Interpretable Random Forest

Sandeep Tandra
Department of computer science
University of Regina
sandeepreddy0987@gmail.com

***Abstract*—In this paper we try to explain Random forest model with some changes in feature selection technique for some business problems (discussed in the problem statement). The probabilistic method chosen to select each feature at every node along with the normal feature selection criteria will give us an Interpretable Random forest model which will be compared to the general random forest model that gives an idea of how feature selection impacts the output from a model.**

***Index Terms*—Ensemble Method, Supervised learning, Decision trees, ANFIS**

I. Introduction

Today's modern business applications make use of various machine learning algorithms for various predictions such as whether a particular user will buy a certain product or if a business is likely to make a profit in the next quarter and so on. These kinds of real-world scenarios require the use of classification and regression algorithms to precisely predict the outcome desired. This classification is necessary to know the target class of an observation and then making decisions accordingly. Ensemble learning is a technique where we join the outputs of different types of algorithms or the same type of algorithm is joined multiple times to form a powerful prediction model. This technique is mainly used on Supervised algorithms that explains what the input is and what the corresponding output is in the training data.

"Random Forest Algorithm"[1] is a supervised algorithm that is based on Ensemble learning. This algorithm uses bagging technique to select random input data also called Row sampling and also column sampling to select features randomly to each base learner and these base learners are aggregated to predict the result[2][3][4]. Decision trees are used as base learners resulting in a forest of trees that has random input data hence the name is given as "Random Forest". This can be used for both classification and regression problems but mainly used for classification problems.

Classification and Regression Trees(CART)- In order to create a model which predicts the value of the target based on the input, decision trees are used in machine learning. Classification and regression decision tree is one of the best machine learning techniques. In non technical terms, CART algorithm is nothing but a sequence of questions that will help in predicting what the next question should be. The result of this will be a tree like structure which ends at a point where there will be no further questions. This algorithm works by continuously searching for the best predictor variable by splitting the data into two subsequent subsets.

A Decision tree is a type of Supervised Machine Learning model where the data samples are continuously split into nodes until the node becomes pure. These are the building blocks of the Random forest algorithm. Once the decision trees are created on the data samples, this algorithm gets the prediction from each of them and finally selects the best solution with the help of the majority vote or mean. So basically multiple decision trees are built and then merged to get a more accurate prediction.

While growing the decision trees, random forest adds randomness to the model. It searches for the best feature instead of the most important feature among the random subset of features while splitting a node. This results in a wide diversity resulting in a better model. Random Forest Algorithm makes it

very easy to measure the relative importance of each feature on the prediction[5]. Another advantage of this algorithm is its stability. Even when a new data point is added into the dataset, the overall algorithm is not affected. This is because this new data might impact one tree but cannot impact all the trees. Because of all its advantages, the Random forest algorithm is widely used in varieties of applications such as banking, medicine, stock market, E-commerce. With its simplicity and diversity, Random forest algorithm will explore many unexplored areas and will find solutions to many of the unsolved machine learning problems.

II. Background

The initial work of Amit and Geman in 1997 on geometric feature selection, Ho's(1998) random subspace method and, Dietterich's(2000) random split selection approach influenced Breiman to come up with an idea of random forest model[6]. At some stage, Random forest model became the competitor for state-of-the-art models like Boosting and SVM. They became competitive because of their fast computation, highly accurate results, easy to implement, can handle large data with missing values.

Let's point down a few things to be considered in the model implementation.

- Row sampling- input samples should be drawn randomly which is a random subset of a population for each base learner.
- Column sampling- input features are randomly selected and given to each base learner. Usually round($\sqrt{M}$) features are given to each decision tree for increasing diversity among the trees and reduce computational load.
- Splitting a node in a decision tree- In each decision tree, nodes are decided using some criteria. Entropy, Gini impurity, Information gain are the most frequently used criteria in most of the libraries. For example, sci-kit learn library uses Gini impurity as default criteria to select nodes because of computational speed.
- Number of base learners- we use techniques like grid search to decide the number of decision trees are based on the evaluation metric.
- Aggregation – the outcomes of all the base learners are combined to give the final output. For classification problems we use majority vote and for regression we use mean or median.

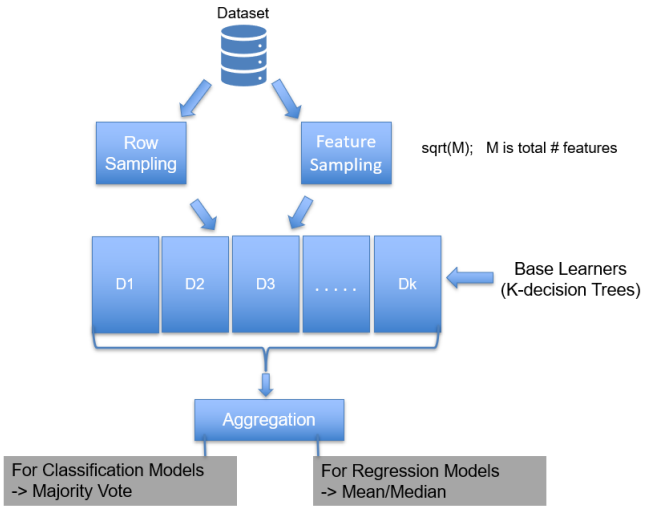

Fig. 1: Random Forest Model

A. Random Forest Algorithm

Lets consider a training dataset S,

$$S = \{(x_i, y_i)_{i=1}^{N} | x_i \in \mathbf{R}^M, y_i \in (1, 2, 3, ...c)\} \quad (1)$$

where x_i is input features, y_i is a class response or target label feature, N is the number of training samples, and M is the number of features and Algorithm 1 is used to illustrate a random forest model RF, given input x let y^k be the output of tree T^k. The final prediction of random forest with Q trees is y.

$$y = majorityvote\{y^k\}_1^k \quad (2)$$

where k is the number of base learners(trees).

In the RF model Bagging method[7] is used to select the sample of data so each tree is built using a subset of a dataset S (up to 75% of the population sample). This subset of the sample is called in-bag samples. And remaining sample in each tree is called the OOB (out-of-bag) sample which will be used to cross-validate the trained model i.e to estimate the training error.

B. Building decision trees

Each base learner is given random set of samples and features. So, there should be a criteria to place each feature at the corresponding node (parent

Algorithm 1: Random forest model

Data: $S = \{(x_i, y_i)^N_{i=1} — x_i \in \mathbf{R}^M, y_i \in (1, 2, 3, ...c)\}$
Result: Random forest model
initialization;
`Def Model(Data):`
for *each tree in [1,2,....,k]* **do**
  `select` S_n `samples (subset of S) randomly`
  `select m features randomly`
  `tree_list t = BUILD_TREE(`S_n`,m)`
end
`Result=Aggregation(t)`
`Def BUILD_TREE(`S_n`,m):`
for *i in range 1 to m* **do**
  `Calculate` $Gini_{split}$ `for all the unselected features at each node -> (i)`
  `Choose the feature with lowest value from (i).`
  `Selected feature at node r is divided into child nodes until the nodes becomes pure`
end
`return tree`

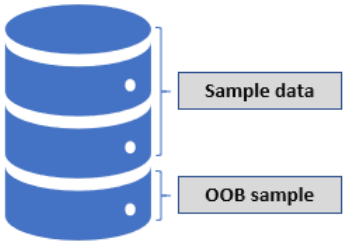

Fig. 2: sampling of dataset for each tree

node/internal nodes). Breiman(1996)[8] introduced a permutation method, called an out-of bag significance ranking[9], to calculate the importance of features in prediction. The basic idea for calculating this kind of significant score of features is to measure the difference between the original mean error in OOB samples and the randomly permuted mean error. The method stochastically rearranges for each tree all values of the jth function in OOB and uses the RF model to estimate this permuted function and get the mean error. The aim of this permutation is to remove the current association between the jth function and Y values and then check the effect of this on the RF model. If the mean error decreases dramatically then the feature is in a strong association. After this, many feature selection methods are used while the nodes split. The most used one is Information gain which is calculated using entropy. This idea was used from Shannon's entropy in information theory[10]. Entropy measures the uncertainty of a random variable X. It can be calculated using:

$$H(X) = -\sum_{i=1}^{C} P_i log_2 P_i \tag{3}$$

where C is the number of classes and p_i is the probability of randomly picking an element of class i.

But there is an alternate to entropy which is called Gini impurity[11]. It measures the probability of uncertainty in identifying a class. An attribute is considered for a better split if it has lower impurity. For every node r in a decision tree, the node

impurity is calculated as I(r). I(r) is defined as

$$I(r) = 1 - \sum_{i=1}^{C} P_i^2 \quad (4)$$

where Pi is the relative frequency of class i in r

When the node r is split into two child nodes r1, r2 with size S1(r) and S2(r). The Gini index of the split can be defined as

$$Gini_{split}(r) = \frac{S1(r)}{S(r)} * I(r) + \frac{S2(r)}{S(r)} * I(r) \quad (5)$$

Attribute with the smallest $Gini_{split}$ value will be chosen as a node at a split. And the reason gini Impurity is used instead of entropy is the computational speed. Due to the presence of log-term in calculating entropy the computational speed is low. The maximum value of entropy is 1 whereas for Gini impurity it is 0.5.

Fig.3 explains the entropy and Gini values for a given probability of an outcome. The maximum value occurs at a probabilty score of 0.5.

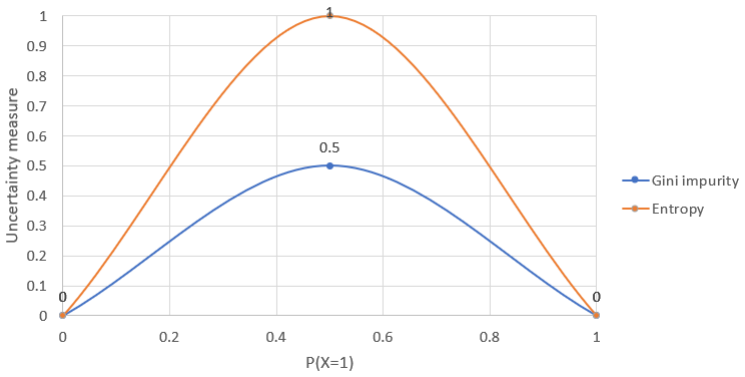

Fig. 3: Entropy vs Gini Plot

C. Pruning

CART algorithm uses the technique called pruning which reduces the size of decision trees and helps in reducing the complexity of resulting classifier which in turn increases the predictive accuracy[12]. Pruning is classified into two types-pruning or post pruning, Early stopping or pre-pruning.

- Post pruning as the name suggests is cutting back the tree after the entire tree is built. In the absence of pre-pruning, the final tree that has been built might be over-fitted. As discussed above in the CART algorithm, the data will be continuously partitioned into smaller and smaller subsets until the final subset is only a few data points or a single data point. By the end of this process the tree will learn the data but there is a high chance that a new data point might get created which cannot be predicted and thus decreasing the accuracy. Here post pruning comes into picture. The tree is cut back to the point where the error is minimum. This error is also called as cross validation error. In order to further increase the accuracy, the tree is pruned back a bit further than the encountered minimum error.
- Another classification of pruning is pre-pruning. Here over-fitting problem is prevented by stopping the tree-building process before the occurrence of minimum error. At each stage of tree building cross validation error is checked. If this error does not decrease in the next stage we stop the process. This pruning should be done carefully as it might result in underfitting if stopped too early. Pre-pruning and post-pruning can be implemented together or separately.

But the pruning technique is limited to only decision trees because of the randomness in features and having few samples random forest algorithm allows the tree to grow as depth as possible. Even if few trees over-fits individually, the final aggregation of all trees will be a good model without over-fitting or underfitting.

III. Bias Variance Trade-off

In the case of decision trees, a single tree is grown with all the data and features which leads to a high depth tree. A tree with a maximum depth will lead to an overfitting problem. In some cases, a DT is said to be biased as they choose a node which has high cardinality feature values, or we can say split of a tree is done on feature with high impurity. But in RF there is no overfitting problem as we only select sub-sample of the data and features which are randomly selected. We may have an overfitting problem in one of the decision trees but when we aggregate all the decision trees there is no overfitting

case. And same as DT we may get biased in some cases.

IV. Problem Statement

We all know that the random forest model gives high accuracy when compared to other models, because of its randomness in data and features. Input features are selected randomly and given to each decision tree. Because each node is selected using Information gain criteria on individual features, some of the features may not be present in the tree or may be pruned as the tree grows deeper. But sometimes we need a good interpretable model that uses some actionable features to predict the output, depending on the output we get from the actionable features we can work on those features values to solve a business problem. This problem can be described with a current problem the whole world is facing i.e. Covid19.

Suppose we are given data of the last 3 months of all people who have symptoms of Corona virus, after isolation and testing them some are declared as negative and some are positive. Now we have a dataset of a classification problem with features such as symptoms, their travel history, country, reporting date, etc. Now if we give this random sample set of inputs and random features to a Random forest model it can give a highly accurate output but this wont give us insight on problem solution. So, when we have some actionable features included in predicting output that can help us prevent the spread of viruses such as travel history.

If we predict by including this feature will give you an insight of the travel places from which the virus is spreading, and government officials can restrict people from those places to control the pandemic. The model which has a feature that can explain the cause of the output can be termed as the Interpretable Model. This can be extended to more business problems as well. Thus, building an interpretable model[13] will benefit the companies or researchers as a tool to evaluate their outcomes.

As Random Forest algorithm is using a feature selection technique to select the features randomly why can't we give interpretable or actionable features to the model with some criteria to select those features. This idea led me to implement the Interpretable Random forest (IRF) model.

V. Proposed Approach

As discussed, feature selection at each node of the decision tree is done using Information gain criteria. But to design our IRF model we need to have some important features to interpret the output. For that, we need to find those important features in the given dataset and assign some probabilities or priorities to select them while the node splits.

This can be done using domain expert analysis and feedback as they know about the features that are important to have for a model to predict so that they can act on those features. With this we can assign probabilities directly or we can use ANFIS (Artificial Network for Fuzzy Inference System) method to assign priority to each feature.

A. ANFIS

Adaptive Neuro-Fuzzy Inference system or in short ANFIS is a data learning technique. The given input will be transformed into a targeted output using a fuzzy interference system. This prediction method makes use of if-then rules, membership functions, and fuzzy logic operators. In ANFIS operation there are five main processing stages.

- Fuzzification
- Application of fuzzy operators
- Application method
- Output aggregation
- Defuzzification

In our case, depending on the rules given for the system, it will give the output as high priority or medium priority or low priority for an input feature[14]. With this we can assign the priority values to each feature instead of giving random probabilities.

VI. Dataset

To experiment on this approach, I have taken the datasets mentioned in table 1. The chosen datasets are diverse in the category they belong to, feature size and data sample size. Even though the random forest model works for both classification and regression models I prefer to work on classification problems in this paper to see how our proposed IRF model varies in the output compared to the random forest model.

Algorithm 2: Interpretable Random forest model

Data: $S = \{(x_i, y_i)^N_{i=1} — x_i \in \mathbf{R}^M, y_i \in (1, 2, 3, ...c)\}$
Result: Random forest model
initialization;
`Def Model(Data):`
for *each tree in [1,2,....,k]* **do**
  `select` S_n `samples (subset of S) randomly`
  `select m features randomly`
  `tree_list t = BUILD_TREE(`S_n`,m)`
end
`Result=Aggregation(t)`
`Def BUILD_TREE(`S_n`,m):`
for *i in range 1 to m* **do**
  `Calculate` $Gini_{split}$ `for all the unselected features at each node`
  `Multiply the corresponding probability value of each feature to the calculated` $Gini_{split}$ `value ->(i)`
  `Choose the feature with lowest value from (i).`
  `Selected feature at node r is divided into child nodes`
  `until the nodes becomes pure`
end
`return tree`

A. Dataset 1- Haberman survival dataset

In the Haberman dataset we have 4 features, 3 are input features and the last column is the output/target label for a binary classification problem. when we draw a pair plot between the features we can conclude Age and auxiliary node are Important features to be considered.

Dataset	Sample size	No. of features	Problem type
Haberman survival dataset	305	4	Classification
Customer churn dataset	650	9	Classification
Sonar dataset	207	61	Classification
Spam email	1200	58	Classification

Table 1. Datasets

B. Dataset 2- Customer churn dataset

In the customer churn dataset we have 9 features Rate, competitor Rate, date, amount, location, risk Score, number of Products, customer Segment and target label funded or not. In this dataset we have rate, competitor rate, risk score as important features. So, we are going to give a high probability to these features to include them in deciding whether a customer gets funds or not.

C. Dataset 3- Sonar dataset

In Sonar dataset we have 61 features which are signals obtained from a variety of different angles, spanning 90 degrees for the metal cylinder (mine) and 180 degrees for the rock over a period. This will be a binary classification problem to check whether the outcome is rock or mine. The major goal in using this dataset is to experiment by giving the probability to some features if they are important will make changes in accuracy or not.

D. Dataset 4- Spam email classification dataset

In this dataset we have a binary classification problem. We have emails with few words that will make it spam and few words make an email as personal or work mail which is not spam. So in this dataset we are given 57 feature engineered

attributes/features to classify the email. Each row is one email. From this experiment we can give few important words which are given as features in some numerical value will be included

VII. Results

In this project, I would like to compare the results of the random forest model and our new proposed interpretable random forest model. In practice, if we are giving a high probability to important features which have low cardinality values then the complexity may increase in building a tree and in a few cases accuracy may increase or decrease depending on the features that are selected randomly.

We will see the below results of both the models which changes with change in features. To check the interpretability we can plot the tree in a random forest to check which features are used to predict the output. But here to evaluate my model I want to use accuracy as my metric because all the datasets used in this project are classification and balanced datasets. To reproduce the results we should not use standard libraries as the feature selection is already implemented.

We should have the code from scratch so that we can write our own feature selection logic. For now I have given a list of probability values to all the features and on checking the Gini impurity at each split probability values are multiplied with Gini impurity values then the feature with the lowest value is selected at the node. This continues for all nodes and trees.

The plots for change in accuracy with respect to increase in number of trees are shown for each dataset and compared with both the models.

In this project 5 cross validation is used on each tree and the accuracy is calculated as the mean of all the results at each validation. So number of trees considered is limited to 100. From all the observations seen from the plots Figure 4-11 and data shown in Table 2, IRF model is giving a little more accuracy than RF model at some point with T_k number of trees.

A. Observation

In haberman surival dataset we have only 4 features. From these 4 features taking 2 random features for each tree can't give good results. So

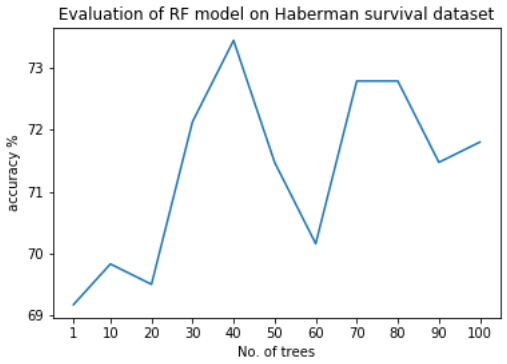

Fig. 4: Plot for dataset-1 using RF model

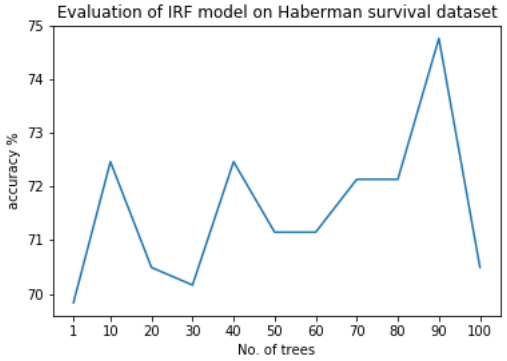

Fig. 5: Plot for dataset-1 using IRF model

dataset with very few features may not work well for RF/IRF model. Whereas spam email classification dataset has large feature set and data samples resulting in good accuracy and same in case of customer churn dataset.

Because of Random selection in features and data samples, model should have atleast some good

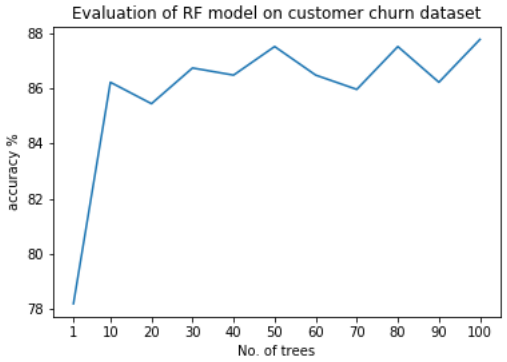

Fig. 6: Plot for dataset-2 using RF model

		No. of trees										
Dataset	model	1	10	20	30	40	50	60	70	80	90	100
Haberman Dataset	RF	**69.1%**	**69.8%**	**69.5%**	**72.1%**	**73.4%**	**71.4%**	**70.1%**	**72.7%**	**72.7%**	**71.4%**	**71.8%**
	IRF	**69.8%**	**72.4%**	**70.4%**	**70.1%**	**72.4%**	**71.1%**	**71.1%**	**72.1%**	**72.1%**	**74.7%**	**70.1%**
Customer Churn Dataset	RF	**78.1%**	**86.2%**	**85.4%**	**86.7%**	**86.4%**	**87.5%**	**86.4%**	**85.9%**	**87.5%**	**86.2%**	**87.7%**
	IRF	**79.2%**	**84.4%**	**86.2%**	**88.8%**	**86.2%**	**89.8%**	**86.4%**	**88.2%**	**87.7%**	**88.5%**	**88.5%**
Sonar Dataset	RF	**68.7%**	**80.4%**	**78.5%**	**79.5%**	**80.9%**	**80.4%**	**80.4%**	**79.0%**	**80.4%**	**82.9%**	**80.4%**
	IRF	**65.8%**	**81.4%**	**81.4%**	**80.9%**	**84.8%**	**82.9%**	**82.9%**	**85.8%**	**86.3%**	**83.9%**	**83.4%**
Spam email Dataset	RF	**83.6%**	**93.5%**	**93.5%**	**94.8%**	**94.5%**	**94.5%**	**93.8%**	**94.1%**	**94.5%**	**94.1%**	**94.1%**
	IRF	**76.5%**	**88.4%**	**88.4%**	**90.7%**	**90.0%**	**90.7%**	**94.8%**	**92.7%**	**91.7%**	**91.7%**	**91.4%**

Table 2. Accuracy for both models as number of trees grow

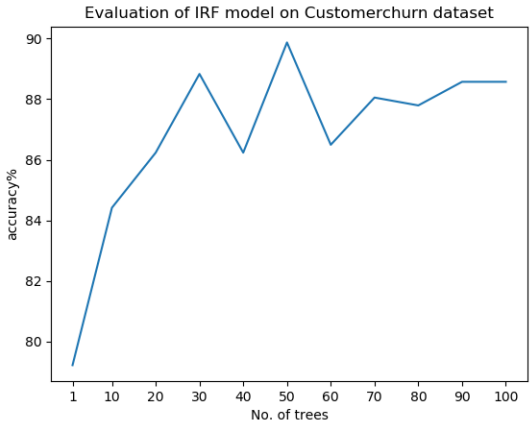

Fig. 7: Plot for dataset-2 using IRF model

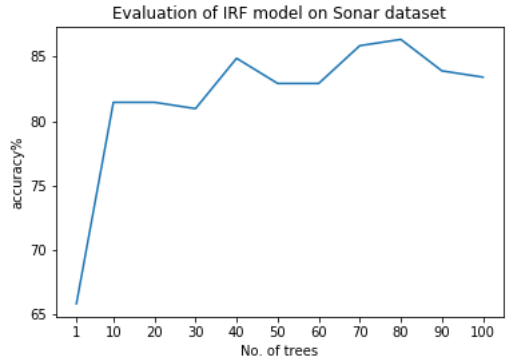

Fig. 9: Plot for dataset-3 using IRF model

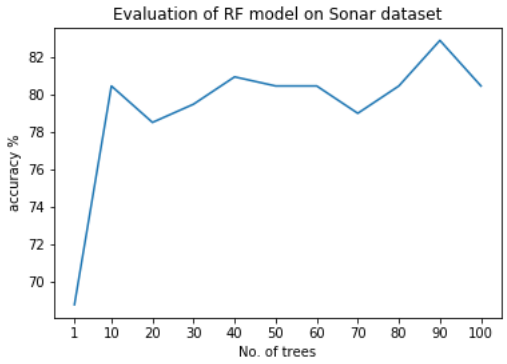

Fig. 8: Plot for dataset-3 using RF model

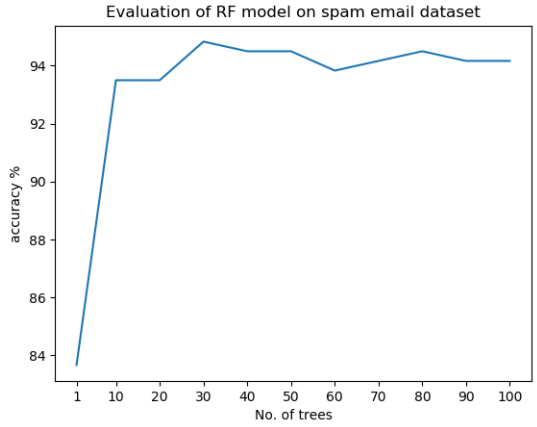

Fig. 10: Plot for dataset-4 using RF model

number of features,samples and T_k number of trees to give the high accuracy.

VIII. Conclusion and Future Work

With this project, we have presented a new method to select interpretable features for a random forest model to make the model output more explainable and it will be easy to act on those features for improvement.In this project we assumed the probabilities to each interpretable feature but this should be implemented using techniques like ANFIS as discussed in this paper and also implementation of the IRF model in federated learning environment will be a part of my future work.

Federated learning[15] is the state-of-the-art data privacy-preserving technique in implementing the

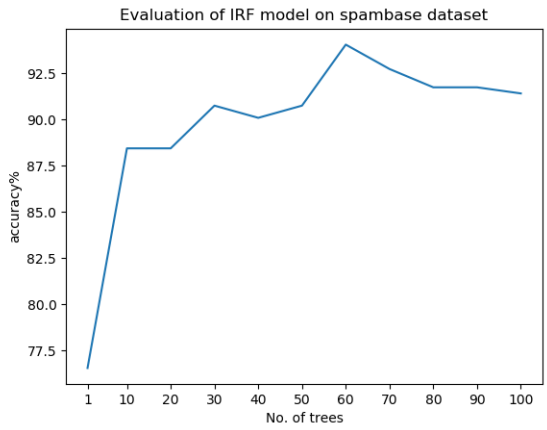

Fig. 11: Plot for dataset-4 using IRF model

machine learning algorithms without the data being brought out of the client's device. This works like Random forest model architecture as each client has different samples of data and random features to train the individual model. This can be termed as 'Interpretable federated forests'.

ACKNOWLEDGMENT

This research project is a part of the machine learning course at the university of Regina under the guidance of Professor Dr. Sandra Zilles. Also, I would like to acknowledge Dr. Alireza Manashty for the support in accessing required resources at the data science laboratory.

REFERENCES

[1] J. M. Klusowski, "Complete Analysis of a Random Forest Model," vol. 13, pp. 1063–1095, 2018.

[2] L. Breiman, "ST4_Method_Random_Forest," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] L. Breiman, "Manual on setting up, using, and understanding random forests v3. 1," *Technical Report, http://oz.berkeley.edu/users/breiman, Statistics Department University of California Berkeley, ...*, p. 29, 2002.

[4] Y. L. Pavlov, "Random forests," *Random Forests*, pp. 1–122, 2019.

[5] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in Forests of randomized trees," *Advances in Neural Information Processing Systems*, pp. 1–9, 2013.

[6] N. Nguyen, A. Subramanian, and B. King, "CS 294-1 Final Project : Benchmarking Random Forests against Na ¨ ıve Bayes," pp. 1–8, 2013.

[7] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[8] D. Smyth, E. Deverall, M. Balm, A. Nesdale, and I. Rosemergy, "Out-of-Bag Estimation," *New Zealand Medical Journal*, vol. 128, no. 1425, pp. 97–100, 2015.

[9] T. T. Nguyen, J. Z. Huang, and T. T. Nguyen, "Unbiased feature selection in learning random forests for high-dimensional data," *Scientific World Journal*, vol. 2015, 2015.

[10] R. X. Chen, "A Brief Introduction on Shannon ' s Information Theory," no. January, 2016.

[11] A. D'Ambrosio and V. A. Tutore, "Conditional classification trees by weighting the gini impurity measure," in *Studies in Classification, Data Analysis, and Knowledge Organization* (S. Ingrassia, R. Rocci, and M. Vichi, eds.), (Berlin, Heidelberg), pp. 273–280, Springer Berlin Heidelberg, 2011.

[12] E. Frank, "Pruning decision trees and lists," *Science*, vol. 300, no. January, p. 204, 2000.

[13] S. Pereira, R. Meier, R. McKinley, R. Wiest, V. Alves, C. A. Silva, and M. Reyes, "Enhancing interpretability of automatically extracted machine learning features: application to a rbm-random forest system on brain lesion segmentation," *Medical image analysis*, vol. 44, pp. 228–244, 2018.

[14] D. S. Badde, A. Gupta, and V. K. Patki, "Comparison of Fuzzy Logic and ANFIS for Prediction of Compressive Strength of RMC," *IOSR Journal of Mechanical and Civil Engineering*, pp. 1–10, 2013.

[15] Y. Liu, Y. Liu, Z. Liu, J. Zhang, C. Meng, and Y. Zheng, "Federated Forest," pp. 1–15, 2019.