A predictive analysis approach using linear regression to estimate software effort

ERBASE 2018

Spc. Antonio Esteves and Dr. Leonardo M. Medeiros

Federal Institute of Alagoas - BRAZIL

22/08/2018

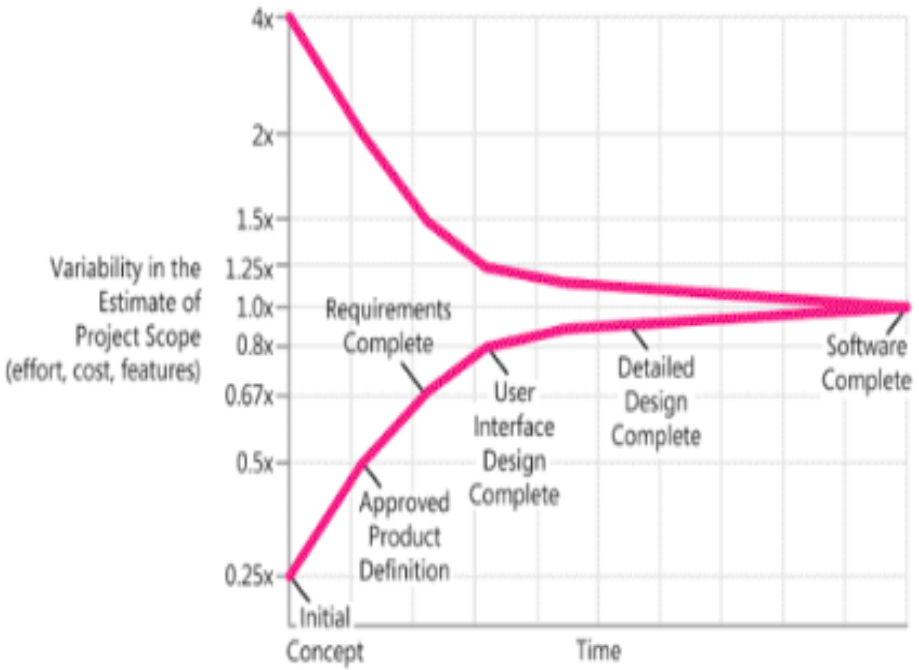
Summary

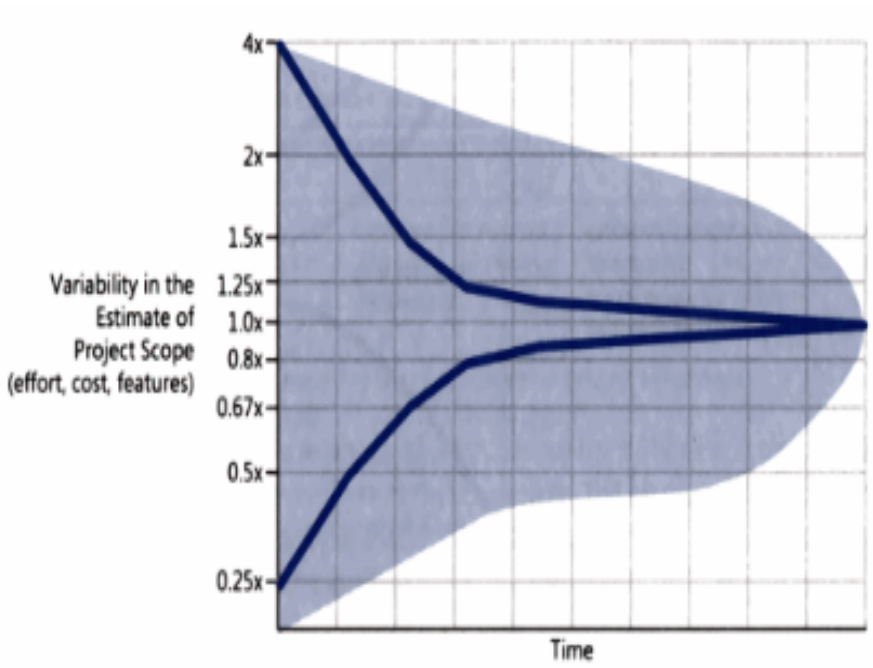
Software development effort and productivity

Nowadays, predict software development time and budget is a real challenge with economic and scientific relevancy. So, the accuracy of a predict software estimation model has a vital role in software development and effort prediction estimation is one of the critical tasks required for developing software.

This work presents regression and machine learning techniques to predict software effort estimation, through data from existing projects.

Effort Estimation





Effort Estimation

One of the fundamental issues in a software project is to know how much effort will be spent in working hours or budget before starts the software development. Actually, companies uses insufficient background of metrics in the area of software estimation.

Learning-oriented models

The Learning-oriented models attempts to automate the estimation process by building autonomous models that can learn from previous estimations experiences. These models do not rely on assumptions and are capable of learning incrementally as new data are provided over time.

AI-based Predictive Models

AI-based predictive models can be a useful tool with an accurate degree that helps on the prediction of software effort based on historical data from software development metrics.

In this study, we built a software effort estimation model to predict this effort using a Linear Regression model.

Data Collection

To perform this study we used Desharnais dataset from the PROMISE software engineer repository. This dataset consists of 81 software projects from a Canadian software house collected by J. M. Desharnais.

Features definition to Desharnais dataset

Feature	**Description**
Project	Project ID which starts by 1 and ends by 81
TeamExp	Team experience measured in years
Manager Exp	Manager experience measured in years
YearEnd	Year the project ended
Length	Duration of the project in months
Effort	ActualEffort is measured in person-hours
Transactions	Transactions is a count of basic logical transactions in the system
Entities	Entities is the number of entities in the systems data model
PointsAdj	Size of the project measured in unadjusted PointsAdj function points.
PointsNonAdjust	Size of the project measured in adjusted function points.
Language	Type of language used in the project expressed as 1, 2 or 3.

Linear Regression

Regression analysis aims to verify the existence of a functional relationship between a variable with one or more variables, obtaining an equation that explains the variation of the dependent variable Y, by the variation of the levels of the independent variables.

K-Nearest Neighbors Regression

- K-Nearest Neighbor Regression is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions);

K-Nearest Neighbors Regression

- K-Nearest Neighbor Regression is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions);
- The input consists of the K closest training examples in the feature space;

K-Nearest Neighbors Regression

- K-Nearest Neighbor Regression is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions);
- The input consists of the K closest training examples in the feature space;
- It's been used in a statistical estimation and pattern recognition as non-parametric technique classifying correctly unknown cases calculating euclidean distance between data points;

K-Nearest Neighbors Choice

Our choice by *K-Nearest Neighbor* Regression was motivated by the absence of a detailed explanation about how effort attribute value is calculated on Desharnais dataset;

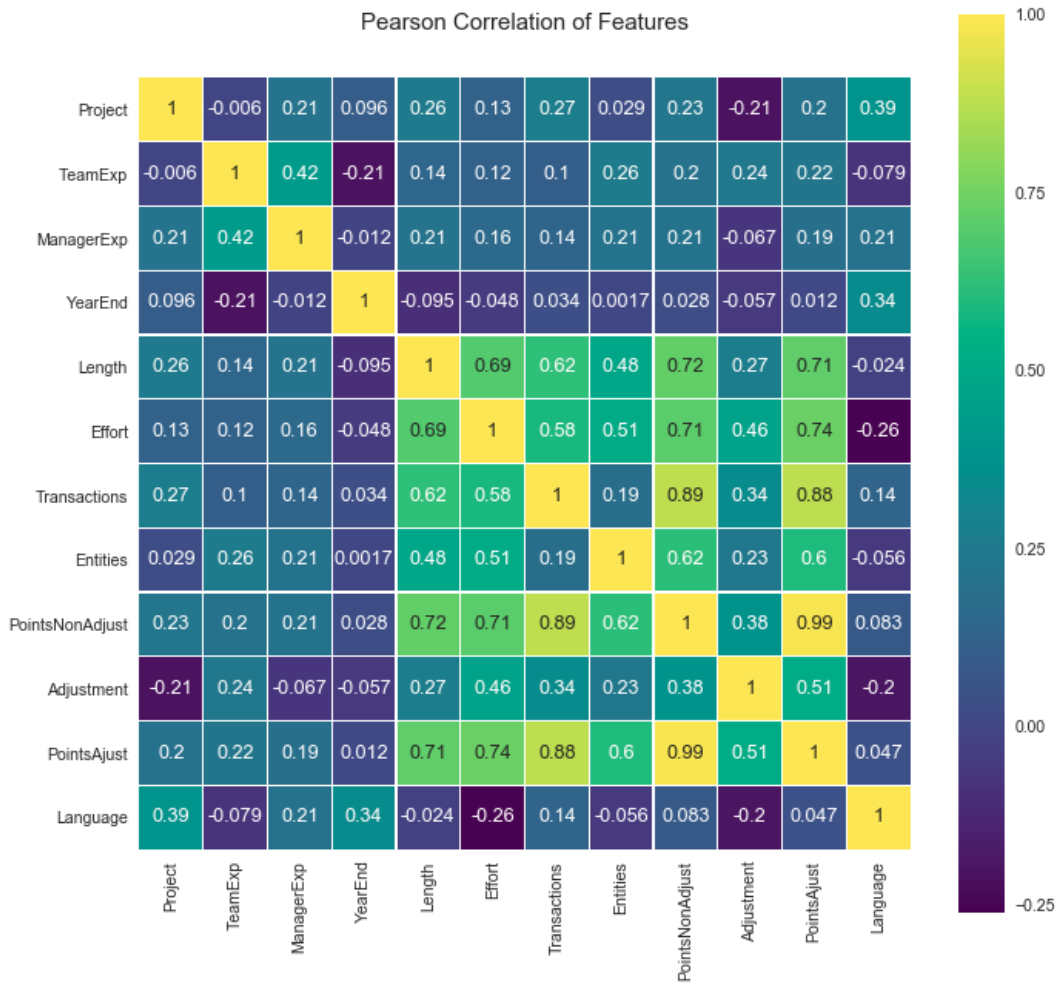
Feature Selection

To address Desharnais dataset the correlations between attributes and software effort were analyzed. The correlation between two variables is a measure of how well the variables are related.

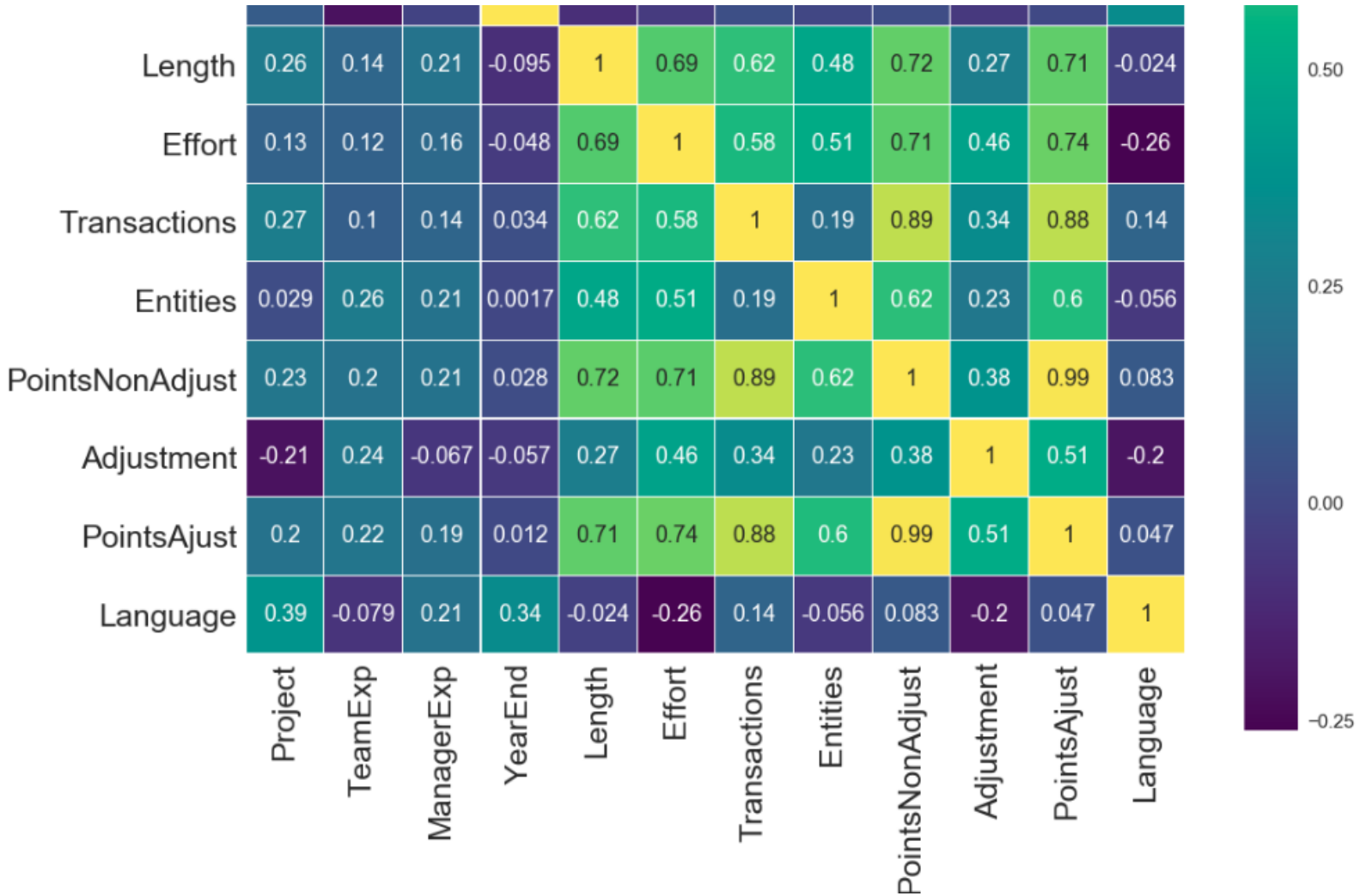
Pearson Correlation

The most common measure of correlation in statistics is the Pearson Correlation which shows the linear Relationship between two variables. Pearson correlation coefficient (PCC) is a statistical metric that measures the strength and direction of a linear relationship between two random variables.

Pearson's Correlation Matrix



Pearson's Correlation Matrix



Model Construction

- *Linear Regression* and *K-Nearest Neighbors Regression*;

Model Construction

- *Linear Regression* and *K-Nearest Neighbors Regression*;
- Linear Regression model consists of generating a regression for the target variable Y;

Model Construction

- *Linear Regression* and *K-Nearest Neighbors Regression*;
- Linear Regression model consists of generating a regression for the target variable Y;
- *K-Nearest Neighbor Regression* goal is to calculate the average of the numerical target of the K nearest neighbors with **k=3**;

Developed Approach

In this work, we focus on analyzing the importance weights of attributes in the estimating of software cost and time and his correlation, so we set out to answer two research questions related to the dataset:

- Which the correlation of each metrics in the estimation of software effort ?

Developed Approach

In this work, we focus on analyzing the importance weights of attributes in the estimating of software cost and time and his correlation, so we set out to answer two research questions related to the dataset:

- Which the correlation of each metrics in the estimation of software effort ?
- How accurate is the model of software effort ?

Developed Approach

The training of regression models models were performed using hold out techniques splitting training and test sets.

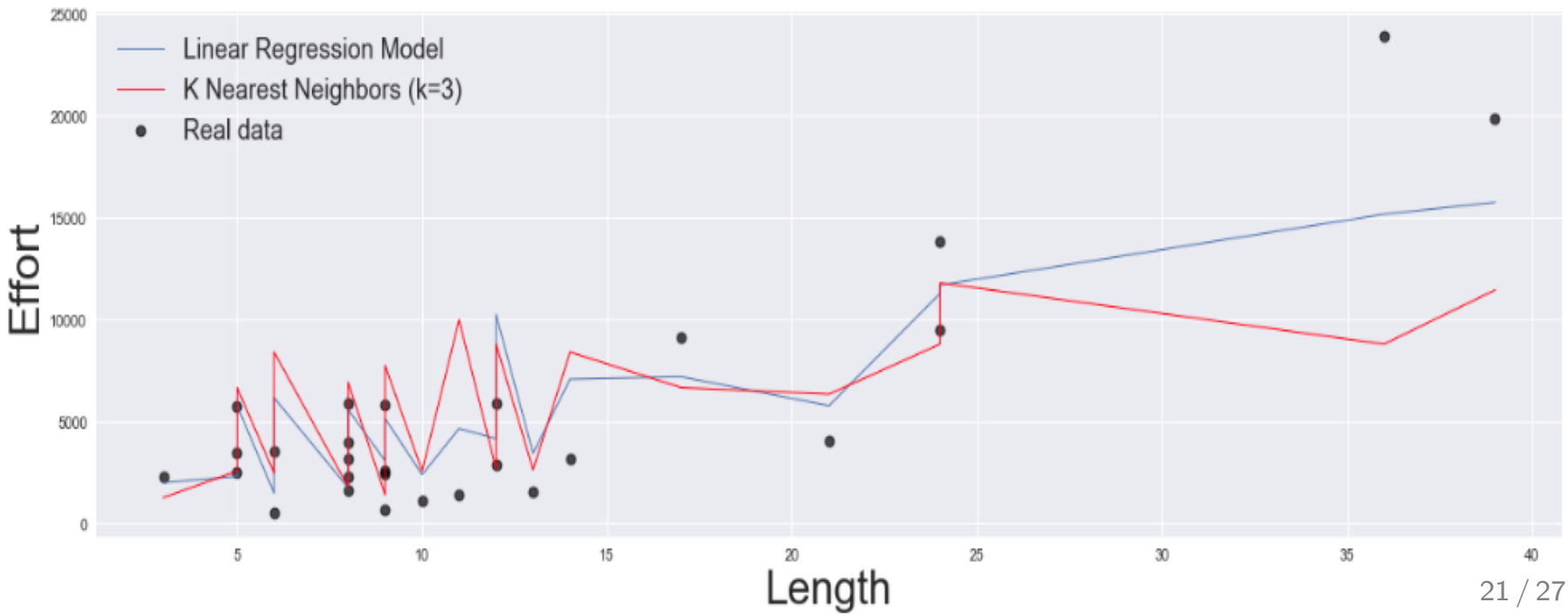
During the training it was necessary to estimate the values of the random state parameter, since they are not previously known.

Developed Approach

On models generated from the training data was applied remaining 33% of the data, previously isolated, and their performances will be evaluated in order to demonstrate how accurate the linear regression model can predict software effort estimation.

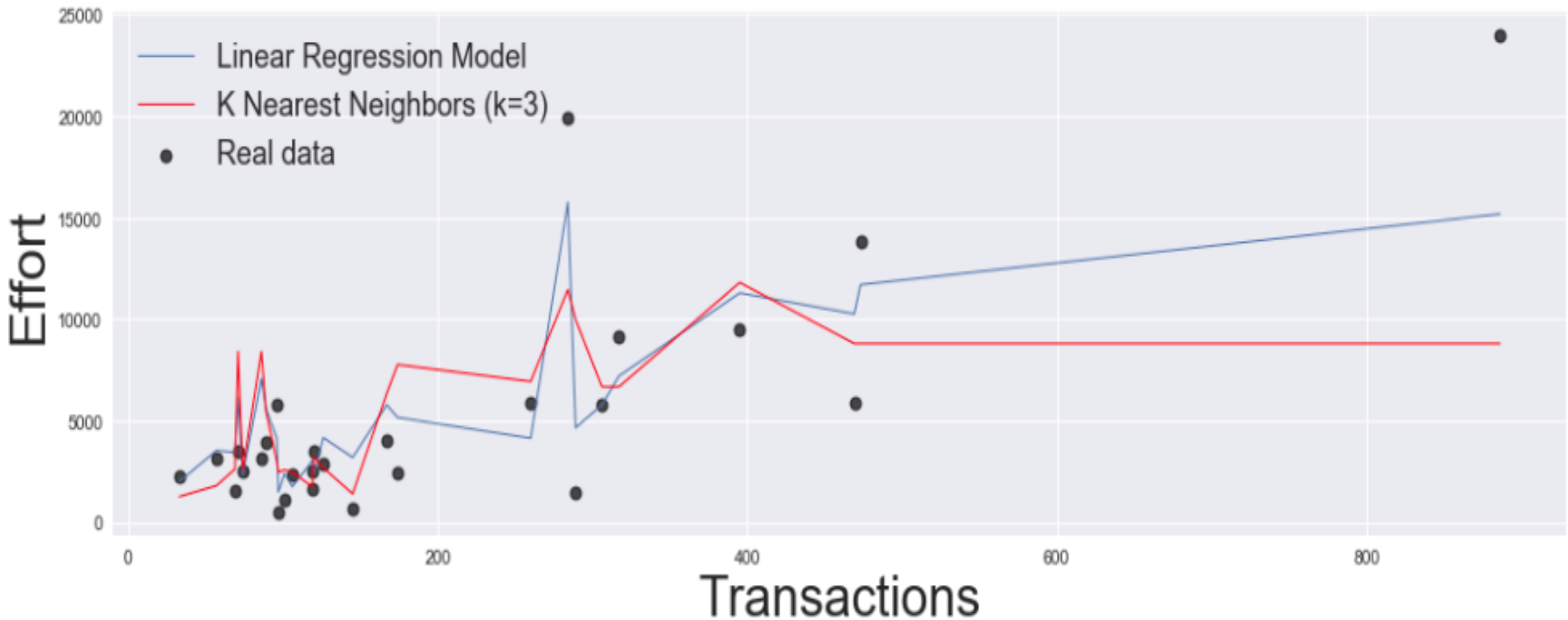
Knn (k=3) x *LR* on Length Metric

According to the plot we observe that LR model (blue line) presents a better performance. Although Knn model (red line) is fairly close to data points, the LR model shows a smaller mean squared error.



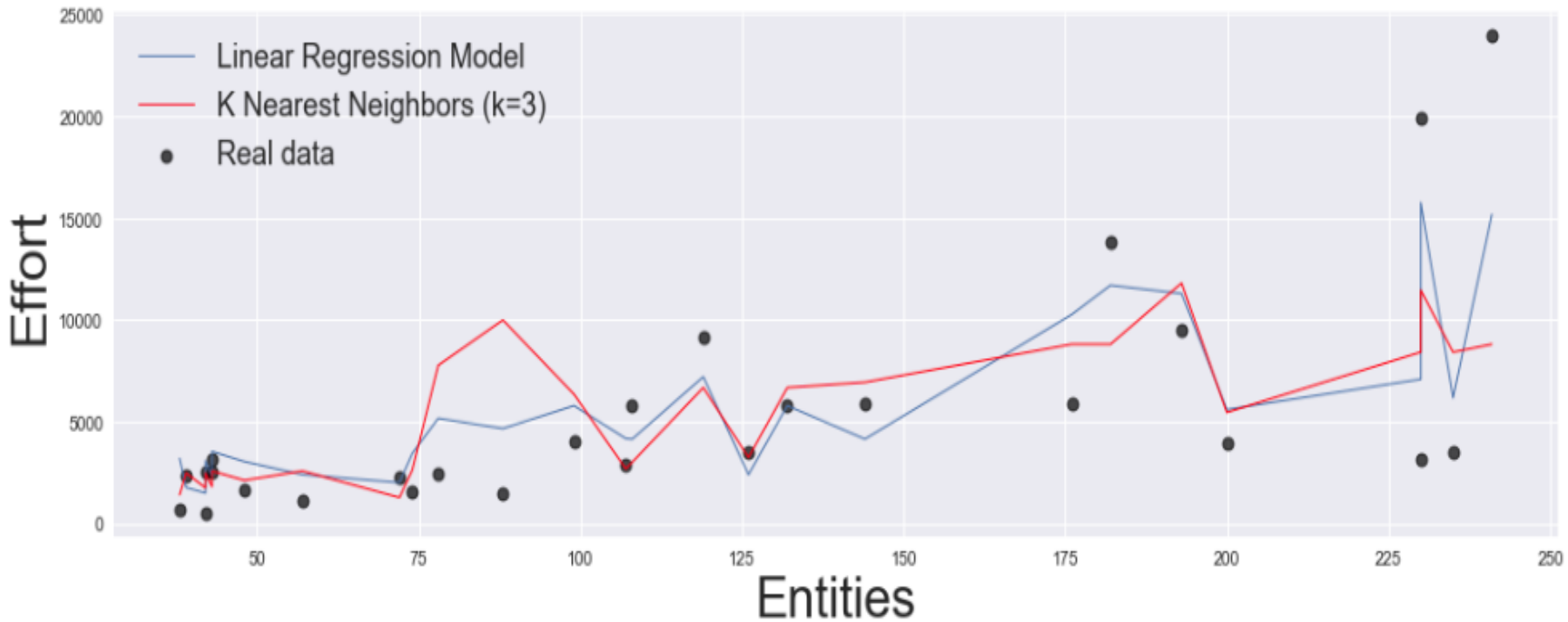
Knn (k=3) x *LR* on Transactions Metric

It is possible to observe that the lines of both models present a slight tendency to rise, which justifies their correlation with the increase in effort.

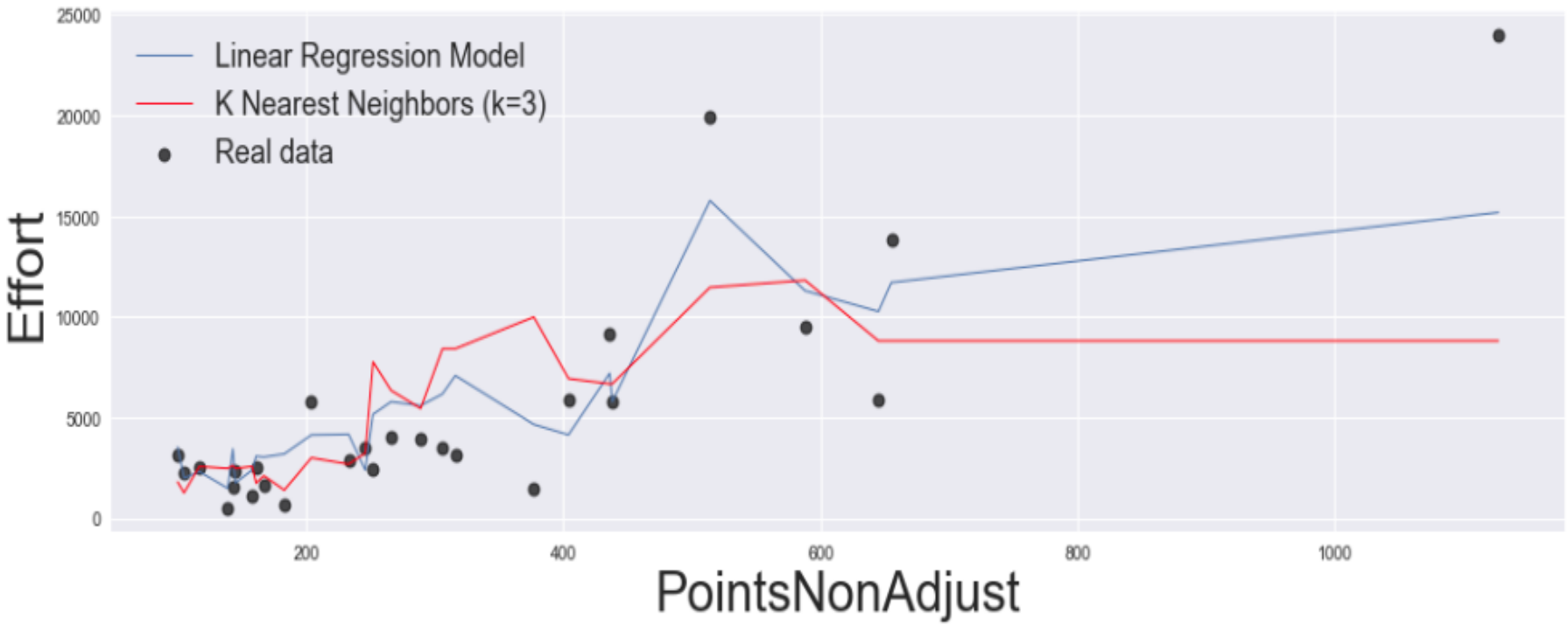


Knn (k=3) x *LR* on Entities Metric

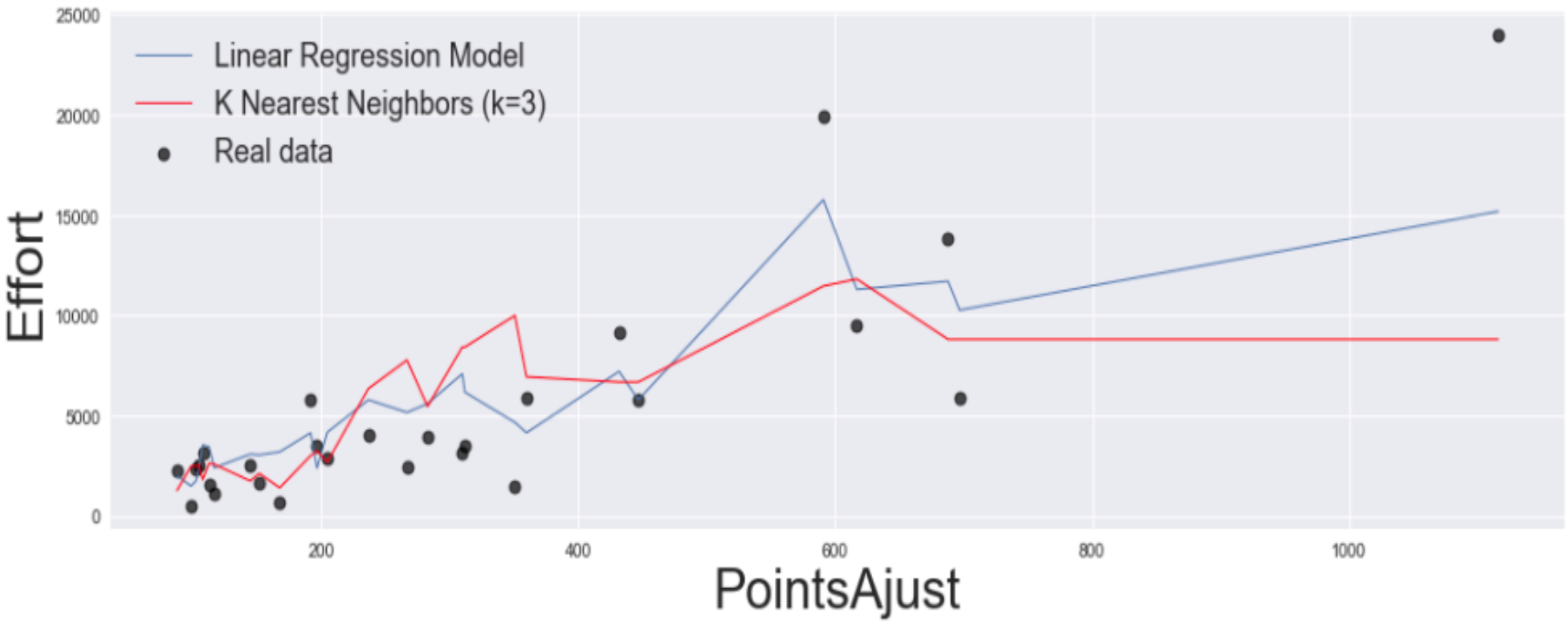
Some metrics are also highlighted by the presence of outliers.



Knn (k=3) x *LR* on Points Non Adjust Metric



Knn (k=3) x *LR* on Points Adjust Metric



R^2 Score Performance

Algorithm Model	**R^2 Score**
Linear Model Regression	0.7680074954440712
K-Nearest Neighbor Regressor	0.7379861869550943

As in simple linear regression, the coefficient of determination (R^2 Score) must take values between and including 0 and 1, where the value 0 indicates that the regression is nonexistent while the value 1 indicates a "perfect" linear relationship [Freund et al. 2006].

Questions

Reproducible Research

All steps in the analytical workflow are coded in Python and available for download from a Github repository – https://github.com/toniesteves/sw-effort-predictive-analysis.information `https://github.com/toniesteves/sw-effort-predictive-analysis`.