Unit -4 Classification & Prediction

Classification & Prediction Issues: Overfitting: Creating a model that is too complex and performs well on the training data but poorly on unseen data. Underfitting: Building a model that is too simple to capture the underlying patterns in the data. Imbalanced Data: When one class in the dataset is significantly more prevalent than others, leading to biased models. Feature Selection: Identifying the most relevant features for classification or prediction. Model Interpretability: Ensuring that the classification or prediction model is understandable and interpretable by users. Evaluation Metrics: Choosing appropriate metrics to evaluate the performance of classification or prediction models. Classification by Decision Tree Induction: Decision tree induction is a popular machine learning technique for classification. It involves constructing a tree-like structure where each internal node represents a decision based on the value of a feature, and each leaf node represents a class label. The process typically involves: Selecting the best attribute to split the data at each node based on criteria like information gain or Gini impurity. Recursively partitioning the data until each subset is pure or a stopping criterion is met. Decision trees are easy to interpret and can handle both categorical and numerical data. Bayesian Classification: Bayesian classification is a probabilistic approach to classification based on Bayes' theorem. It calculates the probability of each class given the input features and assigns the class with the highest probability. Key steps include: Estimating prior probabilities of classes based on training data. Calculating likelihood probabilities of features given each class. Combining prior and likelihood probabilities using Bayes' theorem to compute posterior probabilities. Bayesian classification is robust to noisy data and works well for small

datasets. Prediction: Prediction refers to the process of estimating unknown or future values based on historical data and patterns. It involves: Training a predictive model using historical data. Using the trained model to make predictions on new or unseen data. Evaluating the accuracy and reliability of the predictions using appropriate metrics.

datasets. Classification by Back Propagation: Backpropagation is a supervised learning algorithm commonly used for training artificial neural networks, including multi-layer perceptrons (MLPs), for classification tasks. It involves: Forward pass: Propagating input data through the network to generate predictions. Backward pass: Calculating gradients of the loss function with respect to network parameters using the chain rule. Updating network weights using gradient descent or its variants to minimize the loss function. Backpropagation is effective for complex classification problems but may require careful tuning of hyperparameters and regularization techniques to prevent overfitting. Associative Classification: Associative classification combines association rule mining with classification techniques to build classification models. It involves: Mining association rules from the training data. Using these rules to classify new instances based on their association with classes. Applying pruning and optimization techniques to improve the efficiency and accuracy of the classification process. Associative classification leverages the strengths of both association rule mining and classification, potentially leading to more accurate models. Nearest Neighbor Classification: Nearest neighbor classification is a simple yet effective lazy learning algorithm for classification. It classifies new instances based on the majority class among their nearest neighbors in the feature space. Key steps include: Calculating distances between the new instance and all training instances. Selecting the k nearest neighbors based on distance metrics like Euclidean distance. Assigning the class label based on the majority class among the k neighbors. Nearest neighbor classification is intuitive and easy to implement but may suffer from high computational cost, especially with large