

Model Optimization and Tuning Phase Template

Date	15-09-2024
Team ID	LTVIP2024TMID25000
Project Title	SMS Spam Detection
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

In this phase, the Naive Bayes model was optimized for the SMS Spam Detection task. Hyper parameters were fine-tuned to achieve optimal performance. Although various models were considered, the primary focus was on Naive Bayes, as it is well-suited for text classification tasks due to its simplicity and effectiveness.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters
Model 1: Naive Bayes	<ul style="list-style-type: none">- alpha: Smoothing parameter, tested with values of 0.1, 0.5, and 1.0 to handle zero probabilities in sparse data.- fit_prior: Set to True to learn class priors from the training data.

MODEL BUILDING (NAVIE BAYES)

```
[14] # Initialize and train the model
      model = MultinomialNB()
      model.fit(X_train, y_train)

      # Make predictions
      y_pred = model.predict(X_test)
```

Evaluation of the Model

```
# Accuracy Score
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')

# Classification Report
print(classification_report(y_test, y_pred))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.show()
```

Note: Due to the specific requirements of this project and the dataset's structure, the other models were not implemented in this phase, as Naive Bayes demonstrated superior performance for text classification.

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Model 1:	Naive Bayes was selected because it consistently

Naive Bayes	achieved the highest accuracy and F1-score for the given text-based dataset. Its ability to handle sparse data and its low computational cost make it an excellent choice for real-time spam detection.
-------------	---