

# Project Idea:

## Prediction Projects

### 1. House Price Prediction

- a. Dataset: Boston Housing or Kaggle datasets
- b. Input: Area, location, rooms, age
- c. Output: Predicted price

### 2. Student Score Predictor

- a. Input: Hours studied, attendance
- b. Output: Predicted exam score

### 3. Salary Predictor

- a. Input: Experience, education, job title
- b. Output: Predicted salary

### 4. Stock Price Prediction (Basic)

- a. Use historical stock prices and a linear regression model.
- b. Avoid LSTM for now unless you're more comfortable.

## Classification Projects

### 1. Iris Flower Classification

- a. Classic beginner dataset with 3 flower types
- b. Use Logistic Regression, SVM, or KNN

### 2. Titanic Survival Prediction

- a. Predict survival based on age, gender, class, etc.
- b. Good for learning data preprocessing and missing values

### 3. Email Spam Detection

- a. Input: Email text
- b. Output: Spam or Not Spam
- c. Use Naive Bayes or TF-IDF + Logistic Regression

### 4. Digit Recognition

- a. Dataset: MNIST
- b. Input: Handwritten digit images
- c. Output: Classify 0–9 digits

## ★ Recommendation Projects

### 1. Movie Recommendation System

- a. Use MovieLens dataset
- b. Recommend based on ratings (collaborative filtering)

### 2. Book or Product Recommender

- a. Use cosine similarity on user-item matrix
- b. Can be content-based or collaborative

## 📊 Clustering / Unsupervised Projects

### 1. Customer Segmentation

- a. Dataset: E-commerce user behavior
- b. Algorithm: K-means
- c. Cluster users by purchasing behavior

### 2. Image Compression

- a. Use K-means clustering on image pixels

## 🔧 Bonus Easy Projects

### 1. Fake News Detection

- a. Use a basic text classifier to detect real vs. fake news

### 2. Language Detection

- a. Input: A sentence
- b. Output: Language name (English, Spanish, etc.)

### 3. Weather Condition Classifier

- a. Input: Temp, humidity, pressure
- b. Output: Rainy, Sunny, Cloudy

## Steps for a Machine Learning Project

### 1. Import Libraries

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression # or another model
from sklearn.metrics import mean_squared_error
```

### 2. Load or Create the Dataset

a. If from CSV:

```
df = pd.read_csv('filename.csv')
```

b. Or create your own DataFrame manually.

### 3. Clean the Data

a. Check for missing values:

```
df.isnull().sum()
```

b. Drop or fill missing values:

```
df = df.dropna() # or use df.fillna(value)
```

### 4. Select Features and Target

```
X = df[['feature1', 'feature2', 'feature3']] # input columns
y = df['target'] # what you want to predict
```

### 5. Split the Data

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

## 6. 🏆 Train the Model

```
model = LinearRegression() # or any other model
model.fit(X_train, y_train)
```

## 7. 📈 Make Predictions

```
y_pred = model.predict(X_test)
```

## 8. 📏 Evaluate the Model ( Optional )

```
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

## 9. 📊 (Optional) Visualize Results

a. For regression:

```
import matplotlib.pyplot as plt
plt.scatter(y_test, y_pred)
plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.show()
```

## 10. 💾 (Optional) Save the Model

```
import joblib
joblib.dump(model, 'model.pkl')
```