Documentation

of

UiPath RPA Log Dashboard

Team

Savvy Insights

# **Team Members**

* Swetha Srivarna Selvaganesan(swethasrivarnas@gmail.com**)**
  + - * SuhadevVenkatesh( [Suhadevvenkatesh@gmail.com](mailto:Suhadevvenkatesh@gmail.com))

Contents

[**Team Members** 1](#_Toc522557878)

[**1.** **Problem Statements** 3](#_Toc522557879)

[**2.** **Key Objective** 3](#_Toc522557880)

[**3.** **Functional Requirement** 3](#_Toc522557881)

[**4.** **Software Requirements** 3](#_Toc522557882)

[5. **Use Cases** 4](#_Toc522557883)

[5.1 Descriptive use case: 4](#_Toc522557884)

[5.2 Logical Analytical Insights: 7](#_Toc522557885)

[**6.** **Scope Statement** 7](#_Toc522557886)

[6.1 Failure Prevention 7](#_Toc522557887)

[6.2 Failure mode Prediction 7](#_Toc522557888)

[7. **Approach** 8](#_Toc522557889)

[**8.** **Cost involved** 10](#_Toc522557890)

[**9.** **Assumptions, Risks & Limitations** 10](#_Toc522557891)

[9.1 Assumptions 10](#_Toc522557892)

[9.2 Risks 10](#_Toc522557893)

[9.3 Limitations 10](#_Toc522557894)

# **Problem Statements**

Ready-made dashboards which can be deployed by users in their environments.Will use the UiPath robot’s logs as a data source.Can be made in Kibana or any other reporting/BI platform. It would be great if a mockup is submitted along with the idea, or at least comprehensive descriptions of all the data points. File type: .json files

# **Key Objective**

To provide descriptive analysis from the UiPathrobot’s logs that would help users to take business decisions and bring value. Also to forecast the trends based on analytics model that would help in rendering both predictive and preventive analytics.

Value would include:

* Load planning based on the Bot Availability
* Delivery Planning based on the Bot Performance and Runtime
* Schedule Planning (On-time and Hassle-free execution and Transactions)
* Failure prediction and prevention
* 100% Utilization of the Bots
* Cost savings
* Return on Investments

# **Functional Requirement**

* Data source from UiPath robot’s details
* Data source from UiPath robot’s logs
* Data Storage and Processing
* Connecting Data source with the BI tool
* Identification of all Use cases and Metrics
* Dashboard development using Tableau Public/any BI tool
* Comprehensive descriptions of all the use cases

# **Software Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Software/processor/dependencies | Purpose | Download Link |
| 1. | Python Anaconda3 | Convert .json to csv | https://www.anaconda.com/download |
| 2. | pillow |  |  |
| 3. | lxml | Data Visualization |  |
| 4 | jupyter |  |  |
| 5 | matplotlib |  |  |
| 6 | PaperSpace | Data Processing | https://www.paperspace.com |
| 7 | [NVIDIA TITAN X](https://www.nvidia.com/en-us/geforce/products/10series/titan-x-pascal/#titanx) |  | https://www.nvidia.com/en-us/geforce/products/10series/titan-x-pascal/ |
|  | opencv |  |  |
|  | tensorflow |  |  |
|  | numpy |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **Use Cases**

* DataCollection

A **data set** (or **dataset**) is a collection of data. Most commonly a **data set** corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the **data set** in question.In our project we will be using a dataset of CAID(Child abuse Image Database),and along with our own dictionary with abusive words in it.

* Data Stage for training

In This stage we will be looking into each image with at most attention and marking all the abusive content in it using a software called Labeling, which allows us to select the abusive content so that we can train our system in a better way, the output of this system is a xml file

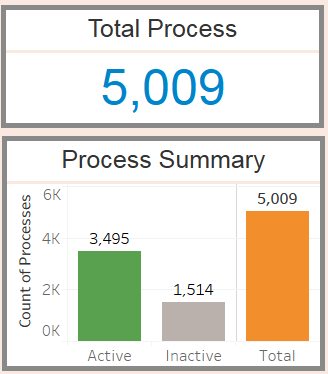
These XML files are converted into singular CSV files that can be then converted to the TFRecord files. To do this, we made use of some of the code from [datitran's github](https://github.com/datitran/raccoon_dataset), with some minor changes. To begin, we used  [xml\_to\_csv.py](https://github.com/datitran/raccoon_dataset/blob/master/xml_to_csv.py). We have two options. We can either use a pre-trained model, and then use transfer learning to learn a new object, or we could learn new objects entirely from scratch. The benefit of transfer learning is that training can be much quicker, and the required data that you might need is much less. For this reason, we're going to be doing transfer learning here.

TensorFlow has quite a few pre-trained models with checkpoint files available, along with configuration files. We can do all of this by ourselves.

Model

* Process Count Summary(Total, Active, Inactive)

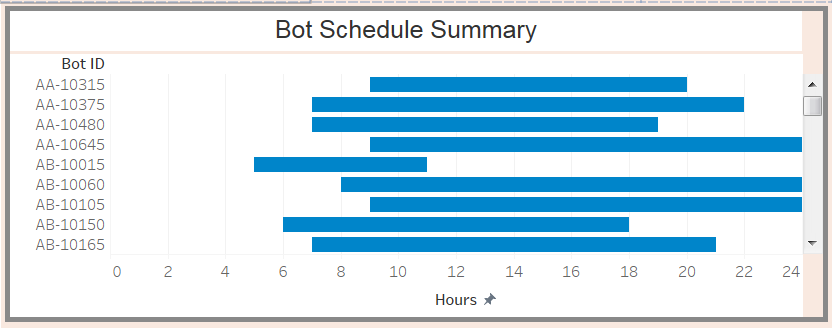
The below allows the Bot Owner to understand the total number of available processes in the business and lets them to plan their delivery and load balancing accordingly. This gives the high level view of the complete process history.



* Schedule Summary

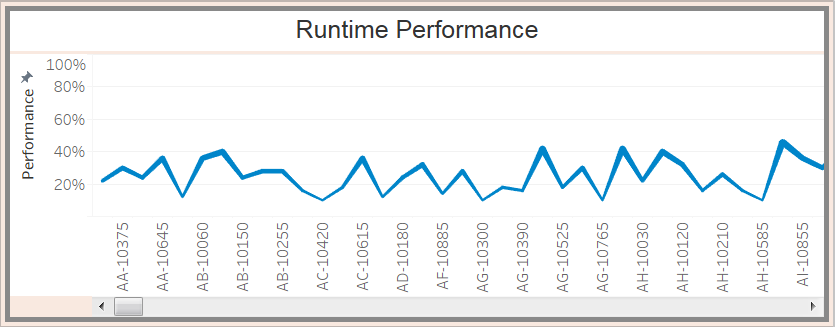
The Schedule summary gives the total allocation of the Bot for a given day over 24 hours. Based on this Gantt chart, the Bot Owner could assign the schedule for any upcoming process.

*The Scope of this chart is to provide recommendations in the future based on the analytics model.*



* Runtime Performance

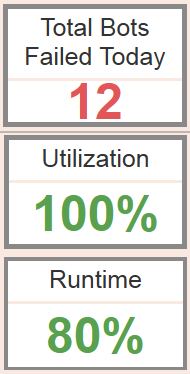
This gives the performance of the Bot per day. The logic would be calculations based on the number of hours a Bot is running to the number of process successfully completed by the Bot.



* Performance details

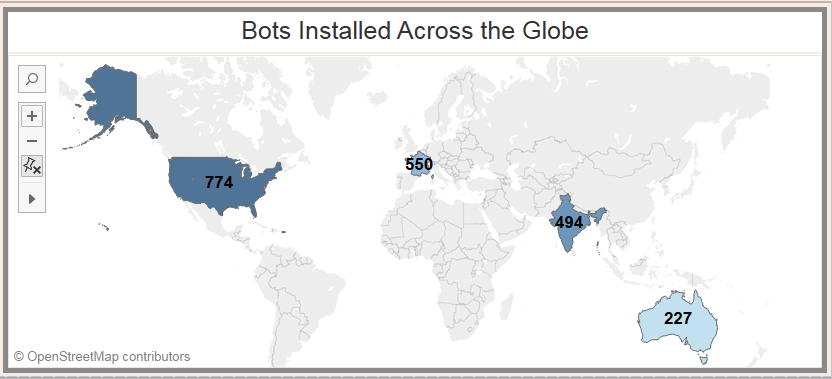
These metrics below allows the users to understand the failure of any Bots on a given day; whether all the Bots are completely utilized and the Average Runtime Performance of all the Bots.

*The Scope of Utilization would allow us to provide the cost savings and return on Investment for all the Bots.*



* Global Distribution

Global Distribution allows the user to have a



* Transaction history

## Logical Analytical Insights:

* Schedule Planning and Recommendations
* Failure alerts
* Cost saving factors
* Trend Analysis
* Error Forecast
* Utilization

# **Scope Statement**

## Failure Prevention

To build analytics model to prevent the failures of Bots in the future based on the metrics identified. Prevention would involve message/email alerts sent to the Bot owner.

## Failure mode Prediction

To build analytics model to predict the failures of Bots. Based on the prediction, the dashboard would forecast the utilization and cost savings incurred to the business.

# **Approach**

**Application Requirements**

(i) **What to Create:**Makers must create an open source reusable components in one of the required categories using the UiPath Studio that automate repetitive tasks in the workplace. (each a “Component”).

(ii) **Categories:**

1. ~~Custom Activities component: covering areas not already handled by default activities that come with UiPath Studio.~~
2. Dashboard component: ready-made dashboards pulling data from UiPath robots logs.
3. Application and Data Connectors: components that enable UiPath to interact with other applications and databases
4. ~~Snippet component: reusable, ready-made workflows that are useful to as many users, environments, and processes as possible.~~
5. ~~Automation Framework (Automation templates): frameworks handling common tasks like error handling, reporting, and environment setup~~
6. ~~Machine Learning models: ready-trained machine learning models which can be used inside a project~~

(iii) **Functionality:**The Component must be capable of being successfully installed and running consistently on the platform for which it is intended, and must function as depicted in the video and/or expressed in the text description.

(iv) **New & Existing:**Components must be either newly created by the Maker or, if the Component existed prior to the Hackathon Submission Period, must have been updated after the start of the Hackathon Submission Period.

(v) **Testing:**The Maker must make the Component available free of charge and without any restriction through a Github or other publicly available code repository, for testing, evaluation and use by the Poster, Administrator and judges during and after the Hackathon.

(vi) **Public Distribution:**The Maker must make the components open source under [MIT license](http://opensource.org/licenses/MIT).

(vii) **Multiple Submissions:** A Maker may submit more than one Submission, however, each Submission must be unique and substantially different from each of the Maker’s other Submissions, as determined by the Poster and/or the Administrator.

(viii) **SDKs, APIs, & Data:**Component may integrate SDKs, APIs and data, provided the Maker is authorized to use them.

(ix) **Intellectual Property:**Your Submission must: (a) be your (or your Team, or Organization’s) original work product; (b) be solely owned by you, your Team, or your Organization with no other person or entity having any right or interest in it; and (c) not violate the intellectual property rights or other rights including but not limited to copyright, trademark, patent, contract, and/or privacy rights, of any other person or entity. A Maker may contract with a third party for technical assistance to create the Submission provided the Submission components are solely the Maker’s work product and the result of the Maker’s ideas and creativity, and the Maker owns all rights to them. A Maker may submit a Submission that includes the use of open source software or hardware, provided the Maker complies with applicable open source licenses and, as part of the Submission, creates software that enhances and builds upon the features and functionality included in the underlying open source product. By entering the Hackathon you represent, warrant, and agree that your Submission meets these requirements and you acknowledge that you will hold Poster harmless of any claim from a third party referring to the breach of its intellectual property rights or other rights. You acknowledge that in case there is a reasonable suspicion that your Submission is not compliant with this section, your Submission will be disqualified.

(x) **Financial or Preferential Support:** A Component must not have been developed, or derived from an Application developed, with financial or preferential support from the Poster or Administrator. Such Applications include, but are not limited to, those that received funding or investment for their development, were developed under contract, or received a commercial license, from the Poster or Administrator any time prior to the end of Hackathon Submission Period. The Poster, at their sole discretion, may disqualify an Application, if awarding a prize to the Component would create a real or apparent conflict of interest.

# **Cost involved**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Software** | **Product Type** | **Cost Per Annum for Basic Implementation** |
| 1 | SQL Server | Standard - Per core | USD 3,717.00 |
| 2 | Python | Open Source | USD 0.00 |
| 3 | Tableau | Tableau Creator | USD 840.00 |
|  |  | **Total** | **USD 4,557.00** |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Software** | **Product Type** | **Cost Per Annum for Enterprise Implementation** |
| 1 | SQL Server | Enterprise | USD 14,256.00 |
| 2 | Python | Open Source | USD 0.00 |
| 3 | Tableau | Enterprise | USD 152,400.00 |
|  |  | **Total** | **USD 166,656.00** |

# **Assumptions, Risks & Limitations**

## Assumptions

* Data source is consistent and unique(has the exact data fields).

## Risks

* Sample dataset would differ from the Actual. Thus data engineering has to be considered before the dashboard is developed.

## Limitations

* Utilized Tableau Public (which involves data in Public cloud). Investment on Tableau tool is high.

Alternative: In-house tools could be used to deploy the same use cases.

* Sample Bot log created for the dashboard. Connection to Actual Bot logs could involve changes in the scripts.