R Model Prediction

Overview

R is a powerful programming language and software environment for statistical computing. By combining scripts created in R with Yellowfin, you can get advanced statistical analysis of your data.

With the R Script transformation step, it is possible to use analytical models and applications created using this language in Yellowfin. You can use an R script via this step to get the results it's designed to generate on your Yellowfin data.

Prerequisites

Before using this step, ensure that you have the following set up:

- Access to an R instance with the Rserve package running.
- At least one valid R script.
- The R plug-in installed in your instance of Yellowfin. You can download the R plug-in from the Marketplace.

It is assumed that the user using the data transformation module to use the R script, knows the script well, and understands its input and output requirements.

Further documentation on Rserve: https://cran.r-project.org/web/packages/Rserve/Rserve.pdf

General workflow

Here is a quick overview of the process. This guide will cover most of these steps in detail.

1. You either need access to an instance of R or set one up yourself. Write your R script and make sure it is valid. This instance will be used to execute this script.
2. Install the R plug-in in your instance of Yellowfin. (You can download this from the Marketplace.) Learn about installing plug-ins here.
3. Use Yellowfin's data transformation module to create a transformation flow. This involves importing data into the module, and if required applying other transformations to it.
4. Use the R transformation step in your flow and configure it. (This guide will cover this in detail.)
5. Execute the step to view the output generated by the script using your data.

Types of data inclusion modes

The result generated by the R transformation step will depend on the following options:

- **Append:** Append is used if additional columns are to be added to existing data. This option is used if your R script is designed to return the original data, along with the appended columns.
- **Replace:** Use this option if your script outputs specific columns, and doesn't necessarily returns the original data.

R Script Criteria

When writing an R script, you need to ensure that it works in Yellowfin. The following guidelines will help you to do that.

Example 1: Append

Here is an example of a basic script that returns output by adding two columns to the input data. This illustrates how the append option can be utilized.

```
outData<-data.frame(c(yfData), yfData[1], yfData[3])
```

Here's a breakdown of this script, and how this should be configured in Yellowfin.

- yfData - yfData is the data frame that contains the input data, or data that is being passed from the previous step into the script. This is the name that we will provide as the input variable when configuring the step.
- yfData[1] / yfData[3] - The script inputs the entire data frame and appends column 1 and 3 to it. This example shows how indexing is used to point out specific data columns.
- outData - The name of the output variable. The combined result is stored in this variable.

Use these variable names when configuring this script in the R step (as shown in the example as well).

blocked URL

Example 2: Replace

This script has been written to retrieve only 1 numeric column from the incoming data and perform a calculation on it. We will use the Replace option when configuring this script.

```
outData<-data.frame(yfData$Income*2.5)
```

Following is an explanation of the script:

- yfData - The incoming data is taken from the yfData data frame, which is the input variable.
- Income - This is the name of the column field that is extracted from the input data. (Here we have directly referred to the column name using the $ symbol, instead of indicating it by its index number.) All the rows of this field will then be multiplied by 2.5.
- outData - The output variable where the resulting calculation will be stored.

When configuring this script, provide yfData as the input name and outData as the output name in the R step configuration panel. Enter 1 as the value of the total number of fields returned.

blocked URL

Guideline: Using R scripts in Yellowfin

Part 1: Setting Up R

It is recommended that you have the latest version of R (at least v3.4.0 or above), with Rserve running. You can download it from the link below: https://cran.r-project.org/mirrors.html. We also recommend using RStudio as your R environment to run Rserve. (https://www.rstudio.com/)

Refer to the commands below to help set up an R instance:

Running R on Windows

1. Install R on your local instance of Windows.
2. Execute the following commands from your R environment (to access Rserve locally):

```
      install.packages("Rserve") #This installs Rserve package
library(Rserve)  #This loads Rserve package
Rserve()   #This starts Rserve
```

3. Once these commands are executed successfully, Rserve should be installed and running. It is now also ready to be integrated into Yellowfin through the R script transformation step.
4. Use the following command to run Rserve on Windows, to allow it to be accessed externally:

```
      run.Rserve(args=" --RS-enable-remote")
```

Installing R and Rserve on Linux CentOS

1. Execute the following commands from your terminal:

```
      yum install epel-release -y #Required to install R
yum install R -y  #Installs R
wget  https://download2.rstudio.org/rstudio-server-rhel-1.0.44-x86_64.rpm  #Installs RStudio. (You might
want to install a different version though.)
```

2. If successful, the R service will be executed automatically.
3. To check the status of your R service, run the following command:

```
      systemctl status rstudio-server.service
```

4. After installation, run the following command to start R:

```
sudo -i R
```

5. Once R starts, run these commands to install Rserve:

```
install.packages("Rserve") #This installs Rserve package

library(Rserve)  #This loads Rserve package
```

6. To execute R so that it could be accessed externally, continue with the following steps:
    a. Once Rserve is installed, quit R.
    b. Start Rserve by using the following command:

    ```
    R CMD Rserve --RS-enable-remote --RS-port port
    ```

    Where port refers to the port number to start R.

    c. Once successful, you will be able to access R remotely.

Installing R and Rserve on Linux Ubuntu

1. Execute the following commands from your terminal:

```
sudo apt-get update
sudo apt-get install r-base r-base-dev
sudo apt-get install gdebi-core
wget https://download1.rstudio.org/rstudio-0.99.896-amd64.deb
sudo gdebi -n rstudio-0.99.896-amd64.deb
```

2. After installation, run the following command to start R:

```
sudo -i R
```

3. Once R starts, run these commands to install Rserve:

```
install.packages("Rserve") #This installs Rserve package

library(Rserve)  #This loads Rserve package
```

4. To execute R to be accessed externally, continue with the following steps:
    a. Once Rserve is installed, quit R.
    b. Start Rserve by using the following command:

```
R CMD Rserve --RS-enable-remote --RS-port port
```

Where port refers to the port number to start R.

c. Once successful, you will be able to access R remotely.

⚠ It is essential to have a properly set up and running R instance and Rserve, for this transformation step to function properly.

Part 2: Using an R script in the transformation flow

Follow these steps to use an R scripts and apply it to your data.

1. Ensure that you meet all of the prerequisites stated above.
2. Navigate to Yellowfin's Data Transformation model. (Create button > Transformation Flow)
3. Create a transformation flow beginning with an input step to extract data. (Click here if you want to learn how to create a basic flow, or here to learn about the different ways data can be extracted).

   blocked URL

4. The extracted data will appear in the data preview panel. You can add more steps to further transform the data.
5. Once you are ready to use your R script, follow the procedure below.
6. Drag the R Script step from the transformation step list onto the canvas. (Note: If you don't see this step, ensure that you have installed the R plug-in.)

   blocked URL

7. Using a connection point, create a connection from the last step to the R script step.

   blocked URL

8. Now you need to configure the R step. (Make sure it is selected to bring up its configurable details.)

   ⚠ Yellowfin utilizes the Rserve package. You must have Rserve installed and running before trying to connect.

   blocked URL

9. You have two options when it comes to connecting to your instance of R: local access (move ahead to step 10) or a remote one (skip to step 11).
10. **Connecting to a local instance of R:** If your Rserve is installed locally on your machine, then choose this option.
    a. For this option, simply click on the Connect to Rserve button. (No need to provide any parameters.)
    b. If successful, additional configuration settings will appear. (Skip ahead to step 12.)
    c. However, if you encounter the following error on trying to connect, then this means that your Rserve isn't working properly.

       blocked URL

11. **Connecting to a remote instance of R:** If the Rserve you're trying to connect to, is installed on a remote machine, then follow the steps below:
    a. Switch on the external connection button. The following fields will appear.

       blocked URL

    b. Provide the IP of the machine hosting Rserve. You can enter either the IP address (for example, 127.0.0.1) or the name of the host machine (e.g. localhost). Note: Ensure not to include "http://" before the IP address or host name, as this will prevent the connection from being established.
    c. Enter the port number of the network hosting R. This is the same port number that we used when starting Rserve.
    d. Note: Ensure that the machine you're trying to connect to is not password protected and does not require access through user credentials, as currently, connections with usernames and passwords are not supported in the R Script transformation step.
    e. Click on the Connect to Rserve button. Continue to step 12.

12. Once a successful connection is established, further configuration details will appear.

    blocked URL

13. Using those, include your R script, through either of the two methods:
    a. Drag your R script into the specified panel.
    b. Or switch on the Load from Path toggle, and submit the path to the file. Ensure that the full path is provided. For example: Windows: C:\Users\admin\Desktop\append.r, or in Linux: /root/append.r

    > ⚠ Make sure the file format is accurate, i.e. has a '.r' extension.

14. Provide configuration details for the script. **Note:** The details provided here are specific to the content of the R script and the function it's designed to perform. Therefore, it is assumed that the user is aware of the functionality included in the script.
15. Choose one of the two methods of including data: append or replace.
    a. **Append:** Select this method if the script returns the new field(s) generated, along with the input data. Then specify the number of new fields that will be added to the data.
    b. **Replace:** Choose this option if the script returns the result that it is designed to. (This could mean only the new fields, or a combination of the new fields and input fields, depending on what result the script creator has designed the script to produce.) Then provide the total number of field(s) the R script is supposed to produce.
16. Make sure that the correct number of fields are specified to continue successfully. The following error will appear on executing this step, if the wrong value is given.

    blocked URL

17. Enter the name of the input variable as mentioned in the script. This specifies where the data is to be read from.
18. Provide the name of the output variable, which states where the result is stored in the script. (The system will return the value stored in this variable as the result generated by the script. This parameter should be a data frame variable.)
19. If incorrect names are provided for either of the variables, error messages will appear when the step is executed. For example:

    blocked URL

20. Then click Apply.
21. If the step executes successfully, the result will appear in the data preview panel. The following example shows the result by appending the result fields to the existing data: (Note: In case of append, the resulting fields from the input data will have their original names and types. And the newly appended fields will be named "newField0", "newField1", etc., with their default type being text.)

    blocked URL

    Similarly, here's an example in which the data is replaced with the result as specified in the script. (Note: All of the fields returned in this case, will have names like "field0", "field1", etc. with the datatype being text.)

    blocked URL

22. If this step fails, however, it could be for a number of reasons. Click here to see the errors some of these issues generate. <link to the step execution issues section>
23. But if successful, you can perform further transformations or save your result in a database. <add links>

Step Execution Issues

In this section we will cover some cases in which the step execution could go wrong. Some of these cases could be:

- The wrong file path to the script has been provided.
- When the number of fields returned by the script does not match the expected number based on your input.
- The connection to Rserve was lost (in this case, you will most probably see an empty table).
- The script you uploaded is not a valid R script.

Rserve connection lost

It is possible that you could lose connection to Rserve, after already setting up the step. The following error message will appear then.

blocked URL

This in itself is not a big issue. You can restart the connection to Rserve and then execute the step again by clicking Apply. The step will run properly, assuming the connection has been properly established.

Incorrect field value

If you provide an incorrect number of added/total returned fields. In this case, the following error will appear on execution.

blocked URL