Faculty of Computing and Information Technology

University of the Punjab,

Lahore

Artificial Intelligence Lab 15

Instructor: Qamar U Zaman

Naive Bayes

Objective

- Understand the basics of the Naive Bayes algorithm.
- Implement a Naive Bayes classifier for binary text classification.
- Train the classifier on a small dataset and evaluate its performance.

Key Concepts

1. **Naive Bayes Algorithm**: A probabilistic classifier based on Bayes' theorem, assuming independence between features.
2. **Probability Calculation**: Use prior probabilities and likelihoods to classify data.
3. **Evaluation Metrics**: Use accuracy to evaluate the classifier's performance.

Dataset

Text	Label
"I love this movie"	1
"This film is fantastic"	1
"What an amazing experience"	1
"I dislike this movie"	0
"Not a great film"	0
"This is terrible"	0

Tasks for Students

1. **Dataset Preparation**
   - Define the dataset with text and labels.
2. **Text Preprocessing**
   - Convert text to lowercase, remove punctuation, and split into words.
3. **Vocabulary Creation**
   - Build a vocabulary of unique words from the dataset.
4. **Probability Calculation**
   - Calculate prior probabilities for each class and likelihoods for words.
5. **Classifier Implementation**
   - Implement the Naive Bayes algorithm to classify new text.
6. **Evaluation**
   - Compute accuracy by testing the classifier on a small test set.

Code Template

```
# Step 1: Preprocess text
def preprocess_text(text):
    """
    Convert text to lowercase, remove punctuation, and split into words.
    """
    pass

# Step 2: Build vocabulary
def build_vocabulary(dataset):
    """
    Create a list of unique words from the dataset.
    """
    pass

# Step 3: Calculate probabilities
def calculate_probabilities(dataset, vocabulary):
    """
    Compute prior probabilities and likelihoods for words.
    """
    pass

# Step 4: Naive Bayes classifier
def naive_bayes_classifier(text, prior_probs, word_likelihoods, vocabulary):
    """
    Classify new text using the Naive Bayes algorithm.
    """
    pass

# Step 5: Evaluate the classifier
def evaluate_classifier(test_data, prior_probs, word_likelihoods,
vocabulary):
    """
    Compute the accuracy of the classifier.
    """
    pass
```