Evaluation of results

1 Datasets

Two datasets were used, china and albrecht. These two have a large amount of numerical attributes and are easily identifiable metric for the effort required of the project (others may have had one, but the information about the attributes was limited and it was often hard to tell what they represented). Albrecht was chosen because it is simple and only have one effort column. The china dataset was choosen for its robustness and size, it has more attributes to consider.

The datasets is composed of numerical attributes, all of which were used to calculate the effort, with the obvious exception of the effort variable. In china, to improve the accuracy of prediction the 'ID', 'Dev.type' attributes were not included in the process, as this would be completely independent of the effort. 'N_effort' was also removed as it was often the same value as effort, and so was assumed to be a similar metric as 'Effort. In albretch, none of the attributes were stripped off. All of which has impact on overall cost of project.

2 Test / Training Data Split

Both data-sets were evaluated using an 80/20 split of the data, with 80% for training and a remaining 20% for testing.

3 Implementation

The model was developed using DEAP and modified to fit the assignment. The primitive set was created using DEAP's creator which held user defined primitives. Primitives were then added, consisting of math functions. Maximum, addition, subtraction, multiplication, minimum, cosine and sine were all added to the primitive set. Log, exponential and division were not added because they needed arguments to be predefined, and the aim of the model was to be as general as possible.

Then, the fitness function was created using the creator. The fitness was chosen to be minimised, due to the user defined fitness function described later during individual evaluation. An individual tree had to be created. This provided information about the characteristics of the tree that had to be created. It was defined to be a primitive tree, with a minimised fitness, and to use the main primitive set for developing functions when making each individual.

After that, the set and fitness function are be registered. Components that were registered were added to a toolbox, that could then change the individuals depending on what tools are being requested. The next set of tools to be registered into the toolbox were the genetic programming(GP) tools. These GP tools were evaluate, select, mate, exp_mut and mutate. The GP was then executed and next generations stored in Panda data frames.

4 Results

Across all runs of the GA, the mutation rate, crossover rate, population size, number of generations:

Population = 300

Crossover probability = 0.5

Mutation probability = 0.5

Number of generations = 100

4.1 Albrecht

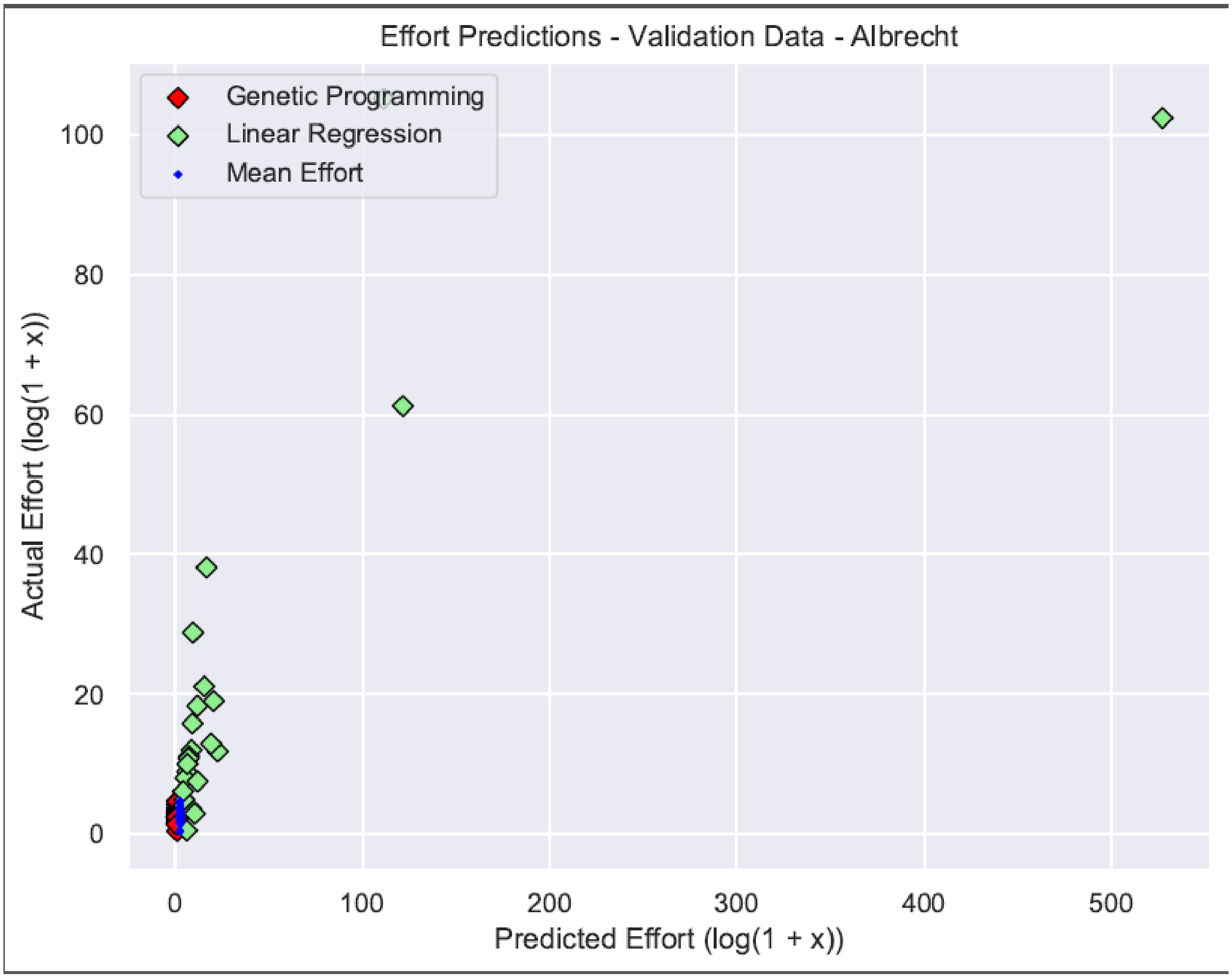

Figure 1: Albrecht Effort prediction

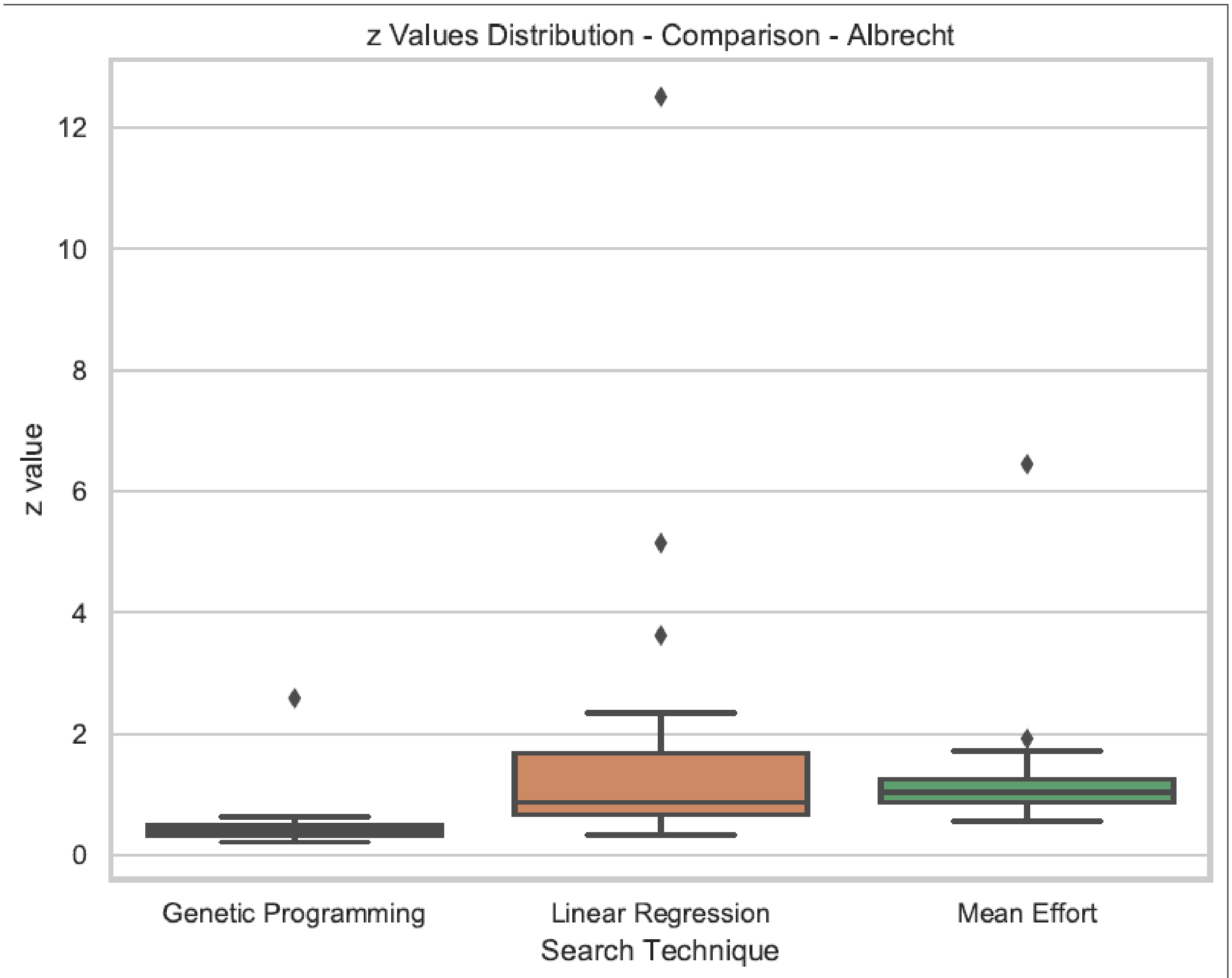

Figure 2: Albrecht z Values Distribution

Genetic programming performed better than Linear Regression. It’s results are closer to optimal that mean effort value. Much of its predicted efforts are closer to to actual values of the efforts. The z Values Distribution shows how closer.

4.2 China

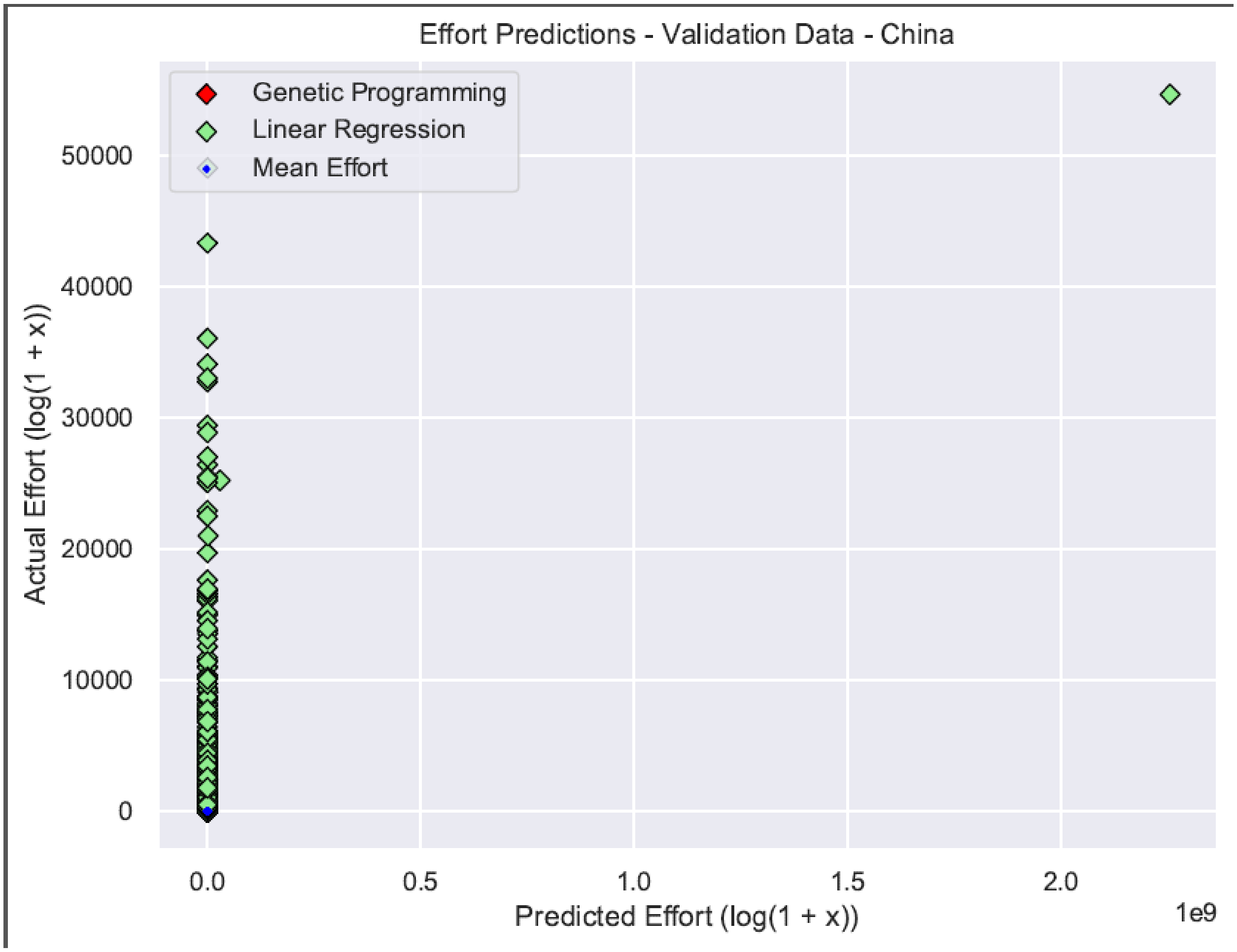

Figure 3: China Effort Predictions

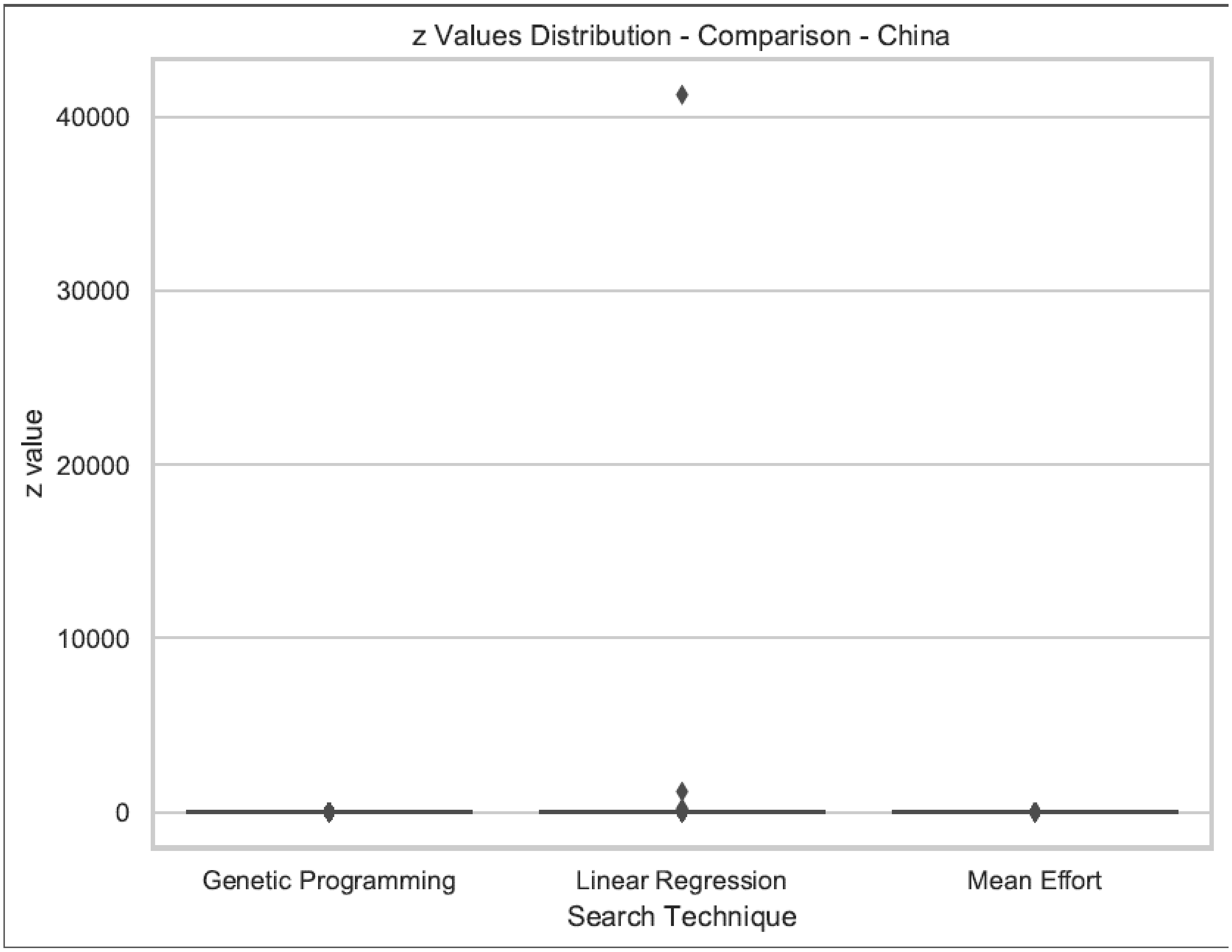

Figure 4: China z Values Distribution

5 Conclusion

The GP model showed it was more reliable at predicting effort values. Applying the developed function onto the training data was useful to establish an estimation of accuracy for the created model. The accuracy could also be found using the mean difference between predicted efforts and actual efforts.