I-80 Closures: An Autonomous Machine Learning Approach*

Clay Carper[1], Aaron McClellan[2], and Craig C. Douglas[3]

[1] Department of Computer Science, University of Wyoming, Laramie, WY 82071, USA ccarper2@uwyo.edu
[2] Department of Computer Science, University of Wyoming, Laramie, WY 82071, USA amcclel2@uwyo.edu
[3] School of Energy Resources and Department of Mathematics and Statistics, University of Wyoming, Laramie, WY 82072-3036, USA craig.c.douglas@gmail.com

Abstract. Road closures due to adverse and severe weather continue to affect Wyoming due to hazardous driving conditions and temporarily suspending interstate commerce. The mountain ranges and elevation in Wyoming makes generating accurate predictions challenging, both from a meteorological and machine learning stand point. In a continuation of prior research, we investigate the 80 kilometer stretch of Interstate-80 between Laramie and Cheyenne using autonomous machine learning to create an improved model that yields a 10% increase in closure prediction accuracy. We explore both serial and parallel implementations run on a supercomputer. We apply auto-sklearn, a popular and well documented autonomous machine learning toolkit, to generate a model utilizing ensemble learning. In the previous study, we applied a linear support vector machine with ensemble learning. We will compare our new found results to previous results.

Keywords: Machine Learning· Autonomous Machine Learning· Road Closure.

1 Introduction

In 2018, the first author commuted between Cheyenne and the University of Wyoming in Laramie, Wyoming. During this nine month period it became apparent while driving between the two towns there were inconsistent classifications concerning road closures. Road conditions sometimes were severe and dangerous while the roadway was classified as open. Road conditions sometimes were clearly safe while the roadway was classified as closed. The experience overall was one that is shared with anyone who regularly travels the corridor between these two towns.

Featuring a maximum elevation differential of nearly 800 meters between Cheyenne and Laramie (1,848 and 2,184 meters above sea level, respectively),

* This research was supported in part by National Science Foundation grant DMS-1722621.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_22

this section of Interstate-80 (I-80) is a predictive nightmare. The misclassification many Wyoming residents and frequent visitors have grown accustomed to leads to needless loss of life, accidents, higher insurances rates for Wyoming residents, and delays in interstate commerce. Prior research [5, 6] has shown that applying machine learning to locally obtained sensor data provides a viable solution to this misclassification problem.

In this paper we will review previous work and results, the details of our application of autonomous using the University of Wyoming's Advanced Research Computing Center (ARCC) [1], our results and comparisons to prior results, and the implication of our new results and future work in sections 2, 3, 4, and 5, respectively.

2 Prior Work

This project begin as a class project in fall 2018. To date, two documents have been published: an international conference paper [5] and a master's thesis [6].

The first publication was in the Proceedings of the 2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Wuhan, China. In this work, we first obtained raw sensor data on the roadway between Laramie and Cheyenne, Wyoming from MesoWest [10]. This data consisted of 29 individual quantities measured at each sensor at irregular time steps and was reduced to six parameters: air temperature, relative humidity, wind speed, wind bust, visibility, and dew point. This was done due to sparsity of parameters, usability, and principle component analysis. Sparse parameters were omitted due to skewing concerns. While replacing missing parameters with the median is standard practice, doing so on millions of missing entries will undoubtedly skew the data in an undesirable fashion.

The weather condition was a feature among the cut parameters. Although this parameter was dense enough to be viable, the datatype presented issues. Since the weather condition parameter contained plaintext values such as *snow squall*, *thunderstorm*, and *mostly cloudy*, we were unable to assign meaningful and consistent numerical weights. Such an endeavor would require insight into meteorology beyond the scope of our work. Applying a one-hot encoding of these parameters may have been possible but would have drastically increased the number of features and each one would be very sparse.

There are many was to ensure selected features maintain predictive power. In this case, we ensured the selected features maintained at least 90% of the energy in the system using principle compenent analysis. To supplement the sensor data, we obtained closure status data from the Wyoming Department of Transportation [12]. This data consisted of binary classifications of weather related roadway closures and ad hoc roadway closures. For consistency, we only used the weather related roadway closures. Using the scikit-learn machine learning framework, we trained a linear support vector machine (LSVM) on a subset of the aforementioned data. Such a model yielded a maximum accuracy of 71% [5].

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_22

Further work occurred for the first author's master's thesis. Ensemble learning was the major addition to the project [2, 4, 9]. Using scikit-learn's implementation of AdaBoost, we obtained a 10% boost in accuracy for a maximum of 81% [6]. Boosting is a common form of ensemble learning, allowing for additional copies of the original classifier to be applied to the same subset of data. This yielded a 13% improvement over a single linear support vector machine model. Additionally, we performed confusion matrix analysis for the standard and boosted models.

Rationale behind all choices can be found in prior publications [5, 6], while results from the previous publications can be found in table 1. Note throughout both analyses, cross validation was the metric of choice.

Table 1. Prior Models, Cross Validation Confusion Matrix

Model	True Negative	False Positive	False Negative	True Positive
Standard	59.05%	10.42%	18.56%	11.97%
Boosted	64.49%	4.98%	14.57%	15.95%

3 Methods

In this section, we present the tools used in this research and their versions, our reasons for choosing these tools, and the methods that led to an improved model. We strive to provide research that is verifiable and reproducible, thus everything needed to recreate our results is provided. Not only do we describe our tools, we include the versions of all software used, the versions of all tools' immediate (but not transient) dependencies, our code and data, and sample execution scripts for use on supercomputers.

The code was written in Python, the most prevalent programming language in academic data science research. This choice was necessitated by the fact that our other tools only provide Python APIs, leaving little choice for other languages.

Our most important tool is auto-sklearn [8]. It is an autonomous machine learning (AML) framework, which attempts to find the best machine learning model automatically with little or no guidance from the researcher. The tool auto-sklearn has undergone stringent comparisons with other AML frameworks and compares favorably in most categories [8]. In order to automatically find optimal models, auto-sklearn creates a parameter space that models can be selected from. Then it algorithmically searches through the parameter space until a termination condition is met. For our purposes, there are two termination conditions: auto-sklearn exhausts its allocated execution time or an optimal model is found (for a user-specified definition of optimal). Several optimality metrics are included with the framework.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_22

The University of Wyoming's supercomputer, ARCC [1], is our execution platform of choice. The ability to run our code in an environment where compute resources were effectively unconstrained allowed for shorter iteration times and the discovery of higher quality models.

As is common for supercomputers, ARCC uses a software scheduler to allocate compute resources. We provide sample scripts to run our code for use with ARCC's scheduler, Slurm [1]. Since many supercomputers schedule with Slurm, we hope our scripts allow researchers to verify and reproduce our results on their supercomputer of choice.

Five individual iterations were ran, using one, two, four, eight, and sixteen CPUs. The reason for this is partially because scikit-learn is built on top of Dask [7] and partially due to the nature of auto-sklearn. Dask allows Python programs to be scalable across multiple nodes by generating a Dask cluster. In doing so, it organizes the workers and handles the management of the cluster. Further, in allocating additional CPUs with a fixed amount of time we are simply running additional models. This does not change running times but yields greater accuracy by checking additional types and variations of models.

Since software is constantly changing, we provide a Conda environment containing all the versions of all software described and their dependencies. At this time, the code and data is available upon request.

4 Results

To benchmark an algorithm using auto-sklearn, we use a fixed subset of data that has previously been tested on. This data is available upon request. For our testing, we used the `AutoSklearnClassifier` [8] with randomized training and test sets. scikit-learn [11, 3] provides a function called `sklearn.metrics.accuracy_score`, which we used to determine the accuracy of the model. Doing so yields an average accuracy of nearly 91%, an improvement of 22% over the base LSVM model [5] and an improvement of 11% over the ADABoosted LSVM [6]. It is important to note that auto-sklearn automatically applies ensemble learning, allowing for a linearly-boosted model to be one of the possible parameters [8]. Accuracy for each of the five variations can be found in table 2. The number of models checked are the number of models that ran successfully. Instances of failed models are either models that crash, exceed the time limit, or exceed the memory limit. Memory usage is the amount of memory reported by Slurm. It is worth noting that the memory allocated for each experiment was 80-100 gigabytes. We are yet to identify why this is necessary, however, without allocating a significant amount of additional memory the multi-core instances will fail to produce output.

The models auto-sklearn found is an ensemble consisting of nine individual models, each with their own weights. Of the models found, 70% of the predictive power is encapsulated in the first three models. Those models are two different ADABoosted linear models and an extra trees model, respectively. The remaining models are a collection of tree-based algorithms and k-nearest neighbors.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_22

Table 2. Accuracy for auto-sklearn models and number of models checked

CPUs	Accuracy	Models Evaluated	Memory Usage (GB)
1	90.69%	72	2.68
2	90.83%	101	4.99
4	90.96%	161	8.16
8	90.85%	191	10.13
16	90.76%	255	14.82

More specifically, they are three random forest models, another ADABoosted linear model, a quadratic discriminant analysis, and k-nearest neighbors.

Another typical metric for analyzing machine learning algorithms is confusion matrix analysis. We elected to do so in this research, the results of which are given in table 3. Please note that in this research, negative refers to a prediction of a nonclosure and positive refers to a closure.

Table 3. Confusion Matrices for auto-sklearn Models

CPUs	True Negative	False Positive	False Negative	True Positive
1	67.55%	1.93%	3.10%	27.43%
2	67.72%	1.75%	3.22%	27.31%
4	67.65%	1.82%	3.13%	27.39%
8	67.74%	1.74%	3.18%	27.34%
16	67.67%	1.81%	3.15%	27.38%

Interestingly, an average of 35% of the incorrect classifications are related to predicting the roadway being closed when it in fact was open and on average 65% of the incorrect classifications are related to predicting the roadway being open when it was in fact closed. This is nearly identical to previous confusion matrix values of 35.96% and 64.04%, respectively. It is worth noting that the variations in the confusion matrices is within variational norms; adding models doesn't necessarily imply an increase in accuracy of predictions.

5 Conclusions and Future Work

As a continuation of previous work, an improvement of 11% in accuracy is substantial, with an average accuracy of 91%. Similar results in the confusion matrix lead us to conclude that the application of auto-sklearn is a viable and meaningful next step for classifying roadway closures. Future directions for this project include dataset subset analysis and validation and automation.

References

1. Advanced Research Computing Center: Teton Computing Environment, Intel x86_64 cluster (2020). https://doi.org/10.15786/M2FY47

DOI: 10.1007/978-3-030-77977-1_22

2. Aurélien, G.: Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. OReilly Media, Sebastopol, CA (2017)
3. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)
4. Burkov, A.: The Hundred-Page Machine Learning Book. Andriy Burkov (2019)
5. Carper, C., Douglas, C.C.: I-80 closures: A support vector machine model. In: Proceedings of DCABES 2019. Wuhan, China (8-10 November 2019 G Yucheng (eds), IEEE Computer Society Press, Los Alamitos, CA, 2019, pp 199-202)
6. Carper, C.: A Support Vector Machine Model for Predicting Closures on Interstate 80. Master's thesis, University of Wyoming, Laramie, WY (2020)
7. Dask: https://dask.org/, last visited 02/18/2021
8. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Efficient and robust automated machine learning. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 2962–2970. Curran Associates, Inc. (2015), http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf
9. Leskovec, J., Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets, Third Edition. Cambridge University Press, New York, NY (2020)
10. MesoWest Data: https://mesowest.utah.edu/, last visited 12/10/2020
11. 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., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
12. Wyoming Department of Transportation: http://www.dot.state.wy.us/home.html, last visited 12/10/2020

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_22