Cross-validation of Machine Learning approach with implementation of Random Forest Classifier Model using Python

Subhasis Mohapatra
Nalanda Institute of Technology, Bhubaneswar, Odisha ,India
Email Id: subhasis22@gmail.com

Aloka Natha
Nalanda Institute of Technology, Bhubaneswar, Odisha ,India
Email Id: aloknath23@gmail.com

Nishanta Ranjan Nanda
Tata Consultancy Services, Bhubaneswar, Odisha, India
Email Id: nishanta.nanda@gmail.com

Abstract: *This paper presents a comprehensive exploration of the cross-validation technique in the context of machine learning, with a focus on the implementation of the Random Forest Classifier model using Python. Cross-validation is a crucial method for assessing the robustness and general ability of a machine-learning model. It involves partitioning the data into subsets, training the model on these subsets, and then testing it on the complementary subset to validate the model's predictions. This study aims to serve as a valuable resource for researchers and practitioners in the field of machine learning, offering detailed guidance on implementing robust machine-learning models using cross-validation techniques and Python programming. The need for cross validating machine learning models is extremely important as we implement it for the problem-solving purpose. Usually in data science assumption is to go through various models to find a better ML model. However, it becomes difficult to find distinction whether this improvement in score is visible because we are capturing the relationship in better approach or we are just over fitting the input data. This model helps us to achieve more generalized relationships and find suitable model for the problem solving. Experimental results underscore the effectiveness of the Random Forest Classifier in various scenarios, providing insights into its performance across different cross-validation schemes. The analysis helps in understanding how cross-validation can be used to fine-tune model parameters and prevent issues like overfitting, thereby enhancing the predictive accuracy of the model.*

Keywords: *cross-validation, over fitting, logistic regression, random forest classifier, decision tree*

Introduction

The Random Forest Classifier is one of the most popular and commonly used algorithms by Data Scientists to find out the most suitable dataset. Random forest comes under the *Supervised* Machine Learning Algorithm which is *widely used in the Classification and Regression problems*. On various samples, it constructs decision trees and uses their average for classification and majority vote for regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set into two ways that is

containing *continuous variables,* as in the case of regression, and *categorical variables,* as in the case of classification. It performs better for classification and regression tasks.

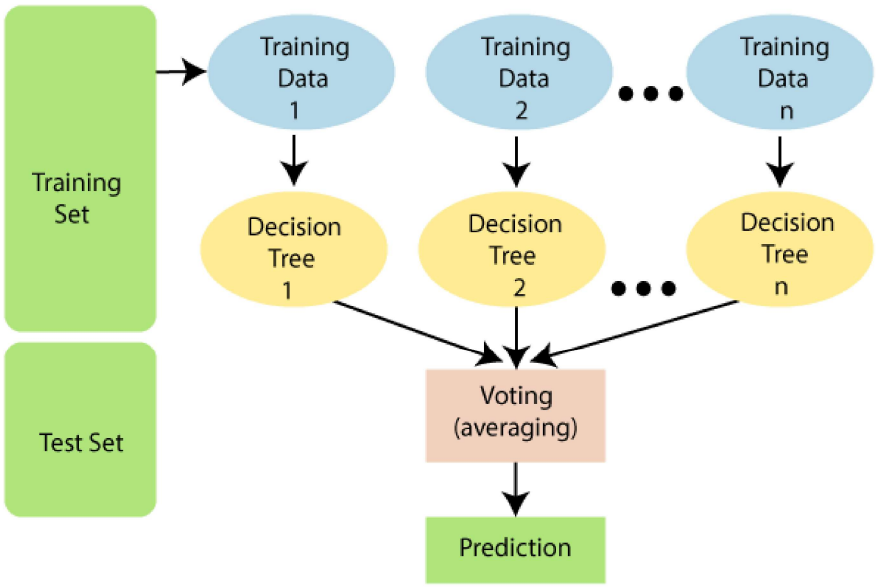

Figure 1: Random Forest Classifier mechanism

Source: Primary data

In the above diagram [19], it depicts that Random Forest approach takes a large dataset then breaks it into several decision trees according to their training data sets. Then applying decision trees techniques, it works on and it aggregates the result of each decision trees as an average and that average result is to be taken as the predicted values or results. It provides the best accuracy, as it takes the average of multiple numbers of decision trees.

Logistic Regression

A supervised machine learning approach known as logistic regression is used mostly for classification issues where the goal is to predict the likelihood that a given instance will belong to a certain predefined class or not. It is considered as a statistical algorithm, which analyses the relationship between a set of independent variables and the dependent variables. It is a powerful tool for decision-making. For example, an email is a spam or not. It is used for classification algorithms which is known as logistic regression. It's referred to as a regression because it takes the output of the linear regression function as input and uses a sigmoid function to calculates the probability for the given class. The difference between linear regression and logistic regression is that in linear regression, the output is a continuous value that might be anything while logistic regression forecasts the likelihood that a particular instance will belong to a specific class or not.

Terminologies involved in Logistic Regression:

- **Independent variables:** The dependent variable's predictions were based on the input characteristics or predictor factors.

- **Dependent variable:** We are attempting to predict the target variable in a logistic regression model.
- **Logistic function:** The equation illustrating the relationship between independent and dependent variables. The dependent variable's likelihood of being 1 or 0 is represented by a probability value between 0 and 1, which is the result of the logistic function, which transforms the input variables.
- **Odds:** It is the proportion of something happening to nothing happening. It differs from probability since probability measures the likelihood of an event happening in relation to all possible outcomes.
- **Log-odds:** The natural logarithm of the chances is the log-odds, sometimes referred to as the logit function. As a linear combination of the independent factors and the intercept, the log chances of the dependent variable are modelled in logistic regression.
- **Coefficient:** The calculated parameters of the logistic regression model demonstrate the relationship between the independent and dependent variables.
- **Intercept:** In the logistic regression model, a constant component that represents the log odds when all independent variables are equal to zero is called the log odds.
- **Maximum likelihood estimation:** The procedure for estimating the logistic regression model's coefficients that maximises the likelihood of actually seeing the data given the model.

Decision Tree:

A type of supervised machine learning known as decision trees constantly divides the data based on a particular parameter. The tree can be explained by two things, namely decision nodes and leaves. The leaves are defined as the decisions or the final outcomes and the decision nodes are where the data is split.

Source: Author

An example of a decision tree can be explained using above hierarchical structure. Let's imagine you want to determine a person's level of fitness based on their age, dietary habits, level of physical activity, etc. The decision nodes here are questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas? And the leaves, which are outcomes like either 'fit', or

'unfit'. This particular classification issue was of the yes-or-no variety. Decision trees can be divided into two categories:

1. Classification trees (Yes/No types)

In the above figure 2 of classification tree, the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is Categorical.

2. Regression trees (Continuous data types)

Here the decision or the outcome variable is Continuous, e.g., a number like 673. There are many algorithms which are used to construct the Decision Trees, in which the nodes are the values of the conditions.

Literature Review

The Random Forest is appropriate for high dimensional data modelling because it can handle missing values and can handle continuous, categorical and binary data. The bootstrapping and ensemble scheme makes Random Forest strong enough to overcome the problems of over fitting and hence there is no need to prune the trees. Besides high prediction accuracy, Random Forest is efficient, interpretable and non-parametric for various types of datasets [18]. The model interpretability and prediction accuracy provided by Random Forest is very unique among popular machine learning methods. Accurate predictions and better generalizations are achieved due to utilization of ensemble strategies and random sampling.

The idea of Random Forests which perform well as compared with other classifiers including Support Vector Machines, Neural Networks and Discriminant Analysis, and overcomes the over fitting problem.

Those methods such as Bagging or Random subspaces [5,6] which are made from ensemble of various classifiers and those which use randomization for producing diversity have proven to be very efficient. In order to introduce diversity and to build classifiers different from each other, they use randomization in the induction process. Random Forests have gained a substantial interest in machine learning because of its efficient discriminative classification [7, 8].

In computer vision community, Random Forests were introduced by Lepetit et. al. [9, 10]. His work in this field provided a foundation for papers such as class recognition [11, 12], bilayer video segmentation [13], image classification [14] and person identification [15], which use Random Forests. A wide range of visual cues are also enabled naturally by the Random Forest including colour, shape, texture and depth. Random Forests are considered general purpose vision tools and considered as efficient.

Random Forest as defined [4] as a generic principle of classifier combination that uses L tree-structured base classifiers $\{h(X, r_n), N=1,2, 3\ldots L\}$, where X denotes the input data and $\{r_n\}$ is a family of identical and dependent distributed random vectors.

In a Random Forest, the features are randomly selected in each decision split. The correlation between trees is reduces by randomly selecting the features which improves the prediction power and results in higher efficiency. As such the advantages of Random Forest are [16]:

1. Overcoming the problem of over fitting
2. In training data, they are less sensitive to outlier data
3. Parameters can be set easily and therefore, eliminates the need for pruning the trees.
4. Variable importance and accuracy are generated automatically.

The Random Forest is appropriate for high dimensional data modelling because it can handle missing values and can handle continuous, categorical and binary data. The bootstrapping and ensemble scheme makes Random Forest strong enough to overcome the problems of over fitting and hence there is no need to prune the trees. Besides high prediction accuracy, Random Forest is efficient, interpretable and non-parametric for various types of datasets [17]. The model interpretability and prediction accuracy

provided by Random Forest is very unique among popular machine learning methods. Accurate predictions and better generalizations are achieved due to utilization of ensemble strategies and random sampling.

Random Forest developed by Leo Breiman [4] is a group of un-pruned classification or regression trees made from the random selection of samples of the training data. Random features are selected in the induction process. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble.

Random Forests Converge

Given an ensemble of classifiers h1(x),h2(x), ...,hK (x), and with the training set drawn at random from the distribution of the random vector Y, X, define the margin function as mg (X, Y) = avkI (hk (X)=Y) "max j `"Y avkI (hk (X)= j). where I (•) is the indicator function. The margin measures the extent to which the average number of votes at X, Y for the right class exceeds the average vote for any other class. The larger the margin, the more confidence in the classification. The generalization error is given by PE* = PX, Y (mg (X, Y) < 0) where the subscripts X, Y indicate that the probability is over the X, Y space. In random forests, hk (X) = h (X,Èk). For a large number of trees, it follows from the Strong Law of Large Numbers and the tree structure that: As the number of trees increases, for almost surely all sequences È1 ... PE* converges to PX, Y (PÈ (h (X, È) =Y) "max j`"Y PÈ (h (X, È) = j) < 0)

Strength and Correlation For random forests, an upper bound can be derived for the generalization error in terms of two parameters that are measures of how accurate the individual classifiers are and of the dependence between them. The interplay between these two gives the foundation for understanding the workings of random forests. We build on the analysis in Amit and Geman [18].

Using Random Features

Some random forests reported in the literature have consistently lower generalization error than others. For instance, random split selection (Dieterrich [2]) does better than bagging. Breiman's introduction of random noise into the outputs (Breiman [4]) also does better. But none of these three forests do as well as Adaboost (Freund and Schapire [1]) or other algorithms that work by adaptive reweighting (arcing) of the training set (see Breiman [4], Dieterrich [2], Bauer and Kohavi [3]). To improve accuracy, the randomness injected has to minimize the correlation ñ while maintaining strength. The forests studied here consist of using randomly selected inputs or combinations of inputs at each node to grow each tree. The resulting forests give accuracy that compare favourably with Adaboost. This class of procedures has desirable characteristics:

i) Its accuracy is as good as Adaboost and sometimes better.

ii) It's relatively robust to outliers and noise.

iii) It's faster than bagging or boosting.

iv) It gives useful internal estimates of error, strength, correlation and variable importance.

v) It's simple and easily parallelized. 8 Amit and Geman [18] grew shallow trees for handwritten character recognition using random selection from a large number of geometrically defined features to define the split at each node. Although my implementation is different and not problem specific, it was their work that provided the start for my ideas.

Proposed work

Now-a-days, in the banking and insurance sector, credit risk is one of the most ongoing issues in any lending institution. It was the ultimate goal of these types of institutions to reduce credit risk even if it was with a small margin. So, we need some machine learning techniques for improving credit scoring methods. In addition, it was becoming very difficult to rely on traditional methods of credit scoring which was given by the influx of invisible clients who barely fit into traditional consumer groups and were easily considered as miss-classified by the traditional

scoring methods, which also misleads to the observation that the risk factor is very high. Given this type of challenges, problems and difficulties, this is a knowledge-based concept that describes how a lending institution can anchorage on the power of machine learning into predicting clients credit ratings. These credit ratings are much more useful for the future enhancements. In this article, we shall use several classification models to predict the likelihood of a customer defaulting on a loan based on past data. The outcome variable is a binary variable have good and bad as the possible outcomes. The feature used include, latitude, longitude, bank branch, employment status, level of education and variables relating to past loan history of the client.

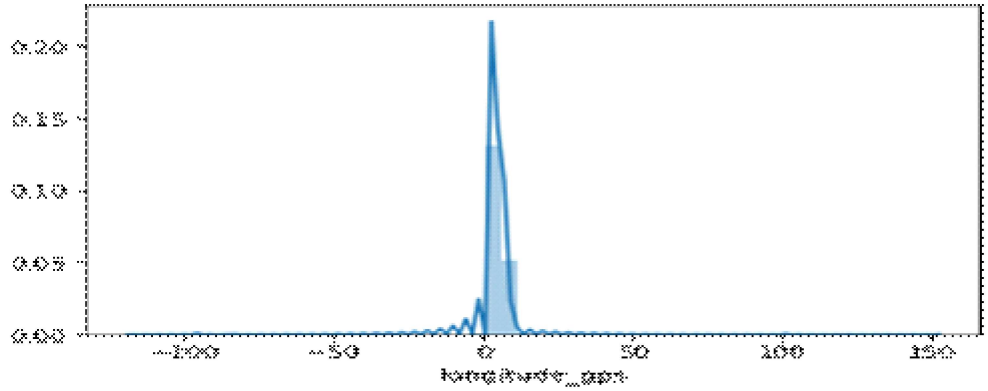

Figure 3: Box Plot Reveals Longitude variables

Source: Author

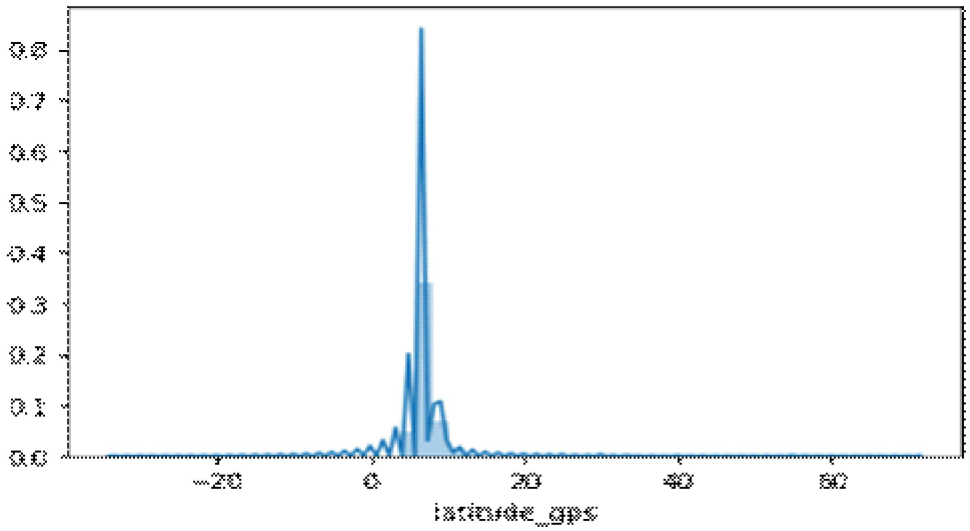

Figure 4: Box Plot Reveals Latitude Variables

Source: Author

In the above two figures (figure 3 and 4), it depicts that the data using box-plots reveals that longitude and latitude variables are highly affected by extreme values which are the most frequently used ones.

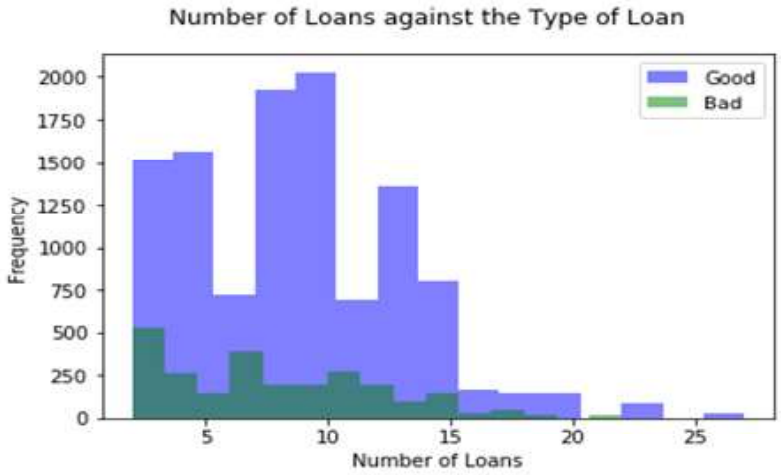

Figure 5: No of Loans against the type of loan

Source: Author

In the above figure (figure 5), it depicts that, the number of loans and the frequency that means it provides the category of good loans and bad loans. It shows the number of variations of different loans against the frequency, that means frequent number of people having different types of loans in different fields.

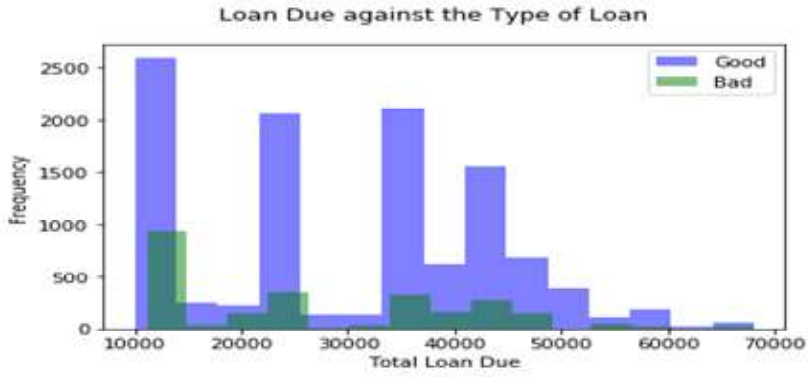

Figure 6: Loan due against the type of Loan

Source: Author

The above-mentioned figure (figure 6), it provides the ratio of total loan due against different types of frequency. Number of people having different types of loans and they are having loan due which they have to pay.

Random Forrest with Cross Validation

irrelevant variables dropped, a cross-validation is used to measure the optimum performance of the random forest model. An average score of 0.923 is obtained.

```
#Random Forest After Removing Redundant variables
score = cross_val_score(ensemble.RandomForestClassifier(random_state= 42),
                        features_n, labels_n, cv= kf, scoring="accuracy")
print(f'Scores for each fold are: {score}')
print(f'Average score: {"{:.3f}".format(score.mean())}')
```

```
Scores for each fold are: [0.9211391  0.91858342 0.92040891 0.92658875 0.
9284149 ]
Average score: 0.923
```

Figure 7: Random Forest with Cross Validation

Source: Author

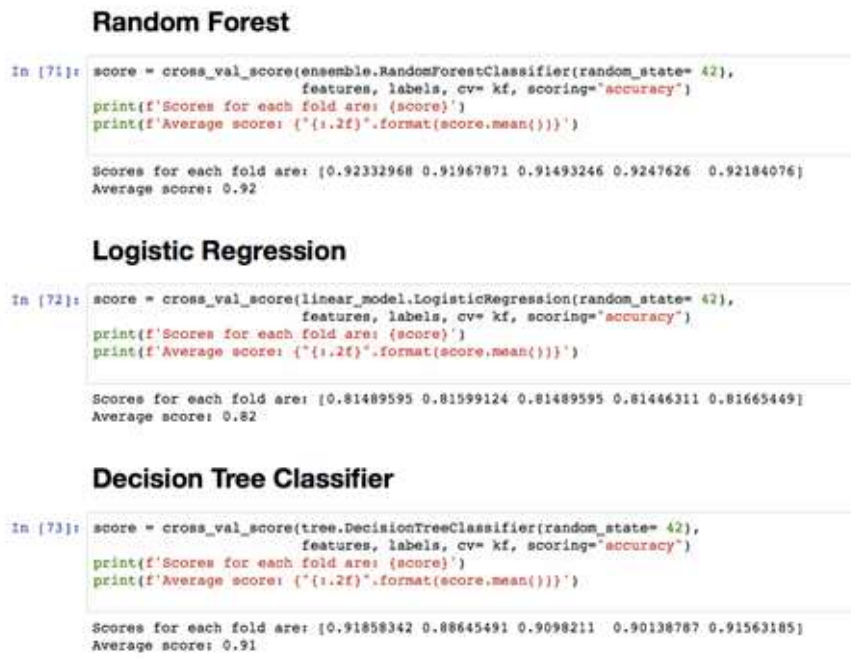

Figure 8

Source: Author

In random forest, Scores for each fold are: [0.92332968 0.91967871 0.91493246 0.9247626 0.92184076] Average score: 0.92

In logistic regression, Scores for each fold are: [0.81489595 0.81599124 0.81489595 0.81446311 0.81665449] Average score: 0.82

In decision tree classifier, Scores for each fold are: [0.91858342 0.88645491 0.9098211 0.90138787 0.91563185] Average score: 0.91

Now drop variables and rerun cross validation and the classifiers, we got scores for each fold are: [0.9211391 0.91858342 0.92040891 0.92658875 0.9284149] average score: 0.923 and the Best Model Accuracy: 0.9288061336254108

Observations

Cross validation is applied to compare and select the best model. Three models are used with cross validation, that is, Random Forest, Logistic

Regression and Decision Trees. Random Forest has the best average score of 0.92 and is selected for building the final model.

Conclusion

The proposed work has a better accuracy for the Random Forest Classifier which provides the credit risk analysis. Cross validation was taken place for showing the better performance over the data. For example, the decision to use cross-validation is adopted after the model over-fitted.

References

1. Freund, Y. and Schapire, R. [1996] Experiments with a new boosting algorithm, Machine Learning: Proceedings of the Thirteenth International Conference, pp. 148-156
2. Dietterich, T. [1998] An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting and Randomization, Machine Learning 1-22
3. Bauer, E. and Kohavi, R. [1999] An Empirical Comparison of Voting Classification Algorithms, Machine Learning, 36, No. 1/2, 105-139.
4. Breiman, L., Random Forests, Machine Learning 45(1), 5-32, 2001.
7. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Computation 9(7), 1545–1588 (1997)
8. Breiman, L.: Random Forests. ML Journal 45(1), 5–32 (2001)
9. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. IEEE Trans. Pattern Anal. Mach. Intell. 28(9), 1465–1479 (2006)
10. Ozuysal, M., Fua, P., Lepetit, V.: Fast keypoint recognition in ten lines of code. In: IEEE CVPR (2007)
11. Winn, J., Criminisi, A.: Object class recognition at a glance. In: IEEE CVPR, video track (2006)
12. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: IEEE CVPR, Anchorage (2008)
13. Yin, P., Criminisi, A., Winn, J.M., Essa, I.A.: Tree-based classifiers for bilayer video segmentation. In: CVPR (2007)
14. Bosh, A., Zisserman, A., Munoz, X.: Image classification using Random Forests and ferns. In: IEEE ICCV (2007)
15. Apostolof, N., Zisserman, A.: Who are you? - real-time person identification. In: BMVC (2007).
16. Introduction to Decision Trees and Random Forests, NedHorning: American Museum of Natural History'sHorning; American Museum of Natural History's
17. Yanjun Qi., "Random Forest for Bioinformatics". www.cs.cmu.edu/~qyj/papersA08/11-rfbook.pdf
18. Amit, Y. and Geman, D. [1997] Shape quantization and recognition with randomized trees, Neural Computation 9, 1545-1588
19. https://www.javatpoint.com/machine-learning-random-forest-algorithm