**Assignment 6**

**Aim:-**Implementation of Navie Bayes on Car dataset.

**Objective:** To use the Naive Bayes algorithm for predicting the price category of used cars based on their attributes such as manufacturing year, kilometers driven, fuel type, seller type, transmission, and ownership history.

**Introduction**

# Naive Bayes is a probabilistic classification algorithm based on Bayes’ Theorem. It assumes that the features are conditionally independent given the class label, which simplifies computation and makes it efficient for high-dimensional datasets. Despite its simplicity, it often performs competitively with more complex models.

# Why Use Naive Bayes on Car Dataset:

# Naive Bayes is suitable for this car dataset because:

# It handles both categorical and numerical data.

# It works well even with relatively small datasets.

# It's fast and interpretable.

# It can provide probabilistic insight into class predictions.

# Advantages of Apriori Algorithm

* **Simplicity**: Easy to implement and fast to train.
* **Scalability**: Works well with large datasets.
* **Categorical Compatibility**: Naturally handles label-encoded data.
* **Effective with Small Data**: Performs decently even with limited data.
* **Probabilistic Output**: Outputs class probabilities, which are useful in decision-making.

**Dataset Description:**

* **Filename**: car\_data.csv
* **Columns**:
  + name – Car model
  + year – Manufacturing year
  + selling\_price – Price at which the car is sold
  + km\_driven – Kilometers driven
  + fuel – Type of fuel used
  + seller\_type – Dealer or Individual
  + transmission – Manual or Automatic
  + owner – Ownership status (e.g., First Owner)

**Steps of Implementation:**

### 1. Importing Libraries

* pandas for data handling
* sklearn for preprocessing, model training, and evaluation
* matplotlib / seaborn for visualization

### 2. Loading and Preparing the Dataset

* Load dataset from CSV into a Pandas DataFrame.
* Inspect the dataset for nulls and feature types.

### 3. Data Preprocessing

* Drop unnecessary features (like name).
* Encode categorical features (fuel, seller\_type, etc.).
* Encode the target variable: selling\_price can be binned into price categories (low, medium, high).

### 4. Feature Encoding

* Apply LabelEncoder to transform categorical features into numeric format.

### 5. Splitting Dataset

* Use train\_test\_split to divide the data into training and testing subsets.

### 6. Model Training

* Use GaussianNB from scikit-learn for training since the features are mostly numeric or can be assumed to follow a Gaussian distribution.

### 7. Prediction

* Predict the class for the test data.

### 8. Model Evaluation

* Use metrics such as Accuracy, Classification Report, and Confusion Matrix.

# Conclusion

* Naive Bayes proved effective in classifying car price categories using a mix of numerical and categorical attributes.
* The model offers interpretability through metrics and confusion matrices.
* This experiment shows how probability-based models can drive insights into feature importance and decision-making.

**References:**

[**https://scikit-learn.org/stable/modules/naive\_bayes.html**](https://scikit-learn.org/stable/modules/naive_bayes.html)

**Github Repository**

<https://github.com/mrnik89/ML_Lab/blob/main/Assignment6_ML.ipynb>