ML1819 Research Assignment 2

Team 02
Task 101

Rajesh Burla - 18306485
Akashdeep Singh Lamba - 18305063
Shanmukha Sai Ram Pavan - 18305688

Each student picked up one library and implemented the three selected algorithms end-to-end in the assigned library, while constantly communicating progress over a Slack channel. We each maintained shared git repos to share code and collectively work. All three of us then compared our results and then discussed the contents of this document. We realized we were supposed to work in a single repo so we migrated our code to a shared repo in a group with write access to only the three of us. This also means the contributor graph isn't a perfect reflection of our activity.

Word count: ####

https://gitlab.scss.tcd.ie/ML1819-team-02/ML1819--task-101--team-02

https://gitlab.scss.tcd.ie/ML1819-team-02/ML1819--task-101--team-02/graphs/master

###screenshot###

Survey of popular Machine Learning Libraries

A comparative analysis of Tensorflow, Sklearn and Weka by implementation of standard algorithms

Rajesh Burla
MSc Computer Science - Intelligent Systems (2018-19)
Trinity College Dublin
Dublin, Ireland
burlar@tcd.ie

Akashdeep Singh Lamba
MSc Computer Science - Intelligent Systems (2018-19)
Trinity College Dublin
Dublin, Ireland
lambaa@tcd.ie

Shanmukha Sai Ram Pavan
MSc Computer Science - Intelligent Systems (2018-19)
Trinity College Dublin
Dublin, Ireland
parvarths@tcd.ie

ABSTRACT

This paper briefly compares 3 popular machine learning libraries by analysing implementation of linear regression, SVM classification and kNN algorithms.

1 INTRODUCTION

Due to growing interest in machine learning, several open-source libraries have become popular for different reasons, each preferred over the other for either ease-of-use, speed, accuracy or other features like GPU support. It's worth asking whether all libraries perform similarly under approximately same conditions or not, and this paper attempts to address this question.

2 RELATED WORK

Analysing the experiments of Bhuvan M Shashidhara et al. in [1] the results shows that Scikit-Learn is best fit for data in comparison with Weka and Apache Spark frameworks.

Scrutinizing results of the Google Brain team [2] reveals that TensorFlow is a flexible dataflow representation that enables power users to achieve excellent performance and scalability.

3 METHODOLOGY

In order to answer the research question, three Machine learning algorithms and frameworks were chosen to be applied to a pre-processed dataset and resultant metrics such RMSE and accuracy were compared.

3.1 Dataset

The dataset used was Google Play Store Apps: web-scraped data of 10k Play Store apps. This dataset contains all the details of the applications on Google Play. There are 13 features that describe an individual app. [3]

3.2 Data Pre-processing

The dataset in the original form obtained from Kaggle wasn't fit for direct use. In order to make it suitable for the experiment, sparse columns were eliminated. Further, string columns *Category* and *Genres* were encoded as numbers.

We also added a new column called *Rated 4.4 or more* derived from the column *Rating* for use with SVM. *Rated 4.4 or more* was defined as 1 if the *Rating* column is equal to or greater than 4.4, and -1 otherwise. The resultant classes were nearly equally distributed.

Scaling was left to implementation specific code since different implementations treat data differently.

3.3 Deployment on different platforms

i) Weka: Weka is a Java based data mining and machine learning suite. It contains robust sequential implementations of many machine learning algorithms, and ships with an easy to use GUI.

ii) Scikit-Learn: Also known as sklearn, this is a Python based library that implements a wide range of algorithms. Due to it's in-memory processing, it's very fast for smaller data which can be easily loaded into memory.

iii) TensorFlow: TensorFlow is a flexible Python framework for building fast and complex machine learning models specifically targeted for deep learning and neural networks. TensorFlow receives data in the form of Tensors, which are arrays of dimensions and ranks. It supports distributed execution over GPUs and CPUs.

The selection of these three implementations was done based of popularity as reported by [4] and [5]. Including Weka helped us make sure our study was not limited to Python-based implementations. While this isn't an exhaustive list of implementations, it's a good starting point.

3.4 Machine learning Algorithms

Three machine learning algorithms were chosen: Linear Regression, Support Vector Machines (SVM) and k-Nearest Neighbour (kNN) to test Root Mean Square Error (RMSE) and Accuracy on the pre-processed dataset using the selected frameworks.

For Linear Regression, the feature *Reviews* was used to predict *Rating*.

For SVM, *Reviews* was used to predict *Rated 4.4 or more*.

For kNN, *Reviews*, *Size*, *Genres* was used to predict *Category*.

Feature selection was done by manually scanning the dataset and reasoning about which features are well-suited for the given model and features that were skewed or sparse were dropped.

3.5 Cross validation

While Weka provides an in-built option to cross-validation, thereby making splitting of data and configuration of CV parameters internal, sklearn requires to explicit configuration of kFold validation with a split dataset. TensorFlow does not directly deal with any of these aspects. For TensorFlow and sklearn, we used sklearn's kFold validation with default parameters while using a random-indexed splitting between train data and test data, keeping a ratio of 7:3.

4 RESULTS & DISCUSSION

The results of our tests, represented by Fig. 1, indicate that using default parameters, the chosen libraries implement the chosen algorithms more or less at par with each other.

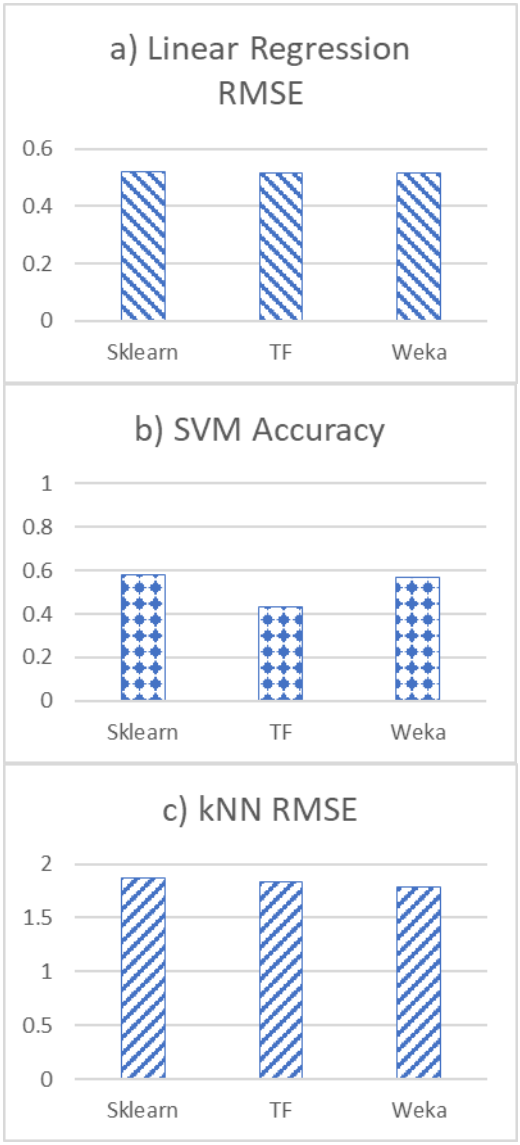

Figure 1: Test results

For linear regression, as seen in Fig. 1a, RMSE (Root mean squared error) was about 0.51 for both TensorFlow and Weka while only marginally higher at about 0.52 for sklearn. The lower the RMSE the better.

In case of SVM, we chose accuracy as the metric to report given that SVM results are binary and comparing MSE or RSME doesn't make sense. Here, accuracy is defined as the number of correctly predicted outcomes over the total number of predictions. Fig. 1b shows that accuracy stood at about 0.57 for sklearn and Weka while TensorFlow performed relatively poorly at 0.43. Accuracy is better when higher. This is probably due to the fact that TensorFlow is targeted towards Neural Networks with large datasets, and the low accuracy here is due to no optimization and the defaults.

Finally, for kNN, we again chose RMSE to represent performance and from Fig. 1c, it's apparent that there isn't much difference as RMSE was about 1.8 for sklearn and TensorFlow, while a tad lower at 1.78 for Weka.

The overall low difference in results seems to suggest that the three implementations are more or less consistent with minor differences. It may be noted that the difference in the level of abstraction also effects how the tests are carried out. To test Weka, the algorithms were implemented in Java code and also verified the results with Weka's own GUI, which hides implementation details. Sklearn exposes an easy API through which we can configure the model and execute it on our data. TensorFlow allows much more flexibility as it provides building blocks for defining the model and executing it on the given primitives.

5 LIMITATIONS & OUTLOOK

While the experiment conducted within the prevailing constraints provides an overview of comparison between the three libraries, it's by no means a comprehensive analysis. Metrics such as training speed haven't been considered, and more sophisticated algorithms haven't been tested. Further, frameworks like Apache Spark's MLib [7] and Orange [8] should be part of such a study of machine learning implementations, and future work on this project will focus on addressing these shortcomings.

ACKNOWLEDGMENTS

This analysis was conducted as part of the 2018/19 Machine Learning module CS7CS4/CS4404 at Trinity College Dublin [6].

REFERENCES

[1] Bhuvan M Shashidhara et al. "Evaluation of Machine Learning Frameworks on Bank Marketing and Higgs Datasets" IEEE 2015 DOI: 10.1109/ICACCE.2015.31

[2] The team Google Brain's "TensorFlow: A system for large-scale machine learning" 12th USENIX Symposium on Operating Systems Design and Implementation in 2016.

[3] Lavanya Gupta, "Google Play Store Apps," [Online] https://www.kaggle.com/lava18/google-play-store-apps

[4] Datanyze, "Machine Learning Market Share Report | Competitor Analysis | TensorFlow, scikit-learn, MLlib," [Online] https://www.datanyze.com/market-share/machine-learning

[5] Joeran Beel, "Experience with and Preference of Machine-Learning Libraries: scikit-learn vs. Tensorflow vs. Weka ... [What Machine-Learning Students Think/Like/Know/Are...]," [Online] https://www.scss.tcd.ie/joeran.beel/blog/2018/01/28/what-machine-learning-students-think-like-know-are-experience-with-and-preference-for-machine-learning-libraries-scikit-learn-vs-tensorflow-vs-weka/

[6] Joeran Beel and Douglas Leith. Machine Learning (CS7CS4/CS4404). Trinity College Dublin, School of Computer Science and Statistics. 2018.

[7] Xiangrui Meng et al. MLlib: Machine Learning in Apache Spark, Journal of Machine Learning Research 17 (2016) 1-7

[8] Demsar J, Curk et al. Orange: Data Mining Toolbox in Python. Journal of Machine Learning Research 14(Aug):2349–2353., https://orange.biolab.si/citation/