Data Driven Project Management: Predicting the Development Time

Marko Prelevikj
Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, Slovenia
Email: mp2638@stuent.uni-lj.si

***Abstract*—**

1. Introduction

The project manager's (PM) main task is to break down the project they manage into tasks which are manageable, not very complex and make a round unit which can be executed with the knowledge of a single person. Once the project is broken down into pieces the PM needs to answer the following questions for each task:

1) how much time the task is going to take to develop; and
2) which project member is the best fit for the task

In this paper we are focusing on the task of estimating the time required to develop a given task. We use data provided from a company's JIRA portal [1], which keeps record of the project's tasks. We made 4 different models of the time required to develop a task, where we changed the unit we are forecasting in: days or hours, and the time span of development time, ranging from all time down to a maximum of 10 days. We evaluated the variations of the model using 4 distinct methods: Naive Bayes, Random Forests, XGBoost [2] and SVM. In the end we uncover which are the most important features of the task which should be considered when estimating the development time for which we used the SHAP [3] method.

2. Model data

The available information used to build the model has been extracted from a company's JIRA portal [1]. All tasks are described by their categorical features: *type*, *priority*, *components* of the project they affect, and *labels* which are specific for the project. Due to their high cardinality, the values of the *components* and *labels* features have been filtered such that there are only left values which have at least 50 entries in the dataset. The filtered values are used in their one-hot-encoded form to reduce the complexity of the model. Additionally, we used the following discrete features: the *number of comments* each task has, the *number of linked issues*, and their *degree of cycling*.

Table 1. DATASET CHARACTERISTICS.

Statistic	All	1 Q	1 M	10 D
count	2935.000	2902.000	2775.000	2451.000
mean	205.252	149.210	96.165	55.762
std	678.103	293.573	132.257	57.019
min	2.000	2.000	2.000	2.000
25%	15.000	15.000	13.000	11.000
50%	49.000	48.000	44.000	33.000
75%	148.000	143.000	121.000	83.000
max	15003.000	2260.000	754.000	228.000

Due to improper usage of JIRA, there is some noise in the data, which causes a strong bias toward low values of the predicted time to develop. To reduce this effect we have filtered out all tasks which have development time lower than $2h$. To further reduce the variance in our dataset we have decided to limit the upper bound of the development time. We have done so in 3 different stages to measure the effect of the variance on our model: 1) there is no upper bound, 2) the upper bound is 30 days, and 3) the upper bound is 10 days. The summary of the datasets is shown in Table 1.

3. Testing model quality

Present different the results (MAE, RMSE, R2) obtained by different regressors. As the variance lowers, so do the metrics of quality of the models.

4. Model Explainability

Write about feature importance and how to explain the made decisions.

5. Conclusion

Quick recap of the problem and how we solved it. XGBoost [2], SHAP [3].

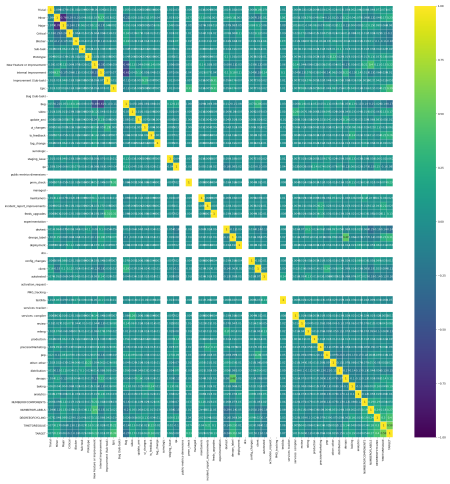


Figure 1. Simulation results for the network.

Table 2. PERFORMANCE OF DIFFERENT METHODS ON THE VARIATIONS OF THE DATASET. THE $*$ SYMBOL INDICATES THAT THE DATASET DOES NOT CONTAIN ALL THE INITIAL ATTRIBUTES, THUS IT IS MORE REALISTIC.

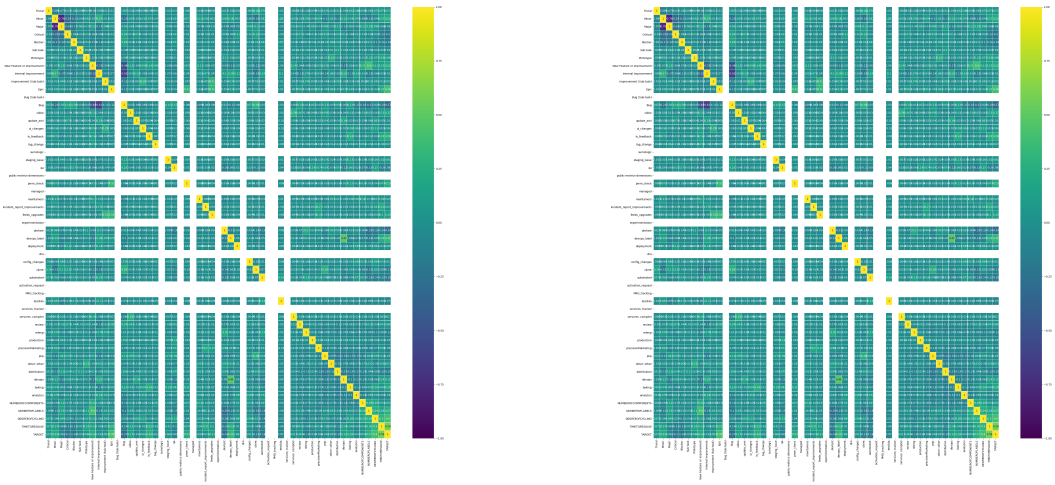
DataSet	Method	RMSE	MAE	R^2
All*	boost	**791.428**	254.770	-0.042
	naive	948.649	420.036	-0.498
	forest	840.027	256.399	-0.174
	SVM	792.344	**191.267**	-0.045
1Q*	boost	**286.835**	160.587	0.018
	naive	577.520	389.602	-2.982
	forest	361.223	179.258	-0.558
	SVM	302.772	**124.272**	-0.095
1M*	boost	**138.034**	92.616	-0.048
	naive	255.083	213.834	-2.579
	forest	178.280	109.402	-0.748
	SVM	143.368	**78.249**	-0.131
10D*	boost	**56.406**	43.847	0.041
	naive	120.824	106.642	-3.398
	forest	77.243	56.629	-0.797
	SVM	60.460	**41.976**	-0.101

References

[1] "Jira - issue tracking software," https://www.atlassian.com/software/jira, accessed: 2020-05-28.

[2] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[3] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.



(a) Case I (b) Case II

Figure 2. Simulation results for the network.