Naive Bayes Spam Classification Report

1. Introduction

Naive Bayes is a simple and fast classification algorithm based on Bayes' Theorem. It works by assuming that the features are independent from each other, which is called the naive assumption.

The general formula of Bayes' Theorem is:

```
P(y|x) = [ P(x|y) * P(y) ] / P(x)
```

In this case, x represents the input features (like words in a message), and y is the target class (spam or ham).

This model is especially useful in text classification problems like spam detection, where it can handle a large number of words as input features. Even though the independence assumption is not always true in real data, Naive Bayes still performs well in many practical cases.

---

2. Dataset Description

The dataset used in this project is a public SMS Spam Collection dataset. It contains 5,572 messages, each labeled as either "ham" (not spam) or "spam".

The dataset has two main columns:

- `label`: shows if the message is spam or not.
- `message`: the text of the SMS message.

We chose this dataset because it is a good example for binary classification. Also, Naive Bayes works well with text data and is commonly used for spam detection tasks.

---

3. Implementation

The model was implemented in Python using the scikit-learn library. The data was cleaned and preprocessed, including:

- Removing irrelevant columns with missing values.
- Renaming features to meaningful names.
- Normalizing the `label` values (ham/spam).

Text messages were converted into numeric vectors using CountVectorizer. Then, the data was split into training and testing sets.

Three Naive Bayes models were used for training:

- MultinomialNB
- BernoulliNB
- GaussianNB

4. Results

The performance of each model was evaluated using accuracy, confusion matrix, and classification report.

Model	Accuracy
MultinomialNB	High
BernoulliNB	Medium
GaussianNB	Low

- **MultinomialNB** performed best, achieving high accuracy.
- **BernoulliNB** was slightly less accurate.
- **GaussianNB** performed poorly due to data type mismatch.

We also visualized the average message lengths for each label. Spam messages tend to be slightly longer.

5. Model Comparison

Each Naive Bayes model was trained and evaluated on the same dataset:

- **MultinomialNB** is designed for count-based text data and showed the best performance.
- **BernoulliNB** is better suited for binary features and gave slightly lower results.
- **GaussianNB** is meant for continuous input features and gave poor results in this task.

6. Discussion

Naive Bayes models are easy to implement and fast to train. In this experiment:

- MultinomialNB was the most suitable model for this task.
- BernoulliNB worked reasonably well but was less effective with count data.
- GaussianNB was not suitable due to the structure of the input features.

Note on Scaling: We did not apply feature scaling (like StandardScaler) because CountVectorizer produces integer counts of word frequencies. Naive Bayes models, especially MultinomialNB, are designed to work with such raw counts. Applying scaling would reduce the interpretability and distort the relationship between features and labels.

7. Conclusion

Naive Bayes is an efficient model for spam message classification. Among the three models tested, MultinomialNB showed the highest accuracy and was the best suited for the SMS Spam dataset.

The project demonstrated the full process from data cleaning and preprocessing to model training, evaluation, and comparison. Despite its simplicity, Naive Bayes remains a strong candidate for many classification tasks involving text data.