Assignment No: 13

**Title:** Trip History Analysis: Use trip history dataset that is from a bike sharing service in the United States. The data is provided quarter-wise from 2010 (Q4) onwards. Each file has 7 columns. Predict the class of user

**Theory:**

Recursive partitioning, or “classification and regression trees,” is a recursive partitioning method which is a fundamental tool in data mining. It is a prediction method often used with dichotomous outcomes that avoids the assumptions of linearity. This technique creates prediction rules by repeatedly dividing the sample into subgroups, with each subdivision being formed by separating the sample on the value of one of the predictor variables. The end result is a set of branching questions that forms a treelike structure in which each final branch provides a yes/no prediction of the outcome. rpart is for modeling decision trees. This section briefly describes ART modeling, conditional inference trees, and random forests.

**CART modeling via rpart**

Classification and regression trees (as described by Brieman, Freidman, Olshen, and Stone) can generated through the rpart package. The general steps are provided below followed by two examples.

**1. Grow the Tree**

To grow a tree, use

rpart(formula, data=, method=,control=) where

formula

is in the format

outcome ~ predictor1+predictor2+predictor3+ect.

data=specifies the data frame

Method: class" for a classification tree

"anova" for a regression tree

**Control:**

Optional parameters for controlling tree growth. For example, control=rpart.control(minsplit=30, cp=0.001) requires that the minimum number of observations in a node be 30 before attempting a split and that a split must decrease the overall lack of fit by a factor of 0.001(cost complexity factor) before being attempted.

**2. Examine the results**

The following functions help us to examine the results.

printcp(fit) display cp table

plotcp(fit) plot cross-validation results

construct a decision tree.

fit <- rpart(Play- Outlook+ Temperature +Humidity+ Wind,

method="class11,data=play\_decision,control=rpart.control(minsplit=l),

parms=list(split=•information•))

The rpart function has four parameters. The first parameter, Play - Out look + Temperature+ Hurnidi ty + Wind, is the model indicating that attribute Play can be predicted based on attributes Outlook, Temperature, Humidity, and Wind. The second parameter, method, is set to "class," telling R it is building a classification tree. The third parameter, data, specifies the dataframe containing those attributes mentioned in the formula. The fourth parameter, control, is optional and controls the tree growth. In the preceding example, control=rpart. control (minsplit=l) requires that each node have at least one observation before attempting a split. The rminsplit= 1 makes sense for the small dataset, but for larger datasets rminsplit could be set to 10% of the dataset size to combat overfitting. Besides minsp 1 it, other parameters are available to control the construction of the decision tree. For example, rpart. control (rnaxdepth=10, cp=O. 001) limits the depth of the tree to no more than 10, and a split must decrease the overall lack of fit by a factor of 0.001 before being attempted. The last parameter (parms) specifies the purity measure being used for the splits. The value of sp 1 it can be either information (for using the information gain) or g ini (for using the Gini index). For the tripadvisory dataset the following code snippet for rpart is used:

fit<-rpart(train$Member.type~.,data=train, method="class")

**rsq.rpart(fit)**

plot approximate R-squared and relative error for different splits (2 plots). labels are only appropriate for the "anova" method.

**print(fit)**

print results

summary(fit) detailed results including surrogate splits

plot(fit) plot decision tree

**text(fit)** label the decision tree plot

**post(fit, file=)** create postscript plot of decision tree

**Output:**

