Introduction of a new Metric Hit Rate and it's Variation with Scaling on Classification Algorithms

Swapnil Ahuja
Guru Gobind Singh
Indraprastha University,New
Delhi,110009

ABSTRACT
This paper aims to introduce a new metric Hit Rate and how it is effected by the introduction of Scaling and also how does scaling effects accuracy of different algorithms and is not always beneficial. To reach our results we have used Python's Machine Learning Library Scikit-learn which is widely popular and to further validate our findings we have taken to completely different datasets from UCI Machine Learning repository.

General Terms
Classification, Machine Learning, Data Science, Python, Scaling, Random Forest Classifier, Decision Tree Classifier

Keywords
Hit Rate, KNN, Scikit

1. INTRODUCTION
Classification refers to assigning a particular category to a set of values based on training set of data. Classification is achieved by the use of various classification algorithms part of machine learning[1].

The various Applications of Classification [2]

- Pattern recognition
- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition: Different handwriting styles.
- Speech recognition: Temporal dependency.
- Medical diagnosis: From symptoms to illnesses
- Biometrics: Recognition/authentication using physical and/or behavioral characteristics: Face, iris, signature, etc

Selecting an algorithm is a crucial step as it decides the performance of our system. Mostly algorithms are judged on prediction accuracy which is the percentage of correct prediction divided by the total number of predictions. In this paper we are suggesting way to increase this accuracy[1].

Another important criteria while performing machine learning is the dataset we choose usually larger datasets give better results in comparison to smaller dataset because the algorithm is trained on this dataset and the more it can learn the better it can perform.

Here we compare various classification algorithms and measure the changes in the accuracy of algorithms after the introduction of scaling.

Here we have also introduced a new metric to compare classification algorithms Hit Rate. It would be explained later that where we can use it and how is it beneficial.

To make our results more accurate and foolproof we have made the use drastically different datasets that will be introduced further.

To implement this concept we have utilized the scikit-learn[3]which is machine learning for Python which has large number of machine learning algorithms that can be easily implemented. The reasons for choosing scikit[3] for this is that it's open source, widely used and the implementations of the machine learning algorithms are also quite stable.

Our aim is to suggest ways to improve the algorithms and comparing them based on hit rate rather than just accuracy so that we can move forward in the right direction.

2. THE ALGORITHMS USED HERE

2.1 K Nearest Neighbors
This algorithm can both be used for classification and regression. The K is the deciding factor. The accuracy of the algorithm can be greatly affected by varying the values of K such as larger usually reduces the noise but not a general rule and is vastly dependent on the dataset.

A good K can be selected by various heuristic methods.

The Nearest Neighbor Rule[4] forms the basis of this algorithm. It classifies an unclassified point by assigning it the value which is nearest to an already classified point

The algorithm does not require any pre-processing[5].The algorithm predicts the label based on previous training data assigns the classifying label of input data based on the observation of K nearest neighbors and assigning the majority label.

One of the main problem in nearest neighbor method is finding the closest point in high dimensional spaces as the method becomes impracticable for datasets having large dimensions[6].

2.2 Decision Tree Classifier

It's one of the approaches that utilizes the concept of trees for classification of various types of datasets and forms an important part of machine learning. A tree is constructed on the basis of training data and the tree has various internal nodes and leaf nodes[7].

Example of Decision Tree Classifier is shown in Fig 1.

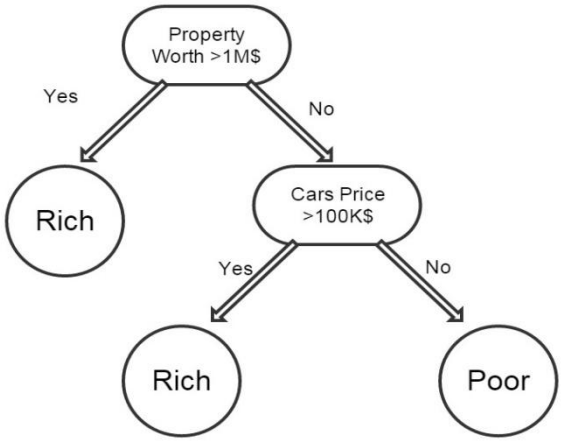

Fig 1: Application of Decision Tree Classifier to Classify People as Rich or Poor

Here each data to be classified is enters from the top and finally gets classified by passing through different levels and this tree is formed by the use of the training dataset.

Drawbacks of Decision Tree Classifier[8]

- In some cases it may waste a lot of memory space when the number of dimension is large
- Error gets added at each level and each level takes the prediction far away from the correct prediction

2.3 Random Forest Classifier

It is an ensemble learning method for classification by ensemble method we mean that uses more than one learning algorithms for better performance. It is based on the idea of random searching when making a decision to split a node for the growth of a tree.

It is one of the algorithms that uses combination of tree predictors and the trees generation is dependent on random values that are generated independently and this randomization factor is absent from most of the algorithms which acts in its favor most of the times. This can also used for regression[9].

Advantages of Random Forest Classifier[9]

- Better than other ensemble methods such bagging or boosting
- It is relatively robust to outlier and noise

Random Forest Classifier can also be applied to unsupervised learning which has been shown by

Tao Shi and Steve Horvath[10] which they applied it in the field of Biostatistics.

3. THE LIBRARY USED SCIKIT-LEARN

Scikit learn[3] is open source machine learning library licensed under the BSD license for Python

Programming. It is one the most popular machine learning library now days and is also vastly popular in the industry.

There is continuous development taking place to improve the implementations of the present machine learning algorithms which are constantly being added to the library which has only been possible because the contribution of large family of developers. This makes it easy to implement machine learning algorithms efficiently and easily and directly applying it to our data.

It does not only provide algorithms but also provides the supporting methods that are required when preparing our dataset. Therefore it acts as total package which makes the first choice for many developers and statisticians.

It also allows us to change the various variables related to an algorithm for advanced users and also providing amateurish users with default values which are calculated by using appropriate methods.

The various machine learning applications that we can perform are

- Classification
- Clustering
- Regression
- Model Selection
- Dimensionality Reduction

4. EXPERIMENT

4.1 Datasets Used

4.1.1 ADULT

The ADULT DATASET[11]used here is from UCI(University of California, Irvine) Machine Learning Repository. The task associated with this dataset is to predict whether the income exceeds 50k dollars per year or not which can easily be done with the help of Classification Algorithms.

It is a dataset that contains dimensions which can be continuous or discrete data. The dataset has training set of 30, 000 observations and testing or validation set of about 15000 observations which is enough to verify and compare the performance of the classification algorithms.

4.1.2 Wine Quality Dataset

The WINE DATASET[11]used here is from UCI(University of California, Irvine) Machine Learning Repository. For this experiment we have used only White wine dataset from the wine dataset which has less number of dimensions and less number of instances.

The task associated with this dataset is to predict the quality of wine which is divided into 10 levels ranging from 1-10 therefore it is not binary classification. All the values are in numeric format which makes our task easy.

4.2 Preprocessing of Data

The dataset cannot be used as given, it requires some pre-processing steps so that it become compatible with the algorithm that we are using to predict.

- First the numerical and non numerical values such as Country name are separated.
- Then these non numerical values must be converted into numerical values as for a machine learning algorithm to work we should have data in the form of numerical values.

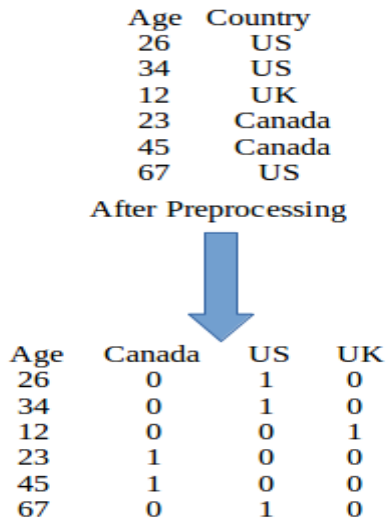

Fig 2: An Example showing how non numeric values are converted to numeric values

The only disadvantage of this method is that it increases the number of attributes by large amounts as in this case of dataset the dimensions went from about 14 to 102 almost 8-10 times which is huge chunk.

4.3 Scaling

All the algorithms were tested in their default implementation on both the datasets ADULT[11]and WINE[11].The following observations were made represented below in the form bar charts. The values for the accuracy of Random forest classifier was obtained by finding the average accuracy for around 50 runs.

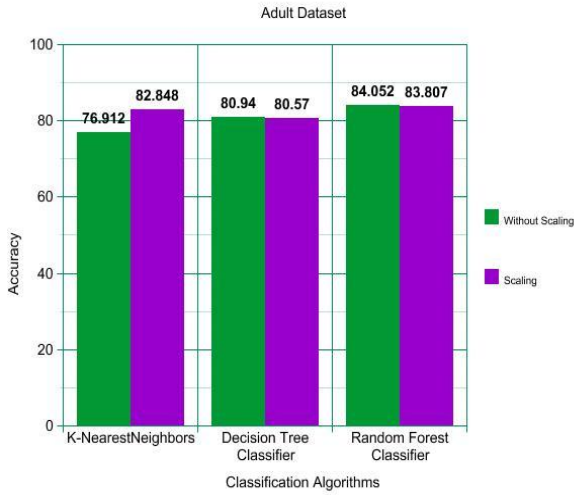

Fig 3: Comparing the performance with and without scaling on classification algorithms

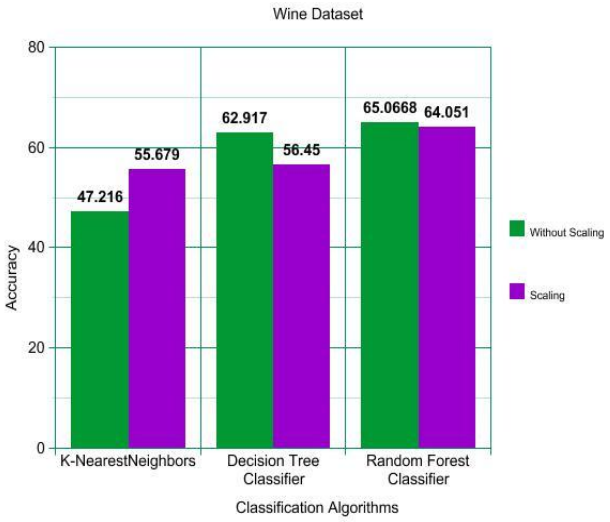

Fig 4: Comparing the performance with and without scaling on classification algorithms

We can clearly observe that scaling has an effect on each of the classification algorithms.

- For K-Nearest neighbors the accuracy increases greatly for both the datasets. Therefore, it has positive effect on it.
- The Decision Tree Classifier when run on ADULT Dataset[11]the accuracy is almost remains same but it is a little lower but when it is run on WINE Dataset[11]the accuracy significantly drops.
- On Random Forest Classifier scaling has a similar effect decreasing it's accuracy.

From this we can understand the statement that scaling always improves accuracy is vague and is not valid for every algorithm as scaling puts every feature on similar scales which makes comparison easier but makes it difficult to differentiate which maybe the reason for lower accuracy.

4.4 Evaluation based on Hit Rate

A new parameter Hit Rate gives the percentage about the common places at which two algorithms make mistakes.

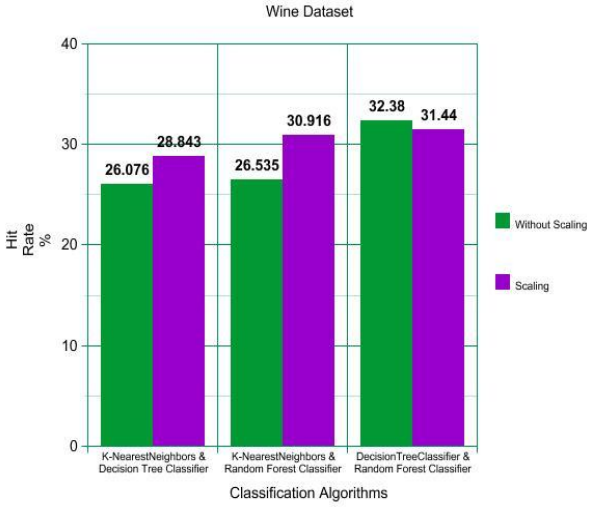

Fig 5: Hit Rate of Classification Algorithms

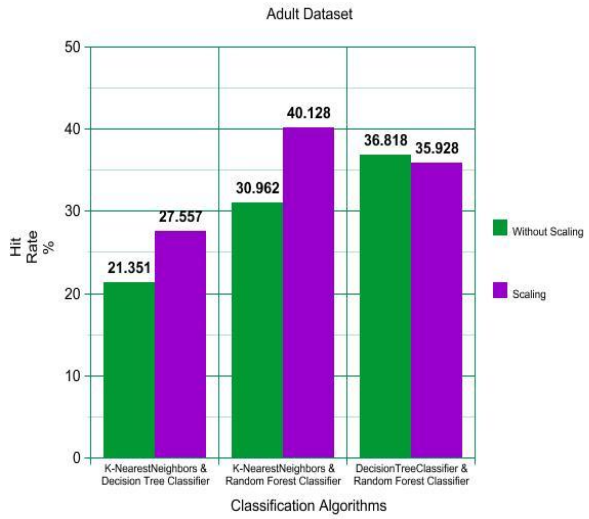

Fig 6: Hit Rate of Classification Algorithms

- Whenever K-Nearest Neighbor algorithms is compared with an another algorithm then the Hit Rate increases with scaling which is mainly because as K-Nearest Neighbor improves the accuracy therefore starts making errors at the same place as better algorithms with same efficiency do.
- The last column for both the datasets does not show any great variation in Hit Rate before scaling which can be understood by the fact scaling does not have much effect on both the algorithms.

Now understanding the Hit Rate we can see that as algorithms become closer accuracy they start making the same errors at the same places which increases the Hit Rate as in the case of K-Nearest Neighbors with both Decision Tree Classifier and Random Forest Classifier .The Hit Rate increases with scaling as with scaling the accuracy of K-Nearest Neighbor algorithms improves which in turn makes it closer in performance with other two. Similarly the other two algorithms when evaluated have similar Hit Rates before and after scaling because scaling does not have a major effect.

Let us take an example to understand the benefit of decreasing the hit rate even if the accuracy of the algorithm is low. If one of the algorithms has accuracy of 60 percent and the other algorithms has an accuracy of 40 percent but the Hit Rate for the two is about 0-5 percent therefore which signifies the fact that they are failing at different cases and therefore can overcome each other's weaknesses and achieve greater accuracy together which signifies the importance of Hit Rate. This becomes even more important in ensemble learning methods.

5. CONCLUSION

In this paper we have introduced a new metric Hit Rate which is an important metric to measure the performance of an algorithm along with the accuracy. Therefore we should involve Hit Rate when involving algorithms as its validity is tested on two completely different datasets.

The other point that we investigated is that Scaling is not always beneficial even if it may appear to be at first in some algorithms therefore it is subjective to each algorithm.

6. ACKNOWLEDGMENTS

I would like to thank UCI Machine Learning Repository for providing us with Datasets on which we could carry out our experiments. The datasets were easily available and were fully documented which further made the process easier.

7. REFERENCES

[1] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques,"Artificial Intelligence Review, vol. 26, no. 3, pp. 159–190,2006.
[2] E. Alpaydin, Introduction to machine learning, 2010.
[3] F. Pedregosa, G. Varoquax, A. Gramfort, V. Michel, B. Thirion,O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and M. Brucher, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[4] T. Cover and P. Hart, "Nearest neighbor pattern classification,"IEEE Transactions on Information Theory, vol. 13, no. 1, pp.21–27, 1967.
[5] J. M. Keller and M. R. Gray, "A fuzzy K-nearest neighbor algorithm,"IEEE transactions on systems, man, and cybernetics,vol. smc-15, no. 4, july/august 1985, no. 4, pp. 580–585, 1985.
[6] S. A. Nene and S. K. Nayar, "A Simple Algorithm for Nearest Neighbor Search in High Dimensions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 9, pp.989–1003, 1997.
[7] S. K. Murthy, "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey," (A2) Data Mining and Knowledge Discovery, vol. 2, pp. 345–389, 1998.
[8] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE Transactions on Systems, Manand Cybernetics, vol. 21, no. 3, pp. 660–674, 1991.
[9] L. Breiman, "Random forests," Machine Learning, vol. 45,no. 1, pp. 5–32, 2001.
[10] T. Shi and S. Horvath, "Unsupervised Learning With Random Forest Predictors," Journal of Computational and Graphical Statistics, vol. 15, no. 1, pp. 118–138, 2006.
[11] M. Lichman, "UCI machine learning repository," 2013.[Online]. Available: http://archive.ics.uci.edu/ml.