Business insight

TIMi suite

integrated data mining solutions

An automated predictive datamining tool

Data Preparation - Propensity to Buy v1.07

Creation Date: June 2011
Last Edited: February 2015

1. Data preparation - Introduction

If you are using "***The Intelligent Mining Machine*" (TIMi)** inside your datamining projects, the data preparation step is the most time-consuming step in the whole project.

In opposition to other classical datamining softwares, **TIMi** has no limitations to the number of columns or rows inside the learning/creation dataset that it can process:

- **TIMi** is able to process dataset containing millions of rows in a few minutes. Sampling (and thus losing information) is (nearly) no longer required with **TIMi**.
- **TIMi** is able to easily process datasets with more than fifty thousands columns. Classical datamining softwares reach their limits around 300 columns (even if they don't "crash" when reaching this limit, the accuracy of the predictive models produced by other softwares is degrading when you increase the number of columns above 300 inside your dataset). This also means that you **do not have** to eliminate arbitrarily (based on some "heuristic") some variables of your dataset in order to stay below the column limitation of the other datamining softwares.

When using **TIMi**, it is strongly suggested to accumulate the highest number of information (rows and columns) about the process to predict: don't reduce arbitraly the number of rows or columns: always keep the "full datatset" (even if the target size is less than one percent). **TIMi** will use this extra information (that is not available to other "limited" datamining softwares) to produce great predictive models that outperform any predictive model constructed with any other datamining software.

Of course, to be completely rigorous, you should also not forget to "let aside" a TEST dataset that will be used to really assert the quality of the delivered predictive models (to be able to compare in an objective way the models constructed with different predictive datamining tools). See this web page that explains the importance of the TEST set:
http://www.business-insight.com/html/intelligence/bi_test_dataset.html

This document describes very briefly the data preparation steps required for a preliminary analysis of your data. This preliminary analysis is just to assess if your databases contains enough useful, exploitable, information about your customers that are actionable from a business perspective.

This document is also useful when you want to setup very rapidly a benchmark to compare the performances of different datamining tools. In this case, the dataset given to **TIMi** should follow the recommendation and guidelines contained within the section 4 ("Creation dataset file format").

2. Set up model design

To build a predictive model you need a database. This database must be presented to TIMi as a large table (the optimal format is as a compressed .csv flat file). The objective of this document is to describe how to build this table.

Let's assume the following situation: You have a database of all your customers. You want to predict for each customer the probability that he will buy the product X (this is a classical "propensity-to-buy" problem, but the same principles applies to any predictive modelling exercice: Cross-Selling, Up-Selling, Purchase prevention, etc. For the simplicity of the exposition, we will now assume that we are in a classical "propensity-to-buy" setting). You will create a table (typically: a flat file). Each line of the table represents one customer. The columns of the table are information about the customer: what did you bought recently?, what's its age, gender, financial income?,… One column of the table has a special meaning: it's the ***target***: It's the column that you want to predict. The ***target*** column contains two values (no missing values are allowed for the ***target*** column):

- value '0': the customer did NOT buy the product X.
- value '1': the customer bought the product X.

To create the above database (in technical this database is named the "creation dataset"), you will need a tool. Of course, you can do all data preparation work with simple SQL queries but I strongly suggest that you use Anatella (i.e. the ETL included inside the TIMi Suite) to prepare you dataset: it will make things a lot faster & easier. Anatella is the only data-processing-tool currently available that is specifically built for data preparation for predictive modeling.

In more details, the "creation dataset" must contain:

- On each row: information related to one customer.

- A special column named the "Target" that contains, basically, two values (no missing values are allowed for the ***target*** column):
  - value '0': the customer did NOT buy the product X.
  - value '1': the customer bought the product X.

  The exact procedure to create the "Target" column is described in the next section (3.2).

- Many different columns that will be used by the predictive model to predict the target (i.e. to predict if an individual will buy or not the product X). These columns contain all the "profiling information" of each of your customers.

  The higher the number of columns in the dataset the higher will be the accuracy of the predictive model. The columns must be somewhat related to the target to predict (for example, the name of the individual has no influence at all on the purchase, so it's not an useful column).

  To create interesting and useful columns, we will, most of the time, follow the RFM approach. The RFM approach is the following: We will create columns (also named "variables" in technical terms) that are related to:

  - R ("R" means "Recency"):
    - How recent is the last purchase of the individual?
    - How recent is the last contact with the website/sale point?

  - F ("F" means "Frequency"):
    - How many purchases on the last time period?
    - How many transactions on the last time period?
    - How many visit to the website/sale point?

    - How much time spend on the last time period?
    - How many purchases in each category (food, groceries, electronics, computer gadets,etc.) on the last time period?
- M ("M" means "Monetary"):
    - For how much money was the last purchase on the last time period?
    - For how much money was the last purchase per category on the last time period?
    - For subscription-based services:
        - For how much time is the current subscription still active?
        - For how much time is the current subscription still binding?
        - How many subscriptions are currently active or binding? What's the ratio between active and binding?
        - Mean Recurring re-charge per month.
    - Which payment method (direct debit, bank transfer, …)?
    - How much discount on the current subscription(s)? (Has the individual some promo code?)
    - Which promotion type?
    - Did the customer buy some specific options on his products?

The RFM columns can be computed at a different granularity:

- At the product level
- At the product category level
- For all products indifferently

The following "M"-type variables should not be used "directly" inside the predictive model:

- For subscription-based services: Which tariff plan?
- What's the LTV (Life-Time-Value) of the individual?

These variables should rather be used to create different customers segments. Thereafter we can create one predictive model for each segment.

Other interesting columns to include are:

- Ratio variables: These variables encode the difference and ratio between two related RFM variables:
    - What's the ratio of purchase inside a specific category?
    - What's the ratio of money spent inside a specific category?

- Delta variables: These variables encode the difference in RFM variables (consumption, frequency of usage,etc.) between two time periods:
    - How much increase (or decrease) in terms of number of purchase between the last time period and the previous one?
    - How much increase (or decrease) in terms of expenses (in euros) between the last time period and the previous one?

- Delta on Ratio Variables: These variables encode the difference in Ratio variables between two time periods:
    - How much increase (or decrease) in terms of ratio of money spent inside a specific category between the last time period and the previous one?

- Standard profiling information:
  - Country of origin
  - ZIP code
  - Language
  - Sex
  - For Telecom Only: Handset:
    - Brand,
    - Age of Handset,
    - Functionality,
    - Adequacy between Handset functionality and subscription.
  - For subscription-based services: Adequacy between current consumption habits and current subscription (this can be computed automatically using a predictive model built with TIMi).

- Social Network Analysis:
  - Are there any recent purchase in the immediate vicinity of the individual?

    The “immediate vicinity concept” can be computed in many different ways: for example: Two individuals are “*close together*” (i.e. in the “immediate vicinity” of each other) if:
    - Their postal address is the same.
    - For Telecom Only :
      - There has been some phone calls between the 2 individuals (this can be computed directly from CDR logs using the LinkAlytics tool from Business-Insight).
      - Their most common location during the evenings and/or the day (estimated using cell-id) is close. (This can be computed directly from CDR logs using the LinkAlytics tool from Business-Insight).
      - They are in the same “group of friends” (see next point).

  - For Telecom Only: Concept related to “group of friends” (all the “group-of-friends” can be computed directly from CDR logs using the LinkAlytics tool from Business-Insight): If we look specifically to the “group of friend” of the current individual:
    - How many purchase in this “group of friends”?
    - Are the buyers “social leaders” (in this “group of friends”)?
    - Is the individual strongly integrated into his “group of friends” or is the connection to his friends “loose”?
    - What’s the distance (expressed in “number-of-friends”) between the current individual and the closest buyer in the “group of friends”?
    - We can also compute all the different RFM, Ratio and Delta variables aggregated at the level of the “group of friends” (this adds many, many interesting variables!). For example:
      - How many subscriptions are currently active or binding in the current “group of friends”?
      - How many MOU on the last time period towards “outside the network” in the current “group of friends”?
      - How much increase (or decrease) in terms of MOU (minute-of-usage) towards “outside the network” between the last

time period and the previous one in the current “group of friends”?

- etc.

- GIS data (geographical data)
    - The ZIP code of an individual is very often related to the revenue of the individual. The “revenue information” is very often a good indicator if an individual can “afford” the product/service/subscription (especially for expensive items). Thus, it’s common to find the ZIP code as a good predictive variable.
    - For Telecom Only:
        - The most common location during the evening/during the day (estimated using cell-id) (for the same reason as the previous point) (This can be computed directly from CDR logs using the LinkAlytics tool from Business-Insight).
        - Are there many disconnections on this part of the country? (This can be computed directly from CDR logs using the LinkAlytics tool from Business-Insight).

- Socio-demographic data: You can usually buy such data from external data providers. These data are usually extrapolated from the postal address of an individual: Based on the postal address, you will usually obtain the following variables:
    - an estimation of the revenue,
    - an estimation of the age,
    - an estimation of the number of cars of owned by an individual,

  These values are all estimated based on the postal address and are thus not very reliable. Socio-demographic data are thus usually not very useful in terms of prediction. The best variables are always behavioural variables.

For Telecom Only:

The LinkAlytics tool from Business-Insight can automatically compute, using CDR logs, the following variables:

- Many RF variables,
- Many Ratios on RF variables,
- Many Deltas on RF variables,
- Many Deltas Ratio on RF variables,
- Many Social Network variables (This is many variables because this also includes aggregates on the RF variables, The Ratios on RF variables, The Deltas on RF variables and The Deltas Ratio on RF variables),
- Some of the GIS data variables (based on Cell-id’s)

Using the LinkAlytics tool from Business-Insight is a good starting point to obtain rapidly an already quite good dataset for cross-selling projects or “propensity-to-buy” projects using predictive analytic techniques.

Usually, the most important variables for cross-selling for the telecom industry are (from the most important one to the least important one):

- Number of recent purchase in the "Group-of-Friend" (and all the related Delta variables)
- Number of binding subscription (but this variable should be removed when the target is defined as the "commercial purchaseers" and not the "standard" purchaseers).
- Ratio of calls "outside the network" compared to calls "inside the network"? (with all the variation around the "outside theme") (and with all the related Delta variables)
- Payment method
- Number of calls to the hot-line
- 2-digit ZIP code
- Handset Brand & Age

The data preparation phase before obtaining a first dataset (i.e. a first the table) required to create a predictive model can be quite long (especially if your operational system does not contain the appropriate information).

There is no actual limit to the amount of information that you can extract from your operational system to build your predictive models (**TIMi** is unlimited in the number of columns that it can process). The only limit is your imagination and creativity! The higher the number of variables available to make the prediction, the better will be your predictive models (and thus, the higher the ROI of your campaigns!). Thus, the temptation to spend still a little bit more time in data preparation to create still another new variable (that can possibly increase substantially the quality of your models) is big.

Before investing time and resources in a very long and expensive data preparation effort, we suggest you to contact our datamining experts. Our team of dataminers will examine the data already available in your operational system and will propose to you the best alternative possible: A data preparation phase as short as possible (selecting only the data the easiest to obtain) but still delivering the most important variables for modelization. Our consultants can also perform all the data preparation phase for you, if required (and if the data confidentiality protection rules of your particular company allows it). The objective here is to rapidly arrive to a preliminary model to rapidly demonstrates, on your specific case, the large ROI delivered by predictive analytics.

3. Preparing the Dataset:

3.1. Defining a good "Target".

Let's assume the following situation: You have a database containing all the recent purchases of your customers. You want to predict, for each customer, the probability that he will purchase the product X in the next time period. Classically, the "Time period" for propensity-models is 1 month.

To create the predictive model, you need a table named the "creation dataset": see the previous section about this table. Each line of the table represents one customer. The columns of the table are information about the customer. One column of the table has a special meaning: it's the ***target***: It's the column you want to predict. The ***target*** column contains two values (no missing values are allowed for the ***target*** column):

- value '0': the customer did NOT buy the product X.
- value '1': the customer bought the product X.

Let's assume that you have:

Database SnapShot at the end of January

id	name	age	revenu	gender	date of purchase	target
1	frank	32	4000	M		0
2	sabrina	29	4000	F		0
3	max	20	2000	M		0

Current Database at the end of February

id	name	age	revenu	gender	date of Purchase	target
1	frank	32	4000	M		0
2	sabrina	29	4000	F	2-14-07	1
3	max	20	2000	M		0

March

No data Yet

Please note that Sabrina decided to purchase the product X at some point in time after the "End-of-January".

The table that you should provide to TIMi to build a predictive model is the following (this table is called the ***model creation/learning dataset***):

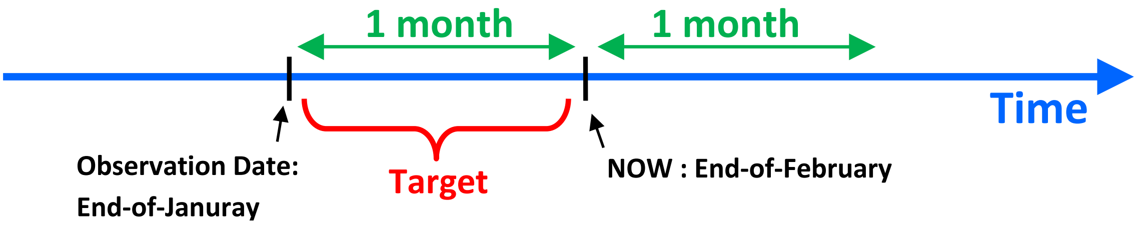
The target is based on what happened **after** January: It is simply the target column extracted from the most recent update of your database

Snapshot from end-of-January

id	name	age	income	gender	date of purchase	target
1	frank	32	4500	M		0
2	sabrina	25	4000	F		1
3	max	33	2000	M		0

The table (i.e. the ***creation/learning dataset***) above is the best option: TIMi will construct a new predictive model ***m*** that will use the customer profiles as they appeared at the end-of-January to predict the ***target*** in {February}. The date "*End-of-January*" is called in technical term the "**Observation date**": it's the date where we "*observed*" the profile of the customers to construct the ***learning dataset***.

Graphically, this can be illustrated in this way:

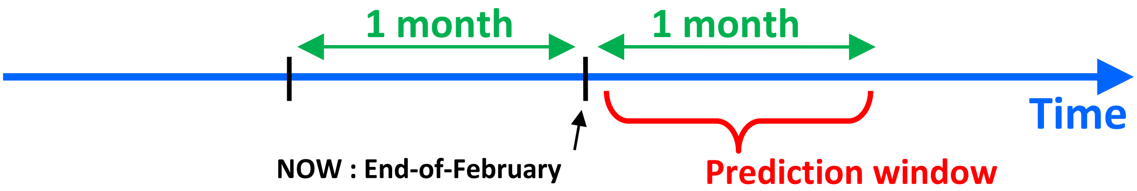


In the above example, the target is defined on a period of 1 month (i.e. it's defined based on the period that extends between "now" and the "observation date"): in technical terms, the "**Prediction Window**" is 1 month.

More formaly, we can also write the predictive model in the following way:

$$\boxed{\boldsymbol{m}(\text{ CustomerProfile}_{\text{Time=T}_{\text{observation}}}) = \text{TargetPrediction}_{\text{Time} \in [\text{T}_{\text{observation}} \ \text{T}_{\text{observation+1 month}}]}}$$

Once you have constructed your predictive model ***m***, you can apply it on the most "up-to-date" version of your customer database (from End-Of-February) to predict who are the customers that will buy the product X in {March}. Graphically, this can be illustrated in this way:



To build such a ***creation/learning dataset***, you need a database structure that supports time logging (or you need to create several “snapshots” of your customer database at different point in time) because you are mixing columns from End-Of-January (containing the profile of your customers) with one column from End-Of-February (containing the target column). Such complex database structure is not always available. As an alternative (but less accurate) solution***,*** you can also use the following ***creation/learning dataset***:

Most Up-To-Date database from End-of-February

id	name	age	income	gender	date of purchase	target
1	frank	32	2000	M		0
2	sabrina	25	4000	F	2-14-07	1
3	max	33	2000	M		0

You should try to avoid this second approach because it has many flaws. Unfortunately, inside the industry, this approach is used 99% of the time because it does not require time logging (nor snapshots).

The first major drawback of this second approach is: when you use TIMi (or any other predictive analytic tool) to construct a predictive model on this ***creation dataset***: you will obtain the following predictive model:

If (“date of purchase” is missing) then target=0 else target=1

The above predictive model has not identified the ***cause*** of the purchase of the product X but the ***consequence***. The column “*date of purchase*” does not contain any information that could be used to predict if a customer will purchase the product X (because this column is initialized ***after*** a purchase). When you use TIMi to construct a predictive model, you must tell to TIMi to ignore all the “consequences” columns. The visual interface to TIMi allows you to ignore a column/variable very easily with only one mouse-click. 99% of the modelization time is usually spent finding these bad “consequences” columns. Usually, you don’t know them “in advance”, before starting the modelization process. TIMi allows you to find these “consequences” columns very easily (because of the very concise and intuitive reports auto-generated by TIMi). At the end, your ***creation dataset*** will be:

Current database from beginning of March
minus all “consequence” columns.

id	name	age	income	gender	target
1	frank	32	2000	M	0
2	sabrina	25	4000	F	1
3	max	33	2000	M	0

To summarize, the ***creation/learning dataset*** to prepare before using TIMi can either be:

- **Approach 1 (this is the best approach):** A mix of different time period:

Snapshot from end of January

the ***target*** column (extracted from the most up-to-date customer database)

id	name	age	income	gender	date of purchase	target
1	frank	32	4500	M		0
2	sabrina	25	4000	F		1
3	max	33	2000	M		0

- **Approach 2 :** The database in its current state **and** a list of "consequence" columns to ignore

Most Up-To-Date customer database

id	name	age	income	gender	date of purchase	target
1	frank	32	2000	M		0
2	sabrina	25	4000	F	2-14-07	1
3	max	33	2000	M		0

The list of "consequence" columns to ignore is: *date of purchase*

NOTE:

Approach 1 systematically delivers higher accuracy models (higher ROI) than Approach 2.

If you analyze closely the creation/learning dataset used in approach 2, you will arrive after some thoughts, to either one of these two predictive models:

If ("age" < 30) then target=1 else target=0

or

If ("income" > 3000) then target=1 else target=0

If you look at the creation/learning dataset used in approach 1, you will notice that only the first model (out of the above 2 model) (the one based on the "age") is correct.

When your customer database is evolving rapidly, you cannot make the assumption that the approximate customer profiles available for “approach 2” are similar to the exact customer profiles available in “approach 1”. When this assumption breaks down, the predictive model generated using the “approach 2” will have poor accuracy (because these are based on wrong profiling information).

3.2. Taking into account the TIME aspect: Prediction window and Target size.

In the above example, the “*prediction window*” is 1 month (it’s a common value found in the industry). What happens if nobody decided to buy the product X during this month? In such situation, the “target” will be “zero” for all the rows of your dataset and it won’t be possible to create any predictive model. You will be forced to define a longer “*prediction window*” (to avoid the “null” target problem). Another (less good) solution is to change slightly you target: instead of creating a model that predicts the purchase of the product X, you create model that predicts any purchase inside the category of the product X.

Let’s assume that your “prediction window” is one year. This also means that your “*observation date*” is one year in the past and that your ***creation/learning dataset*** is based on some data that is already *one year old*. In one year, the whole economic situation of the market might have changed and your predictive model won’t be adapted to this new situation. This is not good. Obviously, you want your “prediction window” to be as short as possible (to be able to capture the latest trend of your market) but still long enough to have a “reasonable target size” (to avoid the “null” or the “nearly-null” target problem).

This is exactly where TIMi has an important added value compared to other datamining tool: TIMi is able to work with extremely small target size. With TIMi, the target size can be as small as half a percent of your database. Typically, other datamining tools are not working properly when the target size is below 5% of the whole database.

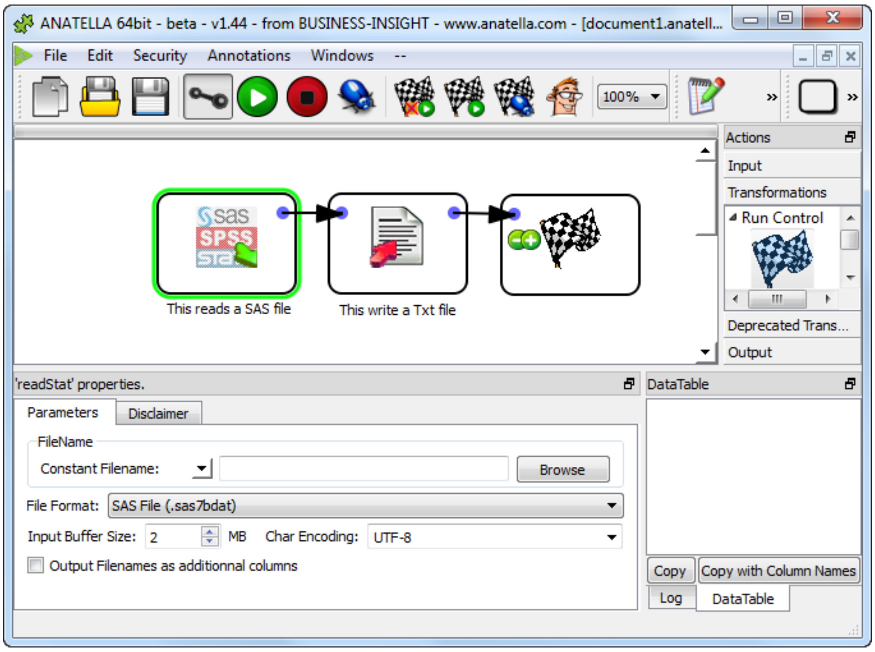
4. Creation dataset file format

TIMi can read datasets from many data sources: simple "txt" or ".csv" flat files, SAS files, ODBC & OleDB links to any databases (Teradata, Oracle, SQLServer,etc.). **But** for a first “quick benchmark”, it‘s suggest to store the ***creation dataset*** inside a simple "txt" or ".csv" flat file (in order to prevent any inter-operability issues).

The "txt" or ".csv" flat file should follow the following format:

- The ***creation dataset*** is a "txt" or ".csv" file where the separator is a dot-comma ‘;’. The first line of the file must contain the column names.

  WARNING: "txt" files exported from SAS have a size limitation: one line cannot exceed 65535 characters. If you encounter this bug in SAS, the easiest solution is the following: Convert the .sas7bdat SAS file to a simple "txt" file (or even better: a .gel_anatella file!) using Anatella:
  Use the following Anatella-data-transformation-graph:



FileName

Constant Filename:

File Format: SAS File (.sas7bdat)

Input Buffer Size: 2 MB Char Encoding: UTF-8

□ Output Filenames as additionnal columns

- Column names must be unique.

  <u>WARNING:</u> TIMI is case IN-SENSITIVE (as is SQL)

- Column names are NOT within quotes.

- The data in the columns are NOT within quotes (never).
- The field separator character (here ‘ ;’) is not allowed (neither in the data, neither in the column names).

- The ***creation dataset*** contains <u>one</u> unique primary key.

- The decimal character is a dot and not a comma (Standard English notation or Scientific notation for numbers).

- If The ***target*** column (the column to predict) is:
  - <u>Binary:</u> then it must contains only ‘0’ and ‘1’ values (and the “one’s” are the value to predict and must be the **minority case**).
  - <u>Continuous:</u> then it should not contain any “missing value”.

- Missing values must always be encoded as empty values (“”).

- <u>OPTIONAL:</u> The ***creation dataset*** should not contain any “consequence columns”. If the dataset nevertheless contains some “consequence columns”, it’s good to know their name in advance. However, you can always use **TIMi** to find all the “consequence columns” easily.

- <u>OPTIONAL:</u> the flat file can be compressed in RAR (.rar), GZip(.gz), Winzip(.zip)

- <u>OPTIONAL:</u> all the columns that represent a “True/False” information may contain only two different value: ‘0’ (for false) or ‘1’ (for true) or are empty (“”) if the value is missing.

- OPTIONAL: all the columns that represent either:
  - a number
  - an information that can be ordered

… should be encoded as pure number. For example:

number of cats	number of cats
missing	
no cat	0
one cat	1
2 cats	2
3 or more	3

Social class	social class
missing	
poor	0
middle	1
rich	2