**Name: Sumit Avinash Solanke**

**Roll No: TYITB121**

**Assignment No. 6**

**AIM:** Assignment on Naïve Bayes.

**PREREQUISITE:** Python programming

**THEORY:**

The Naïve Bayes classification technique is a probabilistic algorithm based on Bayes' theorem. It is widely used in machine learning for classification tasks due to its simplicity and effectiveness. Despite its assumption that all predictor variables are independent, Naïve Bayes often performs well in real-world scenarios.

Naïve Bayes classifiers are particularly useful when dealing with large datasets with high dimensionality. They are commonly applied in text classification, spam filtering, sentiment analysis, and medical diagnosis. The approach is based on estimating probabilities from prior knowledge and observed data, which helps in making predictions efficiently.

**Concept of Naïve Bayes Classification**

To understand Naïve Bayes classification, consider a scenario where objects are classified into two categories based on prior observations. For example, given a dataset where objects are labeled as either GREEN or RED, a new object must be classified based on its surrounding characteristics.

If a dataset contains twice as many GREEN objects as RED, a new object is more likely to belong to the GREEN category. This initial assumption, based on prior observations, is known as prior probability. However, classification is not solely dependent on prior probability but also on the likelihood of an object belonging to a category given its features.

A new object’s classification is determined by analyzing its surrounding objects. If a new object is positioned closer to more RED objects than GREEN ones, it is more likely to belong to the RED category. Combining prior probability and likelihood, Bayes' theorem calculates the posterior probability to classify the new object.

**Working Mechanism of Naïve Bayes**

Naïve Bayes assumes that all predictors contribute independently to the outcome. Although this assumption is not always accurate, it simplifies computation and improves efficiency in classification tasks. The method works as follows:

1. **Calculate Prior Probability**
   * The probability of each class is estimated based on its occurrence in the training dataset.
2. **Compute Likelihood**
   * The likelihood of each feature given a particular class is calculated based on observed data.
3. **Apply Bayes' Theorem**
   * The posterior probability is computed using Bayes' theorem by combining prior probability and likelihood.
4. **Classify New Data**
   * The class with the highest posterior probability is assigned to the new data instance.

**Applications of Naïve Bayes**

Naïve Bayes is widely applied in various domains, including:

* **Spam Filtering:** Identifies spam emails based on the frequency of certain words.
* **Sentiment Analysis:** Determines whether a review is positive, negative, or neutral.
* **Medical Diagnosis:** Helps in predicting diseases based on symptoms.
* **Document Classification:** Categorizes text documents into predefined classes.

**Advantages of Naïve Bayes**

* **Simple and Fast:** The algorithm is easy to implement and computationally efficient.
* **Effective for High-Dimensional Data:** Works well with large datasets and multiple features.
* **Performs Well with Small Datasets:** Provides good classification results even with limited data.
* **Handles Categorical and Continuous Data:** Can process different types of input data effectively.

**Disadvantages of Naïve Bayes**

* **Strong Independence Assumption:** Assumes that features are independent, which may not always be true.
* **Sensitive to Imbalanced Data:** Performance can be affected if one class dominates the dataset.
* **Limited Representation of Complex Relationships:** Struggles with feature interactions and dependencies.

**Dataset Description: Salaries\_pd.csv**

**Dataset Name:** *Salaries\_pd.csv*  
**Purpose:** This dataset is used to analyze salary trends based on experience, education, job title, company, and gender. It’s often used to explore factors affecting pay and detect patterns or inequalities in salary distributions.

**Dataset Characteristics:**

| **Attribute** | **Description** |
| --- | --- |
| **Total Records** | *(e.g., 500–1000 entries, varies per dataset)* |
| **Number of Features** | 6–10 depending on the version |
| **Missing Values** | May contain missing/empty values in salary or experience fields |
| **Target Variable** | Salary (for regression tasks) or Job Title (for classification) |

**Common Columns and Their Descriptions:**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| EmployeeID | Integer | Unique identifier for each employee |
| Name | String | Name of the employee (optional or anonymized) |
| Job Title | String | Title or role of the employee |
| Department | String | Department where the employee works |
| Company | String | Name of the company/organization |
| Education | String | Highest education level attained (e.g., Bachelors, Masters, PhD) |
| Experience | Float/Integer | Number of years of relevant professional experience |
| Gender | Categorical | Gender of the employee (Male, Female, etc.) |
| Salary | Float | Annual salary in currency (e.g., USD or INR) |
| Location | String | City or region of employment (optional) |

**Sample Records (Preview):**

| **EmployeeID** | **Job Title** | **Experience** | **Education** | **Gender** | **Salary** |
| --- | --- | --- | --- | --- | --- |
| 101 | Data Analyst | 3 | Bachelors | Female | 550000.00 |
| 102 | Software Eng. | 5 | Masters | Male | 850000.00 |
| 103 | HR Manager | 7 | Bachelors | Female | 600000.00 |
| 104 | DevOps Eng. | 2 | Masters | Male | 720000.00 |

**CONCLUSION:**

Naïve Bayes is a powerful classification algorithm that provides fast and reliable predictions. While its assumption of feature independence may not always hold, it remains a popular choice for text classification, spam detection, and sentiment analysis. Its efficiency and ease of implementation make it an essential tool in machine learning applications.