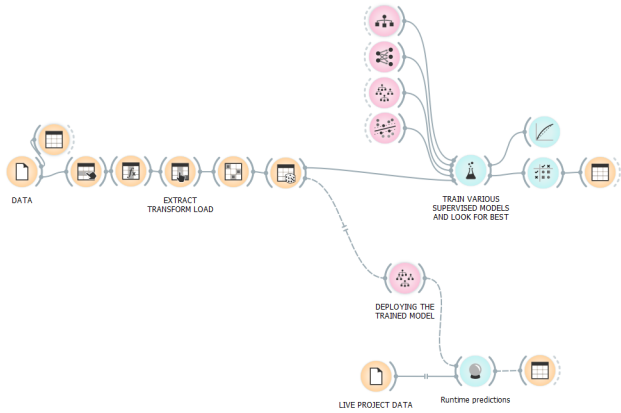
Machine learning Use Case: Project Success Prediction for large portfolios

Exec Summary

Predicting which projects are more likely to succeed would have a high return both in terms of reduced costs and benefits accruing from more successful projects. Here we have trained a machine-learning model on their archive of projects, and which will then predict which projects are most likely to succeed and which to fail. For projects that the model had not seen, the model achieved a classification accuracy of 89%. We believe it is worth exploring further the applications of machine learning for advising companies with large portfolios.

Introduction

We examine the way that a portfolio of projects is overseen by a large Institution. We have started with the World Bank as they have data on 12000 projects and which of these were ultimately successful. Each project has been rated for Bank Performance as Satisfactory, or Unsatisfactory, Highly Satisfactory etc. after the project concluded by the client. This rating label has been used as the target for Learning. Each project has about 20 features recorded for it. The decision we are seeking to improve is in identifying which projects are unlikely to succeed, and whether they should be cancelled, rescoped, supported, or monitored as a result. This task can be selectively generalised to Pcubed Client portfolios.



The data is publically available if you are interested in creating a similar model. We were not engaged by the World Bank for this study - but did it because of the excellent data set.

Main results

We have trained a classification model which predicts which projects will achieve a final rating of Satisfactory.

Our model is trained to understand which combinations of these features are most predictive of success. This is a supervised learning approach, where we have ended up training and selecting a Random Forest Model as providing the best results.

Even taking a small number of the dataset, say 667 records, the model can train so that it predicts 309 as satisfactory, of which 281 prove to be actually satisfactory. It misses another 15 which it does not predict to be satisfactory, but which prove to be actually satisfactory. These results can be tuned to get an appropriate number of false positives and false negatives.

Suggested implications for the World Bank

1. Protect and pay attention to project reviews
2. Consider hard closing down projects that fail these reviews
3. Lessen focus on other metrics
4. Look for latent features together: Can the client see any commonality between the four clusters of project performance from the Unsupervised learning. They may be a latent variable we can discover and track for the future
5. Unsupervised models would warrant investigation of the outlier “OT Rehab Project”, and other outliers shown within the Unsupervised learning models
6. Periodically put through current projects in run-time against model
7. Review together the effectiveness of projects deemed Unsatisfactory during reviews by analysing this subset. Because we dont yet know whether intervention makes a difference to outcomes.
8. Annually retrain model
9. Pay attention to the reference forecast of % failed projects during annual planning

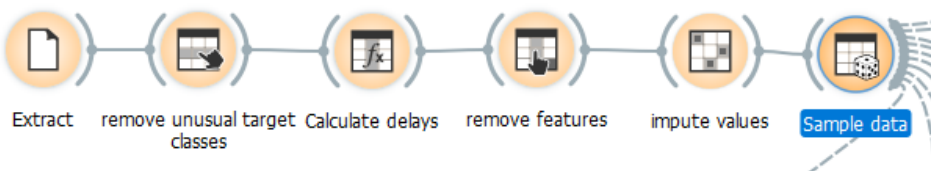
Data

https://www.kaggle.com/theworldbank/ieg-world-bank-project-performance-ratings We took the set from April; this site periodically updates these results. You can take the latest ones and test. Here is a sample record, with some of the lesser columns removed.

Project_Name	ELECTRICITY GENERATION REHAB & RESTRUCTU
IEG_Bank_Perf_Rating	SATISFACTORY
Approval_FY	2006
CANCELLED_USD_AMOUNT	419,815,830
NET_COMMITMENT_AMOUNT	- 83,815,830
ProdLine	IBRD/IDA
LendingInstr	SIL
Agreement_Type	IBRD
FragileState	IBRD non-FCV
Eval_Type	ES
Eval_FY	2011
CLOSING_DATE	40908
REV_CLOSE_DATE	40908
Approval Date	38874
Deactivation_Date	40137
IEG_Outcome_Rating	NOT RATED
IEG_Bank_QAE_Rating	SATISFACTORY
IEG_Bank_QOS_Rating	SATISFACTORY
IEG_ICR_Quality_Rating_Modified	SATISFACTORY

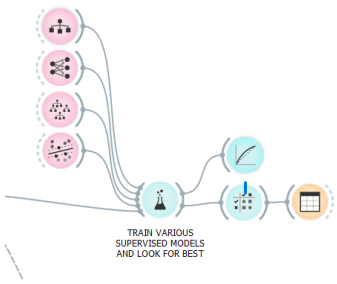
Extract, Transform and Load the data

1. Data extracted. The client outcome per project was tagged as the target feature for the classification.
2. Data transformation: For the target feature, some of the classes are rarely used, so they have been removed. Project delay beyond forecast was calculated as a new feature. We also skipped some features that looked repetitive or unhelpful. The “impute values” widget was used to remove incomplete instances, although we could probably use those instances if we were tight for data.
3. Data Load: Sampling. For speed and simplicity of demonstration, we are using 10% of the projects



Train various supervised models and look for the best (test and score)

Five other models were tested beyond the four shown. For demonstration purposes we have focussed on the top four. This is easier to understand when we look at the Confusion matrix for the best performing, Random Forest for the 667 sampled instances. A client write-up would include an explanation of these different methods

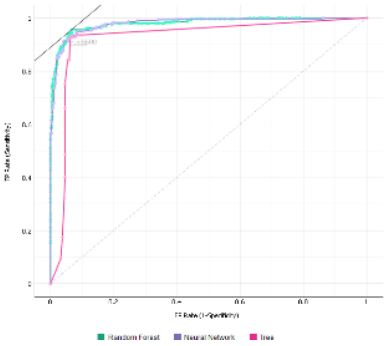


It would be instructive to look at the misclassified instances for a real client case. The more on the diagonal, the better it is. We can see for example, that it predicts 9 as highly satisfactory, although one of them is actually satisfactory. There are also 13 more highly satisfactory in reality, but those have been predicted as satisfactory.

Predicted

Actual	HIGHLY SATISF...	HIGHLY UNSAT...	MODE...	MODE...	SATISF...	UNSAT...	Σ
HIGHLY SATISF...	8	0	0	0	13	0	21
HIGHLY UNSAT...	0	3	0	0	1	3	7
MODE...	0	0	177	4	2	0	183
MODE...	0	0	9	69	2	4	84
SATISF...	1	0	12	0	281	2	296
UNSAT...	0	0	2	4	10	60	76
Σ	9	3	200	77	309	69	667

More generally, this information can be seen by looking at the ROC curves, (curves for Receiver Operating Characteristic), in this case the ROC curve for predicting "satisfactory" projects. The y axis shows "sensitivity, which is the probability that it finds true positives. It shows the false positive rate on the x-axis, which is also called the "specificity". "The best possible prediction method would yield a point in the upper left corner or coordinate (0,1) of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity (no false positives). "



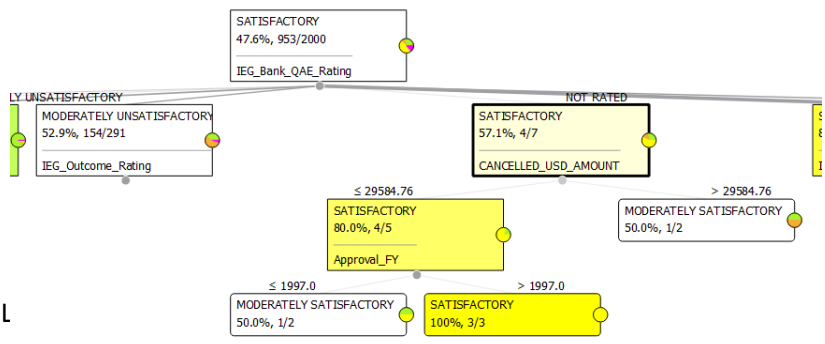
Now it is easier to understand the summary results. AUC is under the ROC curve. The closer to 1 the better, as it is easier to separate false positives from false negatives. Classification accuracy is the proportion of correctly classified examples. Precision is the proportion of true positives among instances classified as positive. Recall is the proportion of true positives among all positive instances in the data (i.e. the predicted "satisfactories" amongst all the "satisfactories"). F1 is a blend of two of the others.

Method	AUC	CA	F1	Precision	Recall
Random Forest	0.979	0.897	0.891	0.897	0.897
Tree	0.915	0.880	0.878	0.877	0.880
Neural Network	0.974	0.877	0.876	0.876	0.877
SVM	0.979	0.864	0.858	0.866	0.864

Data exploration, beyond the classification model

Prediction is different from Understanding. We know we can predict well from within this dataset, but this doesn't mean we understand what is going on. There are some other workflows we can explore.

We ran a decision tree, to get a sense of which variables are most likely to dominate. It is not the model we eventually chose, but it gives a good and interpretable answer. This shows which questions most quickly get an answer as to whether a given project is satisfactory. This is a narrow slice. The way to read this is to keep going down layers until you get an answer that is acurate enough for your



purposes. So its best guess, is that it is Satisfactory, if it knows nothing else. Then it would look at the QAE rating, and if, for example, this was un-rated, then it would be more confident that it would end up being satisfactory. It would then look at how much of the spend was eventually cancelled, and if this was less than $30k then it would be even more confident that it would end up being satisfactory.

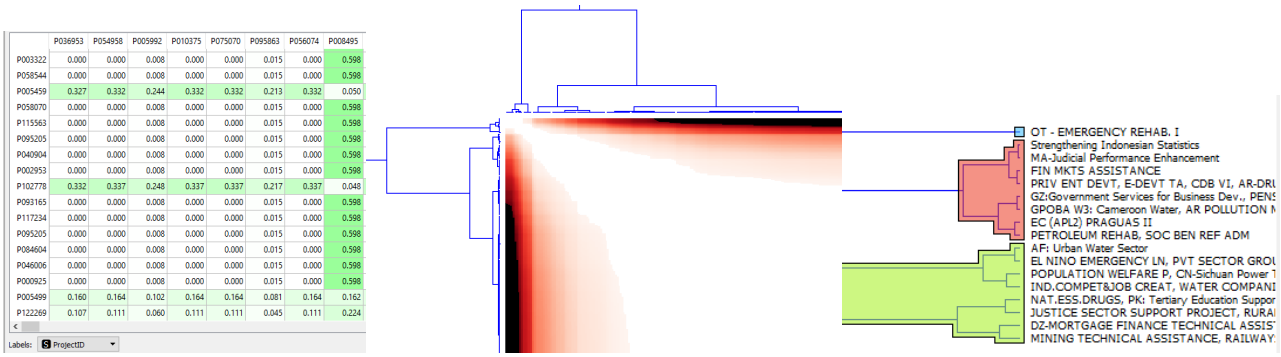
We explored the statistics, which identifies which features are likely to be most important. “The Rank widget considers class-labeled datasets ...and scores the attributes according to their correlation with the class”. I don’t properly understand these yet.

	#	Info. gain	Gain ratio	Gini	ANOVA	χ^2	ReliefF	FCBF
C IEG_Bank_QAE_Rating	7							
C IEG_Bank_QOS_Rating	9							
C IEG_Outcome_Rating	9							
N Approval_FY								
C LendingInstr	16							
C GP_CODE	15	0.115	0.032	NA	NA	22.183	0.054	NA
C IEG_ICR_Quality_Rating_Modified	7	0.107	0.063	0.032	NA	2.828	0.015	0.063
N CANCELLED_USD_AMOUNT		0.064	0.040	0.013	NA	35.085	-0.002	0.000
C Agreement_Type	8	0.038	0.026	NA	NA	1.772	-0.032	NA
C FragileState	5	0.038	0.024	0.007	NA	4.688	-0.014	0.000
C ProdLine	9	0.034	0.059	NA	NA	0.549	-0.011	NA
N NET_COMMITMENT_AMOUNT		0.030	0.015	0.006	NA	13.339	-0.003	0.000
C Region	7	0.026	0.011	NA	NA	4.700	0.004	NA
C Eval_Type	3	0.025	0.036	0.005	NA	17.239	0.017	0.000
N Total elapsed		0.024	0.012	0.003	NA	8.989	0.004	0.000
C PROJECT_STAT	4	0.000	0.000	NA	NA	0.000	0.000	NA
N Close date change		0.000	0.000	0.000	NA	NA	0.000	0.000

Unsupervised learning

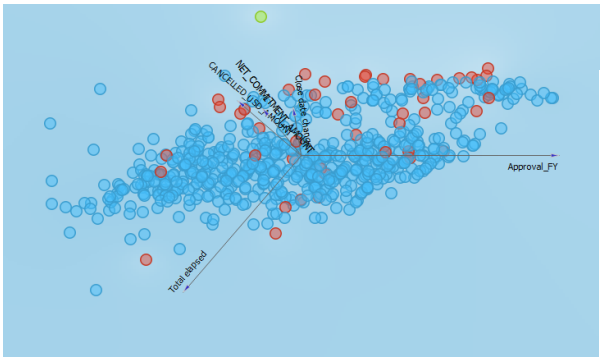
We looked at which projects look similar to each other, without setting a supervised learning task. Categorical features have not been added at this stage. We did this by calculating distances between vector representations of each project:

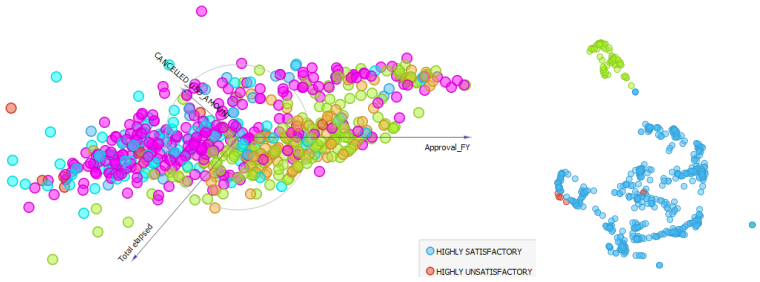
1. project against project distance highlights similar projects
2. dendrogram shows four main clusters of projects
3. zoom in on detail of a couple of clusters



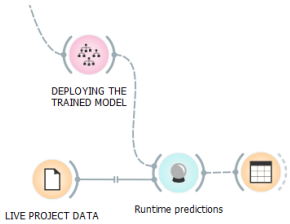
We then ran some other unsupervised learning models:

1. K means clustering identifies two clusters worthy of review plus 1 outlier, crudely stratified by delay in closing date
2. t-SNE Stochastic nearest neighbor, coloured to show evaluation type
3. Free Viz stratifies high performing projects against three useful quantitative variables





Deployment for Client Portfolio Management



Once we had agreed the model with the client, we would make some further small improvements to the Random Forest model and prepare a plan to deploy it against the live portfolio. To do this, we might use the AI canvas as a template to communicate this to the client portfolio manager and related teams. The operational changes could be summarised under the following headings:

Prediction: Predict during project implementation which World Bank projects will be evaluated as Satisfactory after the project completes.

Judgement: The payoffs of being right for the Institution are being able to focus on shutting or intervening in projects that are likely to fail. The impact of false positives is cancelling projects that might otherwise be successful. The impact of false negatives is less serious as it would just lessen the improvement in project coverage

Action: The actions that can be chosen as a result of the judgement would be to:

1. Review a project forecast to fail

2. Strengthen resource on these projects

Outcomes: We would judge whether we are achieving our outcomes by monitoring whether

1. Projects forecast to succeed actually do succeed. For the WB and this dataset, since accuracy achieved is 90%, this should be the target during production.

2. What % projects forecast to fail do fail – anything over 50% would be good

Input: Now that we have trained the model, we would need data on currently running projects as per the feature list used in training. These include forecast to completion, mid-project quality reviews and industry area. This could be fed into the model once a month by the portfolio manager.

Feedback: During production, we will need to use measured outcomes along with input data to generate improvements to our predictive algorithm. This can be done semi-annually, re-running and refitting the whole model. A new model would then need to be issued to the portfolio manager.

How will this AI impact on the overall workflow? Regularly predicting project success will focus management attention on the right projects. It will improve forecasting accuracy, and thereby should increase overall project completion. It may also have a lesser effect on early project scoping and project selection. The impact on the portfolio manager would be a day a month. Since there are likely to be project reviews and interventions for failing projects already, it is likely that the time taken on these from model predictions would be substitution of existing tasks rather than new tasks. As this is implemented, there would be a backlog effect and a higher turnover and reallocation of project teams.

Further developments (making it better and broadening the Use Case)

Improvement and applications	***Benefit***	***Status***
Spend more time understanding the different project reviews, to understand whether Performance review is the best one or whether Outcome review is better.	Check we have understood the problem correctly	Not reviewed
Try a simpler Classification into Successful, unsuccessful	Simpler, makes more sense to client	Not tried, but possible in Orange
Remove the intermediate project reviews to see how much might be explained without these.	Understanding	Not started.
Understand Ranking statistics better	Training	Very rusty with this.
Spend more time testing and optimising across different samples, and reviewing the accuracy.	Optimisation	Not really started
Stack models	Optimisation	1st trial showed small improvement
Try to add Categorical features into the Unsupervised learning	Broader Use case	Needs some thought
Worth looking again at Principal Component Analysis	This was not effective in explaining much of the variance	1st trial
Extend application to large projects w multiple work packages, starting from WBS data	Scale	Not started
Extend application to supplier management for services.	New use case	Would need a little thought
Unsupervised learning: network analysis of the project clusters, once categorical information has been included.	Broader offering	Need some thought
Save the trained model	Makes move into Production better	Witnessed but not tested
Review the Project Success literature in IPMA	Broaden and contextualise the offering	Have seen relevant articles
Test the NYC data set since this has daily data	Would remove hindsight bias	Dataset found.
Extend to other datasets and clients	Generalisability	Not started