

End-to-End Project Roadmap (From 'Let's create...' to 'Summary Roadmap')

Let's create your complete roadmap, step by step

Goal: Build an End-to-End Data + ML Project

Your project should cover these 6 stages:

- 1) Data Collection
- 2) Data Cleaning & Processing
- 3) Data Analysis & Visualization
- 4) Model Building (ML)
- 5) Model Evaluation & Improvement
- 6) Deployment (using Flask or Streamlit)

Example Project Ideas (Pick One)

- E-commerce: Customer Purchase Prediction
- Real Estate: House Price Prediction
- Entertainment: Movie Recommendation System
- Web Data: Scrape & Analyze Product Prices
- Finance: Credit Card Fraud Detection

House Price Prediction (End-to-End Example Roadmap)

STEP 1 - Data Collection

- Option 1: Download dataset from Kaggle (e.g., House Prices: Advanced Regression Techniques)
- Option 2: Scrape data from real-estate sites (requests, BeautifulSoup)
- Option 3: Use SQL to store or query existing data

Tools: Python, Pandas, BeautifulSoup, SQL

STEP 2 - Data Cleaning & Preparation

- Handle missing values (fillna), remove duplicates
- Convert categorical columns using LabelEncoder or OneHotEncoder
- Feature scaling with StandardScaler

Tools: Pandas, NumPy, scikit-learn preprocessing

STEP 3 - Exploratory Data Analysis (EDA)

- Use Matplotlib and Seaborn to explore:

- * Correlation heatmap
 - * Price vs Area scatter plot
 - * Distribution of prices
 - Derive business insights, e.g., location and area most impact price
- Tools: Seaborn, Matplotlib, Pandas

STEP 4 - Model Building

- Split data into train and test sets
- Try models: Linear Regression, Decision Tree, Random Forest, XGBoost (optional)
- Train and tune hyperparameters

Tools: scikit-learn (train_test_split, RandomForestRegressor, etc.)

STEP 5 - Model Evaluation

- Evaluate with RMSE and MAE for regression (or Accuracy/F1-score for classification problems)
- Visualize actual vs predicted values

Tools: scikit-learn metrics, Seaborn

STEP 6 - Deploy with Flask

- Create a Flask web app:
 - * A form to enter inputs (area, rooms, etc.)
 - * Model predicts and returns price
- Save model using joblib or pickle
- Flask loads it and predicts on new data

Example folder structure:

```
project/  
  static/  
  templates/  
  model.pkl  
  app.py  
  requirements.txt  
  README.md
```

Tools: Flask, HTML, Jinja2 Templates

STEP 7 - (Optional) Host It

- Deploy Flask app using Render, Railway, or Vercel
- Upload to GitHub with a clean README (problem, data, model, output, demo link)

Deliverables (What Recruiters Will See)

- Data: CSV or SQL or scraped data
- EDA: Graphs, correlation plots, insights
- Model: Notebook showing model training
- Web App: Flask UI with working prediction
- GitHub: Full project code and documentation
- LinkedIn Post: Summary with screenshot(s) and key learnings

Bonus Tips for Portfolio Impact

- Write a short business problem, e.g., estimate fair house prices for agents
- Use clear visuals (bar charts, heatmaps, model performance plots)
- Post progress on LinkedIn; include visuals and concise insights
- Add the project link to your resume under Projects

Summary Roadmap

Step 1: Data Collection - Python, SQL, BeautifulSoup

Step 2: Data Cleaning - Pandas, NumPy

Step 3: EDA - Seaborn, Matplotlib

Step 4: Model Building - scikit-learn

Step 5: Evaluation - sklearn.metrics

Step 6: Deployment - Flask, HTML

Step 7: Hosting - Render or GitHub