**Estimating the Effects of Major PC Components on the Execution Time of an Image Recognition Task**

submitted to:

**Mr. Leonid Shpaner**

University of San Diego

February 5, 2024

submitted by:

**Arup Chakraborty**

**Issa Ennab**

**Geoffrey Fadera**

Final Project Group 1 Technical Report

AAI-500 Spring 2024

1. **Introduction**

-why we chose this data set

-describe the data set

-describe what caught our interest, and what we are trying to model

-clearly define our exploratory variables and dependent variable

-state our major assumption in doing our analysis (i.e. time has nothing to do with execution time), and state the business application

In deriving the model, we have one major assumption – i.e. the execution time of an image processing task is not affected by time. We assume that any PC will always take the same amount of time to execute an image recognition task no matter how long that PC has been running. This is quite contrary to the general perception that PCs slow down over time as explored in { <https://www.usenix.org/legacy/event/sysml07/tech/full_papers/basu/basu.pdf>}

1. **Data Cleaning/Preparation**

-we prepared two data sets – one without the timestamp, and one with.

In both cases, we (1) checked/removed redundancies, and (2) we removed outliers based on percentile/IQR rule..

Show boxplots for 1 before and after removing outliers for each PC

Show how we changed the timestamp into seconds

Show time series plots for 2, before and after removing outliers for each PC

-1st we compare the distributions of each PC assuming no time-dependency

-2nd, we check for time dependency

-lastly, we create a data frame that includes all the variables that we want to model

combined data with CPU Core, Clock, Ram values

1. **Exploratory Data Analysis**
   1. **Sample Distribution Comparison**

Assuming that the execution time for an image processing task is independent of time, we compared and analyzed the execution time density plots from each PC. Fig 3.1 below shows in one graph the density plots of execution time from each PC.

A graph of data with different colored lines

Description automatically generated

Fig 3.1.1 Combined Density Plots of Execution Times of all PCs

The density plots for MacBookPro1 and VM show bell-shaped distributions as shown in Fig 3.2 below which supports our assumption that time had no effect.

A graph of a computer

Description automatically generated

Fig 3.1.2 Density Plots of Execution Time Values Assuming No Time-Relation of VM and MacBookPro1 PCs

However, the density plots for MacBookPro2 and RaspBerry Pi show irregular shapes (Fig 3.3). Each one appears to be a combination of multiple normal distributions.

A graph of a number of data

Description automatically generated with medium confidence

Fig 3.1.3 Density Plots of Execution Time Values Assuming No Time-Relation of MacBookPro 2 and Raspberry Pi PCs

With N=1000 samples, the sample distribution for each PC should have taken a normal shape by the Central Limit Theorem [page 90 Module 1 Textbook]. The shape of each of the distributions in Fig 3.3 therefore suggests that there must be other variables causing the irregularity. One easy explanation could be of data collection error - i.e maybe other tasks on the PCs were executed during the experiment resulting in a significant variation in the observed execution times.

The other major possibility is that the execution time is time-dependent which would make our assumption incorrect. To check this, we performed a time-series analysis for each PC to check for trend and seasonality.

* 1. **Time-Series Analysis**

In Section 2, we mentioned that we cleaned and prepared another set of data the timestamp is included. To check whether the execution time is time-dependent, we performed seasonal decomposition time-series analysis on each of the data. The resulting trend and seasonality for each are shown in Fig 3.2.1 ~ Fig 3.2.4

A screenshot of a graph

Description automatically generated

Fig 3.2.1 Time Series Decomposition for the Execution Time on MacBookPro1

**A screenshot of a graph

Description automatically generated**

Fig 3.2.2 Time Series Decomposition for the Execution Time on MacBookPro2

**A group of blue lines

Description automatically generated**

Fig 3.2.3 Time Series Decomposition for the Execution Time on Raspberry Pi

**A diagram of a sound wave

Description automatically generated with medium confidence**

Fig 3.2.4 Time Series Decomposition for the Execution Time on the Virtual Machine

There is a clear trend in the execution time of the MacBookPro 2 and Raspberry Pi as shown in Fig 3.2.2 and Fig 3.2.3. The range of the trend values is consistent with that of the density plots shown in Fig 3.1.3.

However, in our opinion, these trends are weak and not enough to affect the execution time dramatically. By visual inspection, the density plots in Fig 3.1.1 show minimal overlap between the 4 sample distributions even when time trend is not removed.

Thinking of our initial objective, and considering that time trend didn’t affect the execution time considerably, we decided to continue our analysis without the time component.

**3.3 Significance Test**

* Result says the difference in execution times between samples are significant.

1. **Model Selection**

Based on our analysis in section 3, we decided to use the Generalized Linear Model Regression to find the effects of each of our exploratory variables.

where the response variable is:

: execution time in seconds

where the explanatory variables are:

: CPU Clock Speed in GHz

: CPU Core count, integer

: RAM size, in GB, {2, 4, 8, 16, 32}

In the following section, we explored two cases:

Case 1: no transformation of the RAM size value

Case 2: with transformation of the RAM size value

Our entire data set consists of 4 different PC configurations. However, the CPU Core Count and CPU Clock speed were not available for the VM data so we elected to remove it. We combined the data from the following three PC configurations and generated our model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PC Label** | **CPU Speed (GHz)** | **CPU Core Count** | **RAM size (Case1)** | **RAM size (Case2)** |
| MacBookPro1 | 1.4 | 4 | 8 | 3 |
| MacBookPro2 | 2.5 | 2 | 8 | 3 |
| Raspberry Pi | 1.8 | 4 | 4 | 2 |

Table 4.a Values of Explanatory Variables From the Available Data Set

* 1. **Case 1: No Transformation of RAM Size**

Table 4.1-1 summarizes the results after running a generalized linear model regression on our data for case 1. We found that an increase of "1GB" in RAM size means a reduction of 0.1706 seconds in the execution time. Considering that RAMs come in fixed sizes of {2, 4, 8, 16, 32GB, etc}, this effect would be even more noticeable when upgrading bigger sized RAMs.

Surprisingly, an increase of 1GHz in the CPU Clock speed, or a unit increase in the CPU Core count, were found to increase the execution time. All coefficients were found significant.

A screenshot of a computer

Description automatically generated

Fig 4.1-1 Generalized Linear Model Regression Results for Case 1

* 1. **Case 2: With Transformation of RAM Size**

Considering that RAM sizes are always a power of 2 - i.e. {2, 4, 8, 16, 32GB}, we decided to generate another GLM model by transforming the RAM size variable into linear values by taking the logarithm of the RAM size (base 2).

**A screenshot of a computer

Description automatically generated**

Fig 4.2-1 Generalized Linear Model Regression Results for Case 2

In the model above, a "unit" increase in RAM configuration (e.g. 2GB to 4GB, or 4GB to 8GB, etc) causes a decrease of 0.6238 seconds in the execution time, adjusting for CPU clock speed and core count.

With this transformation, the RAM size effect becomes higher while the effects of the other variables remained relatively similar to case1. This suggests that any unit change in RAM configuration has a big effect on the improvement of the execution time regardless of size.

1. **Model Analysis**
   1. **Model Comparison via AIC**

|  |  |
| --- | --- |
| **Model** | **AIC value** |
| Case 1 | -3014.85 |
| Case 2 | -3014.85 |

Table 5.1-1 AIC values for both GLM Cases

Since the AICs of both models above are equal in value, we are compelled to conclude that both models are equally effective in predicting the execution time.

However, we would like to highlight that our data only has two variations of the RAM size - i.e. 4GB, or 8GB.

This means that the logarithmic nature of this variable will not be evident to the GLM which caused the AIC to be the same in both cases.

If we have data from all commercially avaiable RAM size variations, we assume that the GLM using log-transformed RAM size would get a better/lower AIC.

* 1. **Residual?**

1. **Conclusions and Recommendations**