DATA SCIENCE PROJECT REPORT

FUTURE GPUs MEMORY PREDICTION

Team Members

Shahtaj Qasim (K142156 – GR3)

Wajeeha Memon (K142272 – GR4)

Submission Date: 29th April 2018

Research Goal:

As the world is advancing, humans are coming up with new and complex technologies and approaches. Some complex technologies and methods may take a lot of time to execute. But we need to have something which helps us execute things on our computer faster, for example; we would not like if our video game is running slow and not properly, and for this purpose we have GPUs (Graphics Processing Unit). It is also very important if our GPU will have enough memory to store and let play our video game. It is always a good idea to predict what will be the state of the GPUs in the coming future. Our main research goal is to predict the future memory size of the GPU by looking at history data of the GPUs. This can give us rough estimation of in which future year we will be having what memory size of a GPU, through which we can also roughly estimate the importance of GPUs for computations throughout the years and how powerful with increasing memory sizes GPUs will be.

Retrieving Data:

We searched GPUs data sets from kaggle.com and chose the most suitable one. In this step, we had to understand the data and figure out the size, attributes of the data and if the data will be meaningful and suitable for our problem of future GPU memory prediction. There are total 33 attributes and 3407 instances in data set. The dataset is in comma separated values file. The chosen data is to be used for further processing. We used external approach to get the data so that we could be able to have access to the suitable data easily. The following below is the dataset description:

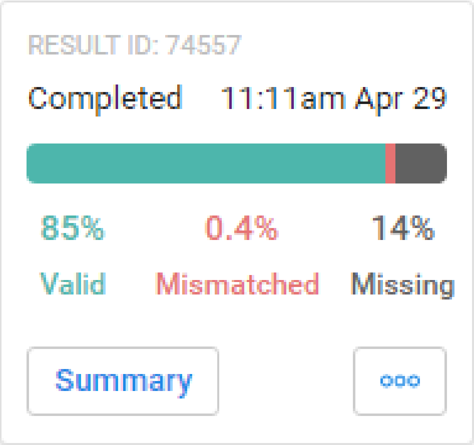
1. **Architecture:** It contains the architecture name of the GPU. The column in in object (string).
2. **Best_Resolution:** It contains the resolution of the GPU which is optimum. It is defined in Width X Height. The column in object (string).
3. **Core_Speed:** It mentions the specific speed of the GPU. It is given in MHz.
4. **DVI_Connection:** Contains the Digital Visual Interface connection in integers.
5. **Dedicated:** It has Boolean entries which tell whether architecture is dedicated or not.
6. **Direct_X:** It contains graphic card software information of the GPU.
7. **DisplayPort_Connection:** It contains information about the port connections in integers.
8. **HDMI_Connection:** It contains information about HDMI connections in GPU in integers.
9. **Integrated:** It contains Boolean information about whether the GPU is integrated or not.
10. **L2_Cache:** Contains the cache size of the GPU.
11. **Manufacturer:** It contains the manufacturer name of the GPU. Nvidia, Intel and AMD are the three values in it.
12. **Max_Power:** It contains the maximum power GPU can have. It is given in Watts.
13. **Boost_Clock:** It tells the boost clock frequency of the GPU. It is provided in MHz.
14. **Memory:** It tells the memory size of the GPU. It is given in MB.
15. **Memory_Bandwidth:** It contains memory bandwidth of the GPU. It is provided in GB/sec.
16. **Memory_Bus:** Contains the size of memory bus of the GPU. It is provided in Bit.
17. **Memory_Speed:** Contains how fast the memory performs which means the memory speed of the GPU. It is given in MHz.
18. **Memory_Type:** GPUs have some memory type of the memory. This attribute contains that information.

19. **Name:** This contains the name of the GPU.
20. **Notebook_GPU:** It is a Boolean column which tells whether GPU is notebook GPU or not.
21. **Open_GL:** It defines the version of OpenGL along with graphics card. It is provided in float.
22. **PSU:** It defines the power supply unit of the GPU. It is given in Watts.
23. **Pixel_Rate:** It defines the pixel rate, which is provided in GPixel/sec, of the GPU.
24. **Power_Connector:** It contains number of power connectors in the GPU or none if there are none.
25. **Process:** It contains the process size of the GPU. It is given in nm (nanometer).
26. 26: **ROPs:** It defines render output unit in integers.
27. **Release_Date:** It tells the release date of a specific GPU.
28. **Release_Price:** It contains the price of a specific GPU.
29. **Resolution_WxH:** It contains the resolution of the GPU which is provided in width x height.
30. **SLI_Crossfire:** It is a Boolean column informing whether or not GPU has SLI cross fire.
31. **Shader:** It contains pixel shader of the GPU in integers.
32. **TMUs:** It contains texture mapping unit of the GPU in integers.
33. **Texture_Rate:** It contains the Texture Rate of the GPU in GTexel/sec.

We found some of the attributes or features to be very meaningful to our problem of predicting memory of GPUs in future so we perform further processing on this dataset.

Data Preparation:

After data was retrieved, it needed to be cleaned. The complete data was cleaned using Trifacta Wrangler tool. Wrangler helps data analysts clean and prepare messy, diverse data more quickly and accurately. It is a simple tool with having a good visualization of the data and the easily identifying the messy data. Here, we cleaned the data and made it suitable for the model to be used without generating any errors or giving bad results. We simply imported the dataset file which needed to be cleaned and exported the file after cleaning the data which was then added in our project to be read and perform further processing. It was found that data contains missing values and mismatched values which were changed according to our desired data suitability. To be exact, the data set had 14% missing values and 0.4% mismatched values as shown in Figure 1. The results were checked and result summary was generated both before and after figure and they are shown in Figure 1 and Figure 2 respectively.



RESULT ID: 74557

Completed 11:11am Apr 29

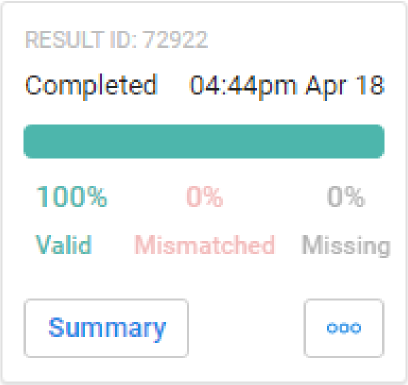
85% Valid

0.4% Mismatched

14% Missing

Figure 1: Dataset results before Data Cleaning –Trifacta Wrangler

The data was 100% cleaned after some processing, like, replacing null values with most frequent values and mismatched values were also resolved. First, we resolved the missing values and then we performed processing for resolving mismatched values. The data was resolved by using the most frequent value (mode) as a technique to clean. The results of cleaning by Trifacta Wrangler are shown in Figure 2.



RESULT ID: 72922

Completed 04:44pm Apr 18

100% Valid

0% Mismatched

0% Missing

Figure 2: Dataset results after Data Cleaning–Trifacta Wrangler

Data Exploration:

In this step, we further analyzed the data. We used JetBrains PyCharm to perform further processing by coding in Python. We took a deep dive into the data to get the information we needed. The attribute named 'Release_Date' was trimmed and from it year was extracted which were then stored in an array. The purpose of doing this is that 'Release_Year' is very important information we need from the data. We can see graph for year and number of GPUs using 'count plot' in Figure 3.

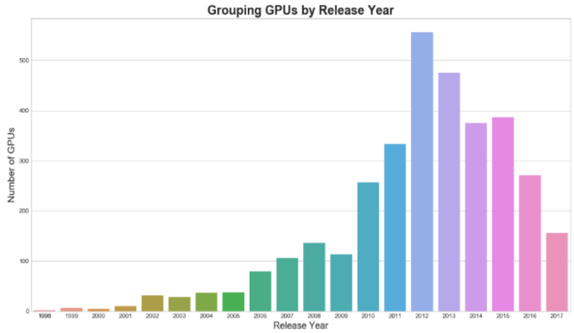


Figure 3: Grouping GPUs by Release Year

The data contains GPU information from 1998 to 2017. Figure 3 visualization showed us the demand of GPUs by their release year. We then created a scatter plot for release year and memory of GPUs, which is shown in Figure 4, to have better understanding about in which year what memory the GPUs had.

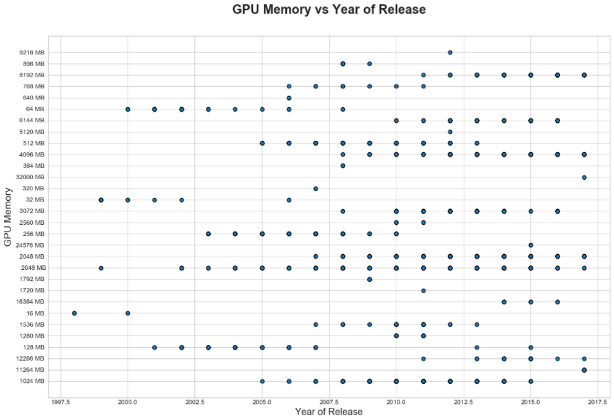


Figure 4: Memory vs. Year of Release using scatter plot

The data set was then grouped by 'Release_Year' and 'Memory' and mean and median was calculated which is shown in Figure 5. We can see that the GPU memory is increasing from year to year.

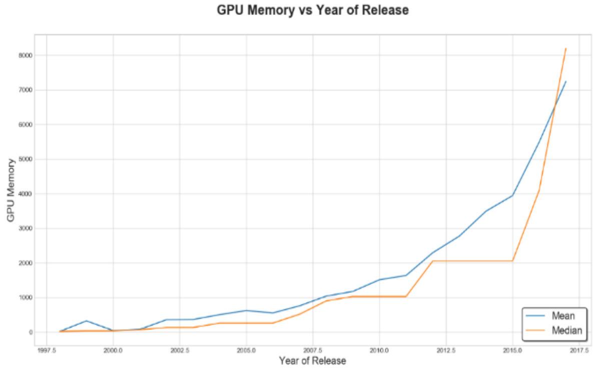


Figure 5: GPU Memory VS Release Year (Mean and Median)

Data Modeling:

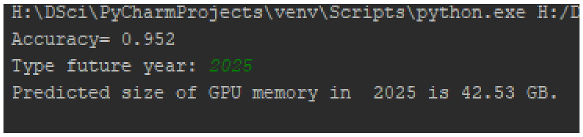
Firstly, we applied polynomial regression in which features were generated using Polynomial Features function and it generated a new feature matrix consisting of all polynomial combinations of the features. Then we apply Linear Regression and then fit it with polynomial features and memory mean by release year. Linear regression is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables). We use exponential curve formula shown below in (1) to calculate the prediction of memory by the specified year.

a*2**((x-c)*b) ----- (1)

This exponential curve function is then used by ‘curve_fit’ function which has a parameterized model function meant to explain some phenomena and wants to adjust the numerical values for the model so that it most closely matches some data. Moreover, we used fitting functions for Polynomial Features and Linear Regression and found out the accuracy by using ‘r2_score’ which gave the answer 0.952.

Presentation and Automation:

All the above steps lead to the output of the system. The system is able to successfully predict the future size of GPU. The user is required to enter any year greater than 2017 as we are trying to predict the future memory size. We get the accuracy of the models to be 0.952. The system then performs the calculations and gives the memory size result in GB. The output sample is shown in Figure 6. As we can see in Figure 6, we gave the input ‘2025’ which gave a predicted size of the GPU memory to be 42.53 GB.



```
H:\DSci\PyCharmProjects\venv\Scripts\python.exe H:/D
Accuracy= 0.952
Type future year: 2025
Predicted size of GPU memory in  2025 is 42.53 GB.
```

Figure 6: Output of the system

Conflicts between the team members:

The only conflict that occurred between the team members was while when we were choosing the topic for the project and also the dataset. But the team agreed on one topic and dataset. Everything went smooth after the topic and dataset was decided. The team put efforts together in succession of this project.