PREDICTION OF DYNAMIC CLOUD RESOURCES PROVISIONING FOR WORKFLOWS

Yashkumar Jain
Department of CSE
Vivekanand education society
Institute of technology
Mumbai, India
yashkumarta@gmail.com

Tumpala Kesava Durga Prasad
Department of ECE
Aditya Engineering College
Surampalem,India
kesavadp0121@gmail.com

Anuj Sharma
Department of CSE
Techno India NJR
Of Technology
Udaipur, India
17etccs002@technonjr.org

Bhumika Agrawal
Department of CSE
Poornima institute of engineering
and technology
Jaipur, India
bhumikaagrawalratlai@gmail.com

Dr. Indrajeet Gupta
Department of CSE
Bennett University
Greater Noida, India
indrajeet7830@gmail.com

ABSTRACT :-This is a project in which we have to predict the resources used by any virtual machine provisioning for workflow. The prediction is affected by many factors including CPU core, CPU capacity, CPU usage, Memory capacity, Memory usage and Network throughput. We have created a Model which predicts a good accuracy score. We have used MLP Classifier, Gradient Boosting Classifier, AdaBoost Classifier, Bagging Classifier and Random Forest Classifier.

Index Term :- machine Learning Algorithm, Data Preprocessing, MLP Classifier, Bagging Classifier, AdaBoost Classifier, GradientBoost Classifier and Random Forest Classifier.

I. INTRODUCTION

Cloud computing is a new trend in Internet computing where resources like storage, computation power, network, applications etc. are delivered as a service. The IaaS delivery model provides storage, hardware and networks as the basic subcomponents of services. It offers on-demand creation of virtual machines (VMs) for various users and applications, which enables dynamic management of VMs for maximizing the resource utilization in the data centers (DCs).
With the increase in cloud computing the load on the cloud has also increased. Every company is using the cloud for many purposes. So the resources available on the cloud for so many companies needs to be predicted otherwise the company will go in loss. In our machine learning model we have used some methods like ensemble and neural network method to predict the

resources available for a company specially on the working days because on working day the resources required is very high and on the weekends its very low.

1.1] Data Preprocessing :- The data provided to us is not standardized so we have to first preprocess the data using the module sklearn.preprocessing. In that module we used the StandardScaler method to process our data. StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation.

1.2] Separating the Column :- The first thing we had to do after importing the dataset was to separate the data in different columns as the data was separated by **';'** so we used a delimiter function to divide the data in separate columns.

II. RELATED WORK

A. PROBLEM STATEMENT :-

From the topic we have clearly understood that we have to predict the resources required by the companies for their work on cloud. We have to check how much resources are available for a particular company when the usage of these resources are at pick level like CPU usage and memory storage.

B. DATASET DESCRIPTION :-

We are using the dataset provided by the GWA. In that dataset there are many features like :-

Table 1: Schema of the GWA-T-12 Bitbrains fastStorage dataset

Schema		
Index	Name	Description
0	Timestamp	Number of milliseconds since start of the trace
1	CPU cores	Number of virtual CPU cores provisioned
2	CPU capacity provisioned (MHZ)	The capacity of the CPUs in terms of MHZ
3	CPU usage (MHZ)	Utilization of the CPU in terms of MHZ
4	CPU usage (%)	Utilization of the CPU in terms of percentage
5	Memory provisioned (KB)	The capacity of the memory of the VM in terms of KB
6	Memory usage (KB)	The memory that is actively used in terms of KB
7	Disk read (KB/s)	Disk read throughput in terms of KB/s
8	Disk write (KB/s)	Disk write throughput in terms of KB/s
9	Network in (KB/s)	Network received throughput in terms of KB/s
10	Network out (KB/s)	Network transmitted throughput in terms of KB/s

From the features it is clear that it is structured data and the network columns are the dependent variables and rest all the other columns are the independent variables.

C. PRELIMINARY DATA ANALYSIS :-

The data was huge so we first divided the dataset into dependent and independent variables. Then we split them into training sets and test sets. We have used the train_test_split method of the model_selection module to divide the dataset. Finally we have used training sets and test sets in different modules as discussed below for prediction.

D. SOFTWARE/LIBRARIES USED :-

Jupyter Notebook :- JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.

sklearn : -Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

numpy :- NumPy is a python library used for working with arrays.
It also has functions for working in domain of linear algebra, fourier transform, and matrices.

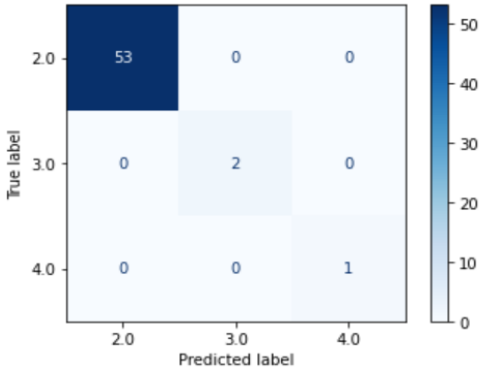
pandas :- pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language.

matplotlib :- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits.

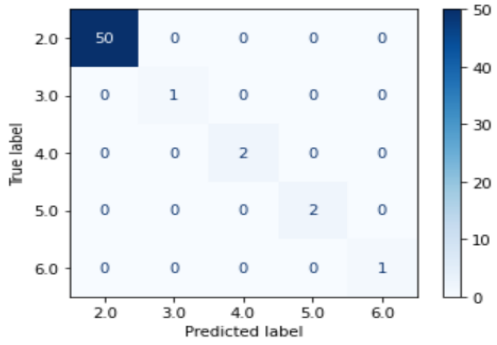
III. METHODOLOGY

While creating the model for prediction we used many methods. Some of the methods we used which has giving an accuracy score above 80% are :-

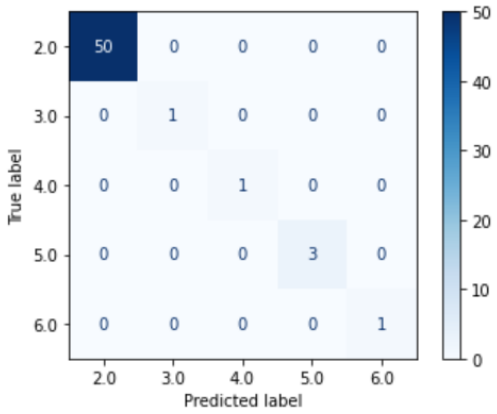
2.1] Bagging Classifier :- Bagging (Bootstrap Aggregating) is a widely used ensemble learning algorithm in machine learning. The algorithm builds multiple models from randomly taken subsets of the train dataset and aggregates learners to build an overall stronger learner. In this post, we'll learn how to classify data with the BaggingClassifier class of a sklearn library in Python. The confusion matrix for this method is:-



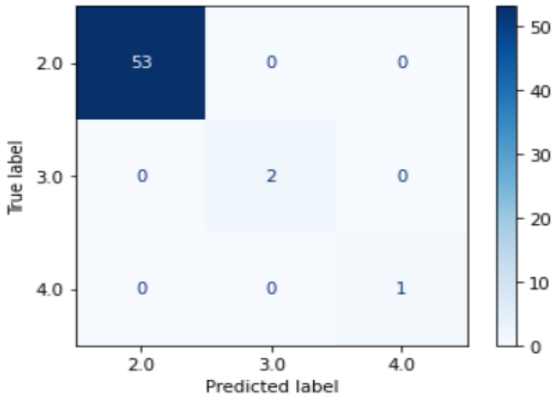
2.2] AdaBoost Classifier :- Ada-boost or Adaptive Boosting is one of the ensemble boosting classifiers proposed by Yoav Freund and Robert Schapire in 1996. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get a high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. The confusion matrix for this method is:-



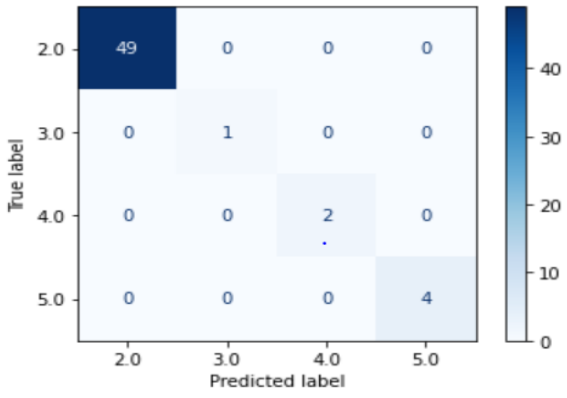
2.3] Gradient Boosting Classifier :- Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. Gradient boosting models are becoming popular because of their effectiveness at classifying complex datasets. The confusion matrix for this method is:-



2.4] Random Forest Classifier :- Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.Basic parameters to Random Forest Classifier can be the total number of trees to be generated and decision tree related parameters like minimum split, split criteria etc. The confusion matrix for this method is:-



2.5] MLP Classifier :- A multilayer perceptron (MLP) is a class of feedforward artificial neural networks (ANN). MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable. The confusion matrix for this method is:-

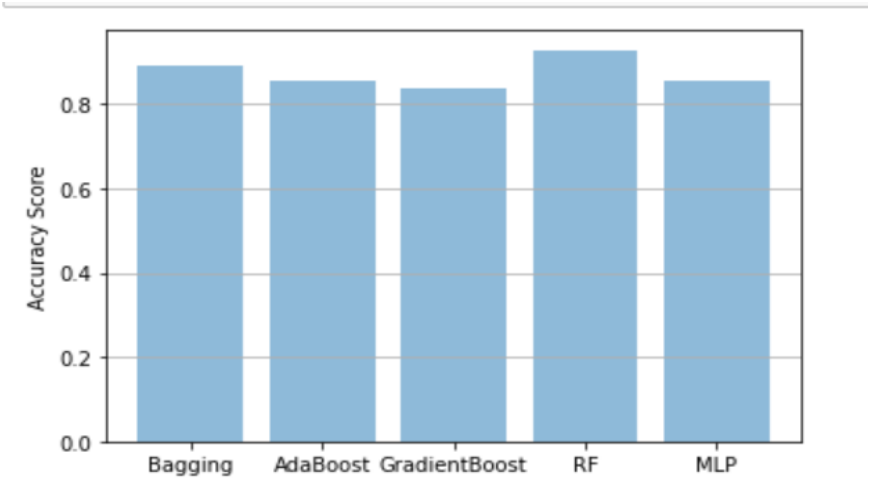


The results we got from the above methods are:-

MODULE NAME	METHODS	ACCURACY SCORE
sklearn.ensemble	Bagging Classifier	0.8928571428571429
sklearn.ensemble	AdaBoost Classifier	0.8571428571428571
sklearn.ensemble	Gradient Boosting Classifier	0.8392857142857143
sklearn.ensemble	Random Forest Classifier	0.9285714285714286
sklearn.nerual_network	MLP Classifier	0.8571428571428571

IV. EXPERIMENTAL RESULTS

From the above table we can clearly see the accuracy score of our model for different modules. It is clearly noted that Gradient Boosting Classifier method gives the least accuracy score of 0.83928. While the Random Forest Classifier method gives the highest accuracy score of 0.92857.



So from the above graph it is clear that Random Forest Classifier gives the best result as compared to the rest of the classifier methods.

V. CONCLUSION

In this paper, the dataset provided by the GWA github was preprocessed and then used in the model. Many modules were used on the dataset (Bagging Classifier, AdaBoost Classifier,

Gradient Boosting Classifier, Random Forest Classifier and MLP Classifier) to get the highest accuracy score or accuracy percentage of the prediction. We have clearly noted that all the methods are giving accuray scores above 80% but the random forest method give's accuracy score of 92.857%. Therefore we recommend a random forest classifier method for this problem statement.

VI. REFERENCE

- https://github.com/kwananth/VMWorkloadPredictor
- Gopal Kirshna Shyam and Sunilkumar S. Manvi, Virtual Resource Prediction in Cloud Environment: A Bayesian Approach, Journal of Network and Co
- https://scikit-learn.org/stable/modules/classes.html
- https://github.com/Azure/AzurePublicDataset