

5/10/2015

TEAM PLUS

Nick D. wondergem, upadhyay ishant, kichukov veselin, velev dan, ngabo francky

Validation and Verification report

ORIKAMI

Document history

Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Status | Date | Changes |
| 0.1 | concept | 07-11-2015 | Template created |
| 0.2 | concept |  |  |
| 0.3 | concept |  |  |
| 0.4 | concept |  |  |
| 1.0 | draft |  |  |

Approval

This document needs to be approved by:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Approval date | Name | Function | Paraph |
| 1.0 |  | Theo Cats | Tutor |  |
| 1.0 |  | Bram den Teuling | Product owner |  |

Distribution

This document has been sent to:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Send date | Name | Function |
| 1.0 | 28-09-2015 | Theo Cats | Tutor |
| 1.0 | 28-09-2015 | Bram den Teuling | Product owner |

Contents

[Introduction 3](#_Toc441581993)

[Bagging 4](#_Toc441581994)

[Boosting 4](#_Toc441581995)

[Stacking (Stacked Generalizations) 5](#_Toc441581996)

[Meta-Learning 5](#_Toc441581997)

[Applying Regression[Dan and Veselin] 6](#_Toc441581998)

[Regression Relations[Ishant and Francky] 6](#_Toc441581999)

# Introduction

The validation and verification stage involves considering various models and choosing the best one based on their predictive performance in other words explaining the variability in question and producing stable results across samples.

This may sound like a simple operation, but in fact, it sometimes involves a very elaborate process. There are a variety of techniques developed to achieve that goal - many of which are based on so-called "competitive evaluation of models," that is, applying different models to the same data set and then comparing their performance to choose the best.

These techniques which are often considered the core of predictive data mining include

* Bagging (Voting, Averaging)
* Boosting
* Stacking (Stacked Generalizations)
* Meta-Learning.

# Bagging

Description ( Voting, Averaging )

The concept of bagging (voting for classification, averaging for regression-type problems with continuous dependent variables of interest) applies to the area of predictive data mining, to combine the predicted classifications (prediction) from multiple models, or from the same type of model for different learning data. It is also used to address the inherent instability of results when applying complex models to relatively small data sets. Suppose your data mining task is to build a model for predictive classification, and the dataset from which to train the model (learning data set, which contains observed classifications) is relatively small. You could repeatedly sub-sample (with replacement) from the dataset, and apply, for example, a tree classifier (e.g., C&RT and CHAID) to the successive samples. In practice, very different trees will often be grown for the different samples, illustrating the instability of models often evident with small data sets. One method of deriving a single prediction (for new observations) is to use all trees found in the different samples, and to apply some simple voting: The final classification is the one most often predicted by the different trees. Note that some weighted combination of predictions (weighted vote, weighted average) is also possible, and commonly used. A sophisticated (machine learning) algorithm for generating weights for weighted prediction or voting is the Boosting procedure.

# Boosting

Description

The concept of boosting applies to the area of predictive data mining, to generate multiple models or classifiers (for prediction or classification), and to derive weights to combine the predictions from those models into a single prediction or predicted classification (see also Bagging).

A simple algorithm for boosting works like this: Start by applying some method (e.g., a tree classifier such as C&RT or CHAID) to the learning data, where each observation is assigned an equal weight. Compute the predicted classifications, and apply weights to the observations in the learning sample that are inversely proportional to the accuracy of the classification. In other words, assign greater weight to those observations that were difficult to classify (where the misclassification rate was high), and lower weights to those that were easy to classify (where the misclassification rate was low). In the context of C&RT for example, different misclassification costs (for the different classes) can be applied, inversely proportional to the accuracy of prediction in each class. Then apply the classifier again to the weighted data (or with different misclassification costs), and continue with the next iteration (application of the analysis method for classification to the re-weighted data).

Boosting will generate a sequence of classifiers, where each consecutive classifier in the sequence is an "expert" in classifying observations that were not well classified by those preceding it. During deployment (for prediction or classification of new cases), the predictions from the different classifiers can then be combined (e.g., via voting, or some weighted voting procedure) to derive a single best prediction or classification.

Note that boosting can also be applied to learning methods that do not explicitly support weights or misclassification costs. In that case, random sub-sampling can be applied to the learning data in the successive steps of the iterative boosting procedure, where the probability for selection of an observation into the subsample is inversely proportional to the accuracy of the prediction for that observation in the previous iteration (in the sequence of iterations of the boosting procedure).

# Stacking (Stacked Generalizations)

Description Stacked Generalizations

The concept of stacking (Stacked Generalization) applies to the area of predictive data mining, to combine the predictions from multiple models. It is particularly useful when the types of models included in the project are very different.

Suppose your data mining project includes tree classifiers, such as C&RT or CHAID, linear discriminant analysis (e.g., see GDA), and Neural Networks. Each computes predicted classifications for a crossvalidation sample, from which overall goodness-of-fit statistics (e.g., misclassification rates) can be computed. Experience has shown that combining the predictions from multiple methods often yields more accurate predictions than can be derived from any one method (e.g., see Witten and Frank, 2000). In stacking, the predictions from different classifiers are used as input into a meta-learner, which attempts to combine the predictions to create a final best predicted classification. So, for example, the predicted classifications from the tree classifiers, linear model, and the neural network classifier(s) can be used as input variables into a neural network meta-classifier, which will attempt to "learn" from the data how to combine the predictions from the different models to yield maximum classification accuracy.

Other methods for combining the prediction from multiple models or methods (e.g., from multiple datasets used for learning) are Boosting and Bagging (Voting).

# Meta-Learning

Description

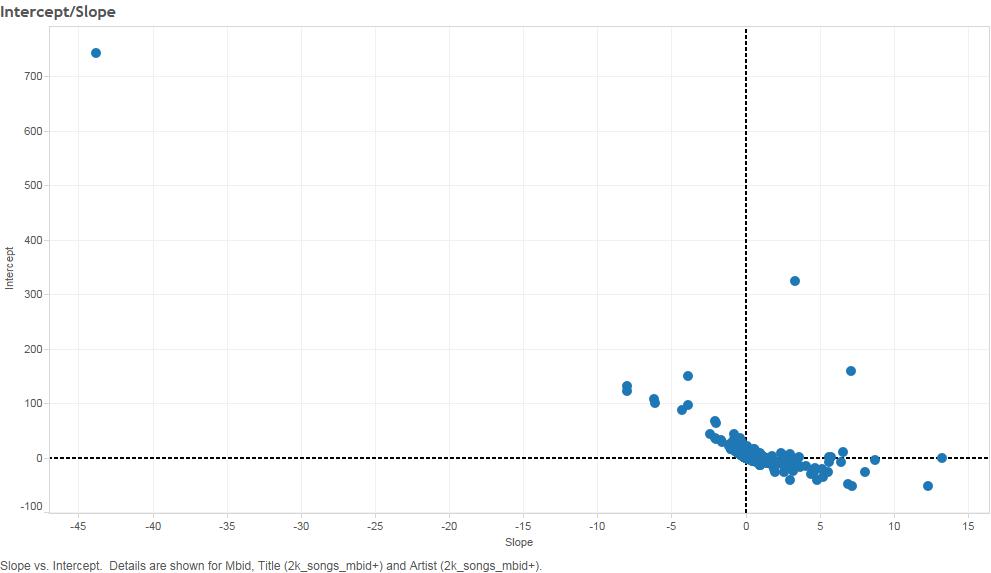
The concept of meta-learning applies to the area of predictive data mining, to combine the predictions from multiple models. It is particularly useful when the types of models included in the project are very different. In this context, this procedure is also referred to as Stacking (Stacked Generalization).

Suppose your data mining project includes tree classifiers, such as C&RT and CHAID, linear discriminant analysis (e.g., see GDA), and Neural Networks. Each computes predicted classifications for a crossvalidation sample, from which overall goodness-of-fit statistics (e.g., misclassification rates) can be computed. Experience has shown that combining the predictions from multiple methods often yields more accurate predictions than can be derived from any one method (e.g., see Witten and Frank, 2000). The predictions from different classifiers can be used as input into a meta-learner, which will attempt to combine the predictions to create a final best predicted classification. So, for example, the predicted classifications from the tree classifiers, linear model, and the neural network classifier(s) can be used as input variables into a neural network meta-classifier, which will attempt to "learn" from the data how to combine the predictions from the different models to yield maximum classification accuracy.

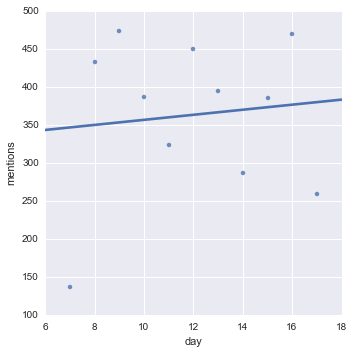
We can apply meta-learners to the results from different meta-learners to create "meta-meta"-learners, and so on; however, in practice such exponential increase in the amount of data processing, in order to derive an accurate prediction, will yield less and less marginal utility.

# Applying Regression

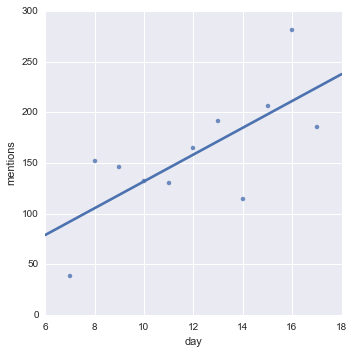
We have decided to apply regression in order to find if we could be able to define songs Popularity through mentions in a given period of time. We have applied the algorithm to each of the 2 000 songs in the period of 7th April 2015 until 17th April 2015. Our first tries gave us errors due to the data type of the date, which was time series, so we had to take out only the days. After applying the algorithm successfully, we found out that there is a relation between the slope and the intercept of each song’s regression line (see graph below).



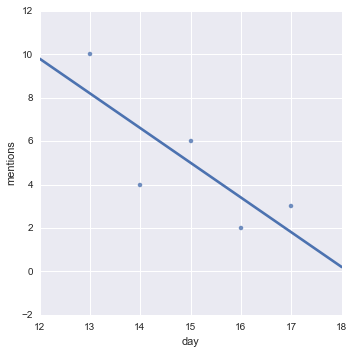
In the upper right area we found out that are located popular songs, which had positive intercept and slope. In another graph below can be seen the regression line of the most popular song in this area. As we can see the song had a lot of mentions during this period and the slope is almost flat. The intercept is with high value as well, so we could assume that the song was popular from before.



In lower right area we can see the songs with increasing popularity based on the number of mentions. In the graph below we plotted the regression line of one song(Ellie Glouding – Love me like you do). It can be seen that the first day of our period it got around 40 mentions. Whereas on 16th it got almost hundred mentions. Compared to the graph of the popular song we see that the slope is much bigger, while the intercept is a negative value, meaning that the song was not popular before, or it may came out more recently.



And finally the upper left area represents the group of songs with decreasing popularity. Those are the songs that have positive intercept and negative slope. In the graph below can be seen the regression line of one song with decreasing popularity which for one day had 10 mentions and through the days they decrease. We can clearly see that the slope goes down gradually.



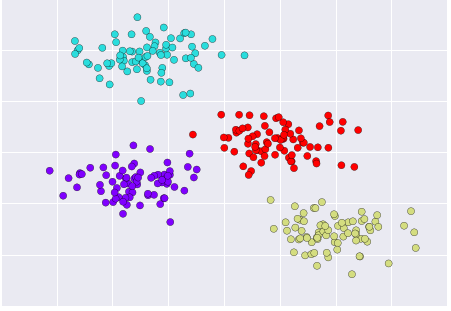
# Regression Relations

Description

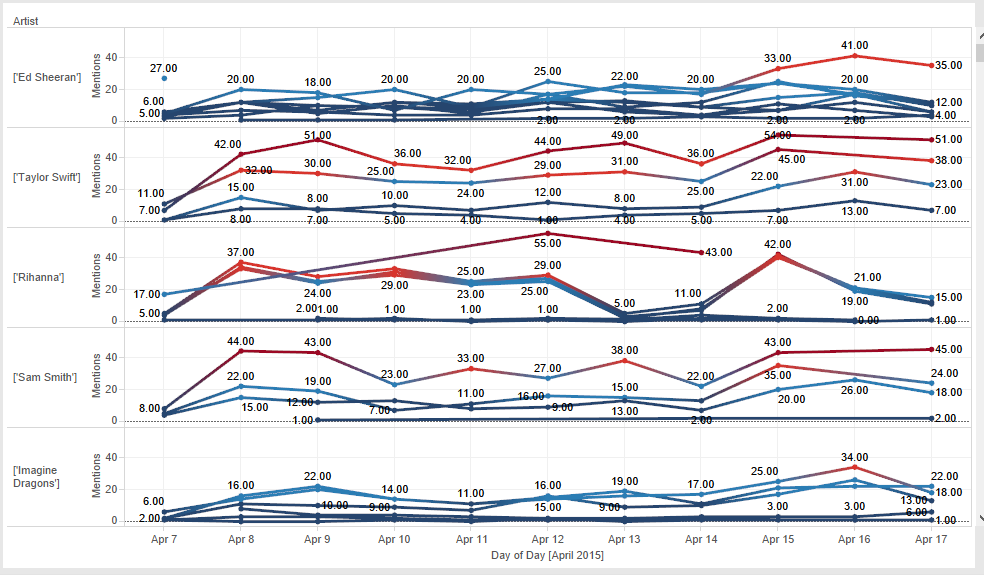
Before choosing the regression algorithm we applied some other algorithm like the k-mean but we couldn’t get far with it due to the fact that k-mean will take the data point randomly which will be resulting in clusters that have roughly the same number of data points. Which was not our goal.

We tried as well to predict based on the mentions what will be the next mention, would be higher or less but we no success due to the fact that we were missing a lot of data.

So we went with another approach after being able to have all the songs in one big file with all the mentions per day.

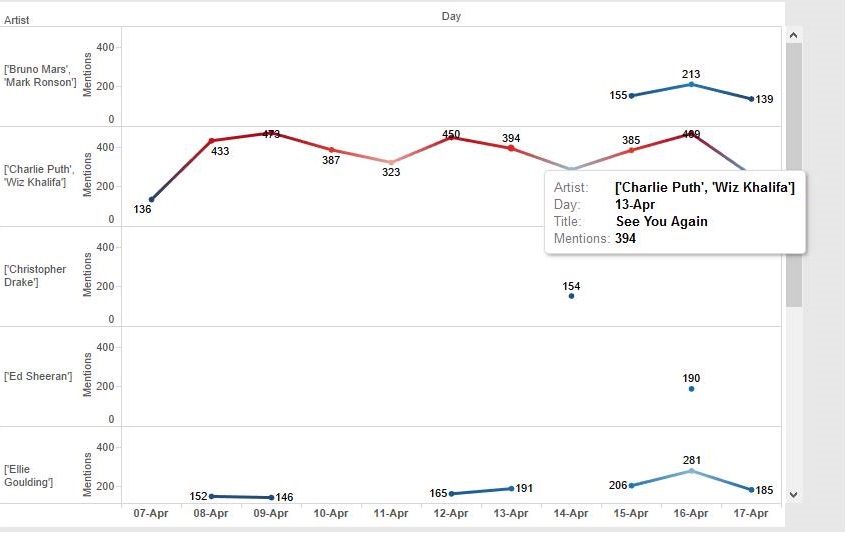


**All the song**

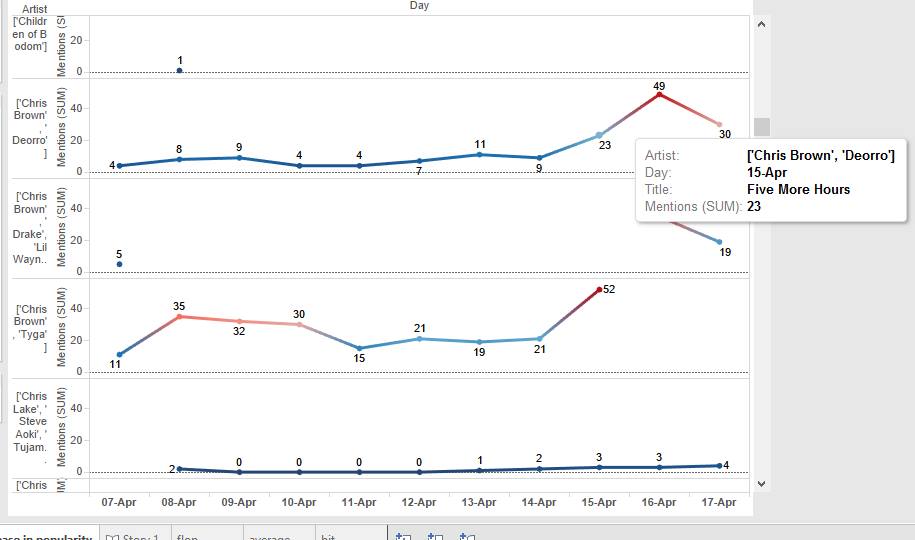
****

So we decided to go for the regression solution which had made, with success, classification on the data based on the slope and the intercept and we could see the life cycle of each song day per day and be able to class them (we had from 07/04/2015 to 17/04/2015) with a increasing popularity, decreasing popularity and popular songs so that our classes will make sense instead of having classes based on the mentions now we had classes based on the slope and the intercept, of the curve how it goes up and down as u can see it on the following graph

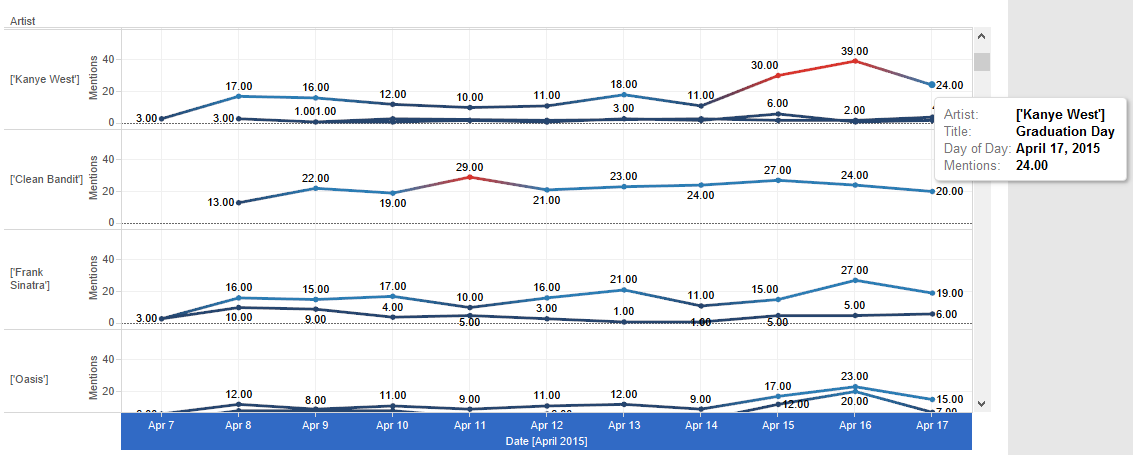
**Most popular song:**



**Increasing in popularity**



**Decreasing in popularity**

****