Bias variance trade off on maternal health risk dataset in K-NN and Decision tree algorithm

[1]Harshil Panchal, [2]Hanoonah Sheikh, [3]Shreyas Muchhal, [4]Siddharth Kabra

[1]Symbiosis International University, Pune, India, [2,4]Medi-Caps University, Indore, India,
[3]Institute of Management, Nirma University, Ahmedabad, India

***Abstract*: One of the most extensively used modelling strategies in the world of machine learning is classification algorithms like K- Nearest Neighbor (KNN) and DECISION TREE Algorithm. Machine learning models using classification algorithms are widely utilized in a variety of domains, including data analytics, image classification, computer vision, exploratory analysis, and game Artificial Intelligence, among others. In this case, the model must be extremely accurate, versatile, and efficient in order to successfully complete the work at hand. However, regardless of technique, the metrics used to assess a model's effectiveness are influenced by a variety of elements such as the confusion matrix, accuracy score, and so on. Among all these factors, the balance between bias and variance must be carefully maintained to optimize the model's performance. The KNN and DECISION TREE algorithms will be used to investigate bias and variance on the Maternal Health Risk Dataset in this research. Furthermore, the paper will focus on the regularization process and its impact on the balance of bias and variance, as well as how to deal with any inconsistencies that may develop owing to minor changes in dataset values. The benefits and drawbacks of variable bias and variance values, respectively, indicate the level of model adaptability on a dataset, regardless of how the training, testing, and validation data are divided.**

***Index Terms*: K-NN, Decision Tree, Variance, Machine Learning**

I. INTRODUCTION

The main goal of classification is to apply our model to the Maternal health risk dataset and deal with the target variable so that we may categorize risk level into three groups based on several independent variables present in our dataset and evaluate them thoroughly. The KNN and DECISION TREE algorithms are the two algorithms employed in this process. Furthermore, while working with datasets in general, there's a chance that some parameters in our machine learning model will produce classification inconsistencies. For example, if a carton of pebbles is given to a few people to guess the number of stones, practically everyone will come up with different numbers. This contradiction is handled in the domain of machine learning by maintaining a good balance between variance and bias. Furthermore, the mean or average of the output data is used to generalize the ultimate judgement. As a result, we have a more efficient and accurate analysis.

KNN AND DECISION TREE

In order to summarize risk level of into three groups based on responsible factors for maternal mortality, we are provided a dataset that is highly comparable. As a result, we proceed with the KNN classification algorithm. The k-nearest neighbors (KNN) algorithm is a data classification approach that estimates the likelihood that a data point will belong to one of two groups based on the data points closest to it. The supervised machine learning algorithm k-nearest neighbor is used to address classification and regression problems. It is, however, mostly employed to solve categorization difficulties. KNN is a non-parametric, slow learning method, because it doesn't perform any training when you submit the training data, it's known as a lazy learning algorithm or lazy learner. Instead, it just saves the data and does not execute any calculations throughout the training period. It doesn't start building a model until the dataset is queried. On the other hand, we deal with bias and variance by applying the decision tree, which in turn plays a critical part in categorizing the training set by creating a legitimate decision tree to improve our prediction accuracy. The ID3, CART, CHID, Hunt's Decision Tree, C4.5, C5.0, and many other forms of decision trees are used for classification. The core procedure of the decision tree algorithm is based on the Sum of Product (Sop) or Disjunctive Normal Form approach, which involves starting at the root node and following a specified path that is reliant on the values of attributes (and distributed recursions). It eventually leads to terminal leaf nodes, completing the categorization. When it comes to prediction and classification error, numerous parameters inside the machine learning model can be tweaked to lessen it. One such object is bias, which is used to improve the model's efficiency by analyzing the difference between the model's exact value and the average predicted value. Higher bias values speed up and make learning easier for a machine learning algorithm, but they also make it more rigorous. When compared to logistic regression, KNN algorithms have a comparatively higher bias value. Variance can be defined in the context of machine learning as the measure of the divergence in values of the target variable if the training data changes. Simply said, variance is the variation in the model's predictions when the dataset changes. The variance of the decision tree algorithm is often considerable (even higher in non-pruned tree). Furthermore, while large variance allows the algorithm to be more flexible, it also increases the influence of the dataset. In general, overfitting occurs when there is a low bias and a high variance. That is to say, the machine learning model considers both the noise and the patterns in the dataset. Because of the intricacies in their structure, decision trees are more prone to data overfitting. When there is a large bias and little variance, however, under-fitting occurs. This indicates that the machine learning model is unable to extract patterns from the data provided. When there is a very small amount of data to train our model with, or when we try to fit non-linear data into a linear model, under-fitting occurs. A machine learning model should ideally be neither overfit nor under-fit. To make the machine learning model more efficient, a healthy balance

between bias and variance must be maintained. The end goal is to reduce overall error and model complexity for efficient and faster prediction and classification, which is known as the bias variance trade-off.

Figure 1- KNN representation

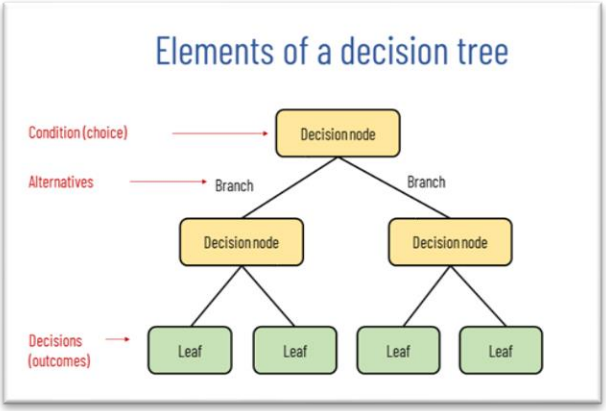

Figure 2 : Flow chart of Decision Tree Algorithm

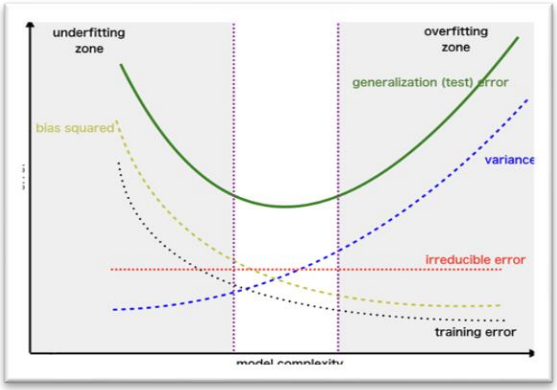

Figure 3 : Variance Bias Tradeoff

II. DATASET

The dataset is obtained from the UCI Machine Learning Repository. Data has been collected by Marzia Ahmed (Daffodil International University, Dhaka, Bangladesh), from different hospitals, community clinics, maternal health cares from the rural areas of Bangladesh through the IoT based risk monitoring system. There are 1014 occurrences in the array with 7 attributes. Continuous and category attributes are included in the dataset.

Risk Level: Predicted Risk Intensity Level during pregnancy considering the previous attribute.

The dataset has seven columns that need to be pre-processed in order to use a machine learning model because they contain missing values and non-integer values. Scikit-Learn is used to do label encoding and one-hot encoding on non-integer values, while simple imputer (with "mean" as a parameter) is used on missing-value data points.

The KNN model was applied on the dataset with varying numbers of test size - 20%, 25%, 30%, 35%, and 40%. It was observed that the accuracy score was almost the same throughout with minute fluctuations. Furthermore, a bar chart was plotted with the help of matplotlib featuring the decimal places of the accuracy score (mean of the two obtained values) on Y axis and the test size on the X axis. The mean of overall accuracy scores was 67.32%.

Decision tree classification was implemented on the same proportions of test and training data as in the k-NN classification. Moreover, the bar graph depicts the comparison of 5 values of test percentages. The mean of overall accuracy scores is 79.21%.

Test Size	Accuracy (k-NN)
20%	64.03%
25%	65.74%
30%	68.85%
35%	71.26%
40%	66.74%

Test Size	Accuracy (Decision Tree)
20%	77.83%
25%	77.55%
30%	80.98%
35%	81.12%
40%	78.57%

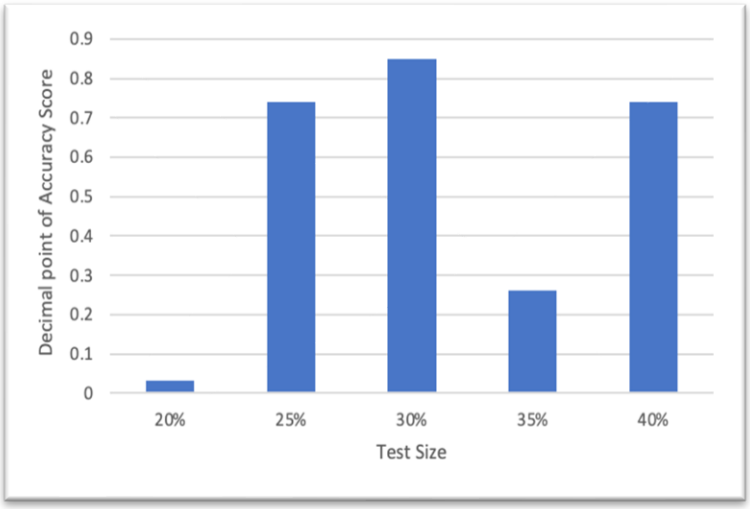

Figure 4 - KNN accuracy plot

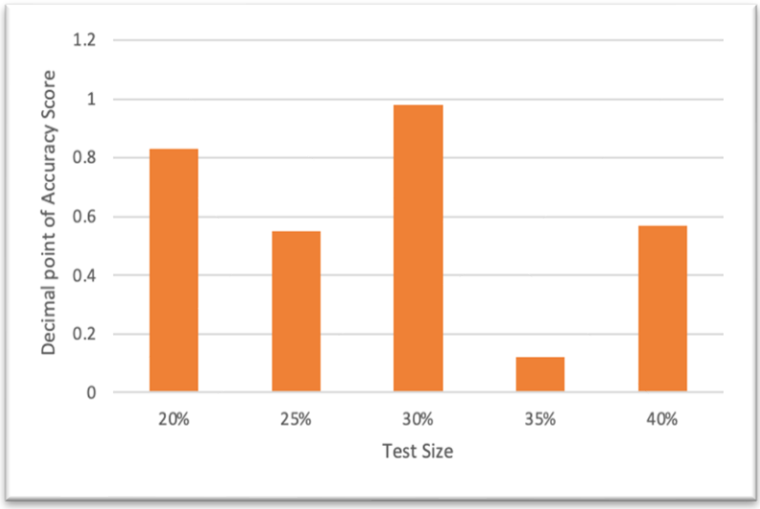

Figure 5 - Decision Tree accuracy plot

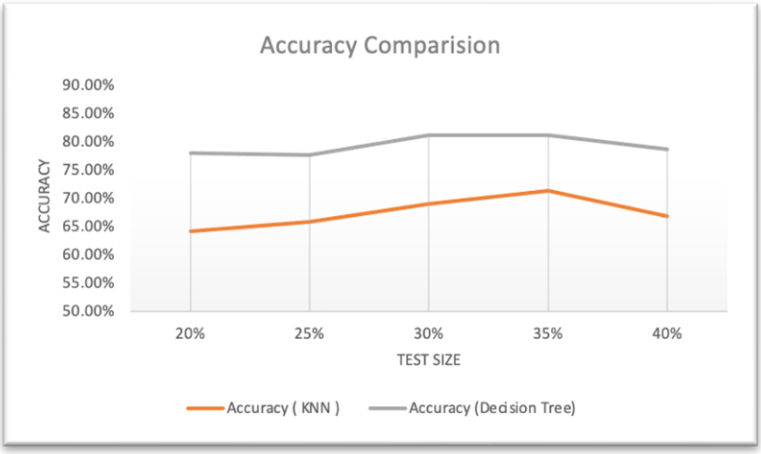

Figure 6 - Decision Tree and KNN Accuracy Comparison

III. Validation and Learning Curve

Validation curve and learning curve are often used to examine the generalization of the model on fluctuating test datasets. Validation curve visualizes the performance over many values for a range of hyper-parameters. On the other hand, the learning curve is used to determine the effect of the number of parameters used for efficiency of model and performance metric. The learning curves of KNN classification model is depicted below. The curve is based upon 50 different sizes of training set.

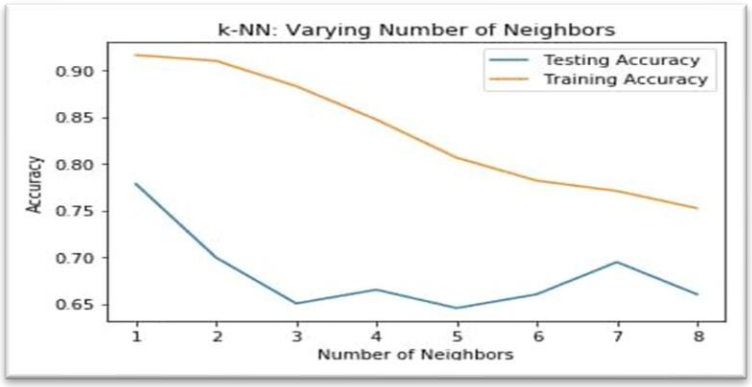

Figure 7 - K-NN Learning Curve

IV. Inference

The logical inference from the two separate models is that both might have been enhanced and made more effective if the dataset had more data points. Moreover, by utilizing additional machine learning techniques, the accuracy ratings can be raised even higher. Additionally, it can be deduced that the accuracy ratings fluctuate with the change in dataset but do so less dramatically.

V. Conclusion

We draw the result that the decision tree has higher testing accuracy than KNN by applying the decision tree model and KNN model to understand the trade-off between bias and variance. While KNN scans the entire dataset to make predictions because it does not generalize the data in advance, decision trees are faster with huge datasets.

Aside from accuracy scores, we may draw the conclusion that the bias variance trade-off in this dataset cannot be eliminated, but that a "sweet spot" can be located after examining the parameters such that the machine learning model produces the best results.

References

1. Pedro Domingos, "*A Unified Bias-Variance Decomposition and its Applications*," *Proceedings of 17th International Conference on Machine Learning*, pp. 231-238, June 2000.
2. Sisay Menji Bekena, "*Using decision tree classifier to predict income levels*," *Munich Personal RePEc Archive*, paper no. 83406, December 2017.
3. Navoneel Chakraborty, Sanket Biswas, "*A Statistical Approach to Adult Census Income Level Prediction*," *arxiv.org*, *arXiv:1810.10076 [cs.LG]*, version 1, October 2018.
4. S. Deepajothi, Dr. S. Selvarajan, "*A Comparative Study of Classification Techniques On Adult Data Set*," *IJERT*, vol. 1, issue 8, Oct. 2012.
5. Lichman M., "*Adult Income Dataset*," *University of California, Irvine*, 2013