COL774 Assignment - 1

Vidushi Maheshwari (2021CS10083)

March 19, 2025

Contents

§1 Text Classification using Naive Bayes

§1.1 Naïve Bayes Multiclass Classifier

(a) Accuracy

- Train Accuracy: 91.33%
- Test Accuracy: 89.11%

(b) Word Cloud Representation

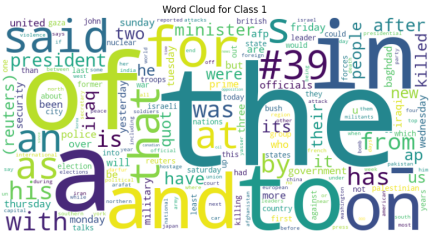


Figure 1: World

Figure 2: Sports

Figure 3: Business

Figure 4: Science

§1.2 Stopword Removal and Stemming

(a) Accuracy

- Train Accuracy: 91.25%
- Test Accuracy: 89.26%

(b) Word Cloud Representation

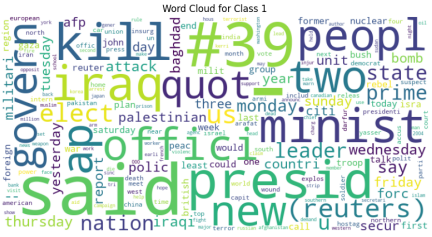


Figure 5: World

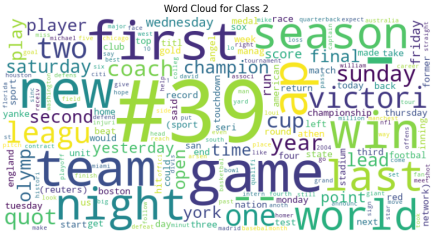


Figure 6: Sports

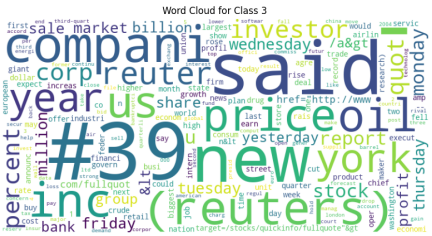


Figure 7: Business

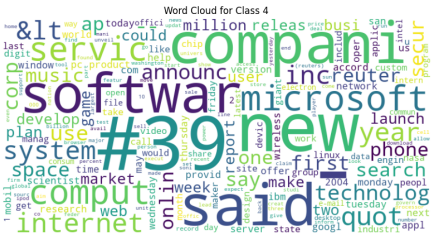


Figure 8: Science

(c) Observations

- Observed increase in test accuracy from 89.11% to 89.26% after applying pre-processing techniques.
- Stopword removal helped remove irrelevant noise, leading to a slight improvement in accuracy.
- Stemming reduced vocabulary size but caused loss of meaning in case of many words, affecting classification performance.
- Since there are lesser features/words in vocabulary, the training accuracy decreases. Parallelly, reduced overfitting leads to higher testing accuracy.
- The word cloud visualization showed that after preprocessing, meaningful words became more prominent, improving class distinction.

§1.3 Feature Engineering using Bigrams

- Train Accuracy: 95.23%
- Test Accuracy: 90.29%

§1.4 Model Comparision

- Model 1: Unigram model without stopword removal or stemming.
- Model 2: Unigram model with stopword removal and stemming.
- Model 3: Bigram model with stopword removal and stemming.

(a) Scores for different models

Model	Accuracy	Precision	Recall	F1-score
Model 1	0.891053	0.890780	0.891053	0.890740
Model 2	0.892632	0.892162	0.892632	0.892311
Model 3	0.902895	0.902650	0.902895	0.902669

Table 1: Performance comparison of different models

(b) Analysis

Model 1 vs. Model 2

Comparing Model 1 and Model 2, we observe a slight improvement across all metrics. This suggests that stopword removal and stemming improve classification performance by reducing noise and standardizing word forms.

Model 2 vs. Model 3

Model 3, which uses bigrams, significantly outperforms both unigram models. The 1% increase in accuracy and F1-score suggests that bigrams capture contextual meaning more effectively, leading to better classification.

(c) Conclusion

Model 3 is the best performing model across all metrics i.e. introduction of bigrams improved classification accuracy significantly.

§1.5 Modelling on Title Feature

(a) Scores for different models

Model	Accuracy	Precision	Recall	F1-score
Model 1	0.862877	0.862347	0.862870	0.862452
Model 2	0.859982	0.859653	0.859975	0.859673
Model 3	0.871957	0.871728	0.871951	0.871698

Table 2: Performance comparison of different models

(b) Analysis

Model 1 vs. Model 2

Comparing Model 1 and Model 2, we observe a slight drop in performance across all metrics. This indicates that while stopword removal and stemming may help reduce noise, they might also remove useful information from the title.

Model 2 vs. Model 3

Model 3, which incorporates bigrams, shows a clear improvement over both unigram models. The 1% increase in accuracy and F1-score suggests that bigrams capture contextual relationships better, leading to improved classification.

(c) Conclusion

Again, Model 3 is the best performing model across all metrics.

(d) Comparision with description based model

- The best description-based model achieves a higher test accuracy (0.9029) than the title-based model (0.8718). This suggests that the description text contains more information than the title, leading to better classification performance.
- The description of an article is usually more detailed and provides better context than the title alone. Titles are often short and may lack sufficient discriminative information for classification.

§1.6 Combining Title and Description Features

Performance by bigram model without preprocessing

- Description: Train Accuracy: 94.54%, Test Accuracy: 89.78%
- Title: Train Accuracy: 94.31%, Test Accuracy: 87.18%

Optimal Approach: Bigram with preprocessing for both description and title.

(a) Concatenated Feature Representation

Model	Train Accuracy	Test Accuracy
Title Only	94.31%	87.18%
Description Only	95.23%	90.29%
Title + Description (Concatenated)	**95.55%**	**91.17%**

Table 3: Accuracy comparison of different Naïve Bayes models.

Concatenating Title & Description improves accuracy:

- Train accuracy increased slightly (+0.32%) compared to using the description alone.
- Test accuracy also improved (+0.88%), showing better generalization.

The improvement from 90.29% to 91.17% suggests that combining both features provides additional useful information.

(b) Learning Separate Parameters

1. Mathematical Formulation

Given a class C, we define two independent probability distributions:

- $P(w_t|C)$ for words in the **title**
- $P(w_d|C)$ for words in the **description**

The final probability of a class given a document is then computed as:

$$P(C|T, D) \propto P(C) \cdot P(T|C) \cdot P(D|C)$$

where:

- $P(C)$ is the prior probability of class C
- $P(T|C)$ is the probability of observing the title given class C
- $P(D|C)$ is the probability of observing the description given class C

Each probability is estimated using MLE with Laplace Smoothing:

$$P(w_t|C) = \frac{\text{count}(w_t, C) + \alpha}{\sum_{w \in V_t}(\text{count}(w, C) + \alpha)}$$

$$P(w_d|C) = \frac{\text{count}(w_d, C) + \alpha}{\sum_{w \in V_d}(\text{count}(w, C) + \alpha)}$$

where:

- $\text{count}(w_t, C)$ and $\text{count}(w_d, C)$ are word occurrences in the title and description, respectively.
- V_t and V_d are vocabulary sizes for title and description.
- α is the Laplace smoothing parameter.

During prediction, the class $\hat{C}$ for a new instance is determined by:

$$\hat{C} = \arg\max_C \log P(C) + \sum_{w_t \in T} \log P(w_t|C) + \sum_{w_d \in D} \log P(w_d|C)$$

2. Results

Model	Train Accuracy	Test Accuracy
Title Only	94.31%	87.18%
Description Only	95.23%	90.29%
Title + Description (Concatenated)	**95.55%**	**91.17%**
Separate Parameters (Title and Description)	**95.76%**	90.96%

Table 4: Accuracy comparison of different Naïve Bayes models.

3. Observations

Accuracy Comparison

- The Concatenated model (91.17% test accuracy) achieves the highest performance.
- The Separate θ model (90.96% test accuracy) slightly underperforms compared to the concatenated model, despite having the highest training accuracy (95.76%).
- Both Title-only (87.18%) and Description-only (90.29%) models perform worse than the combined models, confirming that information gain from combination of both features makes the classification more effective.

Overfitting in the Separate θ Model

- The Separate θ model has a larger gap between training (95.76%) and test (90.96%) accuracy, suggesting mild overfitting.
- Learning separate parameters increases model complexity, leading to better performance on training data but slightly reduced generalization.

Why Does Simple Concatenation Work Better?

- Treating title and description as a single feature set allows the model to learn stronger word associations across both fields.
- Separate probability distributions limit this effect.

The best-performing model is the **Concatenated approach** (91.17% test accuracy).

§1.7 Comparision with Baseline Models

(a) Random Guessing

Expected Accuracy: 25% Obtained Accuracy: 25.29%

(b) Positive Prediction

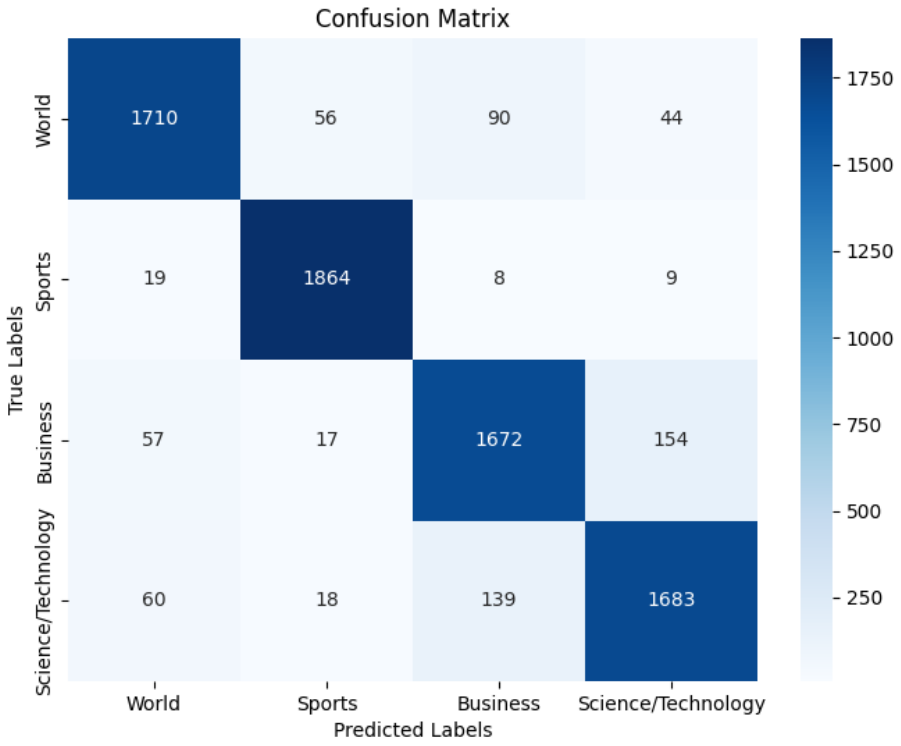
Obtained Accuracy: 25%
We guessed the most frequently occuring class as the prediction for all test datapoints. Since there is an even split among all the classes, the obtained accuracy is 25% as expected.

(c) Concatenation vs baseline models

- Improvement over Random Guessing = 91.17% - 25.29% = 65.88%
- Improvement over Predicting as Positive = 91.17% - 25.00% = 66.17%

The concatenated model shows a substantial improvement of approximately 66% over both baselines, demonstrating that utilizing both the title and description together provides much more informative features compared to simple baseline approaches.

§1.8 Confusion Matrix for Concatenated Model



Sports has the highest number of diagonal entries. This indicates that the model is better at detecting Sports articles over other categories.

§1.9 Feature Engineering

(a) Capitalized Words

News headlines often use capitalized words to emphasize key entities (e.g., company names, locations, or political figures). We hypothesized that tracking the number of capitalized words could provide useful information for classification.

- Train Accuracy: 95.55
- Test Accuracy: 91.07

The slight decrease in test accuracy suggests that capitalization patterns are not a strong distinguishing factor for classification. While some categories may use capitalization more frequently, the variation within each category likely reduces its overall effectiveness as a feature.

(b) Punctuation Count

Different news categories might exhibit varying punctuation usage. For example, opinion pieces or entertainment news might use more exclamation marks, while financial or political news might have more structured punctuation.

- Train Accuracy: 95.56
- Test Accuracy: 91.21

The test accuracy slightly improved, indicating that punctuation carries some distinguishing patterns among categories. However, the improvement is minimal, suggesting that while punctuation might provide some useful signal, it is not a dominant factor in classification performance.

§2 Image Classification using SVM

Binary Classification: Rain vs Rainbow

§2.1 Linear SVM

- Number of Support Vectors: 323
- Percentage of Training Samples as Support Vectors: 53.39%
- Test Accuracy: 82.35%
- Bias = -2.0357

The high percentage of support vectors suggests significant overlap in features between the images, indicating that decision boundary is not easily separable with a linear kernel.

Figure 9: Top-5 Support Vector

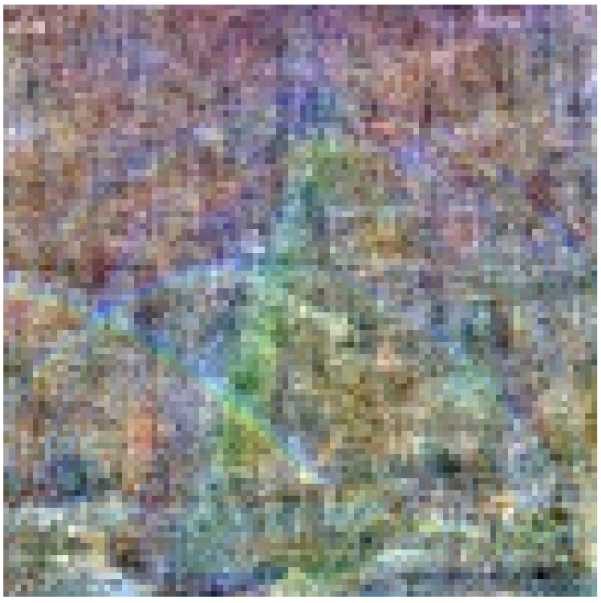


Figure 10: Weight Vector

- The top 5 support vectors seem to be dominated by rainbow images, indicating that these examples are close to the decision boundary. These images exhibit diverse lighting conditions, background variations, and possible blending of rain and rainbow elements, making them difficult to classify confidently. This suggests that the model struggles with distinguishing between rain and rainbow when they appear together or under certain lighting conditions.

- The weight vector visualization appears noisy but seems to capture subtle structural patterns. There are hints of diagonal and curved color gradients, possibly indicating an emphasis on rainbow-like structures or rain streaks as key discriminative features. The noise in the weight vector suggests that some redundant or irrelevant patterns might be influencing classification, meaning feature selection or a non-linear SVM might improve results.

§2.2 SVM using Gaussian Kernel

- Number of Support Vectors: 407
- Percentage of Training Samples as Support Vectors: 67.27%
- Matching Support Vectors: 261
- Test Accuracy: 90.20%

Figure 11: Top-5 Support Vector

Comparision with Linear SVM

- The Gaussian SVM significantly outperforms the Linear SVM, improving accuracy by **7.85%**. This suggests that the decision boundary for distinguishing rain and rainbow images is non-linear, which the Gaussian kernel can model better.
- The Gaussian SVM uses 84 more support vectors than the Linear SVM. The increase in support vectors indicates a more complex model, but this results in **better generalization**.
- 261 out of 323 support vectors from the Linear SVM also appear in the Gaussian SVM. This means that 80.8% of the key linear boundary-defining examples remain the same, but the Gaussian SVM introduces additional vectors for refinement.

§2.3 Scikit-Learn SVM

(a) Support Vectors

Number of Linear Support Vectors: 322
Number of Gaussian Support Vectors: 390

	Linear	Gaussian	Sklearn Linear	Sklearn Gaussian
Linear	323	265	323	260
Gaussian	265	407	266	393
Sklearn Linear	320	264	322	259
Sklearn Gaussian	257	390	258	390

Table 5: Number of Matching Support Vectors Between Models

(b) Linear Weights and Biases

CVXOPT Bias = -2.0357
SkLearn Bias = -1.90426

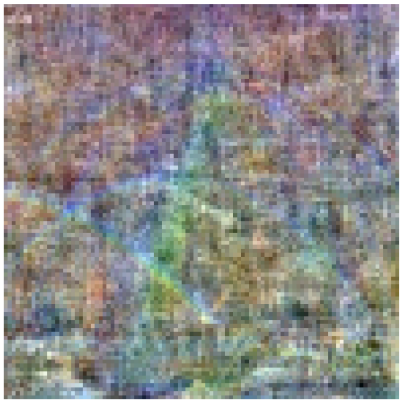


Figure 12: CVXOPT Weight Vector

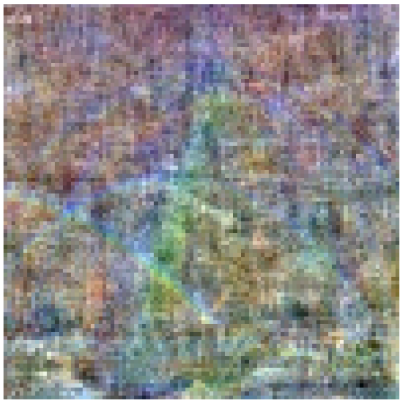


Figure 13: Sklearn Weight Vector

$$\|\mathbf{w}_{\text{CVXOPT}} - \mathbf{w}_{\text{Sklearn}}\|_2 = 0.001613$$

$$|b_{\text{CVXOPT}} - b_{\text{Sklearn}}| = 0.131438$$

This extremely small difference in weights suggests that both solvers are computing nearly identical decision boundaries. The bias difference is slightly larger, which might be due to different solver approaches.

(c) Test Accuracy Comparision

SVM Model	**Test Accuracy (%)**
CVXOPT Linear	82.35
CVXOPT Gaussian	90.20
Sklearn Linear	82.35
Sklearn Gaussian	89.54

The accuracy of both linears models is almost same. Scikit-Learn's Gaussian SVM slight underperforms as compared to CVXOPT one (by 0.66%).

(d) Computation Costs

SVM Model	Training Time (sec)
CVXOPT Linear	0.26
CVXOPT Gaussian	0.27
Sklearn Linear	1.73
Sklearn Gaussian	1.43

It is evident that CVXOPT is significantly faster for both linear and Gaussian kernels compared to Scikit-Learn. This is likely due to the quadratic programming solver in CVXOPT, which is optimized for smaller datasets, whereas Scikit-Learn's solver may involve additional overhead.

§2.4 SVM using SGD algorithm

Metric	LIBLINEAR	SGD
Training Time (sec)	1.8302	1.0707
Test Accuracy (%)	82.35	84.97

- Training Time: The SGD solver is significantly faster (1.07 sec) than LIBLINEAR (1.83 sec), as it performs stochastic gradient descent instead of solving the full optimization problem.
- Test Accuracy: Surprisingly, the SGD solver achieves a higher test accuracy (84.97%) compared to LIBLINEAR (82.35%). This suggests that despite being an approximate method, SGD converges well for this dataset.
- Optimization Approach: LIBLINEAR solves the dual problem exactly, which can be computationally expensive, while SGD updates weights incrementally, making it more scalable to large datasets.
- Variability in Performance: SGD can have fluctuating performance depending on hyperparameters like learning rate and regularization, whereas LIBLINEAR provides more stable results.

Multi-Class Image Classification

§2.5 CVOXPT One-vs-One SVM

- We extend the binary SVM formulation to a multi-class setting using the One-vs-One (OvO) approach.
- Given $k = 11$ classes, we train $\binom{k}{2} = 55$ binary classifiers using our CVXOPT solver with a Gaussian kernel ($C = 1.0, \gamma = 0.001$).
- During inference, each classifier casts a vote, and the final predicted class is determined by the majority vote. In case of ties, we select the class with the highest score.
- Test Accuracy : 66.45%

§2.6 Scikit-Learn SVM

Test Accuracy: 66.30 %

CVOXPT vs Scikit-Learn Multi-Class SVM

Model	Test Accuracy (%)	Training Time (sec)
CVXOPT (Gaussian SVM)	66.45	47.6832
Sklearn (LIBSVM)	66.30	202.8141

Observations

- Accuracy is nearly identical for both models (CVXOPT: 66.45% vs. LIBSVM: 66.30%).
- CVXOPT trains 4.25× faster (47.68 sec vs. 202.81 sec) due to its interior-point QP solver, which efficiently handles small to medium datasets.
- LIBSVM updates iteratively, while CVXOPT directly solves the QP problem, reducing computational overhead but potentially scaling worse for larger datasets.

§2.7 Confusion Matrix

(a) CVOXPT SVM

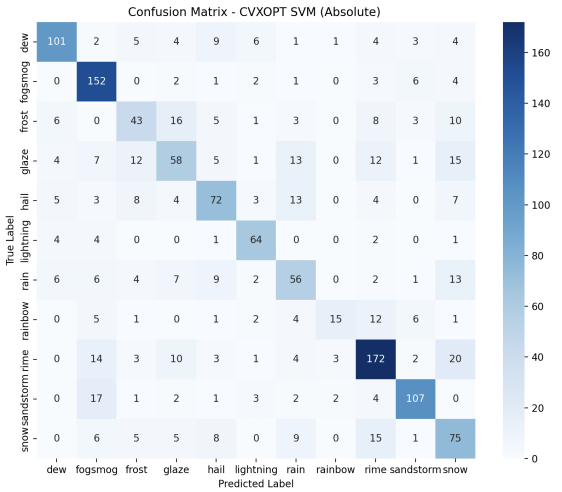


Figure 14: Absolute

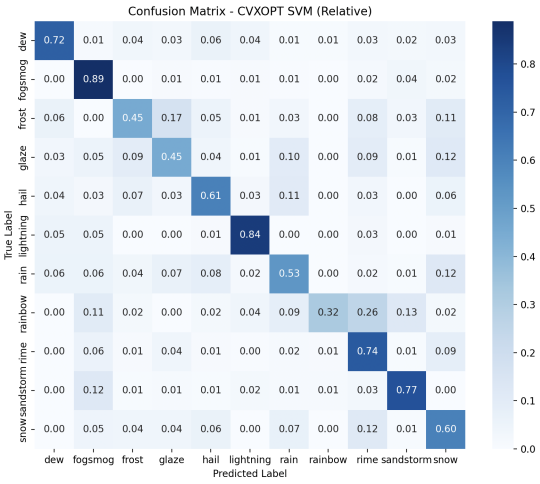
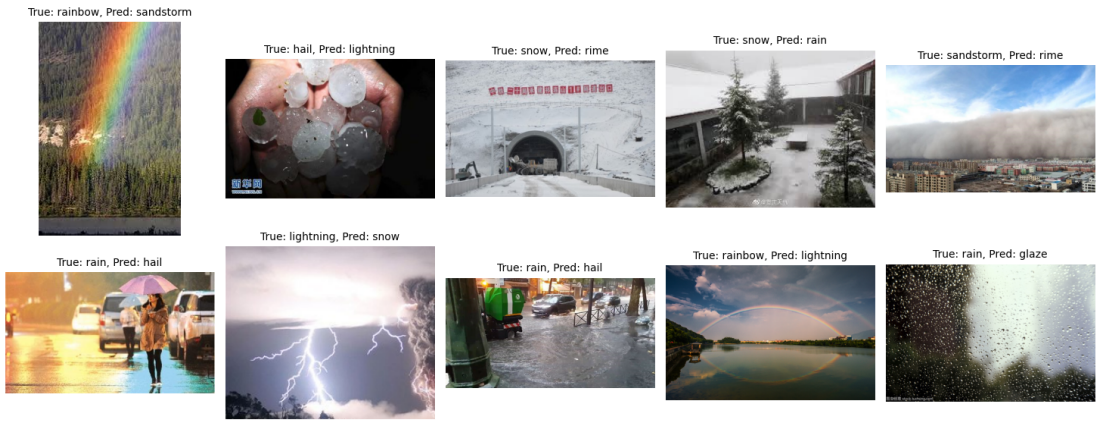


Figure 15: Relative

- **Top Performing Class:** Fogsmog has 152/171 correct (88.9% accuracy)
- **Worst Performing Classes:**
  - Rainbow has 15/47 correct (31.9% accuracy) - frequently confused with rime (12) and rain (4).
  - Frost has 43/95 correct (45.3% accuracy) - confused with glaze (16) and snow (10).
- **Common Misclassification Patterns:**
  - Frost→glaze: 16, glaze→frost: 12, glaze→snow: 15
  - Sandstorm→fogsmog: 17 likely due to similar hazy appearance.
  - Rime→snow: 20, snow→rime: 15 reflects their visual similarity.
  - Glaze→Rain (13) and Hail→Rain(13): problem with precipitation features.



(b) Scikit-Learn SVM

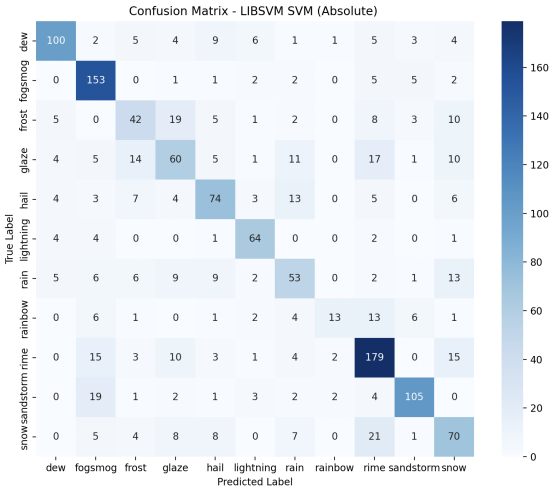


Figure 16: Absolute

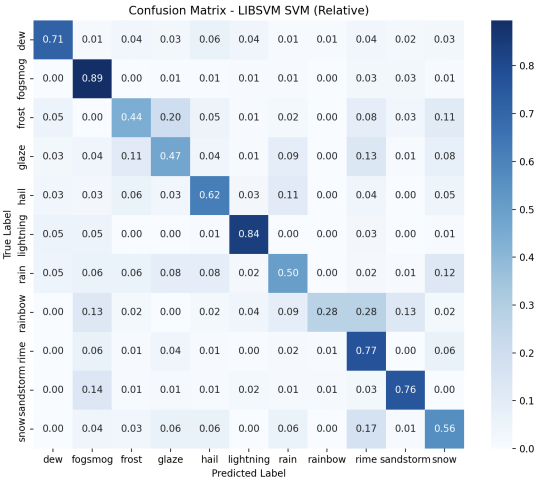


Figure 17: Relative

- **Top Performing Class:** Fogsmog has 153/171 correct (89.5% accuracy)
- **Worst Performing Classes:**
  - Rainbow has 13/47 correct (27.7% accuracy) - frequently confused with rime (13) and rain (4).
  - Frost has 42/95 correct (44.2% accuracy) - confused with glaze (19) and snow (10).
- **Common Misclassification Patterns:**
  - Frost→glaze: 19, glaze→frost: 14, glaze→rime: 17
  - Sandstorm→fogsmog: 19 likely due to similar hazy appearance.
  - Rime→snow: 15, rime→fogsmog: 15, snow→rime,
  - Glaze→Rain (13) and Hail→Rain(13): problem with precipitation features.

§2.8 Optimal C value for Gaussian Kernel SVM

(a) Accuracy for different C values, $\gamma = 0.001$

C	Validation Accuracy (%)	Test Accuracy (%)
10^{-5}	16.93	16.85
10^{-3}	16.93	16.85
1	65.06	66.30
5	66.55	68.34
10	66.15	68.70

Table 6: Validation and Test Accuracy for different values of C

(b) Graphical Analysis

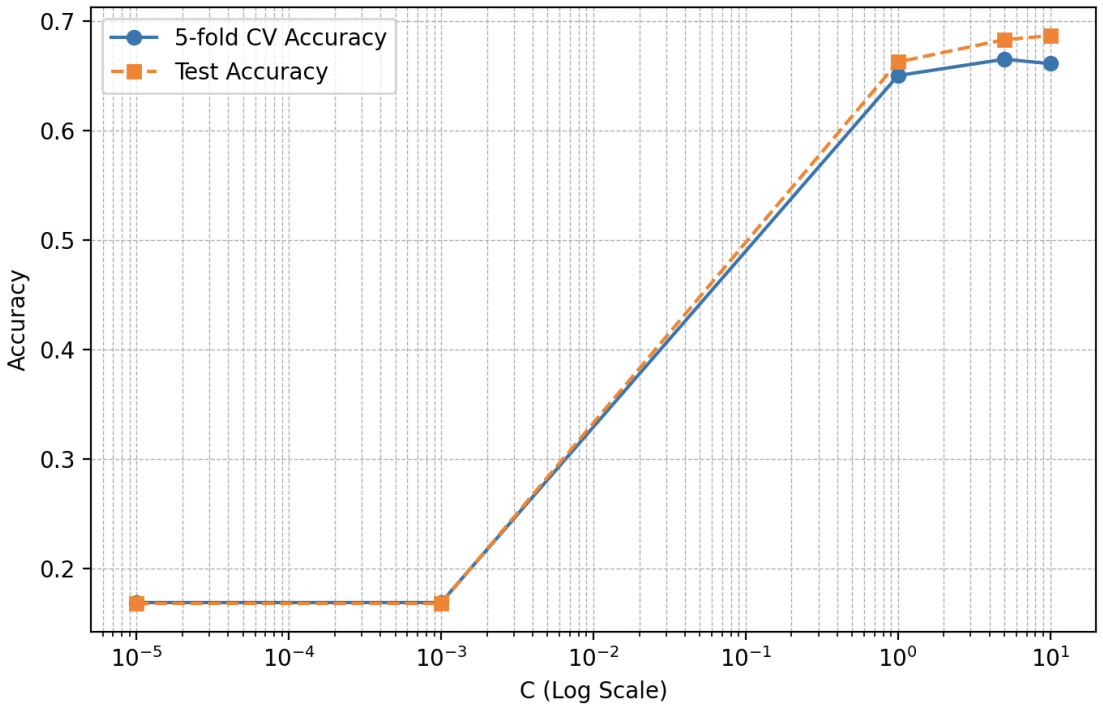


Figure 18: 5-fold CV and Test Accuracy vs C (log scale)

Observations from plot

Small C values $(10^{-5}, 10^{-3})$

- Both validation and test accuracies remain very low (16.9%), indicating that the model is underfitting and unable to capture patterns in the data.

Moderate C values $(C = 1)$

- A significant jump in accuracy occurs, reaching **65.06%** (validation) and **66.30%** (test). This suggests that the model starts learning meaningful decision boundaries.

Higher C values ($C = 5, C = 10$)

- Validation accuracy peaks at $C = 5$ (**66.55%**), while test accuracy continues to increase, reaching a maximum of **68.70%** at $C = 10$.
- The slight drop in validation accuracy for $C = 10$ suggests potential overfitting, but the test accuracy improvement indicates better generalization.

Best C for 5-fold Cross Validation

- The highest validation accuracy is observed at $C = 5$, making it the optimal choice based on cross-validation performance.

(c)Training using SkLearn LIBSVM

- Parameters: C=5, $\gamma = 0.01$
- Test Accuracy: 68.34 %
- Compared to C C=1 (which had 66.30% test accuracy), increasing C to 5 led to an improved test performance.