Comparison of Classification Algorithms using AWS SageMaker

Domain Background

Machine Learning (ML), a branch of Artificial Intelligence (AI), learns from the data using various algorithms and is a self-improving process in terms of performance as make adjustments during the learning process [1]. Supervised learning, unsupervised learning and reinforcement learning are the three main types of ML [2]. ML which is now a data driven approach was mainly knowledge-driven approach before 1990's. Arthur Samuel coined the term 'Machine Learning' in 1952, the Perceptron was introduced in 1957, Nearest Neighbor Algorithm [3] evolved in 1967, concept of multilayers also developed in 1960s and so on [4].

In today's world ML is a lifeblood of every business. ML which itself is a domain has been successfully applied to various domains like healthcare and life sciences, travel and hospitality, financial services, retail, manufacturing, energy etc. Thus, any study related to ML domain can be helpful to multiple domains.

Problem Statement

Every year hundreds of machine learning approaches are developed across different fields with the pursuit of performing better than the existing ones. And someone who wants to implement a developed ML has a multiple choice. It takes to time to decide which ML approach is best for a particular case as one must go through the multiple available choices. In most cases, the process of choosing the right algorithm must pass through the process of writing multiple lines of codes as well. So, the main problem we are trying to focus with this project is "Which state of art ML technology can perform better, when applied to a particular domain and is there way to compare the performance of state of art CF algorithms in a reasonable time with few lines of code?"

Datasets and Inputs

The problem considered for this study is practically applicable to every dataset available in the world. Here, we have chosen a dataset related to the healthcare. The dataset is Pima Indian Diabetes Dataset which is frequently used to evaluate the performance of developed ML techniques [5]. We downloaded the dataset from [5]. This data set has 8 attributes and one class variable. Class variable has possible value of 0 or 1, 1 being interpreted as test positive for diabetes.

Here, we will be applying classification algorithms from the scikit-learn library [6] and AutoGluon library [7]. Scikit-learn is the most successful and robust library for machine learning in Python. This library is primarily written in Python is based on the modules like NumPy [8], SciPy [9] and Matplotlib [10]. And the open source AutoML library AutoGluon-Tabular can train highly accurate different machine learning models with a single line of code [7].

The ML algorithms from scikit-learn library and Auto-Gluon library are implemented with AWS SageMaker [11]. The Amazon SageMaker is capable of building, training, and deploying state of art Machine Learning models with full managed infrastructure tools and workflows[12].

Solution Statement

The solution to find best classifier algorithm for a particular type of data set of a specific domain requires significant amount research on it. We will initiate this approach by applying multiple classifier algorithms to the health care data considered in this study. The problem of finding the best state of art ML algorithm with few lines of code in a reasonable time can be achieved with Auto Gluon-Tabular Library in SageMaker.

Evaluation Metrics and Benchmark Model

Classification ML Algorithms are evaluated using Classification Accuracy Measures like Accuracy, Precision, Recall and F1-score. For the problem considered during this project, let us consider a value of 1 class (having diabetes) be positive and value of 0 in class variable be negative. Let True Positive (TP) be the correctly classified number of positive classes from a CF algorithm. Similarly False Positive (FP) be the number of incorrectly classified as positive classes, True Negative (TN) be the correctly classified number of negative classes and False Negative (FN) be the number of classes incorrectly classified as Negative classes. Various classification accuracy measures are computed based on TP, FP, TN and, FN [13]. The Accuracy can be computed as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{1}$$

the Precision as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

the Recall as

$$\text{Recall} = \frac{\text{Tp}}{\text{TP} + \text{FN}} \tag{3}$$

and the F1-Score as

$$\text{F1} - \text{Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precison} + \text{Recall}} \tag{4}$$

Classification accuracy measures in Eq. (1), (2), (3) and (4) have been used to evaluate the performance of applied classifier algorithms.

The author in [5] mentions that the baseline classification accuracy is about 65 percent when Pima Indian Diabetes Dataset is used for the classification problem and the best accuracy obtained so far is about 77 percent. We will compare our results with these benchmark results.

Evaluation Metrics and Benchmark Model

We will start the project by downloading the Pima Indian Diabetes Dataset [5]. Then, we will create a notebook instance in Amazon SageMaker. When the notebook instance is successfully created, we will open the Jupyter Notebook and upload the downloaded dataset to the instance. We will make sure to use the with most cost-effective instance as our data set is not significantly large. After that we will, import the and download (if needed) the required module and libraries. Next step is to perform Exploratory Data Analysis (EDA). Then, we will apply and evaluate different state of art CF algorithms using scikit-learn and AutoGluon libraries. Finally, we discuss the results obtained and summarize the findings of the project.

Reference

[1] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[2] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Comput. Mater. 2019 51*, vol. 5, no. 1, pp. 1–36, Aug. 2019, doi: 10.1038/s41524-019-0221-0.

[3] T. M. Cover, "This Week's Citation Classic," *IEEE Trans. Inform. Theory IT*, vol. 13, pp. 21–28, 1967.

[4] Keith D. Foote, "A Brief History of Machine Learning - DATAVERSITY." https://www.dataversity.net/a-brief-history-of-machine-learning/ (accessed Jan. 12, 2022).

[5] J. Brownlee, "10 Standard Datasets for Practicing Applied Machine Learning." https://machinelearningmastery.com/standard-machine-learning-datasets/ (accessed Jan. 12, 2022).

[6] F. Pedregosa FABIANPEDREGOSA *et al.*, "Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011, Accessed: Jan. 12, 2022. [Online]. Available: http://scikit-learn.sourceforge.net.

[7] N. Erickson *et al.*, "AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data," Mar. 2020, Accessed: Jan. 12, 2022. [Online]. Available: https://arxiv.org/abs/2003.06505v1.

[8] C. R. Harris *et al.*, "Array programming with NumPy," *Nat. 2020 5857825*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[9] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, doi: 10.1038/S41592-019-0686-2.

[10] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. \& Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[11] SageMaker, "Amazon SageMaker – Machine Learning – Amazon Web Services." https://aws.amazon.com/sagemaker/ (accessed Jan. 13, 2022).

[12] AWS, "Amazon SageMaker Amazon Sagemaker API Reference Amazon SageMaker: Amazon Sagemaker API Reference."

[13] P. Galdi and R. Tagliaferri, "Data Mining: Accuracy and Error Measures for Classification and Prediction Neonatal MRI View project Computational methods for omics data View project Data Mining: Accuracy and Error Measures for Classification and Prediction," doi: 10.1016/B978-0-12-809633-8.20474-3.