Toggle Language:

R

Documentation Menu

Tutorial

Installation

The `Yhat` R client, `yhatr` is distributed on `CRAN`. This means you can install `Yhat` the same way you do any other open source package. `R` makes it pretty easy using the `install.packages` function.

```
> install.packages("yhatr")
```

Your first model

First things first, we need to build a model! For our example use case we'll make a model that predicts the score difference in an NFL game based on information about the home and away team.

Getting the data

You can download the data here.

Building the model

We're going to use a basic `glm` (generalized linear model) to build a regression model.

```
df <- read.csv("~/Downloads/nfl_games.csv")

fit <- glm(score_diff ~ away_win_perc + home_win_perc, data=df)
summary(fit)
```

Wrap it in ŷhat

Now that we have a model we're going to use the ŷhat R Client to deploy it as a REST, streaming, and batch API. To do this, we're going to creat 3 functions: `model.require`, `model.transform`, and `model.predict`. You can think of these functions as your pipeline for processing data.

Step 1: `model.require` (optional)

Import any dependencies you might need for your model.

`Step 2:` model.transform `

Define how you will handle incoming request data. For instance, in our model that will be something like this:

```
{
    "home_win_perc": 0.83,
    "away_win_perc": 0.63
}
```

The value that is returned by the `model.transform` function will become the input to the `model.predict` function.

Step 3: `model.predict`

The `model.predict` function recieves the output from the `transform` function and then execute the neccessary code to make a prediction and formats our response. In this example, we're going to be invoking our `fit` model and then formatting the response into a `data.frame`.

```
library(yhatr)

model.require <- function() {
    # we have no dependencies
}

model.transform  <- function(df) {
    df
}

model.predict <- function(df) {
    df$pred_score_diff <- predict(fit, newdata=df, type="response")
        df
}
```

Create an Instance and Deploy

The hard part is over. All that's left is to log into your ŷhat server and deploy your model. Logging into the server is easy. Just create a `yhat.config` variable that stores your username, apikey, and server url.

```
yhat.config  <- c(
    username="sandipan.dey@gmail.com",
    apikey="46a2a7bd2be69c45ff55f315d13bd1c9",
    env="http://cloud.yhathq.com/"
)
```

To deploy the model, just give call the `yhat.deploy` function. Make sure to pass it a name for your model.

```
yhat. deploy ("nflPredictor")
```

Making a change

Let's say you want to make a change to the model we just deployed. Something fairly common might be tweaking the features in the regression. For our example, let's create an interaction term between `and` . smoother to help prevent overfitting.

```
fit <- glm(score_diff ~ away_win_perc * home_win_perc, data=df)
summary(fit)
```

Putting it all together

To incorporate your change into ŷhat, simply re-run the code from Step 2.

```
yhat. deploy ("nflPredictor")
```

Contact Us

info@yhathq.com
support@yhathq.com
(917) 719-5959

Our Products

Enterprise
Cloud
Get the ŷhat AMI »
Enterprise Signup »

Learn More

About
Blog
FAQ
Jobs
Terms of Service

Newsletter

Email Address

Get Updates

Connect With Us