Lab 5
Decision Tree based Classification

Objective

Use the given data set to build a Decision tree based classification using C5.0 Algorithm and use it to predict the class of given cases. Also, analyze the difference on outcome with and without pruning.

Classification using IBM SPSS Modeler

The IBM SPSS Modeler offer a variety of classification models. These models use the values of one or more input fields to predict the value of one or more output, or target, fields. Some examples of these techniques are:

Decision trees (C&R Tree, QUEST, CHAID and C5.0 algorithms) K-nearest neighbor

Naive Bayesian classifier
Neural networks

Support vector machines
Bayesian networks

Model Nuggets

A model nugget is a container for a model, that is, the set of rules, formulae or equations that represent the results of a model building operations. The main purpose of a nugget is for scoring data to generate predictions, or to allow further analysis of the model properties. Opening a model nugget on the screen enables you to see various details about the model, such as the relative importance of the input fields and/or Rule set used in creating the model. To view the predictions, you need to attach and execute a further process or output node.

Classification using C5.0 Node

C5.0 node works by splitting the sample based on the field that provides the maximum information gain at each level. Each subsample defined by the first split is then split again, usually based on a different field, and the process repeats until the subsamples cannot be split any further. Finally, the lowest-level splits are reexamined, and those that do not contribute significantly to the value of the model are removed or pruned. It can produce following two types of models -

Decision tree

It is a straightforward description of the splits found by the algorithm. Each terminal (or "leaf") node describes a particular subset of the training data, and each case in the training data belongs to exactly one terminal node in the tree.

Rule set

A rule set is a set of rules that tries to make predictions for individual records. In some cases, more than one rule may apply for any particular record, or no rules at all may apply. If multiple rules apply, either prediction of first applicable rule or prediction of majority voting of all applicable rules becomes the class of record. If no rule applies, a default prediction is assigned to the record.

Partitioning the data

1. Select a *"Partition Node"* from *"Field Ops"* tab in nodes palette and connect source node to this node. Double click on partition node and go to settings tab.
2. Name the "Partition Field" as *Partition Field*. Select *"Train and test"* and specify "70:30" ratio for training and testing data splits.
3. Go to *"Generate"* option on menu bar and *"Generate Select node for Training Partition"* and *"Generate Select node for Testing Partition"*. Click *"Apply"* and *"OK"*. Now, a Partition node and two Select nodes will appear on the modeler canvas. View training and testing records using a table node.
4. Connect *Training Select* node and *Testing Select* node to the partition node.

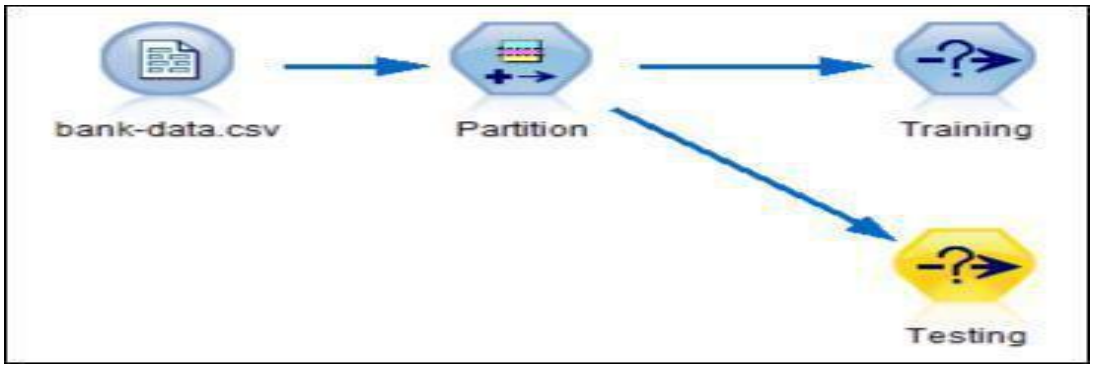

Fig 1- Stream after partitioning

Building Decision Tree

1. Select a *"C5.0 Node"* from *"Classification"* subgroup under *"Modeling"* tab in nodes palette and connect *Training select* node to this node.
2. Go to *"Model"* tab and select *"Decision tree"* as *Output type.* Uncheck *Use partitioned data* and *Build model for each split.* Uncheck *"Calculate predictor importance"* option on *"Analyze"* tab. Click *"Apply"* and *"OK"*. Now, a C5.0 node with target field name will appear on the modeler canvas.
3. *"Run"* this stream from C5.0 node, a *"Decision Tree Model Nugget"* will be created and placed on stream canvas. Double click on model nugget and go to *"Viewer"* tab and see the decision tree. Analyze the tree carefully.
4. Now connect the Testing select node to the model nugget. Run the model node and analyze the result on the testing data.

Generating Rule set

1. Take a new C5.0 node and this time select *"Rule set"* as *Output type*, this time. Select *Mode* as *Simple*. Go to *Annotations* and name the model as "C5.0 Simple mode". Click *"Apply"* and *"OK"*. Copy and paste this C5.0 node and connect *Training Select* node to it.

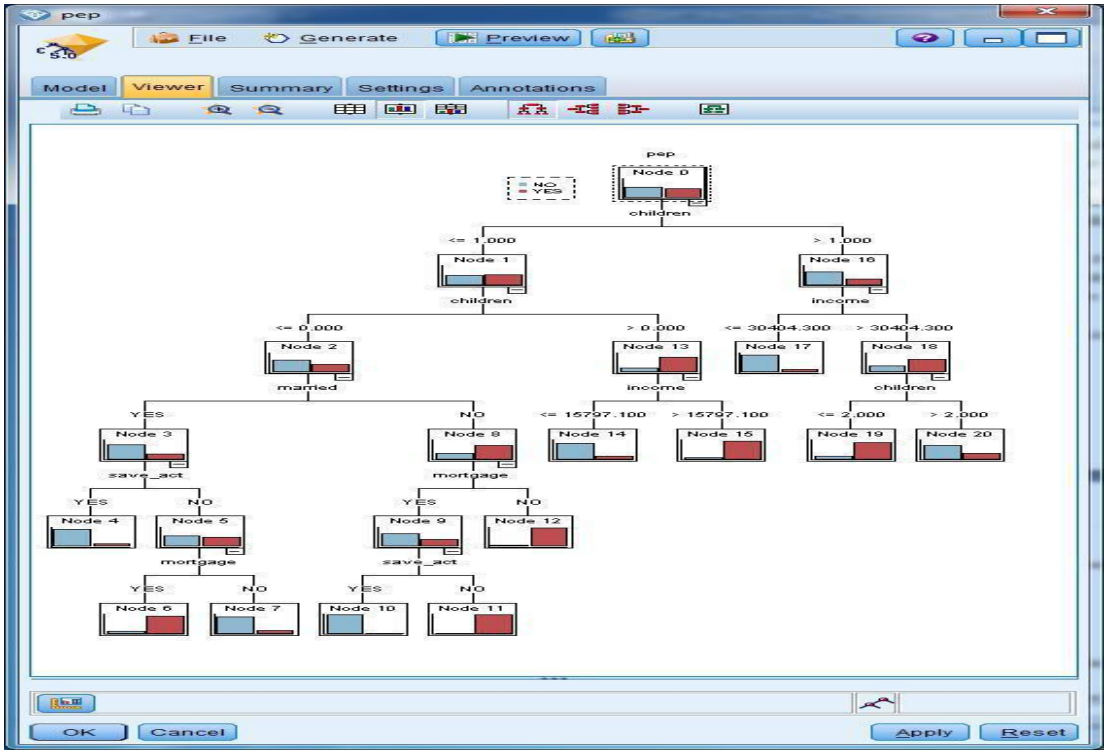

Fig 2- Decision tree

2. Go to *"Model"* tab of this copied C5.0 node and select *"Rule set"* as *Output type*, this time. Select *Mode* as *Expert*. Specify *Pruning severity* as 0 and *Minimum records per child branch* as 1. Uncheck *Use global pruning* and *Winnow attributes*. Uncheck *Use partitioned data* and *Build model for each split.*

3. Go to *Annotations* and name the model as "C5.0 Expert mode". Click *"Apply"* and *"OK"*. *This Expert mode node doesn't perform any type of pruning and creates an unrestricted decision tree (its rule set in this case).*

4. *"Run"* this stream. Two model nuggets will get created. Go to their *Model* tab and observe the difference in Rule sets.

5. Now connect these nuggets to the testing partition. Select node and run the steam to analyze the rule sets

6. Take a *Merge* node from Record Ops tab in nodes palette. Connect both model nuggets to this node. Configure the merge node to merge the data on the basis of *Keys* and select all keys except the predicted class and corresponding confidence fields of both models. Rename these fields accordingly.

7. Connect merge node to an Analysis node. Run the stream and observe the accuracy of both the models.

Predicting class of test cases

1. Go to *"Generate"* option in the menu bar of *Simple mode* model nugget and select *"Rule Trace Node"*. A SuperNode named as *"RULE"* will get generated on modeler canvas. Rename it as *"RULE: Simple mode"*. A SuperNode groups multiple nodes into a single node by encapsulating sections of a data stream, in this case, all the rules specified by Rule Set.

2. Select this SuperNode and choose *"Zoom-in"* option from *"SuperNode"* menu from the menu bar of SPSS modeler. A stream of *Derive node* will appear, each driving a new field based on the conditions in Rule Set. Double-click on the nodes to see the deriving condition.

3. Third-last node derives a filed having concatenation of all the rules or its value is "No rules". (This node uses concatenation operator ><).Second-last node derives another field which contains the concatenated rule set without the trailing semi-colon. Last node will be a *Filter* node which will be filtering out all the intermediate fields generated by different rules from final outcome.

4. Value of these derive fields contains predicted outcome of the rule along with its confidence value. In order to only compare the predicted value, we need to modify the output. [Optional] Add a *Table* node to last/second last node to see the output.

5. To make prediction of first applicable rule as the prediction of the record, select

   a *"Derive"* node from *"Field Ops"* tab under in nodes palette. Insert it in between second-last and last node. Choose *"Flag"* under *Derive as* option. Specify True value as "YES" and False value as "NO". Specify the true condition as

   'RULE' matches "YES*"

   This will put "YES" in newly derived field of all the records which had a string starting with YES in their RULE field.

6. Go to filter node, filter out RULE field and rename newly derived field as *"Simple mode prediction"*. Zoom-out of SuperNode. Repeat all steps from 1 to 9 for *Expert mode* model nugget. Do the renaming correspondingly. Add *Table* nodes to these SuperNodes to see the predicted values.

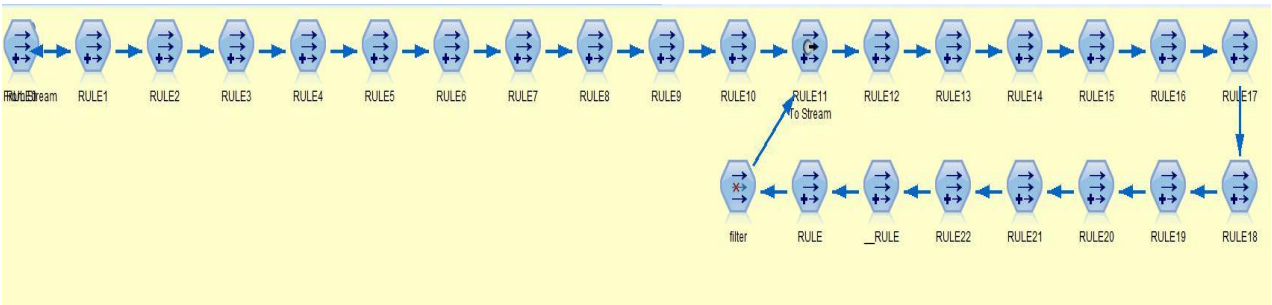

Fig 3- Zoom-in view of Simple mode SuperNode

Comparing performance

1. Select a *"Merge"* node from *"Record Ops"* tab in nodes palette. Connect *Simple mode* and *Expert mode* SuperNodes to this *Merge* node. Go to *Merge* tab and select *"Keys"* as Merge Method. Select all possible keys.
2. Select a *"Matrix"* node from *"Output"* tab in nodes palette. Under *Settings* select *Simple mode prediction* in Rows and *Expert mode prediction* in Columns.
3. Select *Cell-contents* as *Cross-tabulations*. Click on *Run* to see the results. Analyze it carefully.
4. Vary the *"Pruning Severity"*, repeat all steps from beginning and see the change in this matrix.
5. Connect a *Table* node to *Merge* node. Under *"Highlight records where*", write appropriate condition for highlighting records where prediction of both the modes differ. Select *"Output to file"* and file type as *.html.

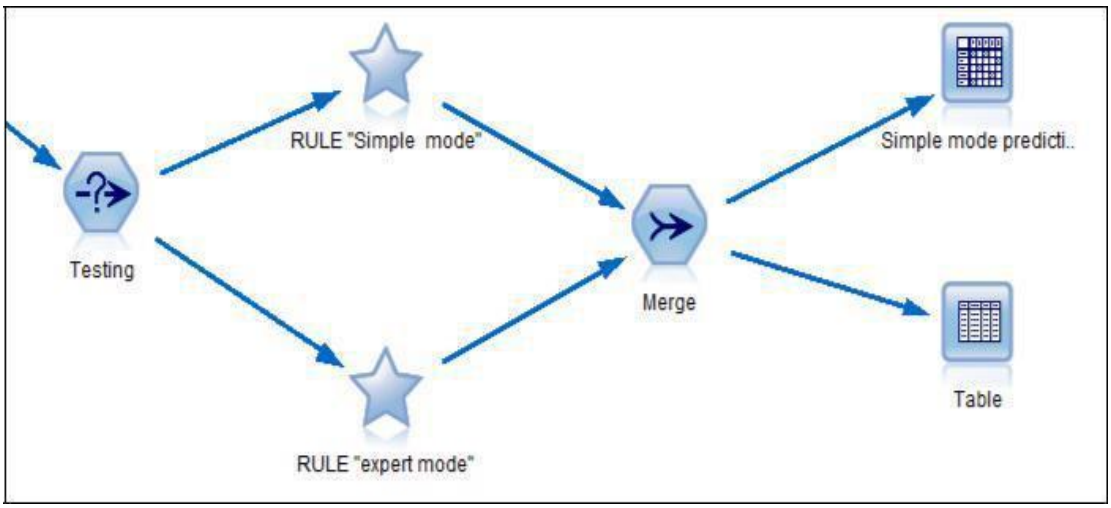

Fig 4- Stream for comparing results on testing data

Assignment

1. Pass both the partitions to the C5.0 models and use Analysis node to know the accuracy of model.
2. Observe the difference in model by performing following operations
   Using misclassification cost to heavily penalize false positives
   Generating a Rule Set from already generated Decision
   Tree Preprocessing the data before applying C5.0
   Performing Global Pruning and Winnow attributes
3. Select a field as Split field and generate models for each split.