RANDOM FOREST METHOD

DATA SCIENCE
INSTITUTE

Classification and Regression Problems

- Classification Technique is used for classifying a dependent variable which is categorical or binary say with 2 categories (0 and 1) Techniques used: Binary Logistic Regression, Naïve Bayes' Classifier, Random Forest Method.
- Regression Technique is used for prediction of the continuous dependent variable.

DATA SCIENCE INSTITUTE

Random Forest: Introduction

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*
- Random Forest is like asking a diverse group of friends for their opinions to make a smarter decision, taking into account different perspectives and preventing biases. It's like taking a vote among your friends.
- Random Forest combines the opinions of all these decision trees to make a final decision.

- Random Forest works in two-phase

- first is to create the random forest by combining N decision tree, and
- second is to make predictions for each tree created in the first phase.

- To create a Random Forest the Method of “Bootstrapping” is used.

Bootstrapping

- Bootstrapping is a method for estimating the sampling distribution of an estimator by resampling with replacement from the original sample.

- The method is especially useful in situations where sampling distribution of estimator is not standard distribution.

- The method can be used in any statistical inference problem.

- The use of the term 'bootstrap' comes from the phrase “To pull oneself up by one's bootstraps ” - generally interpreted as succeeding in spite of limited resources.

Bootstrapping…

- Many conventional statistical methods of analysis make assumptions about normality, including correlation, regression, *t* tests, and analysis of variance. When these assumptions are violated, such methods may fail.

- Bootstrapping, a data-based simulation method, is steadily becoming more popular as a statistical methodology. It is intended to simplify the calculation of statistical inferences, sometimes in situations where no analytical answer can be obtained.

- As computer processors become faster and more powerful, the time and effort required for bootstrapping decreases to levels where it becomes a viable alternative to standard parametric techniques.

Bootstrapping…

Original Sample
12,23,11,29,34, 38,41,45,6
Median=29.00

	Sample 1	Sample 2	Sample 3			Sample B
	23	11	6			41
	23	29	45			45
	29	11	11			11
	29	29	29			34
	34	34	11			34
	38	34	38			38
	41	41	41			41
	45	45	41			45
	41	11	6			6
Median	34.00	29.00	29.00			38.00

Here sampling distribution
of sample median is Generated.
Assuming B=1000,
25th value and 975th value from
ordered Median values will provide
95% confidence Interval for median.

Sample size of original and each bootstrap sample is 9

DATA SCIENCE INSTITUTE

What is “Bagging”?

- The term “bagging” was Introduced by Breiman (1996).

- “Bagging” stands for “bootstrap aggregating”.

- It is an ensemble method: a method of combining results from multiple resamples.

- Ensemble method can also be applied by using different classifiers for a given sample.

DATA SCIENCE INSTITUTE

How Bagging Works?

- Generate B bootstrap samples of the training data: random sampling with replacement.(Example B=500 resamples)
- Train a classifier or a regression function using each bootstrap sample. (Example: Apply Decision Tree Method to 500 resamples)

- For classification: Use majority vote to predict the class
- For regression: Use average to predict value

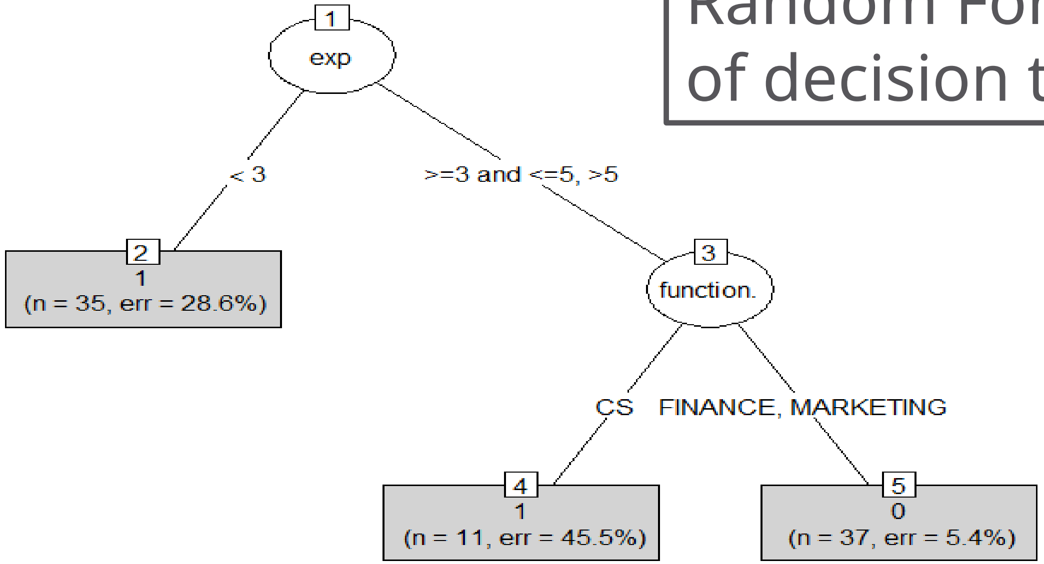
- Bagging improves performance for unstable classifiers which vary significantly with small changes in the data set.

Recap: Decision Tree

35 employees with exp<3.
Out of which 25
Left within 18 months.
25/35=0.714.
Err=28.6%

Exp is the first split variable followed by function. Gender and Source are not appearing in the decision tree.

Random Forest is bagging of decision trees.



DATA SCIENCE INSTITUTE

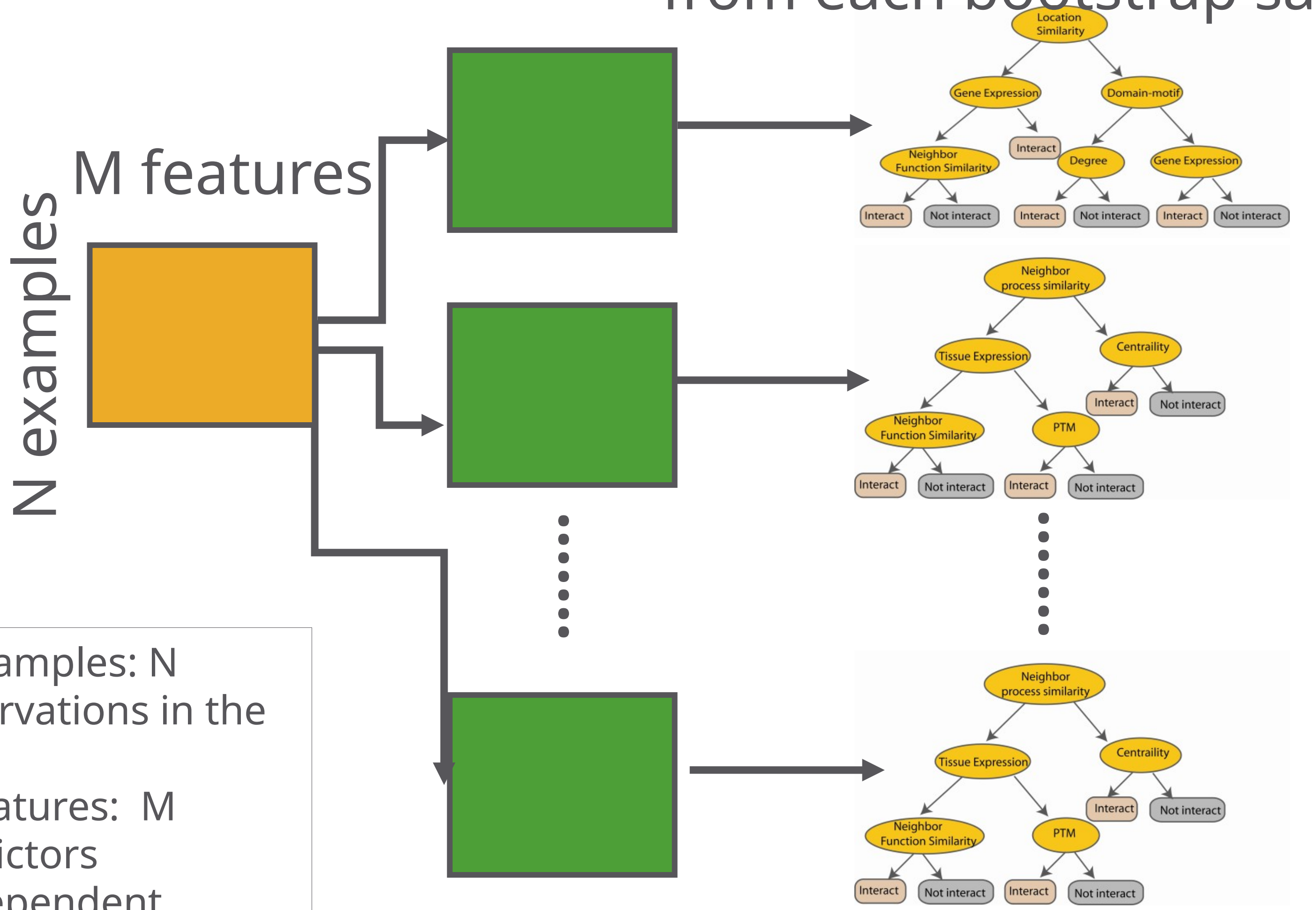
What is Random Forest?

- It is a supervised machine learning method for classification and regression.
- Random forest (or random forests) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees.
- The term came from random decision forests that was first proposed by Tin Kam Ho of Bell Labs in 1995.
- It is a Tree based Model.
- The method combines Breiman's "bagging" idea and the random selection of features.

DATA SCIENCE
INSTITUTE

Random Forest Algorithm

Create decision tree from each bootstrap sample



N examples: N observations in the data,
M features: M predictors (Independent variables)

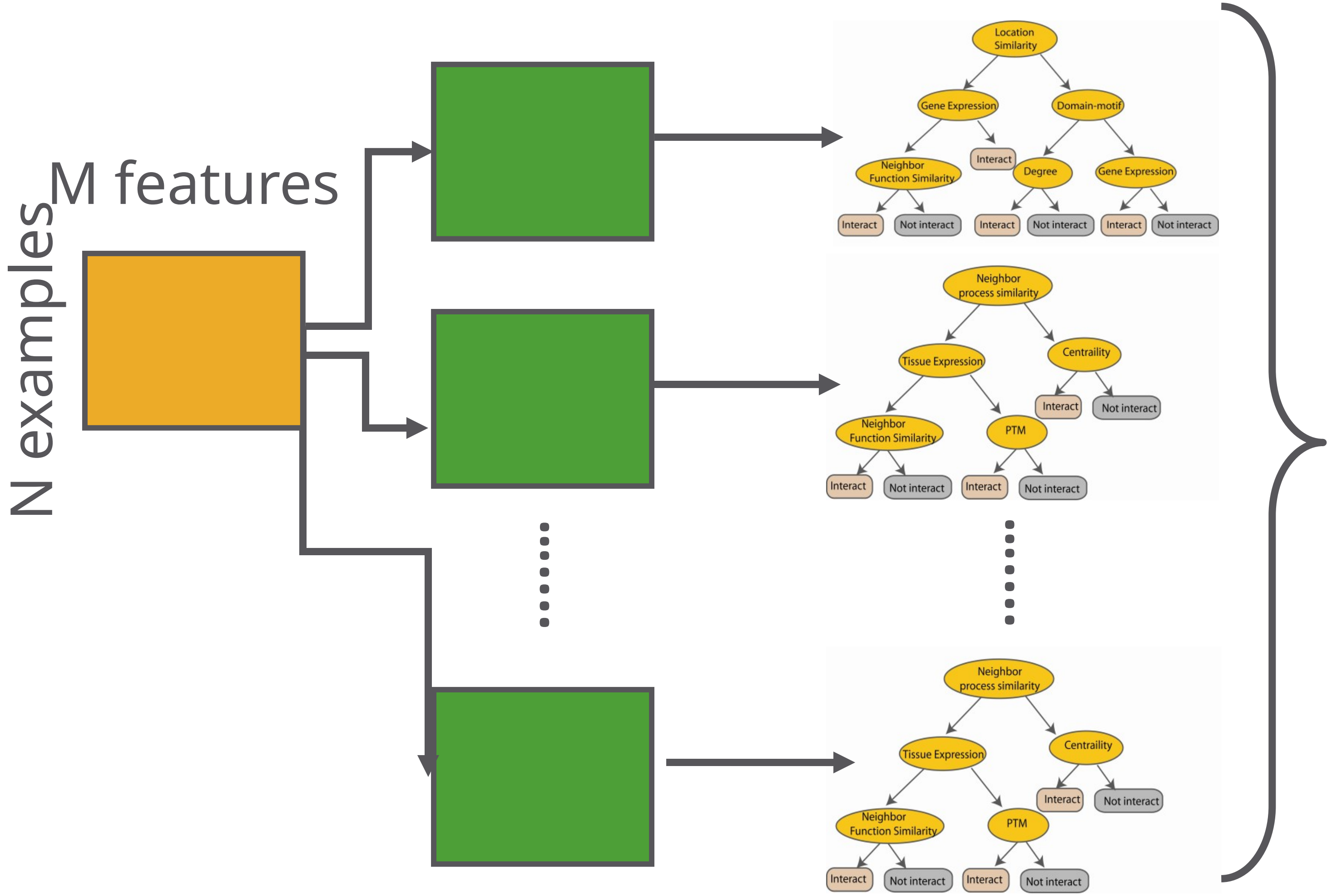
DATA SCIENCE INSTITUTE

Random Forest Algorithm…

- Grow a forest of many trees. (R default is 500)
- Grow each tree on an independent bootstrap sample* from the training data.
- At each node:
  - Select *m* variables at random out of all *M* possible variables (independently for each node).
  - Find the best split on the selected *m* variables.
- Grow the trees to maximum depth (classification).
- Vote/average the trees to get predictions for new data.

*Sample N cases at random with replacement

Random Forest Algorithm…



Take the majority vote

DATA SCIENCE INSTITUTE

Random Forest Algorithm…

- Bootstrap sample of data (sampling with replacement, approx. 2/3 of original values will be in each sample).

- Fit a tree to its greatest depth determining the split at each node through minimizing the loss function considering a random sample of covariates (size is user specified).

- For each tree
  - Predict classification of the leftover 1/3 using the tree, and calculate the misclassification rate = out of bag error rate.
  - For each variable in the tree, permute the variables values and compute the out-of-bag error, compare to the original oob error, the increase is a indication of the variable's importance.

Random Forest Algorithm…

- Aggregate oob error and importance measures from all trees to determine overall oob error rate and Variable Importance measure.
  - Oob Error Rate: Calculate the overall percentage of misclassification
  - Variable Importance: Average increase in oob error over all trees.

Example

Employee churn model.

Independent variables are:

- Gender
- Experience Level (<3, 3-5 and >5 years)
- Function (Marketing, Finance,Client Servicing)
- Source (Internal or External)

Dependent variables is "status" (=1 if employee left within 18 months from Joining date)

Data Snapshot

sn	status	function	exp	gender	source
1	1	CS	<3	M	external
2	1	CS	<3	M	external
3	1	CS	>=3 and <=5	M	internal
4	1	MARKETING	<3	M	external
5	1	FINANCE	<3	F	internal
6	1	CS	>=3 and <=5	F	internal
7	1	MARKETING	<3	F	internal
8	1	FINANCE	<3	F	external
9	1	CS	<3	M	internal
10	1	CS	>5	M	external
11	1	CS	>5	F	external
12	1	CS	<3	F	external
13	1	MARKETING	<3	M	external
14	1	FINANCE	<3	M	external
15	1	FINANCE	<3	M	internal
16	1	CS	<3	M	external
17	1	CS	<3	F	internal
18	1	CS	<3	F	internal
19	1	MARKETING	>=3 and <=5	M	internal
20	1	FINANCE	<3	M	external

Random Forests in R

```
install.packages("randomForest")
library(randomForest)
empdata<-read.csv(file.choose(),header=T)
empdata$status<-as.factor(empdata$status) #classification problem
```

#Run Random Forests

#mtry: Number of variables randomly sampled as candidates at each split
Note that the default values are different for classification (sqrt(p) where p is number of variables in x) and regression (p/3)

#ntree: Number of trees to grow
This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.

```
churn_rf<-randomForest(formula=status~function.+exp+gender+source,
                       data=empdata, mtry = 2, ntree =100,
                       importance=TRUE,cutoff=c(0.6,0.4))
```

1. churn_rf <- randomForest(...): This line creates a Random Forest model and assigns it to the variable churn_rf.
2. formula = status ~ function. + exp + gender + source: Here, you're specifying the formula for your model.
3. data = empdata: You're specifying the dataset empdata from which to build the model.
4. mtry = 2: This parameter determines the number of variables randomly sampled as candidates at each split. In your case, you've set it to 2.
5. ntree = 100: You're specifying the number of trees to grow in the forest. In this case, you're building 100 trees.
6. importance = TRUE: This parameter calculates and stores the importance of variables.
7. cutoff = c(0.6, 0.4): This parameter sets the cutoff values for classifying predictions.

- The cutoff values are applied to the predicted probabilities to determine the final classification of each observation.
- After the model makes predictions, each observation will have a predicted probability of belonging to each class ("churn" or "not churn").
- The cutoff values (0.6 and 0.4) determine the threshold at which these probabilities are converted into class labels.
- These cutoff values allow you to adjust the sensitivity and specificity of your classification based on your specific requirements and the cost associated with misclassification. (You might choose different cutoff values depending on whether you prioritize correctly identifying churn cases or correctly identifying non-churn cases.)

Random Forests in R...
Out of Bag Error Rate

churn_rf

Call:

randomForest(formula = status ~ function. + exp + gender + source, data = empdata,
 mtry = 2, ntree = 100, importance = TRUE, cutoff = c(0.6, 0.4))

Type of random forest: classification

Number of trees: 100

No. of variables tried at each split: 2

OOB estimate of error rate: 24.1%

Confusion matrix:

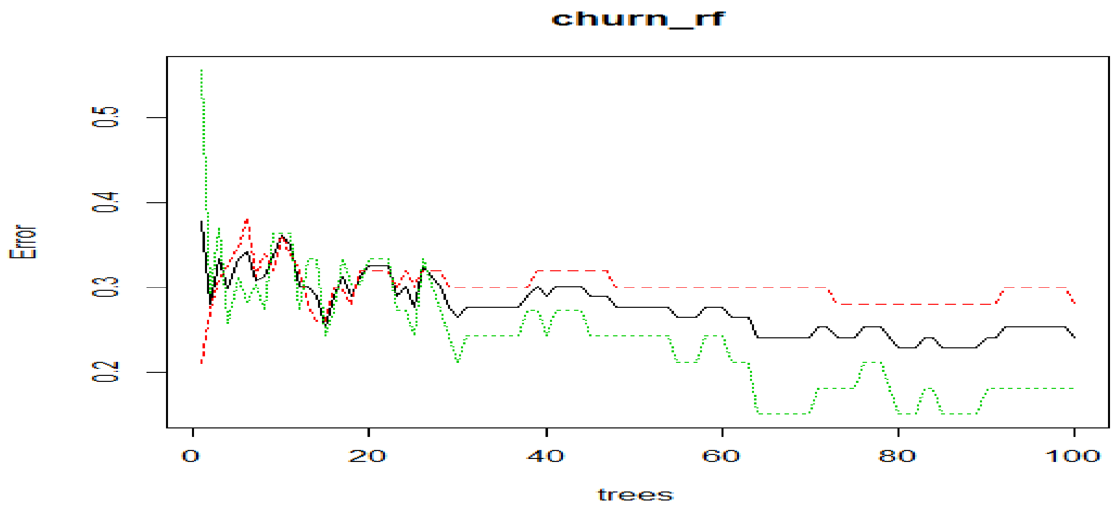
	0	1	class.error
0	36	14	0.2800000
1	6	27	0.1818182

Using bagging method

Note: The model calculates the error using observations not trained on for each decision tree in the forest and aggregates over all so there should be no bias, hence the name out-of-bag.

Random Forests in R…
Decision Trees Error Rate

plot(churn_rf)



Plot shows error rates for all 100 decision trees.

Black line shows the overall OOB error rate

Coloured lines show error rates for each class.

#Also try predict function. You can use new data in predict function

predict(churn_rf,empdata)

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 
 1  1  0  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  1  1 
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 
 1  0  0  1  0  1  0  0  1  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0 
65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 
 0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  1 
Levels: 0 1
```

Random Forests in R…
Variable Importance

#Importance Matrix

churn_rf$importance

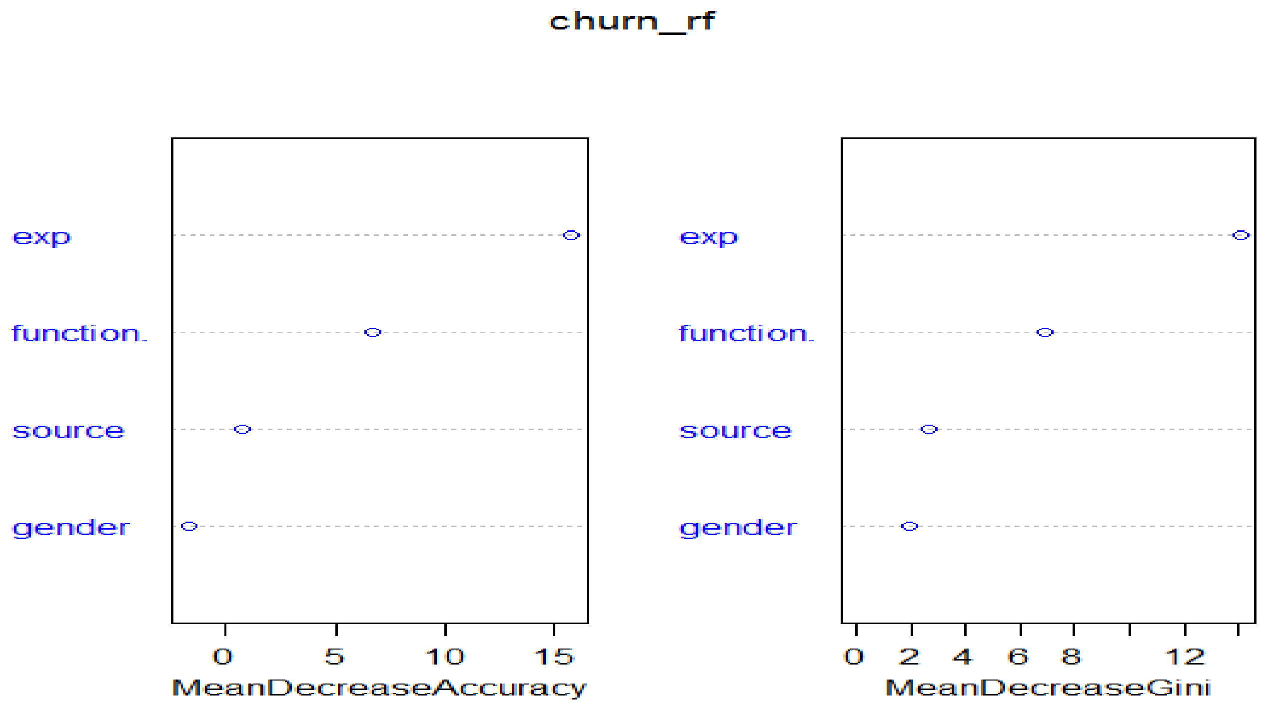
> churn_rf$importance

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
function.	0.060819198	0.03712426	0.051286609	6.899711
exp	**0.142498369**	**0.15427908**	**0.144903998**	**14.072970**
Gender	-0.002759849	-0.01769380	-0.008267061	1.979475
source	0.027194204	-0.02704038	0.004688307	2.620526

#Experience has highest importance since "Mean Decrease Accuracy" after excluding Experience is highest.

Random Forests in R…
Variable Importance Plot

varImpPlot(churn_rf,col="blue")

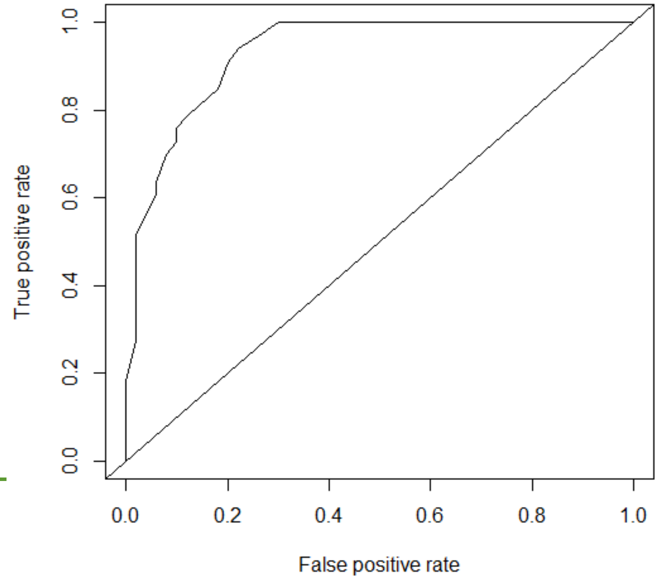


Random Forest in R….
ROC Curve

```
predrf <- predict(churn_rf,empdata, type = "vote", norm.votes = TRUE)

library(ROCR)
pred<-prediction(predrf[,2],empdata$status`
perf<-performance(pred,"tpr","fpr")
plot(perf)
abline(0,1)

## Area under ROC Curve in R (AUC)
auc<-performance(pred,"auc")
auc@y.values
```




```
[1] 0.9327273
```

Advantages and Disadvantages of Random Forest

Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

- The name "Random Forest" reflects both the ensemble nature of the algorithm, consisting of many decision trees, as well as the random selection process used during training to create diversity among the trees.

Three Methods commonly used for Classification

- Binary Logistic Regression
- Naïve Bayes' Classifier
- Random Forest Method

THANK YOU!!

DATA SCIENCE
INSTITUTE