Ensampling Data Prediction Using Sparse Data in Mobile Intelligent System

https://doi.org/10.3991/ijim.v13i10.11311

S. Dhamodaran (✉)
Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India
s.dhamodaran07@gmail.com

M. Lakshmi
Saveetha School of Engineering, Chennai, Tamil Nadu, India

Abstract—Rain is an important source of water. The Indian economy is heavily dependent on agriculture and the livelihood of Indian farmer largely depends on rains. The farms are more dependent on rainfall than any other water resource. As we observe there is a lot of climate change in recent years due to industrialization, because of these climate changes there are a lot of floods and droughts. So, it might be helpful to predict the rainfall beforehand and take necessary precautions to protect crops and other sorts of damages that might occur due to the irregular rainfall. So, we present a model that could reasonably predict the future rainfall using very fewer variables. The reason why we developed a model which uses sparse data is that sometimes it could be hard to obtain a large amount of data due to lack or improper working of recording equipment and so on. So, it will be good to have a working model in those situations and applications are implemented. This paper affords an overview of system learning and offers a brief take a look at on distinctive machine gaining knowledge of strategies together with their programs on mobile devices. It also affords an outline of overall performance-associated parameters of gadget studying techniques useful for mobile devices.

Keywords—Bagging, Boosting, Data analysis, Mobile Application, Mobile Intelligent System, Machine Learning techniques, Rainfall Prediction SVM.

1 Introduction

We all know and seen several examples (like recent floods in Kerala) of floods and drought causing a lot of life, property and resources loss.So, it will be good to have a way to reduce the damage caused by them [10]. To address this issue, we have designed a model, which could predict the future trends of rainfall. So that we can take the necessary precautions to minimize the damage caused by them.

Our model is trained on the rainfall data from 1980 to 2014. The data is collected from across three weather stations across Chennai. The dataset consists of 951 rows and 36 columns, which is significantly fewer data compared to what machine learning algorithms need, to get good accuracy. The saddest part is most of the data is null values

and zeros. So, we have done a lot of data processing to help the model find the relationship between the variables easily. We have removed the majority of variables from the data to help our model grasp the relationships better. That's where the term "sparse data" comes in the title.

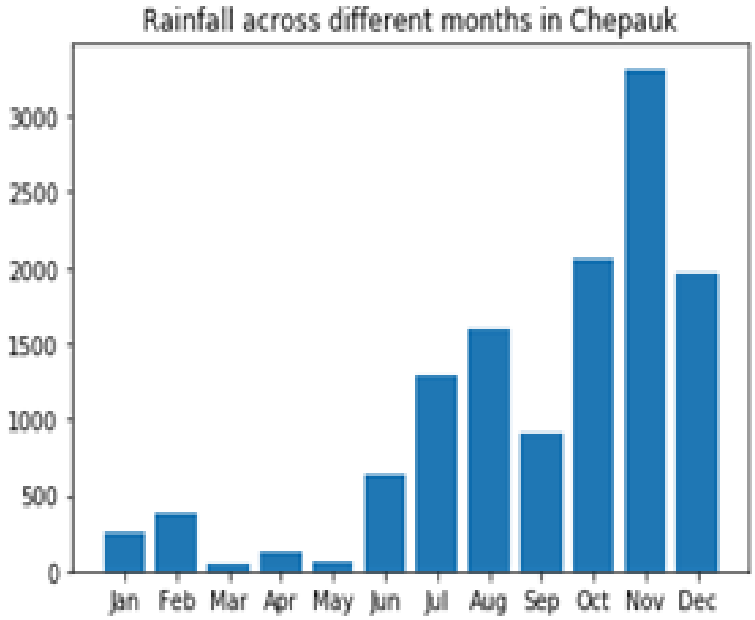

Fig. 1. Rainfall across the chepauk location.

Finally, after a lot of experimentation, we have decided to use ensemble techniques to implement our model. That is because of the procedure the ensemble methods follow to find the relationship between the variables, is efficient and works well for our case

2 Related Work

Our work is closely related to [1]. Which is done by B.Kavitha Rani and A.Govardhan. In their research, they have used Multilayer Perceptrons to make a time series model and used the model to predict the rainfall. The reason why we haven't used Multilayer Perceptrons is that we have very fewer data and our data is a lot noisy(i.e. had many zeros and null values). What happened when we tried to use Multilayer Perceptrons on our data is that the gradient is not converging to the lowest error Our dataset consists of 951 rows and 36 columns. It was collected from across three weather stations in Chennai.The stations are Nungambakkam, Egmore, Chepauk. The data is collected between the years 1980 and 2014. The columns are station id, year, month, functioning status, rainfall status(0-31 days) and total rainfall

point(i.e. The weights are not changing as they should, to get the lowest error).

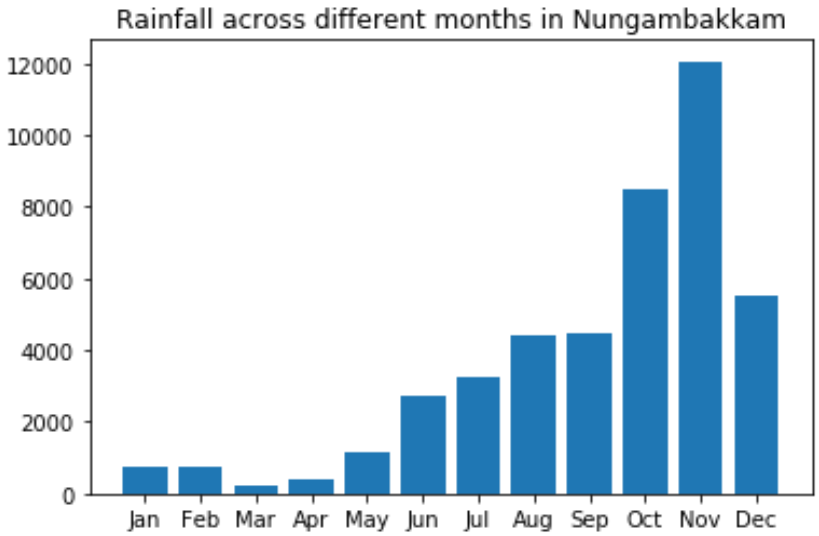

Fig. 2. Rainfall across the nungambakkam location

3 About Dataset

3.1 Data analysis

Data analysis gives us insights about the trends in the dataset. It helps us understand the relationship between the features in the data. It allows us to observe the trends in the data. By observing the relationship and trends in the data, we can get an idea about dependent and independent variables or features in the data. It helps us remove some dependent variables and save computational resources.

We have done some data analysis and visualization to observe the trends in rainfall across the three weather stations and each year [7]. First, we will show some visualizations of how the rainfall is across the three weather stations in different months.

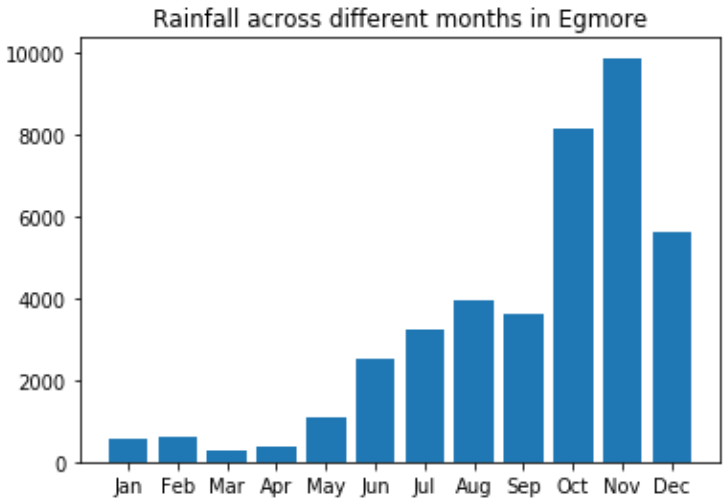

Fig. 3. Rainfall across the Egmore location

As we can see the distribution of trends in rainfall among the three weather stations, they are quite similar, since they are from the same city. Now we will show some visualizations about the rainfall trends are, across the three weather stations in different years.

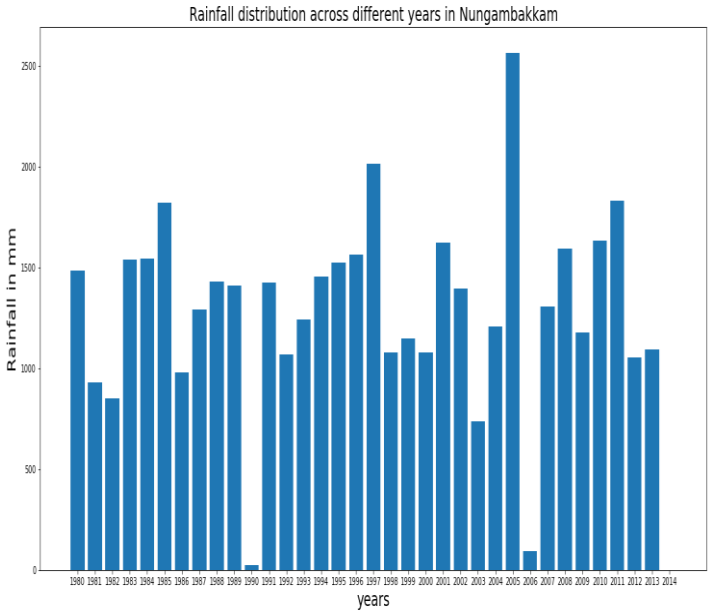

Fig. 4. Distribution of different years in nungabakkam

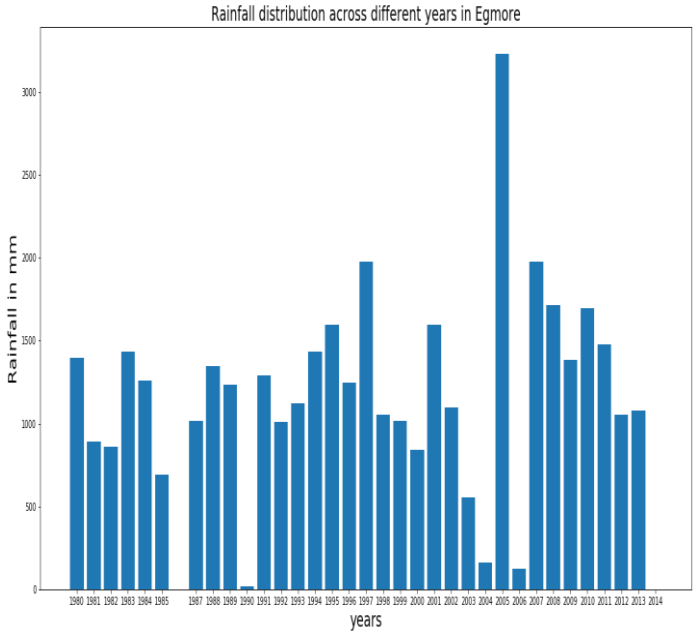

Fig. 5. Distribution of different years in Egmore

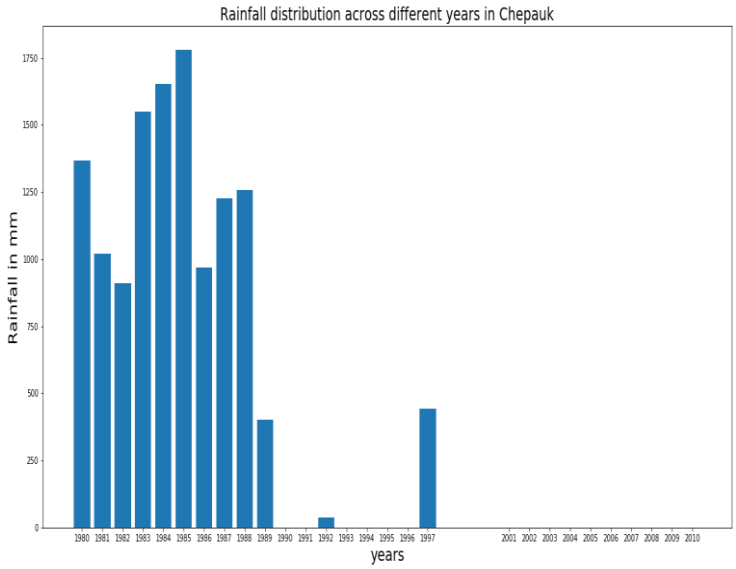

Fig. 6. Distribution of different years in Egmore

By seeing the above three visualizations, we can get that the last visualization is incomplete, this is due to the noise in the data [7]. As we mentioned earlier there are a

lot of "null" values in the dataset, due to them we are unable to plot the trend of rainfall, for that particular station

3.2 Data preprocessing

The data available will not be in a state that our machine learning model can process immediately. There will be a lot of noise and unnecessary features in the data. We need to remove those unnecessary things from the data, to allow our machine learning model to find the relationship between the features easily. The accuracy of our model depends on how we filter our data. In our case, there are three things to be preprocessed.

- Null Values
- Station id in the form of strings instead of integers
- Too many features for fewer data

3.3 Removing null values

Our data is of 951 rows x 36 columns in size. In that about 300 rows are null. These 300 rows are useless. They contribute nothing to model accuracy. By removing these null values, we can save some computational resources and time. For this task, we first replace the null values with zeros using pandas "replace" method and then separate the non zero rows from the dataset. After removing the zeros we are left with 647 rows.

3.4 Converting station ID to integer

In our data, the station ids are in the form of strings. Machine learning models are generally not good at understanding strings, so we need to convert them to integers to feed them into our model. In our case, the station id's are like "^01010121" (example). So here we need to discard**"^"**from the string and then convert the string to an integer. For removing "^" we use python "split ()" method.

3.5 Removing unnecessary features

As said earlier our dataset has 36 features. In these 31 features belongs to rainfall from day0 to day31. Finding the relationship between these 31 variables is difficult for the available 647 rows. So, we need to find some alternative for these columns. There is another column called "total_rainfall", which contains the total rainfall of these 31 days. So, taking this column alone makes sense instead of 31 columns. So, after filtering the features we are left with only 4 features. So, finding a relationship between these 4 features makes sense for the amount of data available. This will also help us improve the accuracy of the model. distinctive machine gaining knowledge of strategies together with their programs on mobile devices.

4 Approaches

We have used different ensemble techniques like bagging and boosting to predict future rainfall. Before digging into bagging and boosting, one must know about bootstrapping. Bootstrapping is a technique of subsampling from a large dataset. Bagging and Boosting use bootstrapping. In this method, we train many models on different subsets of data. The subsets of data are chosen by random resampling of data (i.e. a data item or feature can be chosen multiple times). This reduces the correlation between features in the dataset [10]. This allows each model to find a different relation between features in the dataset. Bootstrapping helps in reducing the variance in the model prediction. Bootstrapping helps in creating a robust model since each model picks different relationships there are fewer chances of overfitting. Bootstrapping has become common recently, because of the increase in computational power.

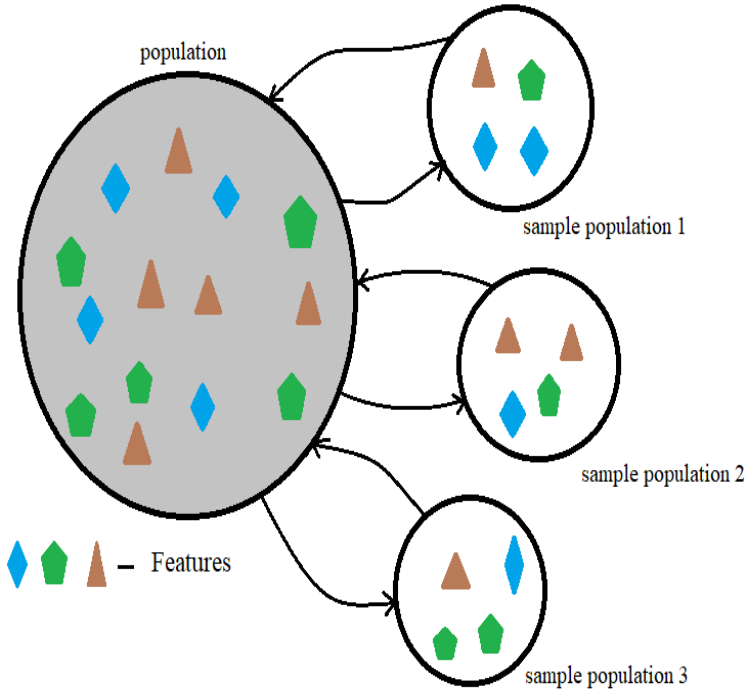

Fig. 7. Boostrapping Methods of Sample Population

4.1 Bagging

Bagging is also called as bootstrap aggregator. This method was introduced by Leo Breiman in 1996. He wrote a paper called "Bagging Predictors". Bagging is a technique used to reduce overfitting due to the high variance of the model. Overfitting occurs when the model's function is too complex, such that it captures even minute details in the dataset (i.e. including outlier's). If the model is overfitting, the model performs well

on the training data but fails to generalize well on the test data. Bagging reduces overfitting by using bootstrapping to draw random subsamples from the dataset with replacement and then training multiple models on the subsamples. Each subsample will be having a different mean, standard deviation, variance. Since each subsample contains different features, each model will pick different constraint to make predictions out of the sample. The bagging is the process of combining multiple models, to produce a robust model. For regression tasks, bagging averages the results from multiple models to make a prediction. For classification tasks, bagging uses voting system to predict the correct class (i.e. outputs a class that is predicted by most of the models).

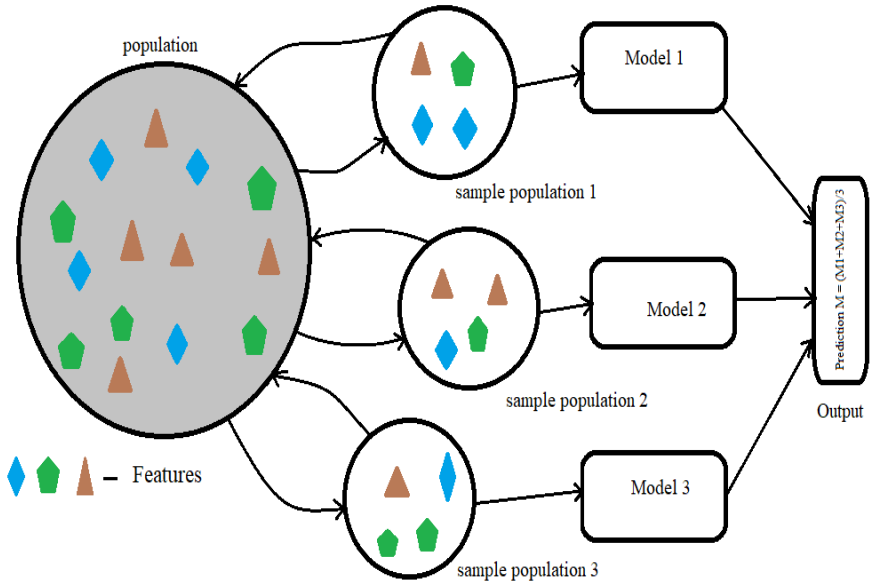

Fig. 8. Fig.7. population with different model with output Boosting

Boosting too uses bootstrapping to draw subsamples from the dataset. Unlike bagging, boosting combines many weak learners to form a strong learner. A weak learner is a model whom's predictions are very less correlated to the actual classes. Its predictions are a bit better than random guessing. Unlike weak learner, a strong learner is a model or a classifier whom's predictions are very strongly correlated to the actual classes.

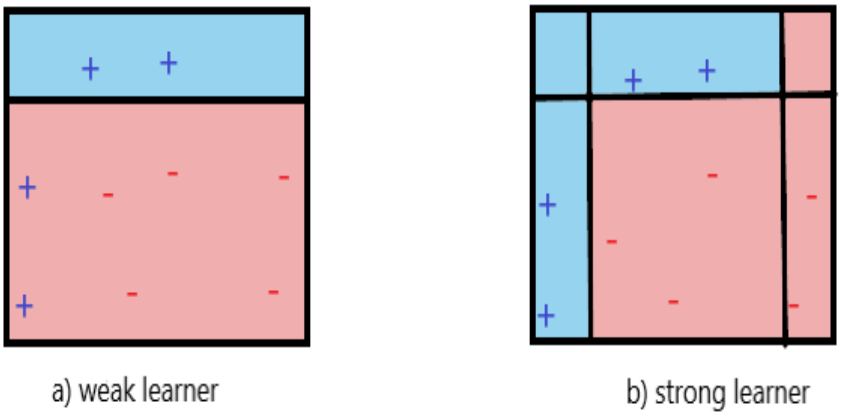

Fig. 9. Prediction of learners

Now, lets' return to boosting. Unlike bagging, in boosting assigns weights to each data sample and uses the weights to decide the samples to be processed. Initially, all the samples are given equal weights. Then some samples are drawn using bootstrapping. These samples will be given to some learner (i.e.algorithm or classifier). The learner will be able to classify or predict some labels correctly by finding some relationship between the features. This is called a weak learner. Now the weights for the correctly predicted samples will be decreased and the weights for ak learner is initialized, it will be made to focus misclassified samples will be increased. When the next wemore on the misclassified samples.

Fig. 10. Learner with new boosting model

This process is repeated until the actual error is decreased. Unlike bagging, in boosting a sample may be processed many times until it is classified correctly. Boosting is extensively used in many machine learning problems because of its ability to reduce the variance and bias.

Here are the steps involved in the boosting,

```
(x – data, y – actual output, y_pred – predicted
output, F.() – Model,
H.() – Error Residual Model, ER – Error Residual, ERP –
Error Residual Prediction)
```

Fit a model to the data and get the predictions, y_pred = F0(x).

Calculate error residuals, ER = y –y_pred.

Fit a model to error residuals and obtain the predictions, ERP = H0(ER).

Update the initial prediction by adding error residual prediction to it, y_pred = y_pred + ERP.

The above process can we represented sequentially as,

```
F(x) = F0(x) →F1(x) = F0(x) + H0(x) …… →Fm(x) = Fm-
1(x) + Hm-1(x).
```

Here H. () is just a model, it need not be a tree-based model. But in most cases, we tend to use a tree-based model. It purely depends on the problem. In most of the cases, a tree-based model works well, but sometimes we need to look at other architectures (like SVM) too.

5 Results

We have implemented both boosting and bagging on the dataset. In both the methods we run multiple weak learners on the data. The only difference is in boosting we run the weak learners sequentially and in bagging we run weak learners in parallel. They both have their own advantages and disadvantages. Bagging is more resource hungry, but very less prone to overfitting. Unlike bagging, boosting consumes fewer resources, but it is more likely to overfit. Finally, we have to tune the hyperparameters of these models such that we end up getting better results.

We have used two versions of boosting algorithms and one version of bagging algorithm. We have observed some significant performance differences between these algorithms. The Gradient-based boosting algorithm outperformed the other boosting algorithm with a significant difference. It had achieved nearly twice the accuracy of the other boosting algorithm.

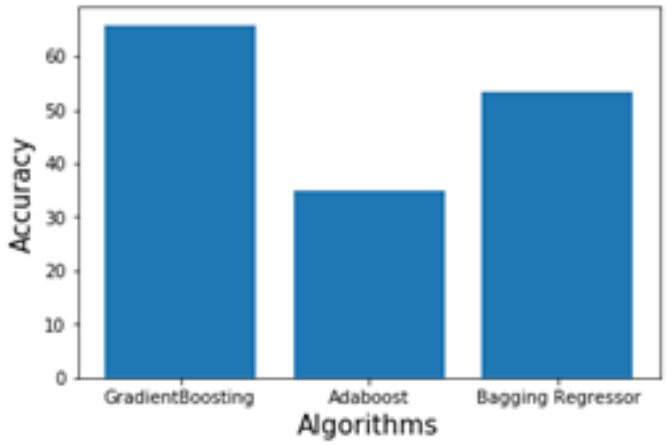

Fig. 11. Accuracy finding based on different algorithms

In this problem, the boosting algorithm managed to surpass bagging algorithm's accuracy, due to its ability to learn from errors. We have visualized the accuracies of the algorithms using matplotlib. The implementation can be modelled and designed in mobile applications .As we can see the gradient boosting model achieved the highest accuracy, which is around 65 percent and Adaboost model scores around 35 percent.On the other hand, the bagging regressor model scored an accuracy of around 55 percent. Considering the amount of data, we have, the accuracies that we have achieved are quite good. We have experimented with a lot of other models (other than ensemble) but, many of them got accuracy less than 10 percent. Some even got less than 5 percent. Since ensemble models are the combination of many models, they will use the best features of various models to make a prediction. So, the resulting prediction will be more accurate. The accurate machine learning algorithm predictions are to be used in machine intelligent systems.

6 Experimentation

Initially, we have built a "Neural Net" model, to predict the rainfall. But since there are too many variables compared to the data available in the dataset our neural net model failed to find the relationship between the variables(features). Then we removed most of the variables (as described in the data preprocessing part) and trained the neural net again. But due to low dataset size, the neural net was unable to pick up the relationship between the variable(features). The neural net model performed worse than random guessing.

We then tried some other models like Logistic Regression, Linear Regression, KNN, Bayesian Ridge, SVR, Decision Trees. The Regression, KNN, Bayesian models have scored an accuracy of around 30 percent. As we can observe these models are not as complex as ensemble models, so the accuracy is much lower.

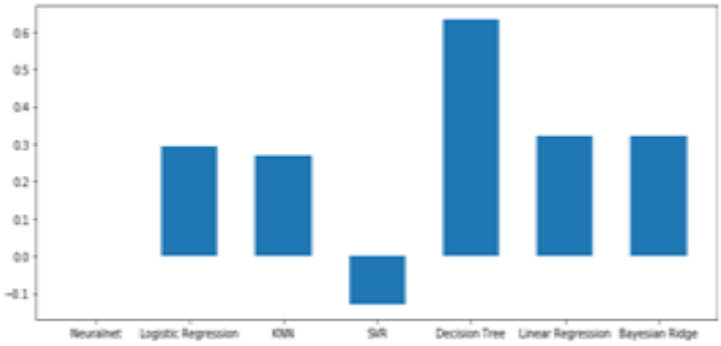

Fig. 12. Accuracy of different Machine learning output

6.1 Why SVR got the least accuracy

SVR is the worst performer among all the other algorithms. To explain, the reason behind its poor performance we should start by knowing about its working. SVR predicts a value, such that the value lies in equal distance from the values lying either size of it. In our case, the variables(features) are very less correlated and there is a lot of variance in the target values. So the SVR is unable to predict a proper value, such that it lies in the equal distance from the surrounding points. In our case, the SVR is predicting the values too far from the actual values. So, the accuracy went to negative 10 percent.

6.2 Problem with decision trees

We can see in Fig 11, that the decision trees has scored an accuracy greater than 60 percent, which is equal to that of aGradient Boosting model. Even though the Decision Tree model scored a good accuracy, we have decided to use Gradient Boosting for the final model. The problem with a decision tree model is that they perform badly during online learning. Online learning is a training process in which a model is trained with the values predicted by the model. Sometimes new variables will be added while training the model. During these dynamic situations, the decision tree tends to pick a wrong feature to split the variables. Due to this, the model will generate a very unreliable result in real-worldscenarios. So, we chose to use a Gradient Boosting model for generating the predictions of future rainfall

7 Conclusion

In this work, we have introduced a task of predicting future rainfall from sparse data using ensemble models. We have designed a rainfall predictor using a Gradient Boosting model. This model can predict future rainfall with reasonable accuracy. This model will be very helpful in the places where the data and features are very less.

8 References

[1] B.Kavitha Rani,A. Govardhan Rainfall Prediction Using Data Mining Techniques.

[2] Rahman, M. M., Bhattacharya, P., & Desai, B. C. (2007). A framework for medical image retrieval using machine learning and statistical similarity matching techniques with relevance feedback. IEEE Transactions on Information Technology in Biomedicine, 11(1), 58-69. https://doi.org/10.1109/titb.2006.884364

[3] Morales, M., Tapia, L., Pearce, R., Rodriguez, S., & Amato, N. M. (2004). A machine learning approach for feature-sensitive motion planning. In Algorithmic Foundations of Robotics VI (pp. 361-376). Springer, Berlin, Heidelberg. https://doi.org/10.1007/10991541_25

[4] Ireland, G., Volpi, M., & Petropoulos, G. P. (2015). Examining the capability of supervised machine learning classifiers in extracting flooded areas from Landsat TM imagery: a case study from a Mediterranean flood. Remote sensing, 7(3), 3372-3399. https://doi.org/10.3390/rs70303372

[5] Huang, C., Davis, L. S., & Townshend, J. R. G. (2002). An assessment of support vector machines for land cover classification. International Journal of remote sensing, 23(4), 725-749. https://doi.org/10.1080/01431160110040323

[6] S. Dhamodaran, A.Shrthi ,Adline Thomas, " High-Resolution Flood Hazard Mapping Using Remote Sensing Data", International Conference on Computation of Power, Energy Information and Communication (ICCPEIC),ISBN**:** 978-1-5090-0901-5,IEEE. https://doi.org/10.1109/iccpeic.2016.7557210

[7] Cheriyadat, A. M. (2014). Unsupervised feature learning for aerial scene classification. IEEE Transactions on Geoscience and Remote Sensing, 52(1), 439-451. https://doi.org/10.1109/tgrs.2013.2241444

[8] Bauer, S., Nolte, L. P., & Reyes, M. (2011, September). Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 354-361). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-23626-6_44

[9] S. Dhamodaran, Dr. M. Lakshmi "Design and Analysis of Spatial–Temporal Model Using Hydrological Techniques" IEEE International Conference on Computing of Power, Energy & Communication" on March 22nd and 23rd, ISBN 978-1-5090-4324-8/17, 2017. https://doi.org/10.1109/iccpeic.2017.8290333

[10] A. Chaudhary et al Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp.913-917

[11] Chou, Y. H., Tiu, C. M., Hung, G. S., Wu, S. C., Chang, T. Y., & Chiang, H. K. (2001). Stepwise logistic regression analysis of tumor contour features for breast ultrasound diagnosis. Ultrasound in medicine & biology, 27(11), 1493-1498. https://doi.org/10.1016/s0301-5629(01)00466-5

[12] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R News, 2(3), 18-22.

[13] Sarhan, A. M. (2009). Cancer classification based on microarray gene expression data using DCT and ANN. Journal of Theoretical & Applied Information Technology, 6(2).

[14] Steven K. Rogers, Dennis W. Ruck, Matthew Kabrisky, Artificial Neural Networks for early detection and diagnosis of cancer, Elsevier Scientific Publishers, pp. 79-83, 1994. https://doi.org/10.1016/0304-3835(94)90089-2

[15] Md. Badrul Alam Miah, Mohammad Abu Tousuf, Detection of Lung Cancer from CT Image Using Image Processing and Neural Network, IEEE, In Proceedings of 2nd Int'l

Conference on Electrical Engineering and Information & Communication Technology, 2015 https://doi.org/10.1109/iceeict.2015.7307530

[16] Lu, R., Marziliano, P., & Thng, C. H. (2006, January). Liver tumor volume estimation by semi-automatic segmentation method. In Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the (pp. 3296-3299). IEEE. https://doi.org/10.1109/iembs.2005.1617181

[17] Haris, K., Efstratiadis, S. N., Maglaveras, N., & Katsaggelos, A. K. (1998). Hybrid image segmentation using watersheds and fast region merging. IEEETransactions on image processing, 7(12), 1684-1699. https://doi.org/10.1109/83.730380

[18] Meng, F., Li, H., Liu, G., & Ngan, K. N. (2013). Image cosegmentation by incorporating color reward strategy and active contour model. IEEE transactions on cybernetics, 43(2), 725-737. https://doi.org/10.1109/tsmcb.2012.2215316

[19] B Revathy, S. and Parvatha Varthini, Decision theoretic Evaulation of Rough Fuzzy Clustering(2014), Arab Gulf journal of Scientific Research, Volume.32,Issue. 2/3Pages:161-167.

[20] K. Geetha, Arputharaj Kannan, Efficient spatial query processing for KNN queries using well organised net-grid partition indexing approach (2018). IJDMMM 10(4): 331-352. https://doi.org/10.1504/ijdmmm.2018.10015876

9 Authors

Mr. S. Dhamodaran is currently working as Assistant Professor of Department of Computer Science and Engineering in Sathyabama University, Chennai. S Dhamodaran is a Research Scholar in Sathyabama Institute of Science and Technology He has more than seven years in Teaching and is working in Sathyabama University from 2011 and has held different positions. He completed his B. Tech (Information Technology) from Anna University (CEG) and M. Tech (Information Technology) from Sathyabama University and is pursuing his Ph.D. in the area of Spatial Data Mining. He has published 23 Papers in International Journals, Conference and national Conferences, out of which five Papers are in web of science publication. He has published one patent. He has coordinated and organized national conferences, faculty development programs and many workshops for students, faculty and Research scholars.

M. Lakshmi works as a Professor at Saveetha School of Engineering, Chennai, Tamil Nadu, India, laks@icadsindia.com

Article submitted 2019-07-16. Resubmitted 2019-08-17. Final acceptance 2019-08-28. Final version published as submitted by the authors.