Deep Thought

SATRAJIT BASU

Goals

- Build framework for predicting project effort based on historical data
- Explore feasibility of using Amazon AWS Machine Learning (Amazon ML) in this scope
- Identify attributes in data essential for building a learning model
- Identify workflow for providing real time predictions for project effort
- Explore D3.js as a data visualization tool

Background

- The Estimation Engine is a stand-alone RESTful Web API for the Kaiser Estimation Model (KEM) tool.
- The current implementation provides estimates for project hours based on the scope of the project and user defined variables termed as cost drivers.
- The goal is to build in a machine learning framework to improve estimations.
- There are two broad categories of optimization that we are currently considering.
    - First, optimize the hours estimated based on historical data (actuals) of similar projects. In this scenario, the Estimation Engine can provide estimates that are more realistic for given the scope and the resources available for the project.
    - Second, optimization scope lies in the cost drivers and their impact on the overall estimation. Currently the impact of a certain cost driver is determined by a modeler defined factor. Such values, often driven by heuristics can be improved by machine learning on historical data.

Related Research

- Substantial amount of research has been conducted on the use of predictive modeling in a software effort estimation setting
- No unequivocal conclusions have been reached in regards to which technique is best suited for tackling the problems of software effort estimation
- It has been found that the success of a technique highly correlates to the datasets being used for predictive modeling
- The variability in the data stemmed from the data quality, the unique set of attributes available
- Those attributes can grouped into following categories:
  - **Size attributes**: e.g. lines of code (LOC), function points, or some other measures.
  - **Environment information**: e.g. the number of developers involved and their experience and the sector of the developing company.
  - **Project data**: e.g. project type
  - **Development related variables**: e.g. programming language, type of database system, resources impacted
- A large scale benchmarking study on various techniques indicates that ordinary least squares regression in combination with a logarithmic transformation performs best.

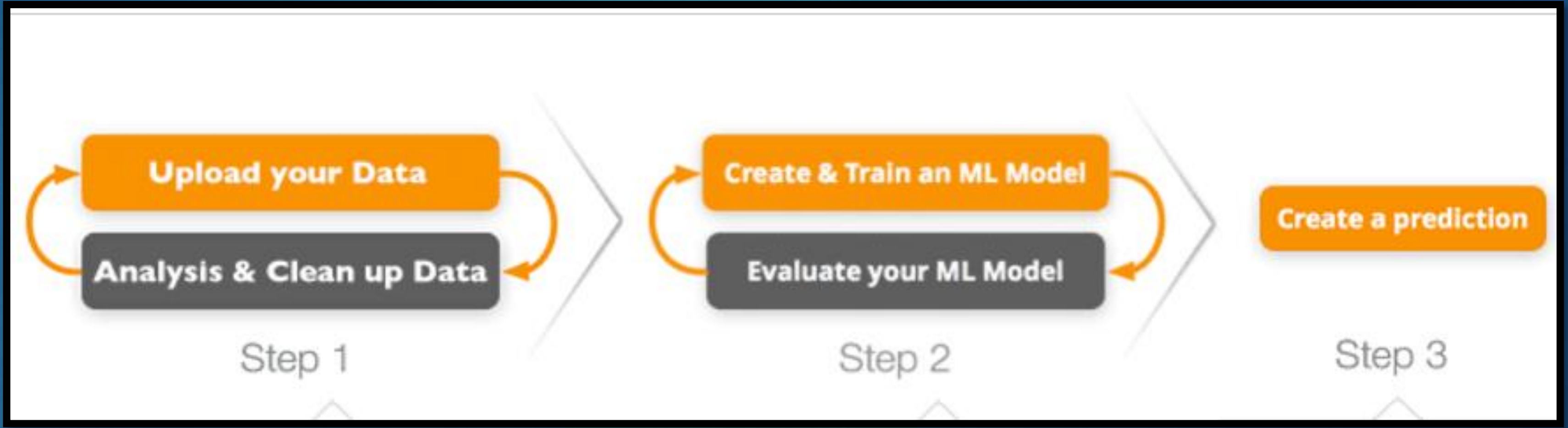
Implementation Plan

- Based on existing research it can be concluded that the performance provided by linear models with certain improvements are as good as any complex data mining models
- For our business problem of predicting estimation inputs, regression analysis using linear models can be considered a good starting point.
- The approach can be further improved once historical data from Kaiser is made available
- Given that our current KEM tool and Estimation Engine API are hosted on Amazon AWS servers, the first instinct is to harness the machine learning capabilities provided on the AWS platform.
- That allows for simpler infrastructure maintenance and utilization of the scalability built into the AWS platform.
- However, the complexity of learning models made available by Amazon Machine Learning (Amazon ML) is limited and not vastly customizable.

AWS Machine Learning

- Amazon Machine Learning (Amazon ML) is a robust, cloud-based service
- Amazon ML provides visualization tools and wizards that guide users through the process of creating machine learning (ML) models without having to learn complex ML algorithms and technology.
- Once models are developed, Amazon ML makes it easy to obtain predictions for the application using **simple APIs**, without having to implement custom prediction generation code, or manage any infrastructure.
- ML models for regression problems predict a numeric value. For training regression models, Amazon ML uses the industry-standard linear regression

Amazon ML Workflow



The workflow for using Amazon ML

Data Used

- Used publicly available Software Engineering Repository
- 81 project instances. 12 attributes
- Attributes:
  - Project Id
  - Team Experience (measured in years)
  - Manager Experience (measured in years)
  - Year End
  - Length
  - **Effort (measured in person-hours)**
  - Transactions (count of basic logical transactions in the system)
  - Entities (number of entities in the systems data model)
  - Points Adjust
  - Span
  - Points Non Adjust
  - Language (labeled as {1,2,3})
- Notes: 4 incomplete projects so often people use the 77 complete cases

Evaluating Models

- For regression tasks, Amazon ML uses the industry standard root mean square error (RMSE) metric.

$$RMSE = \sqrt{1/N) \sum_{i=1}^{N} (actual\, target - predicted\ target)^2}$$

- It is a distance measure between the predicted numeric target and the actual numeric answer (ground truth)
- The smaller the value of the RMSE, the better is the predictive accuracy of the model.
- A model with perfectly correct predictions would have an RMSE of 0
- Amazon ML provides a baseline metric for regression models. It is the RMSE for a hypothetical regression model that would always predict the mean of the target as the answer