**CS256 – Midterm Exam Study Guide**

**By: Zayd Hammoudeh**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification**  Task of assigning objects to one of several predefined categories. | **Training Set**  A collection of records. Each **record** contains a set of attributes one of which is the **class**. | **Model**  A function from the value of record attributes to the class attribute. | **Test Set**  A collection of records used to determine the accuracy of the classification model. | **Example Classification Techniques**   1. **Neural Networks** 2. **Decision Tree** 3. **Rule Based Classifier** 4. **Memory Based Reasoning** 5. **Support Vector Machines** 6. **Naïve Bayes and Bayesian Belief Networks** |

|  |  |  |
| --- | --- | --- |
| **Induction**  Using a training set to generate a model.  **Deduction**  Process of applying a model to a training set.  **Decision Tree Induction**   * **Greedy Strategy** * **Key Decision #1:** Attribute to expand next * **Key Decision #2:** When to stop expanding | **Hunt’s Decision Tree Induction Algorithm:**   * Let *Dt* be the set of training records that reach a node *t*.  1. If *Dt* contains records that **all belong to the same class *yt***, then *t* is a leaf node with class value *yt*. 2. If *Dt* is an **empty set**, then *t* is a leaf node with default value *yd.* 3. If *Dt* contains **records that belong to more than one class and there are no attributes left**, then *t* is a leaf node with default value is a leaf node with default value *yd*. 4. If *Dt* contains **records that belong to more than one class**, then use an attribute test to split the data into smaller subsets. Recursively apply the same procedure above. | **Attribute Types**   * **Binary** – Attribute with exactly two possible values. * **Nominal** – Two or more class values with no intrinsic Order * **Ordinal** – Two or more class values that can be ordered or ranked * **Continuous** – Quantitative attribute that can be measured along a continuum. |

|  |  |  |
| --- | --- | --- |
| **Splitting Nominal and**  **Ordinal Attributes**   * **Binary** – Divides attribute values into two subsets. **This requires the additional step of finding optimal partitioning.** * **Multi-way** – Use as many partitions as distinct values. | **Splitting Based on Continuous Attributes**   * **Discretization** – Form an ordinal categorical attribute.   + **Static** – Discretize once at the beginning   + **Dynamic** – Ranges can be found by equal interval bucketing, equal frequency bucketing, or clustering. * **Binary Decision** (A < v or A >v) – Consider all possible splits and find the best cut.   + **Binary Decision Procedure:** Go between each training set record value and calculate the GINI index if the splitting point was at that value. **Select the splitting point with the lowest GINISPLIT value**.     - **Computationally inefficient** – where is the number of records. | **Homogeneity/Low** **Impurity** – Extent to which nodes in the decision tree have the same class value/distribution.  **Nodes with high levels of homogeneity (i.e. low levels of impurity) are preferred**. |

**Impurity Measures**

**For all of these metrics, a lower score is generally preferable.**

|  |  |  |
| --- | --- | --- |
| **GINI Index**   * – Node in the decision tree * – Class value * – Number of class values * – Probability (i.e. relative frequency) of class value in node   **Minimum Value:** 0 when:  **Maximum Value:** when: | **GINISPLIT**   * – Child node * – Number of records in parent node. Note: * – Number of child nodes (i.e. attribute partitions) * – GINI index value of node .   **Minimum Value:** 0 when:  **Maximum Value:** when: | **Entropy**   * – Node in the decision tree * – Class value * – Number of class values * – Probability (i.e. relative frequency) of class value in node   **Minimum Value:** 0 when:  **Maximum Value:** when: |

|  |  |  |
| --- | --- | --- |
| **Information Gain** | | **Classification Error**   * – Node in the decision tree * – Class value * – Probability (i.e. relative frequency) of class value in node   **Minimum Value:** 0 when:  **Maximum Value:** when: |
| * – Parent node in the decision tree * – Child node in the decision tree * – Number of child nodes * – Number of records in child node * – Number of records in parent node   **Key Note:** A higher is preferable unlike with the other metrics where a lower value was better.  **Disadvantage of Information Gain:** Tends to prefer splits that result in a large number of partitions, each being small but pure (i.e. overfitting) | **Normalizing for Split Size**  **penalizes a large split by reducing the gain.** |