Naive Bayes Classifier Tutorial

1 Introduction to Classification

Classification is a machine learning technique used to categorize data into predefined classes. Some common examples include:

- **Spam Detection**: Classify emails as Spam or Not Spam.
- **Medical Diagnosis**: Predict whether a patient has a disease based on symptoms.

2 Basic Probability Concepts

Before understanding Naive Bayes, let's recall some key probability rules:

2.1 Joint Probability

$$P(A \cap B) = P(A|B) \cdot P(B)$$

if A and B are independent

2.2 Conditional Probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$P(A|B) = P(A)$

$P(A \cap B) = P(A)P(B)$

3 Bayes' Theorem

Bayes' Theorem calculates the probability of a class given certain observed features:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Where:

- $P(A|B)$ is the **posterior probability**.
- $P(B|A)$ is the **likelihood**.
- $P(A)$ is the **prior probability**.
- $P(B)$ is the **evidence** (normalizing factor).

$$p(c|X) = \frac{P(X|C)\,P(C)}{\cancel{P(X)}}$$

4 Naive Bayes Classifier

The Naive Bayes Classifier assumes that all features are independent given the class:

$$P(Class|x_1, x_2, ..., x_n) = P(Class) \times P(x_1|Class) \times ... \times P(x_n|Class)$$

$$p(cs) \times p(g|cs) \times \cdots p(s|cs) \times p(y|cs)$$

5 Argmax in Naive Bayes

After computing probabilities for each class, we select the most likely class using:

$$\hat{y} = \arg\max_{c \in C} P(c) \times P(x_1|c) \times P(x_2|c) \times ... \times P(x_n|c)$$

6 Types of Naive Bayes

1. **Gaussian Naive Bayes (Continuous Data)**

$$P(x_i|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}\right)$$

2. **Multinomial Naive Bayes (Text Classification)**

$$P(x_i|c) = \frac{count(x_i, c) + \alpha}{\sum count(w, c) + \alpha V}$$

3. **Bernoulli Naive Bayes (Binary Features)**

$$P(x_i|c) = p_i^{x_i}(1 - p_i)^{(1-x_i)}$$

7 Log Probability for Numerical Stability

To prevent underflow issues when multiplying small probabilities, we take the logarithm:

$$\log P(c|x_1, ..., x_n) = \log P(c) + \sum_{i=1}^{n} \log P(x_i|c)$$

Final prediction:

$$\hat{y} = \arg\max_{c \in C} \left(\log P(c) + \sum_{i=1}^{n} \log P(x_i|c) \right)$$

8 Python Implementation

Here is a simple implementation of the Naive Bayes classifier using Scikit-learn:

```python
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer

# Sample dataset
emails = ["Win a free lottery now", "Buy money online cheap",
          "You won free cash prize", "Meet me at 5 pm", "Hello, how are you?"]
labels = [1, 1, 1, 0, 0]  # 1 = Spam, 0 = Not Spam

# Convert text into numerical features
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(emails)

# Train Naive Bayes Classifier
model = MultinomialNB()
model.fit(X, labels)

# Predict a new email
new_email = ["Get free cash now"]
X_new = vectorizer.transform(new_email)
prediction = model.predict(X_new)

print("Spam" if prediction[0] == 1 else "Not Spam")
```

9 Summary

- Naive Bayes is a classification algorithm based on Bayes' Theorem.
- It assumes **independence** between features.
- The three main types:
  - **Gaussian Naive Bayes** (for continuous features).
  - **Multinomial Naive Bayes** (for text classification).
  - **Bernoulli Naive Bayes** (for binary features).
- Log probabilities help avoid numerical underflow.

10 Further Reading

- **Pattern Recognition and Machine Learning** – Christopher Bishop
- Scikit-learn documentation on Naive Bayes: `https://scikit-learn.org/stable/modules/naive_bayes.html`