|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ALIBABA 2020 DATASET FOR TASK SCHEDULING** | | | | |
|  |
| **TASK FEATURES** | | | | |  |
|  |
| **memory usage: 38.1+ MB** | | | | |  |
| **RangeIndex: 499999 entries, 0 to 499998** | | | | |  |
|  |
| **Data columns (total 10 columns)** | | | | |  |
| **dtypes: float64(3), int64(3), object(4)** | | | | |  |
|  |
| **ID** | **Columns names** | **Description** | **Format** | **Null Values** |  |
| 1 | job\_name | in this column we have our job names | object | 0 |  |
| 2 | Task\_name | in this column we have our task names | object | 0 |  |
| 3 | Inst\_num | in this column we have instance number | int64 | 0 |  |
| 4 | Status | in the column we have our task status | object | 0 |  |
| 5 | Start\_time | we have start time of our task | int64 | 0 |  |
| 6 | End\_time | end time of our task | int64 | 0 |  |
| 7 | plan\_cpu | in this we have our plan cpu | float64 | 351 |  |
| 8 | plan\_mem | we have our memory | float64 | 351 |  |
| 9 | plan\_gpu | our GPU | float64 | 99,150 |  |
| 10 | gpu\_type | and in this columns we have our gpu type | object | 99,009 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MACHINE METRICS** | | | | | |
|  |
| **memory usage: 45.8+ MB** | | | | | |  |
| **RangeIndex: 499999 entries, 0 to 499998** | | | | | |  |
|  |
| **Data columns (total 12 columns)** | | | | | |  |
| **dtypes: float64(8), int64(2), object(2)** | | | | | |  |
|  |
|  | **ID** | **Columns names** | **Decription** | **Format** | **Null Values** |  |
|  | 1 | worker\_name | in this column we have our work names | object | 0 |  |
|  | 2 | machine | in this column we have our machine names | object | 0 |  |
|  | 3 | start\_time | Start time of our task | int64 | 0 |  |
|  | 4 | end\_time | end time of our task | int64 | 0 |  |
|  | 5 | machine\_cpu\_iowait | machine waits for input/output | float64 | 42,111 |  |
|  | 6 | machine\_cpu\_kernel | end time of our task | float64 | 59,422 |  |
|  | 7 | machine\_cpu\_usr | in this we have our CPU usr | float64 | 43,613 |  |
|  | 8 | machine\_gpu | our GPU | float64 | 53,859 |  |
|  | 9 | machine\_load\_1 | Load on our machine | float64 | 50,347 |  |
|  | 10 | machine\_net\_receive | Networks receive by machine | float64 | 46,597 |  |
|  | 11 | machine\_num\_worker | Number of machine worker | float64 | 52,708 |  |
|  | 12 | machine\_cpu | Machine cpu usage | float64 | 60,953 |  |

**Numerical analysis (summary):**

**Run time:**

We have no column name runtime in our machine metrics, so we calculate our run time using start time minus end time.

**AFTER MERGING BOTH TABLES:**

|  |
| --- |
| **Int64Index: 400849 entries, 44 to 499998** |
| **Data columns (total 16 columns):** |
| **dtypes: float64(8), int64(3), object(5)** |
| **memory usage: 52.0+ MB** |
| **No. Column Non-Null Count Data type** |
|  |
| 1 task\_name 400849 non-null object |
| 2 inst\_num 400849 non-null int64 |
| 3 status 400849 non-null object |
| 4 start\_time 400849 non-null int64 |
| 5 end\_time 400849 non-null int64 |
| 6 plan\_cpu 400849 non-null float64 |
| 7 plan\_mem 400849 non-null float64 |
| 8 plan\_gpu 400849 non-null float64 |
| 9 gpu\_type 400849 non-null object |
| 10 run\_time 328573 non-null float64 |
| 11 machine\_cpu\_iowait 328573 non-null float64 |
| 12 machine\_cpu\_kernel 328573 non-null float64 |
| 13 machine 328573 non-null object |
| 14 machine\_load\_1 328573 non-null float64 |
| 15 machine\_cpu 328573 non-null float64 |

**Data Filtering:**

**Removing Null Values**

* First we calculate our run time because it’s our target column
* After adding both columns we have 351 null values in plan\_cpu we remove this nulls
* we have 351 null values in plan\_mem we remove this nulls.
* we have 99150 null values in plan\_gpu we remove this nulls.
* we have 99009 null values in gpu\_type we remove this nulls.

**Encoding Columns**

* we have categorical data in our three columns gpu\_type, status and machine so we convert our this categorical columns into numeric values using one hot encoding

**Removing Columns:**

* we have no need of columns 'job\_name', 'task\_name' so remove this columns from our dataset.

Because this features relationships is now strong and also there is no need of task name and job name in our model training so we removed that columns.

**Splitting Dataset:**

* we split our data into 30:70 ratio because it’s a good practice for training our model 70% data goes to input during training and remaining 30% data is used for evaluation of our model.

**Evaluation:**

We evaluate our model using the following formulas:

* Mean Absolute Error
* Root Mean Squared Error
* R-squared Score

**THE ANSWERS TO ALL THE QUESTIONS IF YOU GET CONFUSED**

*Q1) I do not understand tha input and output.*

*We have Task feature as input and our output is machine metrics.*

***ANSWER***

**input Columns :**

['plan\_cpu', 'inst\_num', 'status','plan\_mem','plan\_gpu','gpu\_type','start\_time','end\_time']

**Target Columns :**

['run\_time','machine\_cpu','machine\_load\_1']

*Q2) What is the 40 input? It should be task features.*

***ANSWER***

The 40 Input values contains a task features.

['plan\_cpu', 'inst\_num', 'status','plan\_mem','plan\_gpu','gpu\_type','start\_time','end\_time']

So as we have 8 columns in our task features so we make a 5 – 5 sets of each columns to give to our time series model that’s why it’s looks like 40 inputs . our purpose is that to apply time series . so, we give this to our model and we give 5 instances to our model and on the basis of these five instances our model predicts the sixth instance it self

*Q3)And the output should be run Time, cpu, memory.*

***ANSWER***

Yes. In the given requirements we have our target coloumns

['run\_time','machine\_cpu','machine\_load\_1']

So, when we give a previous 5 instances as an input to our model so we will know the next one.

*Q4)There are too much preprocessing data like length it is not clear.*

***ANSWER***

As we have two different tables so we have to normalize and preprocess two tables in a single file of code. So that’s why its look like too much because we have to preprocess two datasets as our Task feature and machine metrics table so we preprocess and normalize both datasets separately that’s why it’s looking too much preprocessing. If we have a single dataset or table then there no need of too much preprocessing. But in the given task there is requirement of two tables machine metrics and task feature that’s why we need to do this large preprocessing because this is our models need.

*Q5)The prediction should be for next future hour? For runtime, cpu, memory- based on the task features.*

***ANSWER***

And in the given problem our model is predicting the next instances. We give input as a task features then our model predicts the next cpu memory and runtime

*Q6)Then the evaluation of Multi output NN regression, how can I say it is perfect? No comparison*

***ANSWER***

On the basis of our previous experiences, we used this model. Because we already did many tasks of time series so we have a lot of experience in the time series so we easily analyze after seeing the dataset that which model is best suited. So we use this model this model is best practice and have very better accuracy

*Q7)My target to predict the future for machine runtime, memory, cpu - based on the task features*

***ANSWER***

Yes, our model is fulfilling your all requirements as we give task feature as a input then our model will having a ability predict the future for machine mertrics

*Q8)How did u calculate run time no step in the code show that:*

***ANSWER***

We calculate runtime using excel formula we used start time and end time to calculate the total run time. So as already we have to much preprocessing in our code so calculating runtime is not a very big deal . so, we do this in the excel. To manage our code.

*Q9)Why u choose to remove null values not to add the average or other methods.*

***ANSWER***

Because already we have too much large dataset to train our model and our model is doing very best performance on the data. So that’s why we don’t a need to try any other methods. We use other methods when we have too much short dataset and if we remove the null values then our dataset get too much short that’s why in that cases we use average , mean or some other techniques but in the given dataset there is no need to do that.

*Q10)Is this give better performance?*

***ANSWER***

Yes. We tried some other techniques but that’s not giving good results. So, we ensure that this accuracy is very best. And as I already told you we make too much models of time series we have past experiences. So, we selected that best model according to our data.

*Q11) The size of the sliding window should be based on the dataset size, we have a long list of the instances, why u choose 5? Not 100 or 1000?*

***ANSWER***

It’s not necessary to make set of 5 instances . we can select more then 5 , 100 or 1000 of sets if we select more instances then our models take a lots of time to train and model get too much complex. So, this is not a fixed value. We can increase if we have more powerful computer machines.

*Q12) The result of the prediction will be just for one single task next? not music tasks prediction.*

***ANSWER***

No, we can predict next to next tasks. This is not a single task next. We can predict more the one task.

*Q13) The prediction will focus on tasks, not on machines.*

***ANSWER***

Yes our prediction is based on our task because task features is our input, and our machine is output.

**QUESTIONS / ANSWERS**

-The tasks are working in one machine or multiple machines in once.

**Ans:** in a one macine

- the tasks are working individually or working concurrently ?

**Ans :** the tasks are working concurrently.

- the prediction wiill be for one node? Or multiple?

**Ans:** for multiple node

- is the data linear or not

**Ans:** data is nonlinear.

- the prediction for example: one machine run 100 in one hour — so the prediction will be how many tasks in next hour? What is our goal or assumption?

**Ans:** Our Goal is (Run time , CPU , Memory)

- how to sort the data based on machines? Or tasks?

**Ans:**  based on task.

- the work will check for one machine or all tasks on all machines.

**Ans:**

- like with predicting we will know the machine idle or busying!?

**Ans:** Machine idle

- how many tasks we have, and how many on each machine? How many machines we have?

**Ans:**

- description of data and why we choose these specific tables and features to work with?

**Ans:** Because this is best for our model.

- what are the types of the tasks and how many tasks we have from each task?

**Ans: Hierarchical data(**integer , Float , string) **Ans:**

- we need to have one more output which is GPU usage because we choose GPU plan and GPU type in the input.

**Ans:** our objectives is (Memory , CPU , Runtime) . GPU usage is not mentioned in our objectives.

- the machines are heterogeneous because the architecture of each one is different, what are the architecture of machines, how many machines we have from each one? How many different architectures we have in the data?

**Ans:**

- how can we add the priority of the tasks in the input? Is it range? Or how?

**Ans:**  it is based on correlation.

- the result will give set of machines with their capabilities as they are good to execute the task? Or how is it looking like?

**Ans:**

-