National University of Singapore

IE5202 - Forecasting Methods

Project 1

# Predicting Social Network posts

*Author:*
Gabriel Azevedo Ferreira

*Supervisor:*
Chen Nan

October 1, 2017

# 1 Introduction

This project aims at predicting the number of comments a social network post will receive on the next hours based on features like:

- Number of hours since it has been posted
- Number of likes the post received
- Number of comments in the last hours
- The day it has been posted

We worked with a data-set based on Facebook pages, where we initially had 41 features to predict and over 40.000 samples.

In order to do such predictions, we focused on linear regression models. In the end, however, we tried a handful of strong non linear models, such as neural networks and random forests, so that we could achieve better results by not assuming any specific linear behavior of our initial set.

This report summarizes the analysis made in the *python* notebook submitted separately, on which the machine learning library "Sci-kit learn" was extensively used.

# 2 Data visualization

Previously to any model fitting, we used basic visualization tools to see if we would have insights about the data.

## 2.1 Principal Component Analysis (PCA)

We applied PCA to the data set so that we could see if any particular pattern would be visually identified. The results are shown in Figure 1. We can see that that the data stays concentrated in around a same spot with low values of dimension 1 and low comment values. there are some outliers with very high values of comments, and some of them with high values of dimension one. a similar pattern is identified regarding dimension 2.

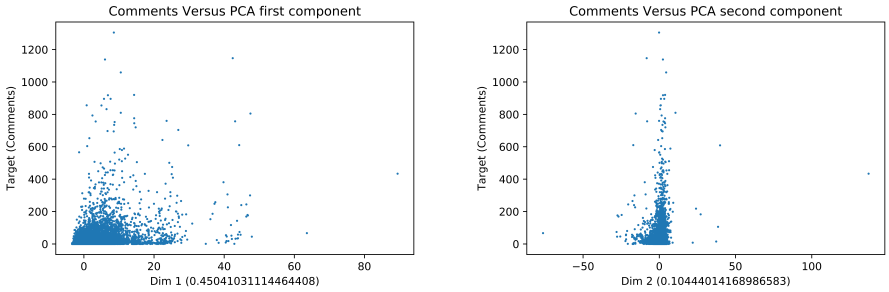


Figure 1: PCA dimensions 1 and 2 versus target variable

## 2.2 Correlation matrix and scatter plot matrix

The plots on Figure 2 allow us to visualize how correlated the features are. We notice that columns from 5 to 29 are very correlated, which is due to the fact that they are transformations of essential features (derivative features). Another interesting remark is that 0 (the target), 35 (the time gap between publishing and the base time), and 31 (number of comments up to the moment) are strongly correlated. That is the equivalent to say:

- The number of comments in the next hours depends strongly on the how old the post is and how many comments it has so far.
- The number of comments of a post depends strongly on how old it is.

Which are expected conclusions from social network posts.

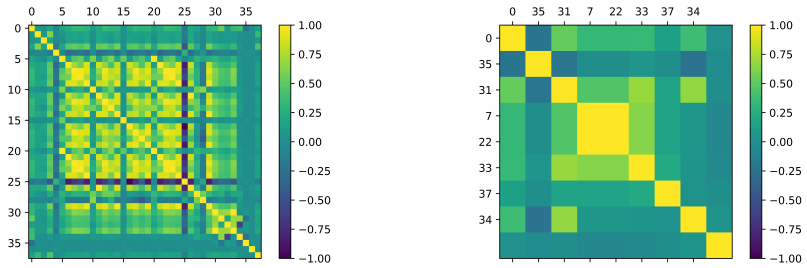

Figure 2: Plotting correlations with all features (left) and the features that will be later selected on the feature selection procedure. the color maps the correlation

Figure 3 shows a scatter plot on each two pair of features, 0 being the target. The conclusions are similar to those taken when considering Figure 2, with more details on the shape involved on the correlation between the variables.

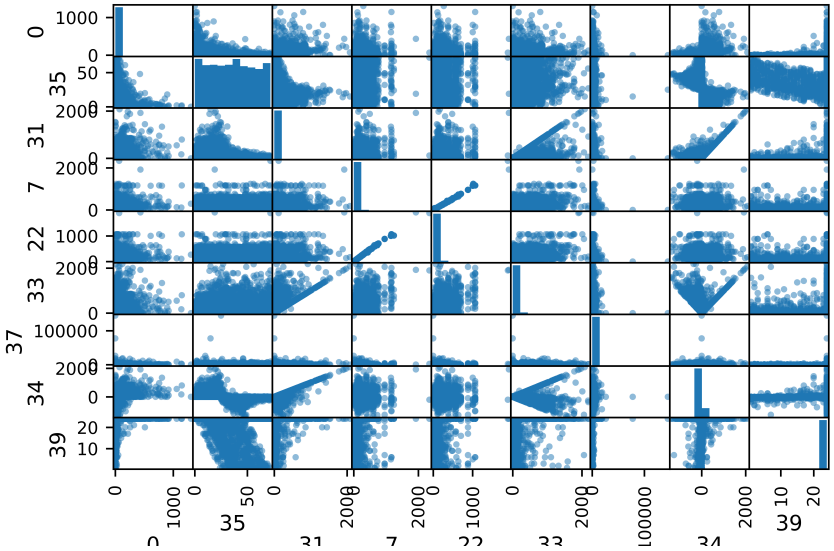


Figure 3: Scatter plot of each 2 of the selected variables

# 3 Single variable fitting

The very first model fitting done on the project was a simple single variable fitting

$$\hat{Y} = \alpha X + \beta$$

Where X is one of the features.

We did not aim to have good results at this point, but rather to be able to visualize dependencies and interpret them.

Two features seemed to be particularly important in this situation:

1. C2 - The number of comments in last 24 hours, relative to base date/time (column number 31). Represented in Figure 4. The $R^2$ score was 0.280

2. C5 - The number of comments in the first 24 hours after the publication of post but before base date/time. (column number 34). Represented in Figure 5. The $R^2$ score was 0.1432

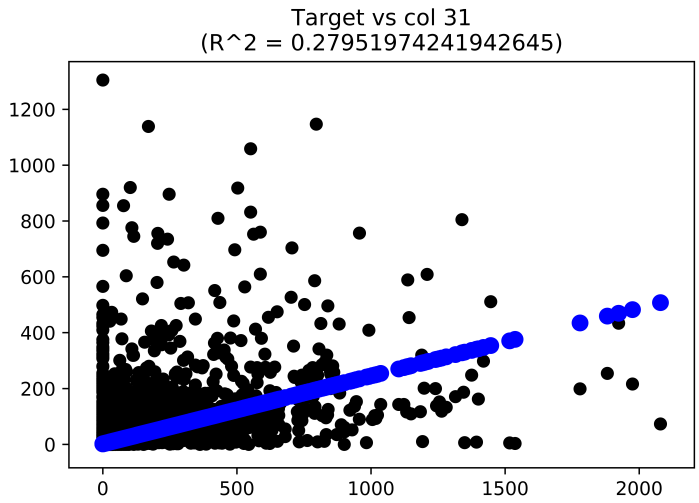


Figure 4: Single variable fitting. C2 VS target

In those cases, even though the prediction was very rough, we can see a relation between the variables. If we look at the meaning of those variables, we can understand why a certain correlation is present: if a post has had many comments lately, it is very likely that it is going to have more comments soon. This is most likely due to the fact that such post is probably "interesting" on the user's opinion so it attracts more people to comment on it, and, in doing so, it causes the post to be shown to more people, increasing the chance of more posting in the future.

On the next step we add more features to the the model, and this demands first a feature selection phase.

# 4 Feature Selection

It is known as "course of dimensionality" the fact that predictions can become unstable and very computationally costly when the data set has too many dimensions. Many algorithms aim at selecting only a few features that are the

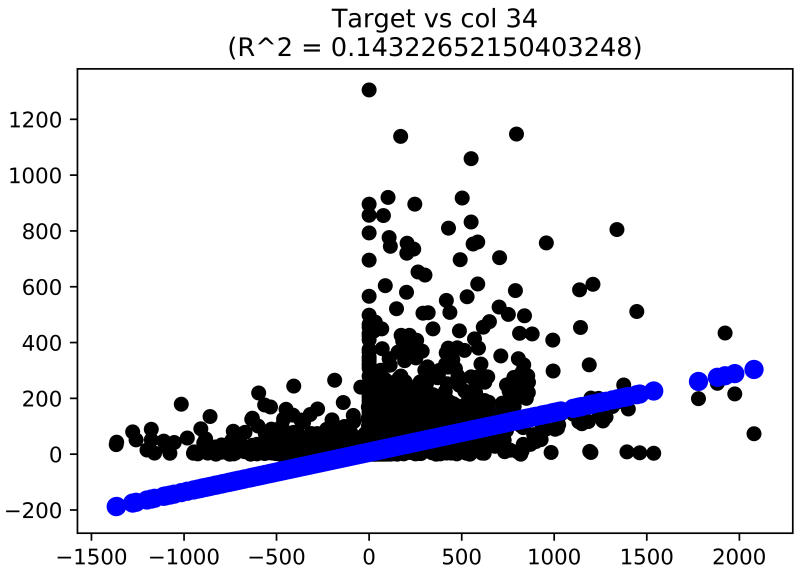


Figure 5: Single variable fitting. C5 VS target

most important when performing the prediction. In fact, when it comes to supervised learning (which is the case of this project), one could think that a simple "brute force" algorithm (where all combinations of features are tested and only the best are selected) is a good approach. However the time involved in such search would be proportional to the factorial number of features, which is usually unfeasible for more than 10 features. Therefore, we usually prefer using some heuristics, such as Recursive Feature Selection (RFE), to do that task. They are not guaranteed to be optimal, but usually take much less time (linear time in many cases) than the "brute force algorithm".

We used two methods for feature selection, and compare the chosen features picking the best of them in the end.

Our first approach consisted on applying the RFE algorithm which recursively removes features (one at a time in our case). The feature removed at each step is the one without which the model final score (in our case the $R^2$ score) is the least affected. Our goal was to find the best model with 6 variables at most.

The second method was a mix of the optimal "testing all combinations" algorithm and the recursive elimination algorithm. We proceeded as follows:

1. We found, by "brute force", the best regressions with 4 variables (the ones with the least $r^2$ core)
2. We took from it only the 3 most frequent variables (the ones that appeared most often on the 10 best regressions)
3. We found by "brute force" the best 6 variables regression where the 3 previously selected variables were present

In order to choose between the two sets of features we made a simple regression model and then we chose the model with best $R^2$ score. The second method performed better that the first one.

The chosen features were the following:

- 35 : The time gap (in hours) between publish time and base time
- 31 : The number of comments in last 24 hours, relative to base date/time.
- 7 : Derived Page feature
- 22 : Derived Page feature
- 33 : The number of comments in the first 24 hours after the publication of post but before base date/time.
- 37 : This features counts the no of shares of the post, that how many peoples had shared this post on to their timeline.

# 5 Simple regression model

In this phase we tried to find the best possible model using the 6 previously selected variables. The model used was:

$$\hat{Y} = \alpha_1 X_{35} + \alpha_2 X_{31} + \alpha_3 X_7 + \alpha_4 X_{22} + \alpha_5 X_{33} + \alpha_6 X_{37} + \beta$$

This model produced a $R^2$ value of 0.316525339173

The coefficients are shown on Table 1.

## 5.1 Cross Validation

We used the KFold method for cross validation, with $K = 10$, and the mean score obtained was 0.306914155465.

# 6 Polynomial regression

We tested polynomials of several orders, including all features with the following form:

$$X_{i_1}^{j_1} * X_{i_2}^{j_2} * ... * X_{i_n}^{j_n} \ s.t. \ j_k \geq 0 \ , \ \sum j_k = n \ and \ i_k \in \{35, 31, 7, 22, 33, 37\}$$

As a general tendency the higher was the order the best was the performance on the training set (that is the $R^2$ score was higher). However, when tested with cross validation (Kfold, $K = 5$) the score becomes very low from a certain degree on. This is a sign that fitting a model on those degrees provides overfitting.

We found that the degree equals two provided a significant improvement on the $R^2$ score and 80% of the validations samples on Kfold had scores greater than 0.3, which is a reasonable trade off between training score and cross-validation score.

Score on training set: 0.494575507269

# 7 Elastic net

In the last section we showed the model might be overfitting when training on polynomials of high degree.

Some adaptations of the linear model were developed to avoid this "good behavior on training set, bad behavior on testing set" phenomena. The **Ridge** and **Lasso** techniques, consist on adding a penalty to the minimization function, relative to the size of the cofficient. The ridge penalty tends to provide coefficient vectors with smaller norm, while the Lasso penalty tends to select sparser coefficients.

Table 1: Coefficients of the regression

| | |
|---|---|
| $\alpha_1$ | -0.18485 |
| $\alpha_2$ | 0.23346 |
| $\alpha_3$ | 0.57228 |
| $\alpha_4$ | -0.53674 |
| $\alpha_5$ | -0.0346 |
| $\beta$ | 6.61008 |

We tried a combined version of those two penalties: the **Elastic Net**, represented by the following equation:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1)$$

Usually $\lambda_1$ and $\lambda_2$ are found via cross validation. The values that worked out in our case were $\lambda_1 = 0.5$ and $\lambda_2 = 0.25$.

They provided the following result, for the features 35,31,7,22,33,37,34 and 39:

- Score = 0.488825446373
- Cross validation score : 0.144550610106 (against -5.8795 on the polynomial regression without penalties)

# 8 Random forest

To try to find the best model, we used one that does not do the assumption of linearity on the model.

The random forest algorithm works by constructing many decision trees and averaging the output of them. This procedure avoids over-fitting

When testing all features with this model (with 200 decision trees), we had the following scores

- $R^2$ score on training set: 0.951353433132
- cross validation $R^2$ mean score: 0.596859982923

This was the best score obtained and this was the model used to predict the test set.