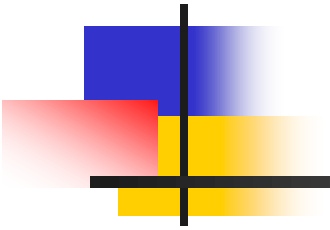


Predictive Models in Text Analytics

Dr. Goutam Chakraborty

Outline

- Recap assessment metrics of predictive models for text analytics
- Explain the basics of Text Categorization models.
- Explain the use of the Text Rule Builder node to categorize and predict text.
- Demonstrate the use of the Text Rule Builder node to categorize and predict text.

Sensitivity, Specificity, Precision, Recall and F1 Score

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Sensitivity =TP/(TP+FN),

Specificity

= TN/(TN+FP)

Overall Correct or, Hit-Ratio

=(TP+TN)/Total

Misclassification = 1-Hit-Ratio

Recall = TP/(TP+FN) , Positive Precision = TP/(TP+FP) = *Positive Predicted Value*, ***PV+***

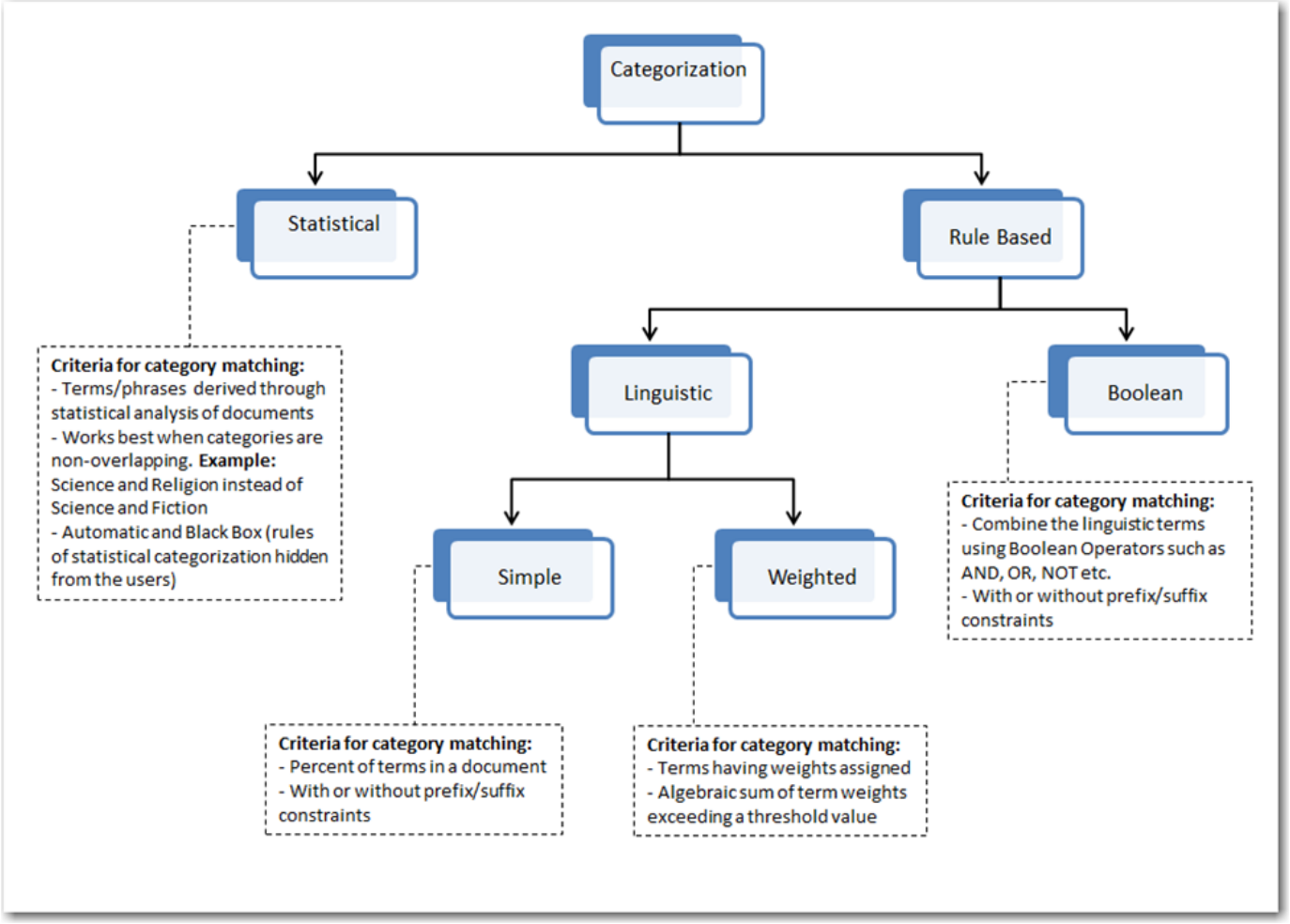
Negative Precision = TN/(TN+FN) = *Negative Predicted Value*, ***PV-***

Overall Precision = (TP+TN)/Total = *Overall Correct or, Hit-Ratio*

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Text Categorization

- It is a process in which text documents are assigned to *a pre-identified* set of categories or topic areas.



Text Rule Builder Node

- It is a Boolean rule-base categorizer that automatically generates an ordered set of rules for describing and predicting a target variable.
- The Text Rule Builder node is a standard SAS Enterprise Miner modeling tool, complete with the standard reporting features such as lift metrics and ROC curves.
  - In addition, it has special features that allows *active learning* so that a user can interact with the algorithm to iteratively build a better predictive model.

Text Rule Builder Properties Panel

- **Generalization Error** — determines the predicted probability for rules that use an untrained data set. This is to *prevent overtraining*. Higher values do a better job of preventing overtraining at a cost of not finding potentially useful rules.
- **Purity of Rules** — determines *how selective each rule is* by controlling the maximum *p*-value necessary to add a term to a rule.
- **Exhaustiveness** — determines the exhaustiveness of the rule search process, or *how many potential rules are considered at each step*. As you increase the exhaustiveness, you increase the amount of time that the node requires and increase the probability of overtraining the model.

Analysis Plan

- **Case** : Categorizing and Predicting Drivers' Feedback.
- **Data**: Professional drivers' feedback about a fuel stop were captured via a mobile app and were classified as positive or negative by the fuel company experts.
- **Goals of this Demonstration**:
  - Use the Text Rule Builder node to automatically classify feedbacks into positive versus negative.
  - Use adaptive learning features of the Text Rule Builder node to improve models.
  - Use the Score node to apply the rules obtained from the Text Rule Builder node to score new feedbacks.

Procedure

- Follow handout titled “Demo with Driver Feedback”

Improving Accuracy of the Model Built So Far

- Refine the dictionary/vocabulary.
  - Revise the start/stop list.
  - Add synonyms.
  - Add entities.
- Use different frequency weights, term weights, or both.
- Add more SVD dimensions.
- Improve SVD weights.
  - Customize term weights in the Text Topic node.
- Continue to refine the Text Rule Builder node by changing assigned target values.