1. Installing Required Libraries
2. Loading the Dataset =

size5875 rows × 22 columns

1. Checking for Missing Values & Data Summary

 **df.isnull().sum()**: Counts missing values in each column.

 **df.info()**: Displays column names, data types, and non-null values.

 **df.corr()**: Computes correlation between features

1. EDA + PREPROCESSING

**Feature Checking**

* **Feature checking** refers to analyzing the dataset's features (columns) to understand their relationships, significance, and distribution.

Feature visualization helps understand data distribution, relationships, and patterns.

sns.distplot(df[i])

sns.histplot(df[i])

### ****Correlation and Dependent Features: Handling Multicollinearity****

1. **Correlation Between Features**
   * The correlation between features helps identify **multicollinearity** (when two or more features are highly correlated).
   * For example, if two features (e.g., age and years\_of\_experience) are highly correlated, they might convey redundant information.
   * **High correlation** means one feature can be predicted from the other, which may not add value to the model and could lead to overfitting.
2. **Correlation with the Target Variable (y)**
   * The **correlation of each feature with the target variable (motor\_UPDRS)** shows how much a feature contributes to predicting the target.
   * Strong correlations (positive or negative) indicate that a feature is likely important for predicting the outcome. Features with weak correlations with y may be less informative and might be removed or transformed.
3. **Dealing with High Correlation**
   * **Removing highly correlated features**: If two features have a correlation greater than a certain threshold , you can remove one of them to avoid multicollinearity. This simplifies the model and prevents it from overfitting.

Deleted feature with understanding A=["motor\_UPDRS","HNR","total\_UPDRS","sex","test\_time"]

Final dataset size = 5875 rows × 17 columns

1. Splitting Data

Splits data into **training (70%)** and **testing (30%)** sets

1. Model building for comparison

'Random Forest',

'Naive Bayes',

'Decision Tree',

'SVM',

'KNN'

1. Proposal model building
   * 1. DEEPFM use model
     2. LIGHTGBM use model
     3. meta-model (Gradient Boosting Regressor)

Hybrid Model Works: Taking Predictions from DeepFM & LightGBM and Using Them for the Meta-Model-

This hybrid stacking model follows a three-step process where DeepFM and LightGBM generate predictions, and these predictions are then used as input to a final meta-model (Gradient Boosting Regressor).

1. Result