Improving AWS EC2 Spot Instance Price Prediction Accuracy using XGBoost - Configuration Manual

Research Project
MSc Cloud Computing

Ankit Dutta
Student ID: X20185502

School of Computing
National College of Ireland

Supervisor: Mr Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ankit Dutta
Student ID:	X20185502
Programme:	MSc Cloud Computing
Year:	2022
Module:	Research Project
Supervisor:	Mr Vikas Sahni
Submission Due Date:	15/08/2018
Project Title:	Improving AWS EC2 Spot Instance Price Prediction Accuracy using XGBoost - Configuration Manual
Word Count:	811
Page Count:	3

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Ankit Dutta
Date:	15th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Improving AWS EC2 Spot Instance Price Prediction Accuracy using XGBoost - Configuration Manual

Ankit Dutta
X20185502

1 Introduction

The main objective of this document is to help the readers to setup and replicate the process of implementing XGBoost Machine Learning model on Amazon Web Services (AWS) Spot Instance (SI) Pricing Data. This document complements the code artefacts.

2 Project Setup

This project is carried out using Python. So, the first step is to install Python. In this project, Python version 3.10.5 has been used1. Furthermore, the installation has been configured with 'pip' to install packages. Following this, the next step is to access the AWS SI price dataset.

2.1 2.1 AWS Spot Instance Price History Dataset

One of the first tasks in this project is to retrieve the AWS SI pricing data. This is public dataset and is available to users if they are signed up on AWS. There are multiple methods to implement this.

2.1.1 Retrieving SI pricing data using AWS console

Step 1: Open Elastic Compute Cloud (EC2) console[1].
Step 2: Select Spot Requests on the navigation pane.
Step 3: Select Pricing History.
Step 4: View Data which can be sorted by Instance Type and Availability Zone.

2.1.2 Retrieving SI pricing data using the command line

Step 1: Install AWS SDK[2] using 'pip install cli'.
Step 2: Setup AWS SDK with AWS User details (i.e., 'aws_access_key_id', 'aws_secret_access_key') using script 'aws configure'.
Step 3: Use the AWS Command Line Interface (CLI) function 'describe-spot-price-history' to retrieve prices grouped by Instance Type and Availability Zone.

[1] https://aws.amazon.com/console/

[2] https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/describe-spot-price-history.html

2.1.3 Retrieving SI pricing data using boto3

Step 1: Install boto3[3] using ‘pip install boto3’.
Step 2: In Python (Jupyter Notebook) file, import the ‘boto3’ (import boto3) and ‘pandas’ (import pandas as pd) packages. Then, setup the AWS credentials using the function, s3=boto3.resource(‘s3’, aws_access_key_id=”, aws_secret_access_key=”). Step 3: Finally, in the code artefacts there are two functions called the ‘handler’ and ‘wrapper’ function. The ‘wrapper’ takes user input like ‘InstanceList’, ‘ProductDescriptionList’, and ‘Region’. It then subsequently calls the ‘handler’ function which retrieved AWS SI pricing data for the past 90 days.

In this project ‘boto3’ and the AWS CLI SDK have been used to retrieve data. The SDK was setup to provide access to the AWS services with the right access permissions and ‘boto3’ is used to retrieve the data.

2.2 Machine Learning Libraries

Once the dataset has been retrieved, it is necessary to install the machine learning libraries. For this purpose, ‘sci-kit learn’ and ‘xgboost’ are being used. These have all the tools to setup training data, test data, model creation, model fitting, generating predictions and generating evaluation metrics.

Step 1: Install Sci-kit Learn Pedregosa et al. (2011) using command ‘pip install scikit-learn’.
Step 2: Install XGBoost[4] using command ‘pip install xgboost’.
Step 3: Import necessary libraries (list mentioned below).

List of other libraries:

- ‘from sklearn.model_selection import train_test_split’.
- ‘from sklearn.ensemble import RandomForestRegressor’.
- ‘import xgboost as xgb’.
- ‘from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error, mean_absolute_percentage_error’.
- ‘import datetime’
- ‘import numpy as np’Harris et al. (2020).

This completes the setup; the next section is a deep dive into the implementation.

3 Project Implementation

The project is split into two Jupyter notebooks, namely, ‘data_eval.ipynb’ and ’model_perf.ipynb’. The first one is for establishing whether the AWS SI price data is stationary or not, and the second one is for running the machine learning models on the AWS SI price dataset.

[3] https://boto3.amazonaws.com/v1/documentation/api/latest/index.html

[4] https://xgboost.readthedocs.io/en/stable/

3.1 'data_eval.ipynb'

Step 1: In the wrapper function, set an appropriate name to the .csv file to save and load the dataset.
Step 2: In the wrapper function call, set the 'InstanceType', 'ProductDescription', and 'Region'.
Step 3: Read the .csv file.
Step 4: Run all remaining cells in the Jupyter notebook.

Note: In the case of re-running, the wrapper() function calls can be skipped and the datasets can be read directly using the pd.read_csv() function using the appropriate dataset.

3.2 'model_perf.ipynb'

Step 1: In the wrapper function, set an appropriate name to the .csv file to save and load the dataset.
Step 2: In the wrapper function call, set the 'InstanceType', 'ProductDescription', and 'Region'.
Step 3: Read the .csv file.
Step 4: Run all remaining cells in the Jupyter notebook to model the five different regions with RFF and XGBoost.

Note: In the case of re-running, the wrapper() function calls can be skipped and the datasets can be read directly using the pd.read_csv() function using the appropriate dataset.

References

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. and Oliphant, T. E. (2020). Array programming with NumPy, *Nature* **585**(7825): 357–362.
URL: *https://doi.org/10.1038/s41586-020-2649-2*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830.