enough e-mails because such folders would not be significant for training our classifier.

3.3 ML methods

As a training model for ML I use logistic regression that works with the case of a binary dependent variable. In training a multiclass classification problem, we have to train n models where n is the number of unique folders present. Using a one-vs-all approach, we need to train models where all e-mails belonging to a folder are classified as positive (1) or True and all e-mails not belonging to a folder are classified as negative (0) or False.

For the folders first, second and third, then we train 3 models with the following conditions:

- All the e-mails belonging to first are positive(1) and all e-mails belonging to other folders are negative(0)
- All the e-mails belonging to second are positive(1) and all e-mails belonging to other folders are negative(0)
- All the e-mails belonging to third are positive(1) and all e-mails belonging to other folders are negative(0)

3.4 Examples

I illustrate the e-mail categorization process on a data collected from a collapsed company Enron. You can find the correspondent code at the Github.



The Enron email dataset contains approximately 500,000 emails generated by employees of

the Enron Corporation. It was obtained by the Federal Energy Regulatory Commission during its investigation of Enron's collapse.

This is the May 7, 2015 Version of dataset, as published at <https://www.cs.cmu.edu/~enron/>

4. Repeat purchase probability

Consumer brands often offer discounts to attract new customers to buy their products. The most valuable customers are those who return after this initial incented purchase. With enough purchase history, it is possible to predict which customers, when presented an offer, will buy a new item. However, identifying the customer who will become a loyal buyer – prior to the initial purchase – is a more challenging task. Customer lifetime value (CLV), “discounted value of future profits generated by a customer” is a helpful parameter in this problem. Generally speaking the CLV can be calculated as: $(\text{Avg Monthly Revenue per Customer} * \text{Gross Margin per Customer}) / \text{Monthly Churn Rate}$.

4.1 Data reduction

The necessary dataset for the problem contains the transaction information that contains customer ID, date and value of transactions. It is usually enormously huge (over 10 GB) making feature engineering process hard and time consuming. Just for getting started, to get the data down to a more manageable size, it is possible to extract from transaction information only transactions where the category was related to at least one of the made offers. This reduction process gets the transactions data down from about 22GB to about 1GB in my example.

4.2 Feature Engineering

For the general task we can generate the following features:

- the number of times a shopper has bought from the company on offer
- the total amount the shopper has bought from the company on offer
- the quantity of items bought from the company on offer.
- the number of times a shopper has bought from the company on offer in the 30 days before the date the coupon was offered. The same can be done for 60,90,120 days.
- the number of times a shopper has bought from the category on offer
- the number of times a shopper has bought from the brand on offer
- if this feature is present the shopper has bought from the company, brand, and category on offer.
- a negative feature for when the shopper has never bought from the company on offer before.

- a negative feature for the combination of brand and company purchase history.

As well we can add some offer-related features:

- The value of the coupon offer
- The number of products to redeem with the coupon

And finally some general monetization features:

- the total amount spend by the shopper

To simplify the discussion we can instead deal with only three main features using the python 'Lifetimes' library:

- age of the customer i.e. the duration between a customer's first purchase and the end of the period under study.
- recency or age at the last customer purchase. This is equal to the duration between a customer's first purchase and their latest purchase.
- frequency or the number of repeat transactions the customer has made. It is the count of time periods the customer had a purchase in. So if using days as units, then it's the count of days the customer had a purchase on.

Lifetimes library has easy utility functions to transform our transactional data (one row per purchase) into summary data (a frequency, recency and age dataset).


4.3 Methods

For the general formulation we can treat the task as a classification problem and apply ML classification algorithms like Random Tree Classifier. Then we can predict future revenues, based on estimation about future products purchased and price paid and finally calculate the CLV.

If we can work with only 3 features we can easily visualize the CLV by computing the Frequency/Recency matrix i.e. the expected number of transactions a artificial customer is to make in the next time period, given his or her recency and frequency. Then as an ML method it is convenient to use the BG/NPD model for future purchase predictions. This method assumes an exponentially distributed active duration and Poisson distributed purchase frequency. For the CLV prediction we can use the Gamma-Gamma model with the key assumption about the independence between the number of transactions of a customer and the related profit per transaction.

4.4 Examples

I illustrate this discussion with a realistic example based on the Acquire Valued customers Challenge from kaggle.

-  The Acquire Valued customers Challenge asks participants to predict which customers are most likely to repeat purchase. To aid with algorithmic development, we have provided complete, basket-level, pre-offer shopping history for a large set of customers who were targeted for an acquisition campaign. The incentive offered to that customer and their post-incentive behavior is also provided.

<https://www.kaggle.com/c/acquire-valued-customers-challenge>.

5. Inventory Management Optimization

Tasks of the data scientist do not only include machine learning problems but spread beyond as I mentioned in the beginning of the report. One alternative task is based on distributing, over time, a limited amount of inventory across the company stores in a retail network. Challenges specific to that environment include very short product life cycles, and store policies whereby an article is removed from display whenever one of its key sizes stocks out.

To solve this problem it is necessary to introduce:

- A stochastic model predicting the sales of an article in a single store during a replenishment period
- Determine demand forecasts, the inventory of each size initially available, and the store inventory management policy that are important for the model
- Perform a linear optimization of the model applied to every store in the network,
- Compute store shipment quantities maximizing overall predicted sales, subject to inventory availability and other constraints.

5.1 Input of the optimization model

In many clothing retail stores, an important source of negative customer experience stems from customers who have identified (perhaps after spending much time searching a crowded store) a specific article they would like to buy, but then cannot find their size on the shelf/rack. Proper management of size inventory seems even more critical to a modern retailer that offers a large number of articles produced in small series throughout the season. The presence of many articles with missing sizes would thus be particularly detrimental to the customers store experience.

R For simplicity of the discussion we will omit now the differentiating the sizes between major (S,M,L) and minor (XXS,XXL) and possible manager actions related to this two groups of articles. Practically the problem will also have a dynamic component because of the shipment decisions any given week. In addition the problem may involve connections between different articles that I neglect here.

5.2 Linearizing the problem

A key concept for estimating the total number of sales from an initial profile of inventory will depend now on a so-called virtual stock-out time i.e. the time at which the store will run out of the chosen size. The primary input for the optimization model includes a demand forecast for every retail store and for every given size and the inventory available in stores. I also assume that all inventory is removed from customer view as soon as one of the major sizes runs out (less than four articles) at any point between successive replenishments.

5.3 Mixed Integer Problem

The main objective to implement an optimization model for distributing a limited amount of warehouse inventory between all stores of retailer with the goal of maximizing total expected revenue. The primary decision variables represent the shipment quantities of each size to each store. These quantities are subjected to the warehouse inventory constraint that insures the total shipment of a given size across all stores never exceeds the inventory. The secondary decision variables correspond to approximately expected sales across all sizes in each store for the current period under consideration. Now we can introduce a maximization problem of the sum of expected revenues in the current period. This problem considers each article independently and captures dynamic effects in a heuristic manner.

5.4 Control values

As control values for quantifying missed sales due to lack of inventory (i.e., under-stock) we can use the ratio of cumulative sales to cumulative demand – demand cover ration. On the other hand for quantifying any excess inventory (i.e., overstock) we use the ratio of cumulative sales to cumulative shipments or shipment success ratio.

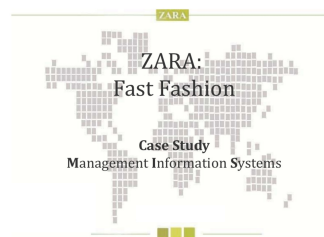
5.5 Methods

The solution of the Mixed Integer Problem can be performed with the help of python PULP library. An example is provided in the github repository.

5.6 Examples



The case of Inventory Management of a Fast-Fashion Retail Network was studied by F. Caro and J. Gallien for the case of Zara Retail



Inventory optimization models are put in place in Zara to help the company to determine the quantity that should be delivered to every single one of its retail stores via shipments that go out twice every week.

<https://pdfs.semanticscholar.org/80d2/4bcf23391dff0907d406cf1466d4b8aab007.pdf>.

Wish you all the best, Sergey Platonov