**ML1819 Research Assignment 1**

1. **Team ID**

Team - 06

1. **Task ID and Title**

Task - 101 “How consistent are the implementations of machine learning algorithms in different ML libraries?”

1. **Student Names and IDs**

Abhishek Jain Jeff Blackburn Siddharth Dhillon

18306320 12305379 18300735

1. **Contributions:**

* Abhishek Jain
* Manage GitHub repository
* Pre-processing of data set
* Implement linear regression, logistic regression and K-nearest neighbour algorithms with TensorFlow library
* Write Results & Discussion and Limitations & Outlook sections
* Jeff Blackburn
* Implement linear regression, logistic regression and K-nearest neighbour algorithms with Sci-Kit Learn library
* Implement linear regression, logistic regression and K-nearest neighbour algorithms with Pytorch library
* Run all code on personal machine for comparison of results
* Create all relevant figures
* Write Methodology and Abstract sections
* Siddharth Dhillon
* Write Introduction and Background sections

1. **Word Count**

Introduction – 105 Background Reading – 104 Methodology -288 Results & Discussion Limitations & Outlook Total

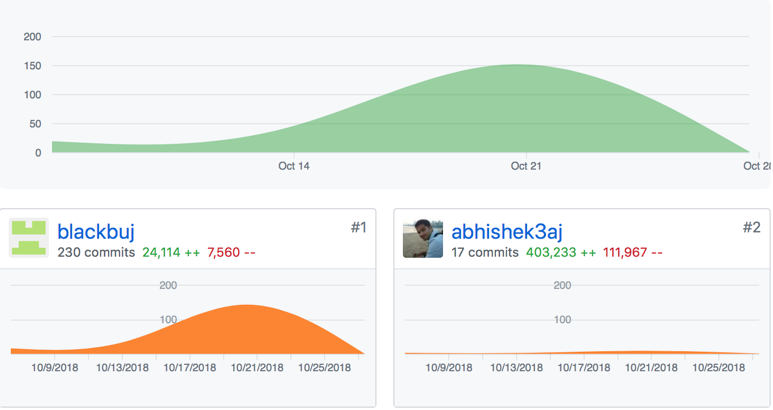
1. **URL to Source Code Repository**

[**https://github.com/abhishek3aj/ML1819--task-101--team-06**](https://github.com/abhishek3aj/ML1819--task-101--team-06)

1. **URL to source code repository activity**

[**https://github.com/abhishek3aj/ML1819--task-101--team-06/graphs/contributors**](https://github.com/abhishek3aj/ML1819--task-101--team-06/graphs/contributors)

1. **Screenshot showing the commit activity of all student**

A Comparative Study of Open Source Machine Learning Frameworks

|  |  |  |
| --- | --- | --- |
| Jeff Blackburn School of Computer Science and Statistics Trinity College Dublin Ireland blackbuj@tcd.ie | Abhishek Jain School of Computer Science and Statistics Trinity College Dublin Ireland jainab@tcd.ie | Siddharth Dillon School of Computer Science and Statistics Trinity College Dublin Ireland dhillons@tcd.ie |

# Introduction

Machine learning utilizes man-made logic to gain insights from data. With increasing amounts of data available, the benefit that machine learning creates for organisations has grown hugely. [1] Many machine learning tools are open source. This can both encourage innovation and allow for faster problem solving and troubleshooting of issues. [2] This is a comparative study of open source frameworks, focusing on three of the most popular tools in 2018; TensorFlow, Sci-Kit Learn and PyTorch. [3] The performance of these tools will be compared for three commonly used algorithms; Linear Regression, Logistic Regression and K-Nearest Neighbours Classification (KNN). The end goal is to make a recommendation for users.

# Related Work

There have been many studies in the past few years comparing the performance of different machine learning frameworks. These studies have focused on a variety of different frameworks including TensorFlow, Theano, CNTK, Keras and Ski-Kit Learn. Studies often focused on run time as a key measure of performance. [4-6] CPU and GPU utilization was another measure of performance used in studies. [5, 6] Across this scientific research there has not been a conclusive answer for an “optimal” framework. There is also a large amount of unpublished research available. PyTorch and TensorFlow are often recommended for their ease of use and large amount of community support. [7]

# Methodology

## Dataset and Pre-Processing

The dataset “Weather in Szeged 2006-2016” is publicly available on Kaggle. It contains 96453 entries with the following information.

* Formatted Date
* Summary
* Precip Type
* Temperature (C)
* Apparent Temperature (C)
* Humidity
* Wind Speed (km/h)
* Wind Bearing (degrees)
* Visibility (km)
* Loud Cover
* Pressure (millibars)
* Daily Summary

Three random samples of the dataset were taken of size – 15582 (small), 25014 (medium) and 55857 (large).

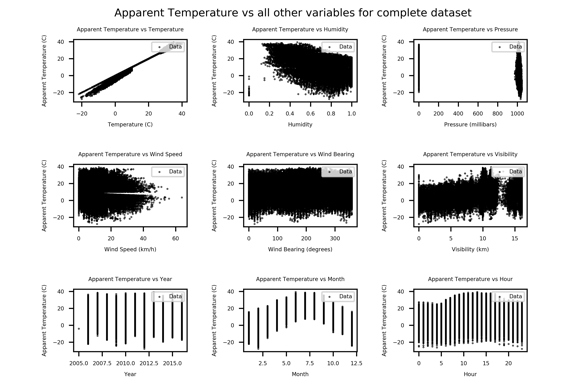
“Formatted Date” – was split into 7 new columns “Year”, “Month”, “Month\_Sin”, “Month\_Cos”, “Hour”, “Hour\_Sin” and “Hour\_Cos”. The sin and cosine of month and hour were calculated to account for the cyclical nature of the variables. A “Heavy\_Cloud” column was created denoting the presence of heavy cloud or not for logistic regression. A “New \_Summary” column was created reducing the number of categories in “Summary” to three for KNN. Continuous variables were standardised for logistic regression and KNN.

## Machine Learning Algorithms

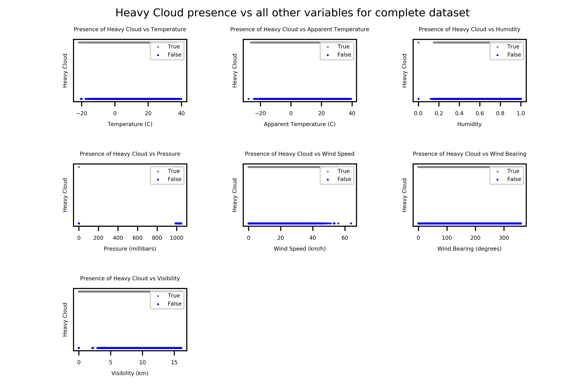
*3.2.1 Linear Regression.* Based on exploratory plots performed (Figure 1), Temperature and Apparent Temperature could be seen to have a clear relationship.

*3.2.2 Logistic Regression*. Based on exploratory plots (Figures 2-5) it was not clear which variables influenced “Heavy\_Cloud”. Using the rfe\_model in Sci-Kit learn the most influential variables were found to be Temperature, Apparent Temperature, Humidity, Wind Speed, Month\_Cos and Visibilty.

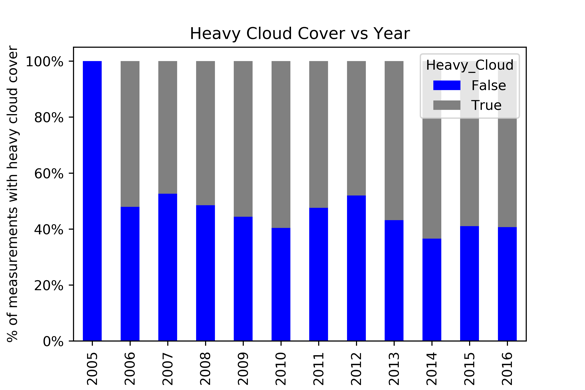
*3.2.2 KNN.* Based on exploratory plots (Figures 6-9) it was not clear what variables influenced “New\_Summary”. KNN was performed for each individual variable and the most influential were found to be Temperature, Apparent Temperature, Humidity, Month\_Sin, Month\_Cos. These were combined for training the models.



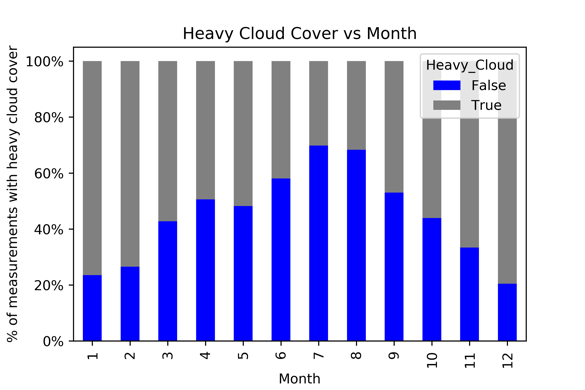
**Figure 1: Individual plots of Apparent Temperature vs all other variables**



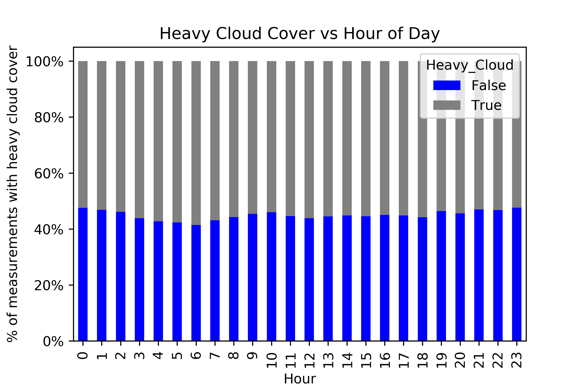
**Figure 2: Individual plots of Heavy Cloud vs continuous variables**



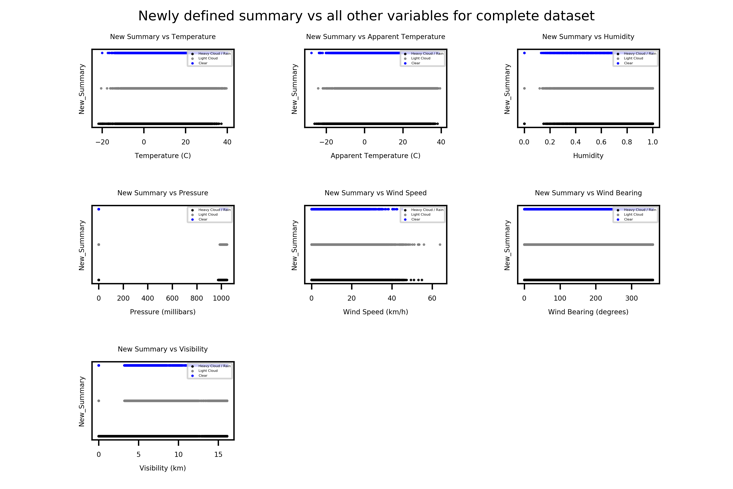
**Figure 3: Heavy Cloud vs Year**



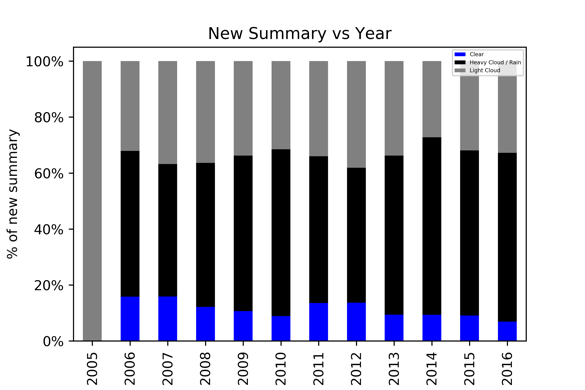
**Figure 4: Heavy Cloud vs Month**



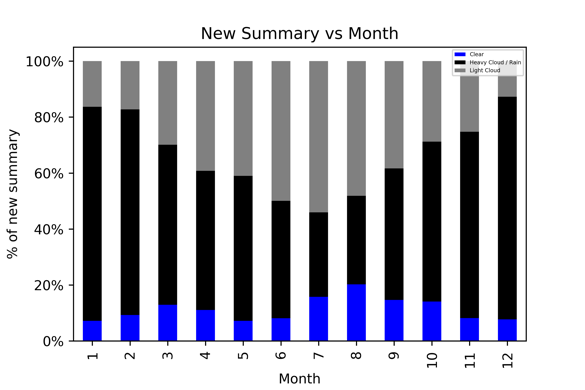
**Figure 5: Heavy Cloud vs Hour of Day**



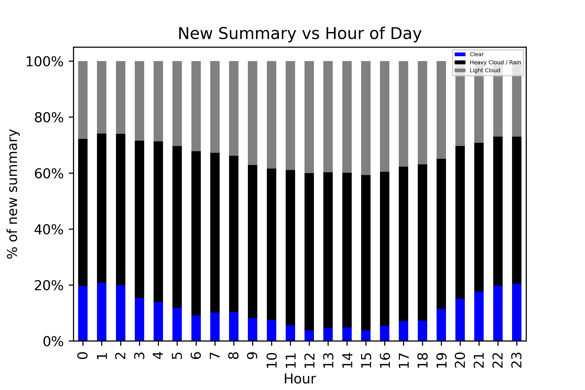
**Figure 6: Individual plots of New Summary vs continuous variables**



**Figure 7: New Summary vs Year**



**Figure 8: New Summary vs Month**



**Figure 9: New Summary vs Hour of Day**

## Evaluation

Data was randomly split into three sections training (70%), validation (20%) and test (10%). All models were evaluated on training time and prediction accuracy of validation data for small, medium and large datasets. Additional methods of evaluation were used depending on model type. (See Results & Discussion)

# Results & DiscussioN

For comparing results across different frameworks and different size of dataset we have used evaluation metrics like MAE, RMSE, MSE for Linear regression. Accuracy, confusion metrics and AUC for Logistic Regression and KNN.

Implementation of linear regression produced similar results in different metrics, but there was a significant difference in training time as Sci-Kit Learn was able to train linear regression model  in 0.07 seconds, tensorflow took 3 seconds whereas PyTourch completed training in 11 seconds for our medium size dataset. Tensorflow and Sci-Kit learn training time was almost same for different size of dataset whereas PyTourch training time increased when size of data increased.

For Logistic Regression model we used evaluation  metrics like accuracy, Error rate, FNR, FPR, AUC and Recall/TPR. Logistic regression implementation produced similar results in all frameworks. For this algorithm scikit learn was faster when compared to Tensorflow and PyTourch. Both Tensorflow and Sci-Kit Learn maintained almost constant training time accross different size of datasets

For KNN comparison we used evaluation metrics like Error rate, AUC and confusion matrix. PyTourch offered slightly better accuracy of 0.6 compared to scikit learn which offered 0.5 for our small dataset, but when compared with medium and larger dataset it was similar for both framework. AUC and confusion metric produced similar results for scikit pyTourch and tensorflow, Training time was similar when using Sci-Kit Learn across dataset of different size.Tensorflow took 2.12 and 3.96 seconds for small and medium dataset respectively but when used for training large dataset it took 17 Seconds whereas PyTourch took 62 Seconds. Detailed comparision data can be seen in *Results Summary.xlsx in git repository.*

In conclusion, Implementing Linear Regression, Logistic Regression and KNN in Sci-Kit Learn was easy and quick as it provide built in Methods, whereas Tensorflow and PyTourch also provide built in estimator/ functions but they work with Tensors which required us to convert data from data frames or numpy array to tensors. Advantage tensor offere over numpay arrays or dataframe is that both Tensorflow and PyTourch tensor can utilize GPU to accelarate numeric computations.

# Limitations & Outlook

Sci-Kit Learn was able to perform better in term of training time and accuracy than tensorflow and Pytourch in our implementation of Linear regression, Logistic Regression and KNN. Altought further comparison can be carried out in implementation of neural networks using tensorflow and PyTourch. In terms of GPU support Tensorflow and PyTourch provide out of the box support for GPU while Sci-kit Learn do not provide support for GPU [8]. Also Tensorflow provide specialized library for web development - Tensorflow.JS and Tensorflow Lite for mobile and embedded devices. both tensorflow and pytourch provide advance features like Distributed training which can provide significant training boost when dealing with giant datasets and models. We can further compare performance of these frameworks in distributed mode and using GPUs.

Many other popular frameworks like Microsoft CNTK, Caffe, MXNet which are not used in this experiment can also be included. Future comparison can be carried out to measure performance of  neural networks like RNN and CNN using above mentioned libraries

# References

[1] A. Mathew. (2018, 28th October). *Importance of Machine Learning*. Available: https://dzone.com/articles/importance-of-machine-learning

[2] K. Matthews. (2018, 28th October). *Why are so many machine learning tools open source?* Available: https://jaxenter.com/machine-learning-tools-open-source-145934.html

[3] I. Reinstein. (2018, 28th October). *Top 20 Python AI and Machine Learning Open Source Projects*. Available: https://www.kdnuggets.com/2018/02/top-20-python-ai-machine-learning-open-source-projects.html

[4] S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, "Comparative Study of Deep Learning Software Frameworks," *eprint arXiv:1511.06435,* p. arXiv:1511.06435, 2015.

[5] K. Mupparaju, A. Soni, P. Gujela, and M. Lanham, "A Comparative Study of Machine Learning Frameworks for Demand Forecasting," 2018.

[6] A. Shatnawi, G. Al-Bdour, R. Al-Qurran, and M. Al-Ayyou, "A Comparative Study of Machine Learning Frameworks for Demand Forecasting," presented at the 9th International Conference on Information and Communication Systems (ICICS), 2018.

[7] A. Ahmed. (2018, 28th October). *Choosing a Machine Learning Framework in 2018*. Available: https://agi.io/2018/02/09/survey-machine-learning-frameworks/

[8] http://scikit-learn.org/stable/faq.html#will-you-add-gpu-support