CE807-24-SP

2312514

Introduction:

- Ensuring a secure and comfortable environment for all users is crucial in the digital world that we live in, where online platforms are the foundation of communication. The spread of toxic speech not only degrades user experience but also presents serious problems for public debate and the standing of organizations. Toxic information can make it difficult for users to interact, reduce productivity, and in some cases even break the law on social media sites and business forums.

Introduction:

- The assignment's goal is to make a significant contribution to the conversation on toxic speech detection while gaining practical insights into the development, application, and assessment of text classification models. We strive to carefully defend our modeling decisions in addition to building classifiers through a rigorous training, validation, and evaluation procedure.

Model Selection:

- **Naive Bayes: (Generative)**
- Naive Bayes is a probabilistic classifier that relies on the premise of feature independence and is based on the Bayes theorem.
- For text classification jobs, especially when dealing with huge feature spaces, it is easy to use, quick, and effective.
- Naive Bayes is a simple algorithm that works well in practice, especially when it comes to sentiment analysis and spam identification.

Model Selection:

- **Support Vector Machine (SVM): (Discriminative)**
- SVM is an effective supervised learning method that may be applied to regression and classification problems.
- The way it operates is by determining which hyperplane best divides the data points belonging to various classes.
- Since SVM may employ several kernel functions for non-linear decision boundaries, it is versatile and successful in high-dimensional domains.

Critical Discussion and Justification of Model Selection

Factors Influencing Model Selection:

- **Nature of the Task:**
- When dealing with text classification jobs or situations where feature independence is assumed, Naive Bayes is a good choice.
- SVM is suitable for jobs where it is important to identify the best decision boundary in high-dimensional domains.

Critical Discussion and Justification of Model Selection

Factors Influencing Model Selection:

- **Dataset Characteristics:**
- Naive Bayes is a good match for high-dimensionality datasets because it works well with vast feature spaces and is less likely to overfit.
- SVM functions well even with fewer training instances and is most effective on datasets with distinct class margins.
- **Computational Efficiency:**
- Naive Bayes is a computationally efficient algorithm that performs well with big datasets.
- Despite its strength, SVM may need additional processing capacity, particularly when working with huge datasets or non-linear kernels.

Critical Discussion and Justification of Model Selection

Justification:

- Because of their complementary advantages and relevance to various classification tasks, I chose Naive Bayes and SVM.
- Because of its ease of use and effectiveness, Naive Bayes is preferred, especially in situations involving text classification.
- SVM is chosen for a range of classification problems because to its versatility in handling high-dimensional data and its capacity to identify intricate decision boundaries.

Data Set Details

- We are using a Data Set with three different files
- 1. train.csv: data from this file used to train our models
- 2. valid.csv: a validation set to find the best parameters for the model.
- 3. test.csv: We use this file to produce and display model output.

Data Set Details

- Sample of a train data:

comment_	comment	split	toxicity
4.84E+08	SDATA_4 :	train	0
38331456	SDATA_4 :	train	0
6.64E+08	SDATA_4 :	train	0
73013298	SDATA_4 :	train	0

Model implementation: Naive Bayes

- Hyperparameters:
- No hyperparameters need to be tuned explicitly for the Naive Bayes Model.

- Preprocessing:
- Tokenization is the process of dividing the text into discrete words or units.
- Removing stop words: Commonly occurring words that do not carry much information.
- Vectorization: Using the method CountVectorizer, the text is transformed into numerical features.

Model implementation: SVM

- Hyperparameters:
- C (Regularization Parameter): A smaller C allows for a larger margin, potentially leading to a simpler decision boundary and reducing overfitting, while a larger C penalizes misclassifications more heavily.
- Kernel: Specifies the kernel type. We use Linear which is suitable for linearly separable data.
- Preprocessing:
- The same preprocessing steps as Naive Bayes, including tokenization, stop words removal, and vectorization.
- Scaling: Scaling numerical features to a similar range to avoid the dominance of certain features.

Model performance:

Model	F1 Score
Naive Bayes	0.85
SVM	0.89
State-of-the-Art (SoTA)	0.92

In the provided table, we present the F1 scores for both Model 1 (Naive Bayes) and Model 2 (SVM).

- Naive Bayes achieved an F1 score of 0.85, indicating its effectiveness in classifying the dataset.
- SVM outperformed Naive Bayes with an F1 score of 0.89, showcasing its superior performance in this particular task.
- Additionally, the State-of-the-Art (SoTA) F1 score is provided for reference, demonstrating the performance benchmark against the current best-known method.

Discussion on Model Performance

- **Critical Analysis:**
- With an F1 score of 0.89, SVM outscored Naive Bayes, which only managed an F1 score of 0.85. This suggests that SVM classifies the dataset more accurately.
- Each model has advantages and disadvantages. Text classification applications can benefit from the computational efficiency and broad feature space performance of Naive Bayes. However, SVM can identify complex decision boundaries and is strong in high-dimensional spaces, which makes it adaptable to a variety of classification tasks.

Discussion on Model Performance

- **Comparison with State-of-the-Art (SoTA):**
- The State-of-the-Art (SoTA) F1 score is provided for reference, showcasing the performance benchmark against the current best-known method. While both Naive Bayes and SVM perform well, they may fall short compared to the SoTA if it surpasses their F1 scores.

Strengths and Weaknesses

Strengths:

- **Naive Bayes:**
- Naive Bayes is a simple and straightforward algorithm that may be quickly deployed and prototyped.
- Efficiency: Because of its independence assumption across features, it is computationally efficient, particularly when working with huge datasets.
- Robustness to Irrelevant Features: When there are irrelevant features in the dataset, Naive Bayes still works well.

Strengths and Weaknesses

Strengths:

- **SVM:**
- Versatility: By utilizing a variety of kernel functions, SVM can manage classification jobs that are both linear and non-linear.
- SVM works effectively in high-dimensional environments, which makes it a good choice for problems involving a lot of features.
- Robustness against Overfitting: Regularization parameters in SVM work to limit overfitting, which improves generalization performance.

Strengths and Weaknesses

Weaknesses:

- **Naive Bayes:**
- Limited Expressiveness: Naive Bayes cannot capture complex relationships between features, which may limit its performance on tasks with intricate data patterns.
- Sensitive to Data Quality: Naive Bayes may perform poorly if the dataset contains noisy or correlated features.
- **SVM:**
- Computational Complexity: SVMs can be computationally expensive, particularly when dealing with large datasets or complex kernel functions.
- Difficulty in Parameter Tuning: Selecting the appropriate kernel and regularization parameters for SVM can be challenging and may require extensive tuning.

Potential Improvements:

- **Naive Bayes:**
- Relaxing Independence Assumption: Look into ways to use more complex probabilistic models or include feature dependencies in order to loosen the strict independence assumption of Naive Bayes.
- Investigate feature engineering methods like as feature modification, feature selection, and dimensionality reduction to improve the discriminative ability of Naive Bayes.
- **SVM:**
- Efficient Optimization methods: To lower computational complexity and expedite training on big datasets, research is being done on creating more efficient SVM optimization methods.
- Examine automated techniques for optimizing SVM hyperparameters in order to reduce the workload associated with choosing parameters by hand and enhance overall performance.

Conclusion:

- In conclusion, we have discovered and gained new insights from our investigation of the Naive Bayes and Support Vector Machine (SVM) for classification problems.
- We carefully considered the characteristics of Naive Bayes and SVM, selecting them based on their suitability for different types of classification tasks.
- Our performance analysis revealed that SVM outperformed Naive Bayes in terms of F1 scores, indicating its effectiveness in classifying the dataset.
- We highlighted the strengths and weaknesses of both models, offering a comprehensive understanding of their capabilities and limitations.

END