# Report: funding school projects on DonorsChoose.org
## Using Machine Learning to help students in need

The online platform Donors.org helps fund school projects for students in need. The platform provides a space where donors can find school projects and contribute to them. Although all projects seek to fulfill some kind of educational need, not all projects meet their funding goal. The client for this project seeks to help those projects which are at the highest risk of not getting fully funded to intervene on those projects.

For this setting, a Machine Learning solution should seek to predict, at the time of posting, which projects are at the highest risk of not getting fully funded within 60 days. Specifically, our client seeks to identify the 5% of the posted projects that are at the highest risk to not get fully funded, to intervene on those projects.

### Data

The data available spans Jan 1, 2012 to Dec 31, 2013. It has 124,976 entries. Each entry is a project. The data includes information at three levels: 1) school context (location, charter, focus subjects and areas, poverty level), 2) project (resource type, grade, total amount of money, number of students reached, date posted, date fully funded, etc.) and 3) teacher (prefix).

To process the data, for each entry, the numeric columns with missing information were imputed using median values for the relevant testing/training dataset[1] to avoid leakage.

The data used to train the solution had 53 variables and covered:

- At the school level: if school charter; if school magnet; level of poverty; and type of metro area (rural, suburban, urban).
- At project level: level of the total price; level of the students reached; which grades the project serves; the type of resources the project will use; whether eligible for a matching program by a corporate partner; and primary and secondary focus areas.
- At teacher level, their gender (whether female).

### Machine learning solution

Using the data previously mentioned, the solution aims to predict whether a project will be not fully funded within 60 days. The prediction will be made at the time of posting the project. For this specific problem, the best solution will be selected based on its precision. In other words, the solution is more concerned with *False Positives*: projects that we predicted as not getting fully funded within 60 days but actually got fully funded within 60 days. False Positives are more concerning because the client wants to avoid intervening and investing resources in projects which were going to get funded anyway.

[1] This means train data was imputed with train data, and test data was imputed with test data, in their relevant test sets. See more in Methodology in the explanation of test sets on a 6 month rolling window.

The solution will be trained across different time-splits as data permits to assess the models' performance through time, as it gets more data. The dates allow for three validation/testing sets with a rolling window of 6 months. Since the prediction has a gap (60 days to get funding) some observations will be lost. For example, the first test/validation test will have a split date of 7/1/2012. The model will be trained with data on projects posted the first semester of 2012 minus 60 days of prediction gap (from 1/1/2012 to 5/2/2012), to ensure all projects have the correct label "not funded within 60 days". Testing data projects posted in the second semester minus 60 days of prediction gap, to assure they also have the correct label. See Annex for full details.

As for the type of models, we will run the following classifiers and several combinations of their parameters: Logistic Regression (LR), K-Nearest Neighbor (KNN), Decision Trees (DT), Support-vector Machine (SVM), Random Forests (RF), Boosting (B), and Bagging. In total, 558 models were trained.

**Analysis**

The best models are those that have consistently high precision across time, as these models  Models with consistently high precision across time splits will be considered to be best, as they improve their performance as data becomes available. SVM classifiers had the highest median precision at 5% (0.468).

We also analyzed the performance by type of classifier through time. The SVM models were consistently among the best performing models in terms of precision[2].

For each split, the best models (in terms of precision) were:

- First split: the best five models were two variations of SVM and three variations of Logistic Regression.
- Second split: the best five models were four variations of SVM and one variation of Gradient Boosting.
- Third split: the best five models were one variation of Gradient Boosting, three variations of Logistic Regression, and one variation of SVM.

Figure 4 in annex shows the five best performing models throughout time. We can see that SVM models ranked highest. The precision for all models increased when the data was split  and their precision

**Recommendation**

For this given problem, the best performing model is a SVM with a regularization parameter (C) equal to 10. Its precision for the first, second and third split was 0.480, 0.547, and 0.459, respectively. In other words, for the last split, out of the projects the model predicted as not getting funded in 60 days, 45.9% of them did not get funded in

[2] See Figure 3 in Annex for plot of the best performing classifiers by type of model by year.

that timespan. This measure is better performance than the random baseline, which is around 0.280.

Our second best performing model is also a SVM whose regularization parameter is only 0.1.

Given that, at best, the data only had three quarters to test its performance, the recommendation is to use a SVM classifier with a C=10 and systematically test its performance every 6 months along with Grading Boosting and Logistic Regression classifiers. As data becomes available, the performance should be monitored over time. More data should translate to higher precision and this should be monitored closely over time.

# Annex

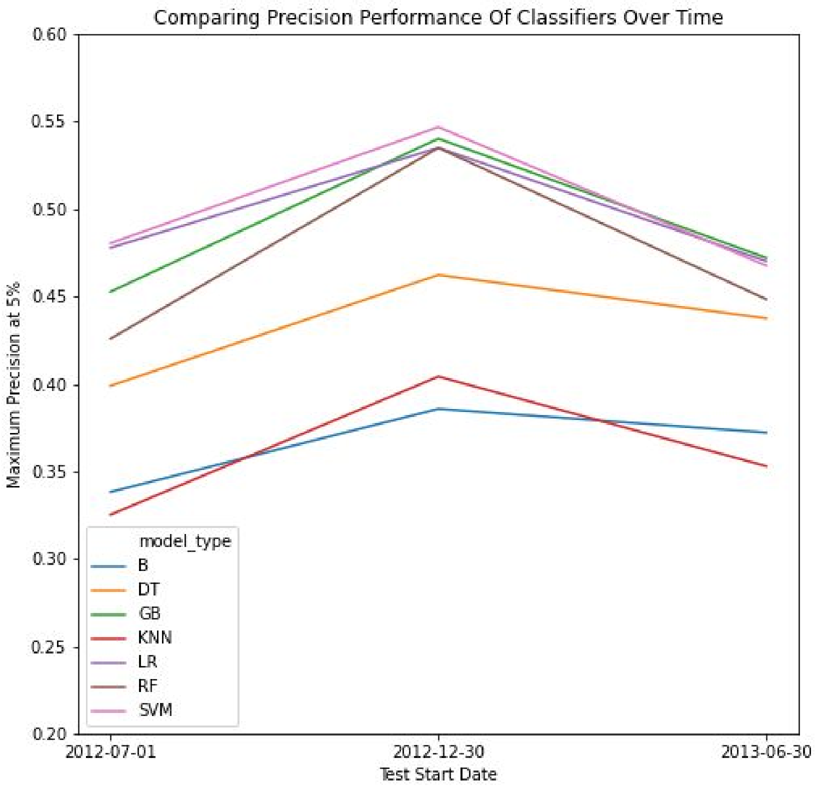
## Figure 1: Calendar of test/validation sets

| 2012 | | | | 2013 | | | |
|---|---|---|---|---|---|---|---|
| 1st semester | | 2nd semester | | 1st semester | | 2nd semester | |
| Train | Prediction Gap* | Test | Prediction Gap* | | | | |
| Train | | Train | Prediction Gap* | Test | Prediction Gap* | | |
| Train | | Train | | Train | Prediction Gap* | Test | Prediction Gap* |

## Figure 2:Exact dates of test/validation sets

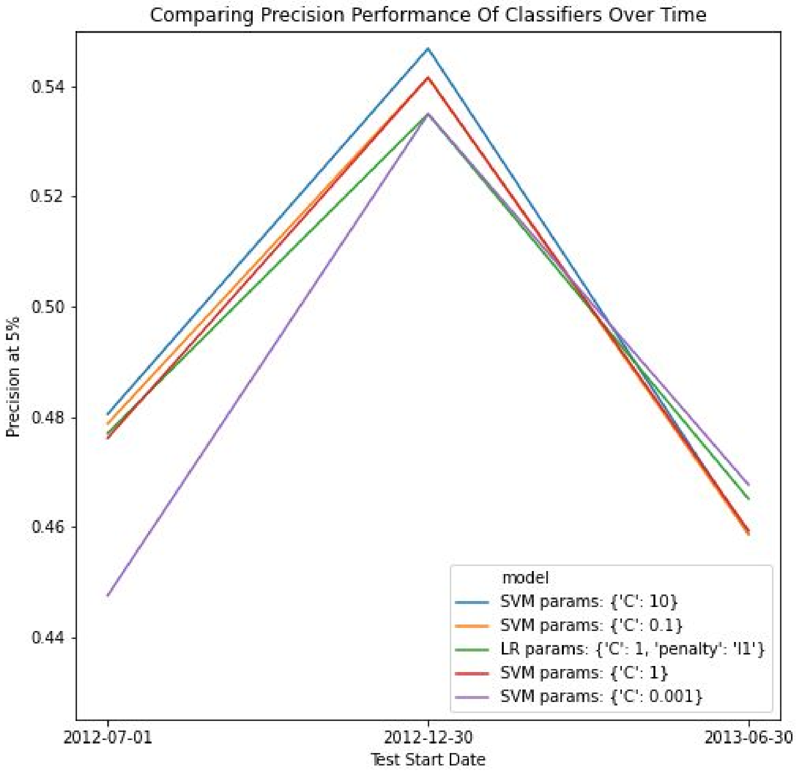
| Train Start Date | Effective Train End Date (due to the prediction gap) | Test Start Date | Effective Test End Date (due to the prediction gap) |
|---|---|---|---|
| January 1, 2012 | May 2, 2012 | July 2, 2012 | October 31, 2012 |
| January 1, 2012 | October 31, 2012 | December 31, 2012 | May 1, 2013 |
| January 1, 2012 | May 1, 2013 | July 1, 2013 | October 30, 2013 |

## Figure 3: Type of Models' Performance Across Time-Splits



Note that the best classifier by type of model by split is plotted. Y-Scale was zoomed in to appreciate better the differences.

**Figure 4: Classifiers performance across time-splits**



These specific models were chosen based on their ranking. The models were ranked based on their precision at 5% on each split. Then, the “average” rank of the models was computed and the lowest rank “average” models were selected. The same analysis was made with medians, and results were consistent for the top 2 classifiers, which is considered enough to support the top 1 model. Note that the Y axis was zoomed-in given the small differences in precision.