SHORT TITLE

Create an assistant that predicts real estate prices

Create an assistant that predicts the price of your future house

LONG TITLE

Deploy an app in Kubernetes that controls the communication between your data science workbench and your chatbot and demonstrate your predictions

Deploy an app in Kubernetes that communicate with your data science workbench and demonstrate your findings in a customized chatbot

Get the code

Will be published in github then inserted link here

SUMMARY

In this code pattern, we will create an assistant that estimates real estate prices with a data science platform in the background and under the control of a containerized application in Kubernetes. The user will provide the features of a house that he or she wants and the result is calculated and displayed to user immediately. In this way, users can quickly obtain the price information of the house that they are seeking for without any research.

TECHNOLOGIES

* Artificial Intelligence
* Containers
* Conversation
* Data Science
* Machine Learning
* Microservices
* Object Storage
* Python

DESCRIPTION

Sometimes while applying Data Science into our everyday applications, we have insufficient resources to demonstrate our work. For example, when we want to share our research findings with our teammates in workplace, we may not have our laptop with us at that moment, or we may not access slides that contains our results with good visuals. Therefore, we may not be able to impress our colleagues and we feel frustrated. However, we forget the fact that we always have our mobile phones with ourselves and they can be the perfect tool for displaying our efforts in anywhere at any time. In this pattern, we will walk through how to show our hard work easily with IBM Watson Assistant, Watson Studio, Node-RED, Kubernetes Service and machine learning libraries such as pandas and scikit-learn.

This pattern is based on the dataset of House Sales in King County which includes Seattle, USA. It includes homes sold between May 2014 and May 2015. The reason behind choosing this dataset is that when we want to search for a new house for ourselves, we need to go to real estate agencies and the process of visiting each one of them require lots of effort and we may want to have a quick look at the approximate prices for our desired house.

After going through this pattern, you will be able to:

* Use Watson Studio
* Prepare a dataset from scratch with machine learning libraries
* Visualize your data
* Use IBM Watson Machine Learning Model Builder Service
* Understand and implement WebSocket communication protocol
* Deploy a Node-RED application in IBM Kubernetes Service
* Use Watson Assistant and integrate it into a Telegram Bot

This pattern created for data science enthusiasts who seeks for different ways of delivering and displaying results with ease. This pattern does not guarantee that the price will be exactly the same as predicted in the pattern but close to the real price offered by real estate agencies.

ARCHITECTURE DIAGRAM

A close up of a map

Description automatically generated

FLOW

1. Open Watson Studio and create a notebook.
2. Download the dataset into Watson Studio and prepare it for modeling.
3. Export the prepared dataset and give it to IBM Machine Learning Model Builder Service (create one) for choosing the regression algorithm.
4. Implement the regression algorithm in Watson Studio and calculate the accuracy.
5. Implement WebSocket client inside Watson Studio.
6. Create a cluster from IBM Kubernetes Service and install Node-RED service into it.
7. Create a Watson Assistant.
8. Create a Telegram Bot.
9. Create a Node-RED flow that connects Watson Assistant to Telegram Bot.
10. Add nodes to current Node-RED flow that implements WebSocket communication (both server and client) and connects Watson Assistant to Telegram Bot.
11. Calculate and display the estimated price in Telegram bot based on house features given by user.

INSTRUCTIONS

Find the detailed steps for this pattern in the README. This code pattern consists of three primary activities:

1. Run a Jupyter notebook in Watson Studio and use Watson Machine Learning.
2. Create IBM Kubernetes Service cluster and install Node-RED into it.
3. Create a Watson Assistant Chatbot and connect it to Telegram via Node-RED. Add necessary flows to Node-RED.

GITHUB README

Run a Jupyter notebook in Watson Studio and use Watson Machine Learning.

1. Login or Sign up for the Watson Studio
2. Create a new Watson Studio project

* Select the New Project option from the Watson Studio landing page and choose the Standard option.
* To create a project in Watson Studio, give the project a name and either create a new Cloud Object Storage service or select an existing one from your IBM Cloud account.
* Upon a successful project creation, you are taken to a dashboard view of your project. Take note of the Add to project button and 1001 tab on the top right, we'll be using them to associate our project with any external assets (datasets and notebooks) and any IBM cloud services.

1. Create the Notebook
   * From the project dashboard view, click the Add to project button, then click Notebook.
   * Give your notebook a name and select your runtime as Default Python 3.5 XS (2 vCPU and 8 GB RAM)
   * Select Language as Python 3.5
   * Now select the From URL tab to specify the URL to the notebook in this repository.
   * Enter this URL: !!!!!Need to be filled!!!!!
   * Click the Create button.
2. Upload data
   * Extract the zip file in this repo: !!!csv zip file needed here!!!

* Go to Jupyter Notebook that you created previously above, click 1001 tab on the top right and click Files tab and click on browse.
* Select the file that is extracted.
* You will see your csv file in the right sidebar click insert to code and choose Insert pandas DataFrame
* Make sure your kc\_house\_data.csv is saved as df (convert df\_data\_1 to df), so that it is consistent with my notebook and so you do not have to change the code.

1. Run the notebook

* On the top left corner below the project name, notice that there are tabs like File, Edit and so on.
* Click on Cell Tab and click Run All to execute the code.

1. Export the data from Notebook and give it to newly created IBM Watson Machine Learning Service.
   * Take a look at the cell with “In [13]:”. Click the Download CSV File and download the csv file into your computer.
   * Go to your project main mage by clicking your project name in the top left corner.
   * Click Settings Tab in top right. Go below to associated services. Click Add service and click Watson. Choose Watson Machine Learning by clicking Add.
   * Go to New Tab and Click round option next to Lite. Then Click create.
   * Go to Assests tab and notice the Models Dropdown menu. Click New Watson Machine Learning Model on the right of that menu.
   * Give your model a name of your choice. Then Select Model Type as model builder.
   * Select runtime as Default Spark Scala 2.11.
   * Go below and Click Manual.
   * Click Create.
   * You will be directed to Select data asset. Click Add Data Assests and import your previously exported data.
   * Choose the round option next to csv data and click next.
   * You will see Select a technique menu. Under Column value to predict (Label Col) select price. Leave Feature Columns as All(default).
   * Click Add Estimator on top right. Click on every regression type. And click next.
   * Wait for the trainings to be finished and take a look at the results. You will see the best option as Random Forest Regressor since it has less error and more r2 compared to other regression types.

Create IBM Kubernetes Service cluster and install Node-RED into it

* Go to <https://cloud.ibm.com/kubernetes/catalog/cluster/create>
* We need to create a free cluster. You just need to upgrade your account by giving your credit card information. There will be no charge. Click on Free.
* Choose Geography as North America.
* Leave resource group as default.
* Give your cluster a unique name and click create cluster button on the right.
* Cluster Main Page opens. Wait few minutes for the deployment of the cluster.
* After you observe that the cluster is at normal stage, follow the main page guidelines to set up your recently created Kubernetes Cluster.
  + 1. Log in to your IBM Cloud account.
  + ibmcloud login -a https://api.ng.bluemix.net
  + If you have a federated ID, use ibmcloud login --sso to get started.
  + 2. Target the Kubernetes Service region in which you want to work.
  + ibmcloud ks region-set <your region abbreviation>
  + 3. Get the command to set the environment variable and download the Kubernetes configuration files.
  + ibmcloud ks cluster-config <your cluster name>
  + 4. Set the KUBECONFIG environment variable. Copy the output from the previous command and paste it in your terminal. The command output should look similar to the following.
  + export KUBECONFIG=/Users/$USER/.bluemix/plugins/container-service/clusters/<your cluster name>/kube-config-dal10-<your cluster name>.yml
  + Alternatively, you can directly download your kubeconfig files to manually configure the cluster context.
  + 5. Verify that you can connect to your cluster by listing your worker nodes.
  + Open your terminal and enter following commands
  + kubectl get nodes
* kubectl run mynodered --image=nodered/node-red-docker
* kubectl expose deployment/mynodered --type=NodePort --port=1880 --name=mynodered-service --target-port=1880
* kubectl describe service mynodered
  + 1. Notice the NodePort section copy the five digit number next to TCP
* ibmcloud ks workers --cluster <your cluster name>
* Notice the IP address under Public IP Section. Copy the IP address.
* Go to nodered.org and see Run Locally section. Click Docker.
* Go to your cluster ip address and port that you copied earlier above.Paste the ipaddress/port into your browser. You will see your Node-RED is installed and running.
* Copy the ip address to your favourite text editor.

Create a Watson Assistant Chatbot and connect it to Telegram via Node-RED. Add necessary flows to Node-RED.

* Go to cloud.ibm.com and login to IBM Cloud.
* Click Catalog on top middle.
* Write assistant inside search bar.
* Click Watson Assistant.
* Give a name to Watson Assistant Service. Leave every other area as it is.
* Click create. You will be directed to Watson Assistant Service main page.
* Click Launch Tool.
* Click on Skills Tab above.
* Click Create New.
* Click Import Skill. Click Choose JSON File and choose the Watson assistant skill in the GitHub page. (assistant.json file)
* Click Skills again on top.
* Notice your Watson Assistant Skill and click the 3 dots next to your skill’s name. Click view API details.
* Copy the workspace Id in a text editor.
* Go to your Node-RED instance. Click the 3 lines top right. Go Import click clipboard.
* Choose Select a file to import. Put the nodered flow.json file from the github page.
* Click Deploy in top right.
* Click on the top-right menu and select Manage pallete.
* Click on the Install tab
* Enter telegram on the search module textbox
* Click install for node-red-contrib-telegrambot
* Then click Install again to confirm
* Once the node has been installed, click Done
  + Note: There will be 4 telegram nodes installed
* Double Click on assistant node. Paste your workspace id in the workspace id field (you copied that earlier). Note: You may need to fill other fields of your Watson assistant nodes, if your flow will give an error. Go to your Watson Assistant Service home page and copy paste the credentials that is given to the same spots in Watson assistant node in nodered.
* Double Click on websocket nodes (the ones that write “ws” on them)
* Choose Listen on and click pencil icon. Give a url path name to your websocket nodes. Copy the path to your favourite text editor.
* Go to your Jupyter Notebook and search for “WebSocketApp”. You will see "ws://YourIPaddressandport/YourPath" part. Replace the inside of quotes YourIPaddressandport/YourPath with your real ipaddress and port that you copied before and url path that you also copied before. So the inside of quotes should look like this: “ws://1.1.1.1:3000/test” , while 1.1.1.1:3000 corresponds to your ip address and “/test/” corresponds to your path.

Create TelegramBot and give credentials to Node-RED.

* Open Telegram app.
* Search for @BotFather at the search bar on top and select it.
* Send /newbot command message to BotFather
* Enter the name and username of your bot as in the guidance.
* Once created, you’ll be given a token string.
* Save the token on your favorite text editor
* Open the Bot that you created.
* Go to your Node-RED Flow.
* Click on telegram receiver. Click on pencil icon.
* Fill the Bot-Name with your TelegramBot name and token field with your previously copied token.
* Click Deploy.
* Go back to Jupyter Notebook and Click Run All from Cell again.
* Write “what is the cost of the house” to Telegram Bot. Fill the necessary fields and observe the result.
* Important Note: After some prediction value occurs, go to your node-red flow and click the square button next to Clear Context Inject Node. This will erase the previous prediction.