Winter Precipitation Forecasting Based on Random Forest Algorithm

Kelly Lavertu
Kennesaw State University
Marietta, GA USA
klavertu@students.kennesaw.edu

Christopher Sefcik
Kennesaw State University
Woodstock Georgia USA
csefcik@students.kennesaw.edu

Brian Rawls
Kennesaw State University
Norcross Georgia USA
brawls3@students.kennesaw.edu

ABSTRACT

Our paper will build on the research performed by Kanavos et al. in the research paper "Forecasting Winter Precipitation based on Weather Sensors Data in Apache Spark". We will be using data gathered from weather sensors to forecast precipitation outcomes: rain, freezing rain and snow. We will use the classical random forest algorithm and compare the results with the algorithms: Random Forest, Naive Bayes and the Decision Tree. The performance of the algorithm will be measured based on accuracy and computation time. We will then compare our results with the results found in their research paper.

RESEARCH STATEMENT AND CONJECTURE

Our research team is comprised of several computer science master's students with an interest in algorithms, computing, and machine learning.

Predictive Forecasting in Modeling Precipitation Outcomes

This area of our work, modeled using automated surface observing systems (ASOS) weather datasets, observes the outcome of performance of random forest and several other algorithms against the proposed dataset. Our research is aiming to answer questions such as: is Random Forest more accurate in predictive power given our model is trained using a system with sufficient storage and memory? Does Random Forest prove to take less computational time than any of the observed algorithms in the "Forecasting Winter Precipitation based on Weather Sensors Data in Apache Spark" research paper?

We hypothesize that random forest will prove to be more optimal, and performant based on accuracy in predictive power and computation time. However, there is a known drawback: a trained forest will require significant memory for storage due to the need to retain information from a vast number of individual trees. We will additionally need to deploy our model on a system with sufficient resources given our project budget.

RELATED WORK

The "Forecasting Daily Stock Trends Using Random Forest Optimization" research paper discusses using random forest to predict daily stock market trends. The research team captures its stock price datasets from the Korea Composite Stock Price Index (KOSPI) list and compares the stocks of Celltrion and Pharmicell. One of the challenges is cleaning the initial dataset and selecting a strong set of predictors for training. The random forest classifier adaptively determines the optimization parameters to improve the predictive accuracy while avoiding over-fitting issues. The

researchers found it best to reduce its stock variables down to a small subset with the least number of uncertainties. Due to the highly volatile nature of stock prices, all variables still had an unacceptable amount of uncertainty. This paper is relevant in our research in that our team is researching trends, specifically weather forecast, and looking to predict the forecast outcomes using random forest. The paper details the importance of cleaning the dataset and reducing any imbalance before utilizing the random forest algorithm. Our research is relying on a consistent trend among the datasets we will train, reduction of noisy variables. and grouping of seasons. This will also help effectively reduce the possibility of an overfitting issue. It is our belief that with the given preliminary parameters input for our algorithm we should have a trainable model and an acceptable predictive output of forecast.

In "A comparison between Machine learning algorithms for the application of micro-grids Energy management", Khoshlessan et al. compares the performance of support vector machine, logistic regression, decision tree, and gradient boosting with random forest with respect to classifying the operation modes of microgrids. Khoshlessan et al. found that the model with the best performance is random forest - which was able to classify with 93-95% accuracy when used with the scikit learn PolynomialFeatures package for preprocessing data. For our research, this paper suggests an optimal preprocessing technique to use in combination with random forest.

In the research performed by Dai et al. “Using Random Forest Algorithm for Breast Cancer Diagnosis”, the random forest algorithm is used to diagnose breast cancer. The type of random forest algorithm used, classification and regression trees (CART), consists of four steps: determining the number of attributes, splitting the node using the Gini coefficient, performing training tasks on the decision tree, and voting for the best solution. The researchers chose the random forest algorithm due to its high predictive accuracy in combining results of multiple decision trees. Also, weak predictors can be combined to produce an accurate predictor. The overall result of the paper was that the random forest algorithm performed well and produced accurate predictors. The results of this paper helped reinforce our decision to apply the random forest algorithm to predicting weather precipitation.

METHODOLOGY

Our objective, which is to compare performance measurements of different classification algorithms against the Random Forest algorithm, will require us to create a controlled environment in order to make the comparisons. The measurements used: lowest accuracy, highest accuracy, difference, average and time, will be computed for each algorithm after completing a fixed number of trials. Overall, the measurements will provide detail on which algorithms are faster to run, how well they create classifications and how well they create the classifications consistently. The performance measurements of the Random Forest algorithm will be compared against the performance statistics for: Gradient Boost, Naive Bayes and the Decision Tree algorithm.

The algorithms will be run using the Spark ecosystem. The hardware that our Spark ecosystem resides on consists of 64 GB RAM, 1 TB storage SSD, M1 Max Chip, 10 Core CPU and a 32 Core GPU.

Within our controlled environment, all our algorithms will be run against fifteen standard sized datasets. Five of the datasets will be 10,000 records, five of the datasets will be 25,000 records and the last five will be 50,000 records from the (ASOS) weather data provider. We will be using the following columns: Air Temperature in Fahrenheit, Dew Point Temperature in Fahrenheit, Relative Humidity, Wind Direction in Degrees, Wind Speed in Knots, Pressure Altimeter in Inches, Sea Level Pressure in Millibar, Visibility in Miles, Sky Level 1 Altitude in Feet, Apparent Temperature in Fahrenheit. The variables will be used to predict the dependent variable WXCodes. Running the algorithms against five different sized datasets will allow us to capture any performance loss due to an increasing record count. While our performance loss might not be substantial on a smaller test level, when dealing with millions of rows, the overall performance loss could be substantial if exponential. During the execution of each algorithm, the dataset will be split into a training set and a test set. We will split the dataset randomly using an 80% training split and a 20% test split.

While executing our algorithms on the standard datasets within our controlled environment, measurements will be computed during the execution of each run. These measurements will then be combined, averaged and tabulated by each algorithm for each of the three different sized datasets.

EXPERIMENTAL DESIGN, RESULTS, ANALYSIS, AND COMPARISON

Experimental Design:

The algorithms will be run using the Spark ecosystem. The hardware that our Spark ecosystem resides on consists of 64 GB RAM, 1 TB storage SSD, M1 Max Chip, 10 Core CPU and a 32 Core GPU. Each run, the results will be computed and averaged.

Results:

In evaluating the observation set of 10k samples, the Naïve Bayes algorithm performed the fastest with a time of 0.17 seconds, however, the algorithm was not that accurate. The Decision Tree had the highest performance in regard to accuracy with a 93.4362 average and time, completing in 0.29 seconds, compared to the Random Forest algorithm which had an average of 93.0041 accuracy measure and completed in 2.4 seconds.

In observing the run of 25,000 records, the Decision Tree was again the most performant in terms of accuracy and speed when compared to the Random Forest algorithm. The Decision Tree had an accuracy average of 95.7491 compared to the Random forest with an accuracy measure of 95.5429. The Naïve Bayes algorithm was faster but it was not consistent in terms of accuracy. The Naïve Bayes completed in 0.13 seconds and had an accuracy of 63.3045.

When the algorithms were run against the datasets that contained 50,000 records, we again had the same results. The Naïve Bayes was the fastest with a time of 0.15 but the algorithm was not consistent in terms of accuracy with an average of 72.3558. The Decision Tree was again faster and more accurate when compared to the Random Forest with an average of 94.079 accuracy compared to 93.5545 and completed in 0.42 seconds compared to 3.2 seconds.

Table I
LOWEST, HIGHEST, AND AVERAGE ACCURACY OUTCOMES FOR DATASET = 10,000

Algorithm	Lowest	Highest	Difference	Average	Time
Random Forest	91.531	95.768	4.237	93.0041	2.4
NAÏVE Bayes	53.3906	67.516	14.1255	61.288	0.17
Decision Tree	91.6528	96.277	4.5742	93.4362	0.29

Table 2
LOWEST, HIGHEST, AND AVERAGE ACCURACY OUTCOMES FOR DATASET = 25,000

Algorithm	Lowest	Highest	Difference	Average	Time
Random Forest	92.9956	98.5226	5.527	95.5429	2.6
NAÏVE Bayes	47.7942	81.0861	33.2919	63.3045	0.13
Decision Tree	92.8675	98.4543	5.5868	95.7491	0.44

Table 3
LOWEST, HIGHEST, AND AVERAGE ACCURACY OUTCOMES FOR DATASET = 50,000

Algorithm	Lowest	Highest	Difference	Average	Time
Random Forest	88.494	96.5227	8.0285	93.5545	3.2
NAÏVE Bayes	57.02	87.0232	30.0032	72.3558	0.15
Decision Tree	89.4838	96.7452	7.2614	94.0797	0.42

Analysis and Comparison:

It's important to note the normalized dataset was comprised of numerical features, no missing values, and a dependent variable that is yielded from the interdependency of features and outcomes.

In optimizing our spark environment, we configured the spark session to have the following settings: all available cores, 32gb executer memory, and 32gb driver memory. Datasets were randomized and 5 instances for each algorithm per evaluation size of 10k, 25k, and 50k were generated.

The Naïve Bayes algorithm performed well in time and poorly in accuracy across all given datasets. The fast time is largely due to the small observation size and the algorithm only needing to calculate the probability of each class and it's given different input. No coefficients need to be fitted by optimization procedures. The poor accuracy was yielded due to the algorithm's method of negating the interdependency of features which highly correlates to the output of the dependent variable in this given dataset. This algorithm works best with categorical features and a holding assumption of independence on the given dataset.

The Decision Tree algorithm performed best in accuracy and relatively well in time. The faster time was yielded from the Decision Tree's method of not computing all possible trees and making greedy decisions in the fitting process. The accuracy is largely due to the given dataset being normalized with high interdependency among the features. Given the dataset had more observations, missing values, and low interdependency among the features this algorithm may not have performed as well. In general, the algorithm only performs well due to the low complexity of the dataset and a small set of observations.

The Random Forest algorithm performed slowest in speed and average in accuracy. The slow time in Random Forest is a disadvantage that is known given how the algorithm computes probability and yields its accuracy over the given dataset. Random Forest generally yields the best accuracy and worst computing times when the dataset is large with a fair number of complexities. This dataset was small for the given environment, thus Random Forest's ability to utilize multiple trees and solve NP-hard problems is not optimal. If the observation size is increased, this algorithm would be much more consistent with high accuracy in comparison to the other proposed algorithms.

CONCLUSION

This work focused on comparing the performance of the Random Forest Algorithm against two other algorithms to accurately forecast weather precipitation. The three classification algorithms were applied to evaluate five instances each of three different dataset sizes. Ultimately, the Random Forest Algorithm did not perform the best in terms of both time and accuracy whereas the Decision Tree proved to perform optimally in each dataset. This outcome does not adhere to our original hypothesis that the Random Forest Algorithm would perform better than the Decision Tree and NAÏVE Bayes algorithms. Some focal points of our future work would be to expand our results to include other classification algorithms, at least double the size of our datasets, and incorporate a gradient boosted tree to optimize performance.

REFERENCES

[1] B. Dai, R. Chen, S. Zhu and W. Zhang, "Using Random Forest Algorithm for Breast Cancer Diagnosis," in 2018 International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, 2018 pp. 449-452.doi: 10.1109/IS3C.2018.00119keywords: {decision trees;classification algorithms;training;machine learning algorithms;machine learning;breast cancer}url: https://doi.ieeecomputersociety.org/10.1109/IS3C.2018.00119

[2] A. Kanavos, T. Panagiotakopoulos, G. Vonitsanos, M. Maragoudakis and Y. Kiouvrekis, "Forecasting Winter Precipitation based on Weather Sensors Data in Apache Spark," 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), 2021, pp. 1-6, doi: 10.1109/IISA52424.2021.9555553.

[3] M. Khoshlessan, B. Fahimi and M. Kiani, "A comparison between Machine learning algorithms for the application of micro-grids Energy management," 2020 IEEE International Conference on Industrial Technology (ICIT), 2020, pp. 805-809, doi: 10.1109/ICIT45562.2020.9067203.

[4] J. S. Park, H. Sung Cho, J. Sung Lee, K. I. Chung, J. M. Kim and D. J. Kim, "Forecasting Daily Stock Trends Using Random Forest Optimization," 2019 International Conference on Information and Communication Technology Convergence (ICTC), 2019, pp. 1152-1155, doi: 10.1109/ICTC46691.2019.8939729.