

## 1. Linear Regression Project: House Price Prediction

### □ Description:

Predict the price of a house based on features like square footage, number of bedrooms, bathrooms, and location.

### □ Steps:

1. **Collect Data:** Use a dataset like the Boston Housing dataset or Kaggle's housing datasets.
  2. **Preprocess Data:** Handle missing values, encode categorical data (e.g., location).
  3. **Feature Selection:** Choose relevant features like size, location, number of rooms, etc.
  4. **Model Building:** Use `LinearRegression()` from `scikit-learn`.
  5. **Train/Test Split:** Split your dataset (e.g., 80/20).
  6. **Train the Model:** Fit the model on the training set.
  7. **Evaluate:** Use metrics like RMSE,  $R^2$  score.
  8. **Visualize:** Plot predictions vs actual prices.
- 

## □ 2. Logistic Regression Project: Email Spam Detection

### □ Description:

Classify emails as **spam or not spam** using logistic regression.

### □ Steps:

1. **Collect Data:** Use a dataset like the SpamAssassin public corpus.
  2. **Text Preprocessing:** Remove stopwords, punctuation, lowercase text, etc.
  3. **Feature Extraction:** Use `TfidfVectorizer` or `CountVectorizer` to convert text to numbers.
  4. **Model Building:** Use `LogisticRegression()` from `scikit-learn`.
  5. **Train/Test Split:** Separate your dataset into training and testing sets.
  6. **Train the Model:** Fit on training data.
  7. **Evaluate:** Accuracy, precision, recall, confusion matrix.
  8. **Tune Threshold (Optional):** Adjust the decision threshold for better balance.
- 

## □ 3. Simple ML Project: Iris Flower Classification (Multiple Algorithms)

### □ Description:

Classify iris flowers into 3 species (Setosa, Versicolor, Virginica) based on petal and sepal length/width.

### □ Steps:

1. **Load Data:** Use `load_iris()` from `sklearn.datasets`.
2. **Explore Data:** Plot using `matplotlib/seaborn`.
3. **Preprocess:** Normalize or standardize features.
4. **Try Multiple Models:**
  - Logistic Regression
  - Decision Tree
  - K-Nearest Neighbors
5. **Train/Test Split**
6. **Train and Compare:** Train each model and compare their accuracy.
7. **Visualize:** Use confusion matrices and classification reports.