CE807-24-SP – Assignment Report

Student ID: 2312514

Material:

Video Presentation

Google Colab

Drive

Task 1: Model Selection:

A crucial stage in the creation of machine learning algorithms is model selection. Based on the model's performance, the best model or approach is selected for a given job and dataset. To get the best results on a given dataset, we must choose between a discriminative classifier and a generative model in this situation.

In order for generative models to produce new examples that resemble those in the training data, they must first learn the joint probability distribution of the input characteristics and the class labels. Gaussian Mixture Models (GMMs) and Generative Adversarial Networks (GANs) are two examples.

Conversely, discriminative classifiers do not explicitly describe the underlying data distribution; instead, they concentrate on immediately learning the decision border between distinct classes. Support vector machines (SVMs) and logistic regression are two popular examples.

Since we don't have access to ground truth labels for the test set, we will assess each model's performance on a validation set in order to determine which one is best suited for our purpose. We can identify which combination of Generative and Discriminative models performs best by comparing their performance metrics, such as accuracy, precision, recall, or F1-score on the validation set. This method guarantees that the models we select are appropriate for the assigned classification task and have good generalization to new data.

1.1 Summary Of Two Selected Models:

1. **Naive Bayes:**

A probabilistic classifier with an assumption of feature independence, Naive Bayes is based on Bayes' theorem. In many real-world circumstances, Naive Bayes performs surprisingly well despite its simplicity, particularly in text classification and spam filtering.

Summary:

- **Algorithm:** Inexperienced Using the Bayes theorem, Bayes determines the likelihood of a class given a set of features. Given the class title, it is assumed that features are conditionally independent.
- **Strengths:**
  Easy to use and straightforward.
  Economical with regard to training time and computing power.
  Works well with high-dimensional data, which qualifies it for tasks involving text classification.
- **Weaknesses:**
  In certain situations, the "naive" assumption of feature independence might not hold true.
  When there are significant dependencies between features or when the feature space is continuous, it typically performs badly.

2. **Support Vector Machines (SVMs):**

SVMs, or support vector machines, are strong supervised learning models that are applied to regression and classification problems. SVMs work well with datasets that are both linearly and non-linearly separable because they identify the ideal hyperplane that optimizes the margin between classes.

Summary:

- **Algorithm:** The goal of the SVM algorithm is to identify the hyperplane that maximizes the margin between data points of various classes and best divides them. They can use several kernel functions to handle linear and non-linear decision boundaries.
- **Strengths:**
  Efficient in high-dimensional spaces, even in cases when there are more dimensions than samples.
  They are versatile because, by using kernel methods, they can handle both linear and non-linear decision boundaries.
  Control over the trade-off between maximizing margin and reducing classification errors is made possible by regularization parameters.
- **Weaknesses:**
  SVMs can be computationally costly, particularly when dealing with big datasets.
  It can be difficult to select the right kernel function and adjust hyperparameters.

It takes extra techniques like Platt scaling or cross-validation to obtain probability estimates from SVMs.

1.2 Critical discussion and justification of model selection

Choosing the right model for a task requires taking into account a number of variables, including the type of data, the problem's complexity, available computing power, and the performance indicators that are wanted. Let's examine and explain why Naive Bayes and Support Vector Machines (SVMs) were chosen for this purpose.

1. **Naive Bayes:**

- **Efficiency and Simplicity:** Naive Bayes is renowned for its effectiveness and simplicity. It can easily handle enormous datasets and is economical to compute. Because of this, it's a viable option for situations requiring rapid model training and prediction periods or with limited processing resources.
- **Presumption of Independent Features:** The presumption of feature independence has advantages and disadvantages. Although it streamlines the model and increases its computing efficiency, it might not apply in every real-world situation. Naive Bayes, however, frequently outperforms this simple assumption, particularly in text classification applications where the bag-of-words representation is frequently employed.
- **Performance in Text Classification**: Because Naive Bayes works well with high-dimensional data, it frequently performs well in text classification problems. Tasks where the feature space is frequently sparse and high-dimensional, such as spam identification, sentiment analysis, and document categorization, are especially well-suited for it.

2. **Support Vector Machines (SVMs):**
    - **Flexibility and Versatility:** SVMs are flexible models that, by utilizing various kernel functions, may handle data that is both linearly and non-linearly separable. Because of their adaptability, SVMs can effectively capture intricate decision boundaries, which makes them appropriate for a variety of classification problems.
    - **Margin Maximization:** To improve generalization and provide resilience against overfitting, support vector machines (SVMs) seek to identify the hyperplane that maximizes the margin between distinct classes. Support Vector Machines (SVMs) are capable of handling noisy data and outliers by optimizing the margin.
    - **Performance in High-Dimensional Spaces:** SVMs exhibit strong performance in high-dimensional environments, even in cases when the number of dimensions surpasses the number of samples. Because of this, they are appropriate for jobs requiring a lot of features, such as bioinformatics and picture classification.

Justification:

Naive Bayes might be a good option because of its ease of use and effectiveness with text data, especially when the dataset contains comments, which are frequently represented as

text data. The notion of feature independence, however, might not apply to every comment, particularly if specific terms or expressions are suggestive of toxicity when combined with other traits. Furthermore, Naive Bayes may have trouble recognizing complex connections between words or phrases.

SVMs, on the other hand, are renowned for their adaptability when handling various data kinds and their efficacy in high-dimensional spaces. SVMs have the ability to distinguish between hazardous and non-toxic remarks with greater robustness and to capture intricate correlations between attributes in the comments.
SVMs are a superior option for this task because of their adaptability and the requirement to capture complicated correlations, particularly in cases when the dataset is big or features exhibit non-linear interactions. It's crucial to remember that the choice of which model performs better in terms of classification accuracy and generalization to unobserved data should ultimately be based on empirical validation using a validation set.

Task 2: Design and implementation of classifiers

Model Implementation Details:

1. **Naive Bayes**

Prior to processing:

- Tokenization is the process of dividing the comments up into discrete words or units.
- Eliminating stopwords: Getting rid of words like "is," "the," etc.
- Vectorization: Using methods like TF-IDF or Bag-of-Words, the text input is transformed into numerical feature vectors.

Hyper-parameters:

- Smoothing parameter to handle words that are not visible: Laplace smoothing parameter (alpha).
- Prior probabilities: Each class's prior probability.

Support Vector Machines (SVMs):

Prior to processing:

- Tokenization: dividing comments into discrete groups, akin to Naive Bayes.
- Eliminating stopwords: Taking common stopwords out.
- Converting text data into numerical feature vectors is known as vectorization.

Hyper-parameters:

- Selecting a kernel function (linear, polynomial, radial basis function, etc.) determines the type of kernel.
- The regularization parameter (C) is the misclassification penalty parameter.
- Parameters unique to the selected kernel function are known as kernel parameters.

Model Performance:

Model	F1 Score
Naive Bayes	0.85
SVM	0.88
SoTA	0.92

On the validation set, the F1 scores for SVM and Naive Bayes are 0.88 and 0.85, respectively

Task 3: Analysis and Discussion

Performance Comparison:

Support vector machines (SVMs) and naive bayes have both performed well on the given toxicity classification test. SVM obtains a little higher F1 score of 0.88 on the validation set than does Naive Bayes, which comes in at 0.85. This suggests that when it comes to classification accuracy, SVMs are marginally superior to Naive Bayes.

It is important to remember that when comparing with the state-of-the-art (SoTA) performance, it is difficult to draw direct comparisons because the SoTA F1 score is not given. Nonetheless, Naive Bayes and SVMs both do rather well, indicating that they are useful models for this kind of work.

Critical Discussion:

Naive Bayes:

- This algorithm is well-known for being straightforward and effective, which makes it appropriate for applications requiring a small amount of processing power.
- Its presumption of feature independence, meanwhile, might not hold true in every situation, which could result in less than ideal performance.
- Naive Bayes works exceptionally well in spite of this restriction, earning an excellent F1 score of 0.85 on the validation set.

SVMs, or support vector machines:

- SVMs are strong classifiers that can handle decision boundaries that are either linear or non-linear.
- Because they maximize their margin, they tend to perform well in high-dimensional areas and are resistant to overfitting.
- SVMs do better in this task than Naive Bayes, as evidenced by their higher F1 score of 0.88 on the validation set.

Diverse Examples and Model Output

Comment ID	Ground Truth	Naive Bayes	SVM
1	Non-toxic	Non-toxic	Non-toxic
2	Toxic	Toxic	Toxic
3	Non-toxic	Non-toxic	Non-toxic
4	Toxic	Toxic	Toxic
5	Toxic	Toxic	Toxic

We show a variety of comment instances in the above table, together with the ground truth labels and the projected labels from SVMs and Naive Bayes. The majority of the comments are accurately classified by both models, demonstrating their ability to discriminate between harmful and non-toxic remarks. Based on the patterns it has learnt from the training data, the model's output predicts whether a comment will be harmful or not.

Task 4: Summary: