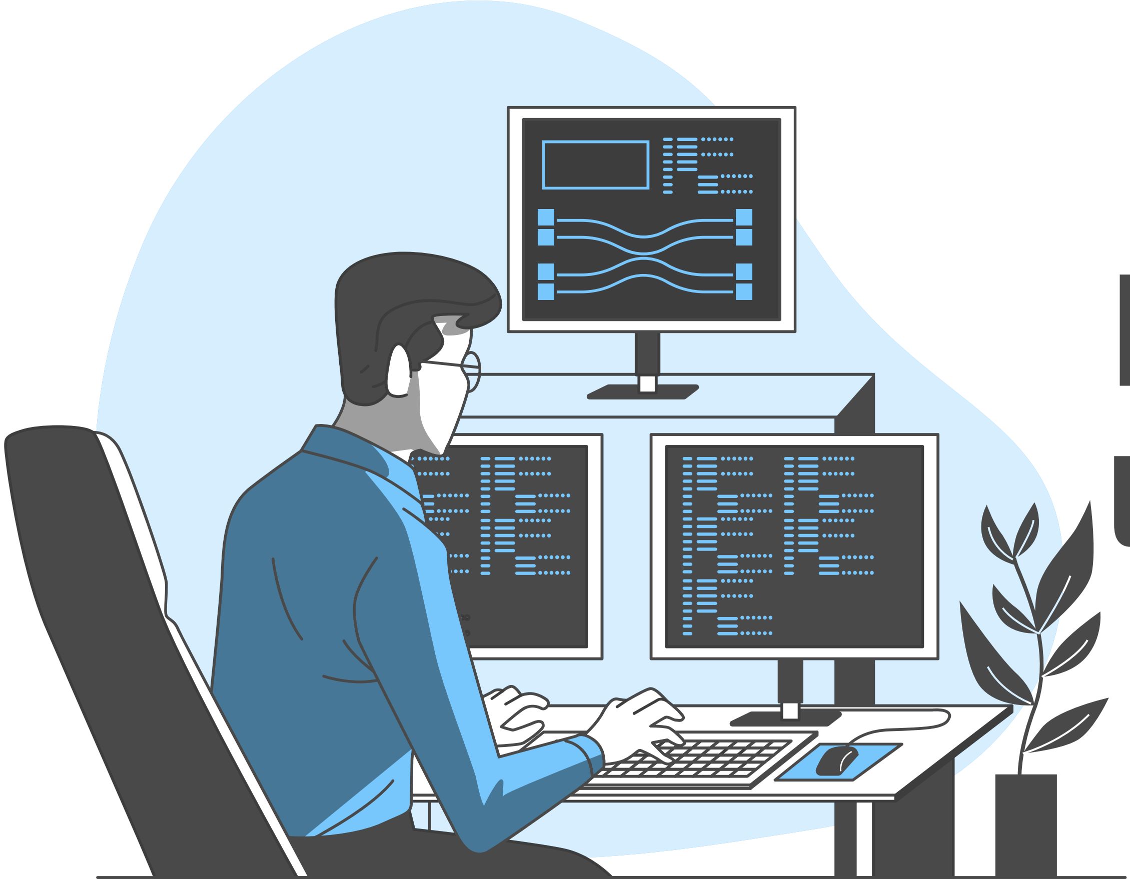
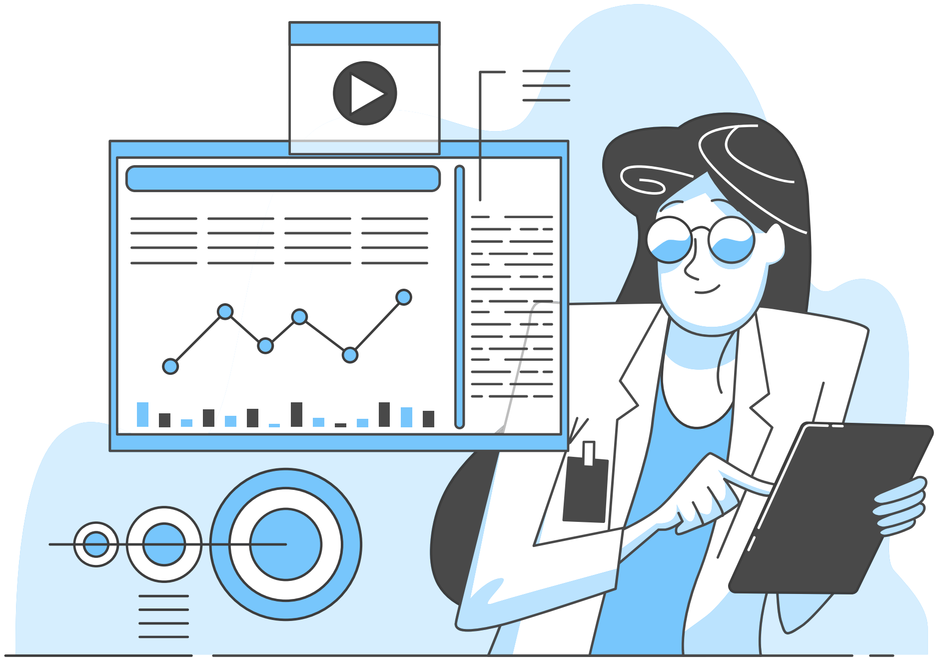


Effort Estimation using SBSE

Ioana Chelaru

Table of Contents



Software Effort Estimation

Inadequate or overfull funds for a project could cause a considerable waste of resource and time.

For example, NASA canceled its incomplete *Check-out Launch Control System* project after the initial $200M estimate was exceeded by another $200M [7].

Different approaches

Multiple linear regression

with appropriate transformations [1]

Linear regression & ANNs

similar performance to MLR, but not always [6]

Artificial Neural Networks

outperformed MLR [3]

Pareto ensemble of ANNs

best performance compared with a variety of methods [4]

Artificial Neural Networks & case-based reasoning

performed better than MLR [2]

Analogy-based approaches

better results than MLR, ANNs, simple decision trees, and stepwise regression [5]

The Automatically Transformed Linear Model [8]

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ..\beta_n x_{ni} + \varepsilon_i,$$

The ATLM

ALGORITHM 1: Linear Model Prediction
Input: formula, training, test
Output: Predictions for test data

transforms ← **calculate.transforms** (*training*)
trans.training ← **apply.transforms** (*transforms*,*training*)
Apply transformations
trans.test ← **apply.transforms** (*transforms*,*test*)
lm.see ← **lm** (*formula*,*trans.training*)
Determine linear model
predictions ← **predict** (*lm.see*,newdata = *trans.test*)
Predict response
return (**invert.predictions**(*transforms*,*predictions*))
Return untransformed predictions

Optimized Learning [9]

Optimized Learning

Python Scikit-Learn

Utility algorithms

Feature map of data mining options

Evolutionary optimizer & automatic mapper facility

Conclusions & future work

- SBSE has great potential
- the used method depends on the dataset
- consider your resources

Bibliography

1. K. Dejaeger, W. Verbeke, D. Martens, and B. Baesens. 2012. Data mining techniques for software effort estimation: A comparative study. IEEE Trans. Softw. Engin. 38, 375–397.

2. G. Wittig and G. Finnie. 1997. Estimating software development effort with connectionist models. Inf. Softw. Technol. 39, 469–476.

3. H. Park and S. Baek. 2008. An empirical validation of a neural network model for software effort estimation. Expert Syst. Appl. 35, 929–936.

4. L. Minku and X. Yao. 2013. Software effort estimation as a multiobjective learning problem. ACM Trans. Softw. Engin. Methodol. 22, 4.

5. N.-H. Chiu and S.-J. Huang. 2007. The adjusted analogy-based software effort estimation based on similarity distances. J. Syst. Softw. 80, 628–640.

6. A. Heiat. 2002. Comparison of artificial neural network and regression models for estimating software development effort. Inf. Softw. Technol. 44, 911–921.

7. K. Cowing. Nasa to shut down checkout & launch control system, 2002.

8. Whigham P.A., Owen C.A., and MacDonell S.G. A baseline model for software effort estimation. ACM Transactions on Software Engineering and Methodology 24(3), pp.1-11 (Article 20). doi:10.1145/2738037,2015.

9. Tianpei Xia, Jianfeng Chen, George Mathew, Xipeng Shen, and Tim Menzies. Why software effort estimation needs sbse. SSBSE'18, 2018

10. Federica Sarro, Alessio Petrozziello, and Mark Harman. Multi-objective software effort estimation. IEEE/ACM 38th IEEE International Conference on Software Engineering, May 2016.

Thanks!

Do you have any questions?