

AI+ Prompt Engineer

Hands-on

Title: Introduction to Project-Based Learning in AI

Objective:

This module introduces learners to project-based learning (PBL) in AI education, helping them understand its benefits and how to implement it in their learning process.

Problem Statement:

How can we leverage project-based learning to build an AI system that analyzes real-world data.

Tools Used:

1. AI GPT Tools: [ChatGPT - AI Education](#)



AI Education

Usage:

Use this GPT to design project-based learning experiences that teach students how to build AI systems for analyzing real-world data. It helps refine prompt engineering, optimize AI model inputs, and interpret data effectively. By leveraging AI ethics, interdisciplinary collaboration, and structured problem-solving, it ensures responsible and impactful AI applications. This GPT also provides best practices for hands-on learning, enabling students to develop AI-driven insights with accuracy and efficiency.

When to Use:

Use this GPT when developing an AI system through project-based learning that involves:

- **Prompt Engineering:** Optimizing AI inputs for reliable outputs.
- **Data Analysis:** Structuring, interpreting, and visualizing real-world data effectively.
- **AI Ethics & Bias Mitigation:** Understanding AI fairness and responsible AI development.
- **Interdisciplinary Learning:** Combining AI, domain expertise, and problem-solving strategies.

Why to Use:

- **Guidance on AI Implementation:** Offers insights on AI models, prompt tuning, and real-world applications.
- **Educational Support:** Helps structure AI learning projects with best practices in project-based learning.
- **Practical Application:** Provides methods to enhance data-driven AI projects for better decision-making.

Steps to be Followed:

Step 1: Data Collection or Creation

Step 2: Data Preprocessing and Feature Engineering

Step 3: Model Selection and Training

Step 4: Model Deployment and Integration'

Step 5: Testing and Performance Evaluation

Step 6: Develop an Interactive Dashboard

Steps in Detail:

Step 1: Data Collection or Creation

Sample prompt:

create a new synthetic dataset with 300 records and 10 features or attributes and identify a real-world problem that requires AI-driven insights. What data sources are relevant to this problem, and where can they be obtained? If real-world data is unavailable, what key variables should be included in a synthetic dataset and share the dataset here with dataset summary

Sample output:

Synthetic Customer Churn Dataset				
	Customer_ID	Age	Gender	Annual_Spen
1	CUST1	56	Female	7921
2	CUST2	69	Male	4952
3	CUST3	46	Male	6381
4	CUST4	32	Male	7165
5	CUST5	60	Male	3349
6	CUST6	25	Male	7890

Dataset Summary

- **Total Records:** 300
- **Columns:** 10
- **Data Types:** Mix of numerical, categorical, and text data

Key Features



1. **Customer_ID** – Unique identifier for each customer
2. **Age** – Age of the customer
3. **Gender** – Male/Female
4. **Annual_Spending (\$)** – Total amount spent by the customer in a year
5. **Purchase_Frequency** – Number of purchases made per month
6. **Last_Purchase_Days_Ago** – Days since the last purchase
7. **Customer_Support_Calls** – Number of customer support interactions
8. **Website_Visits_Per_Month** – Frequency of website visits
9. **Discount_Usage (%)** – Percentage of purchases made using discounts
10. **Churn** – Whether the customer churned (1 = Yes, 0 = No)

Step 2: Data Preprocessing and Feature Engineering

Sample prompt:

Analyze the collected dataset and identify missing, inconsistent, or redundant data. What preprocessing techniques should be applied to clean and standardize the dataset? What new features can be engineered to improve model performance and share the revised dataset here with dataset summary

Sample output:

Cleaned Synthetic Customer Churn Dataset						
		Customer_ID	Age	Gender		
1	0	CUST1	56	Female		
2	1	CUST2	69	Male		
3	2	CUST3	46	Male		
4	3	CUST4	32	Male		
5	4	CUST5	60	Male		
6	5	CUST6	25	Male		

Dataset Summary

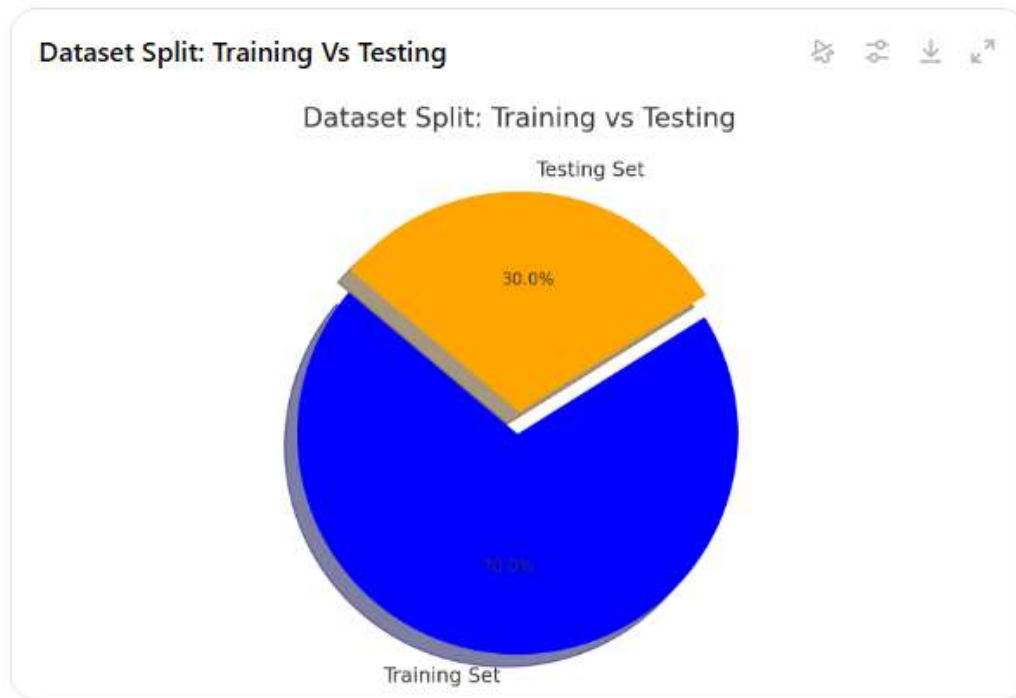
- Total Records: 300
- Columns: 13
- Data Types:
 - Customer_ID: Categorical (Object)
 - Age: Integer
 - Gender: Categorical (Object)
 - Annual_Spending (\$): Integer
 - Purchase_Frequency: Integer
 - Last_Purchase_Days_Ago: Integer
 - Customer_Support_Calls: Integer
 - Website_Visits_Per_Month: Integer
 - Discount_Usage (%): Integer
 - Churn: Integer (Target Variable)
 - Loyalty_Score: Float (Engineered Feature)
 - Spending_Behavior: Categorical (Low, Medium, High)
 - Engagement_Level: Categorical (Low, Medium, High)

Step 3: Model Selection and Training

Sample prompt:

Use train and testing split 70% and 30% and based on the problem type, what machine learning models are best suited for this task? What criteria should be used to compare their performance? How should the dataset be split for training and validation and share it here in visualization form with the dataset

Sample output:



Dataset Split and Machine Learning Model Selection

Dataset Split:

- Training Set: 70%
- Testing Set: 30%

Training Dataset				
		Age	Annual_Spending (\$)	Purchase_Fre
1	232	54	8675	15
2	47	35	8087	14
3	195	69	6542	17
4	219	29	8915	5
5	281	42	5354	5
6	208	61	755	7

Testing Dataset				
		Age	Annual_Spending (\$)	Purchase_Fre
1	224	54	7438	5
2	268	37	5173	5
3	136	44	3423	11
4	211	49	509	10
5	67	57	5468	17
6	274	57	7441	19

Step 4: Model Deployment and Integration'

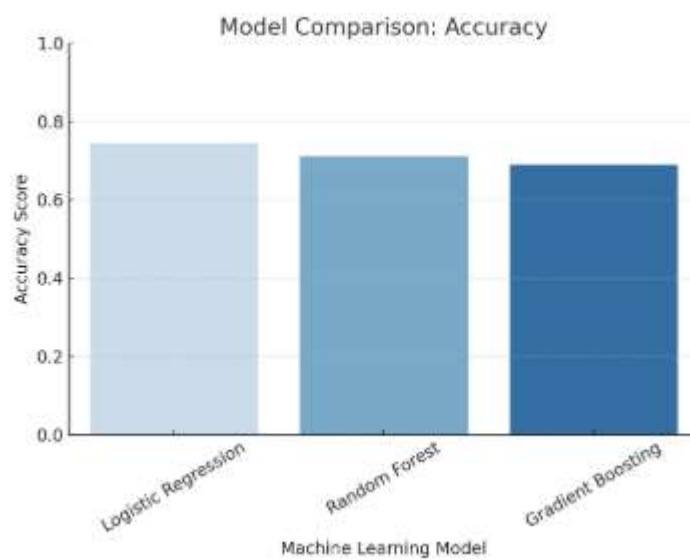
Sample prompt:

Use the best suited algorithm or try 2-3 algorithms that are best suited for this dataset and how should the trained AI model be deployed for real-world use? What platform or framework is best suited for hosting the model? What input-output formats need to be considered for seamless integration and share it here in visualization format with the dataset

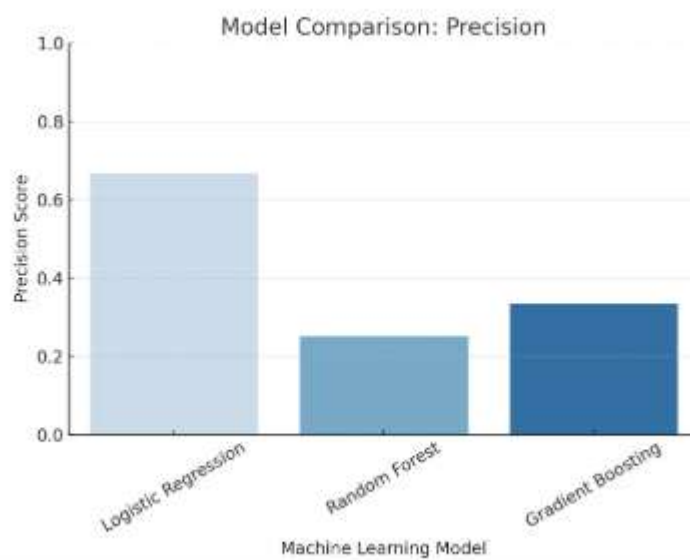
Sample output:

Model Performance Results				
	Model	Accuracy	Precision	Recall
1	Logistic Regression	0.74444444444444445	0.66666666666666666	0.08333333333333333
2	Random Forest	0.71111111111111111	0.25	0.04166666666666666
3	Gradient Boosting	0.68888888888888888	0.33333333333333333	0.16666666666666666

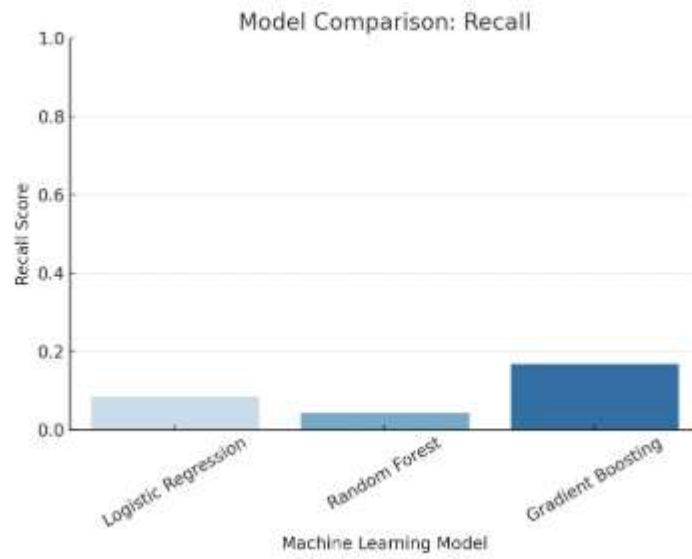
Model Comparison: Accuracy



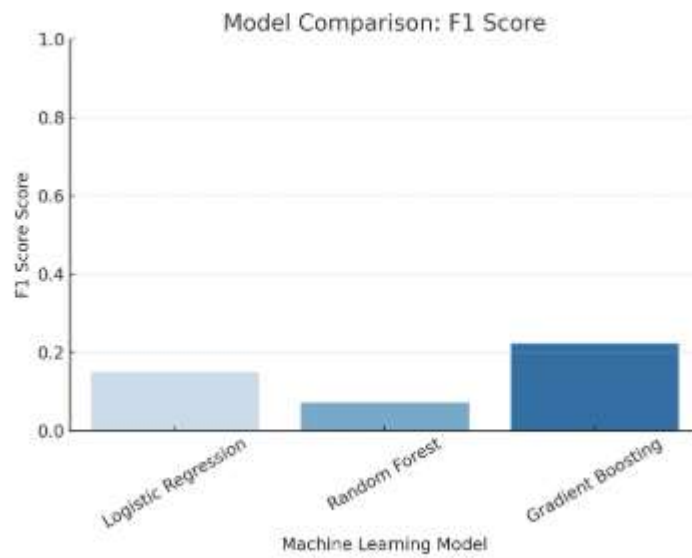
Model Comparison: Precision

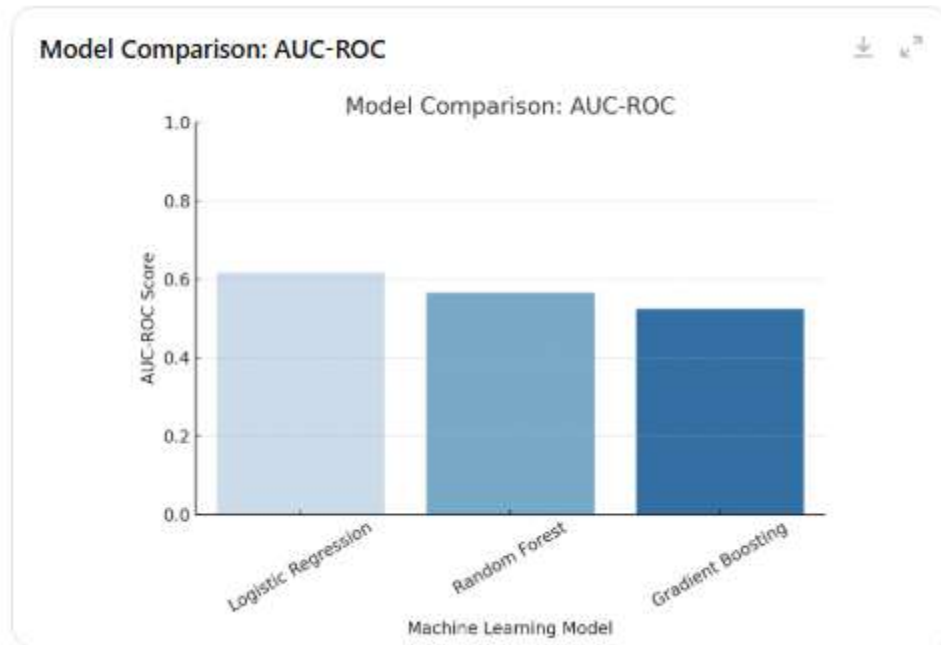


Model Comparison: Recall



Model Comparison: F1 Score





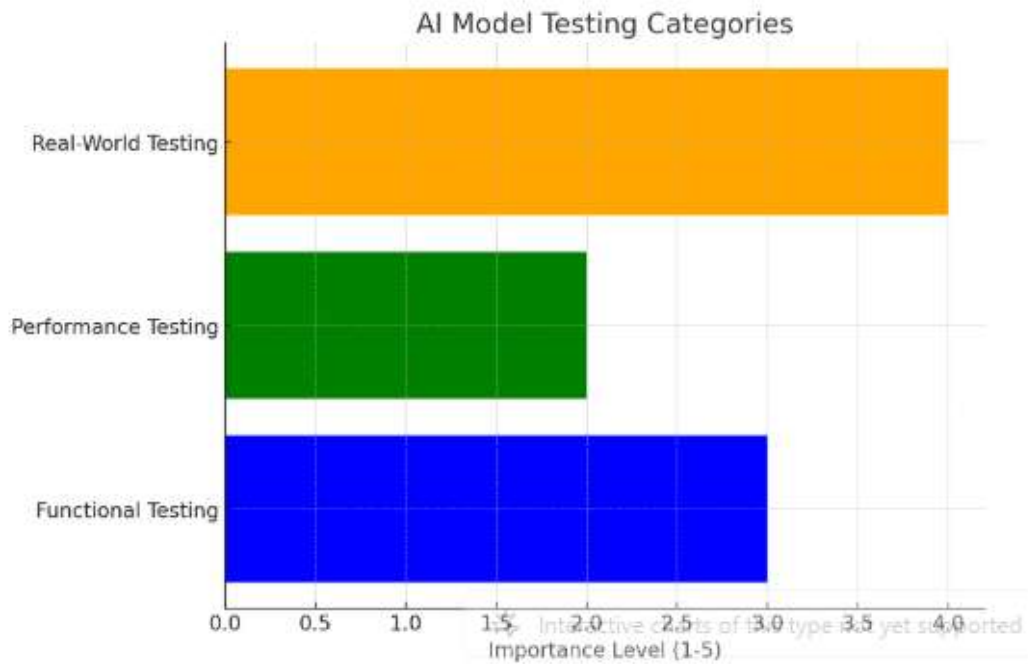
Step 5: Testing and Performance Evaluation

Sample prompt:

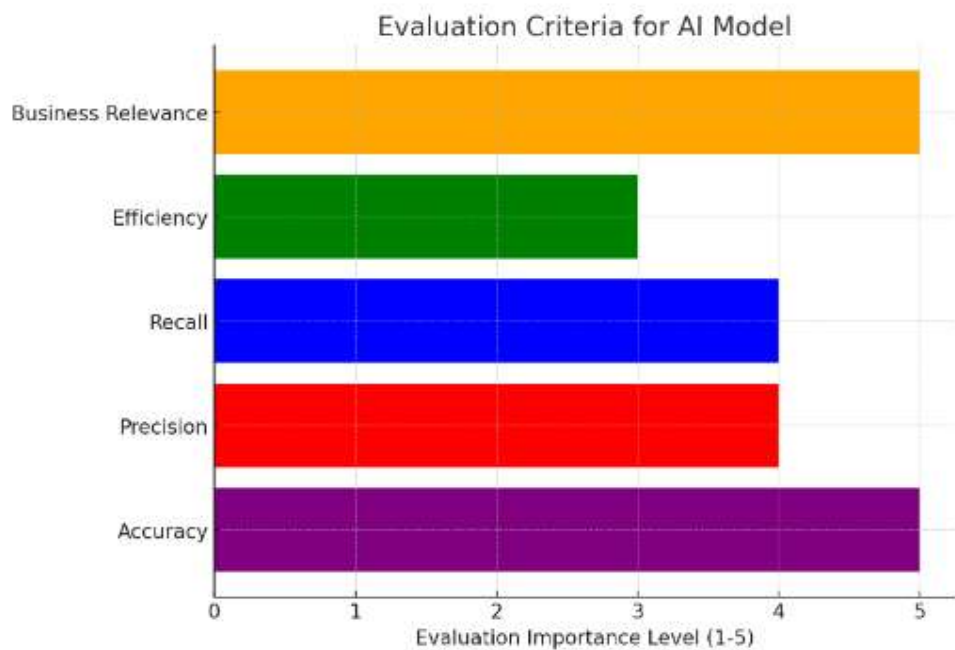
What test cases should be conducted to ensure the AI system works as expected? How should the model be evaluated based on accuracy, efficiency, and real-world applicability? What feedback mechanisms should be implemented to improve performance and share it here in visualization format with the dataset

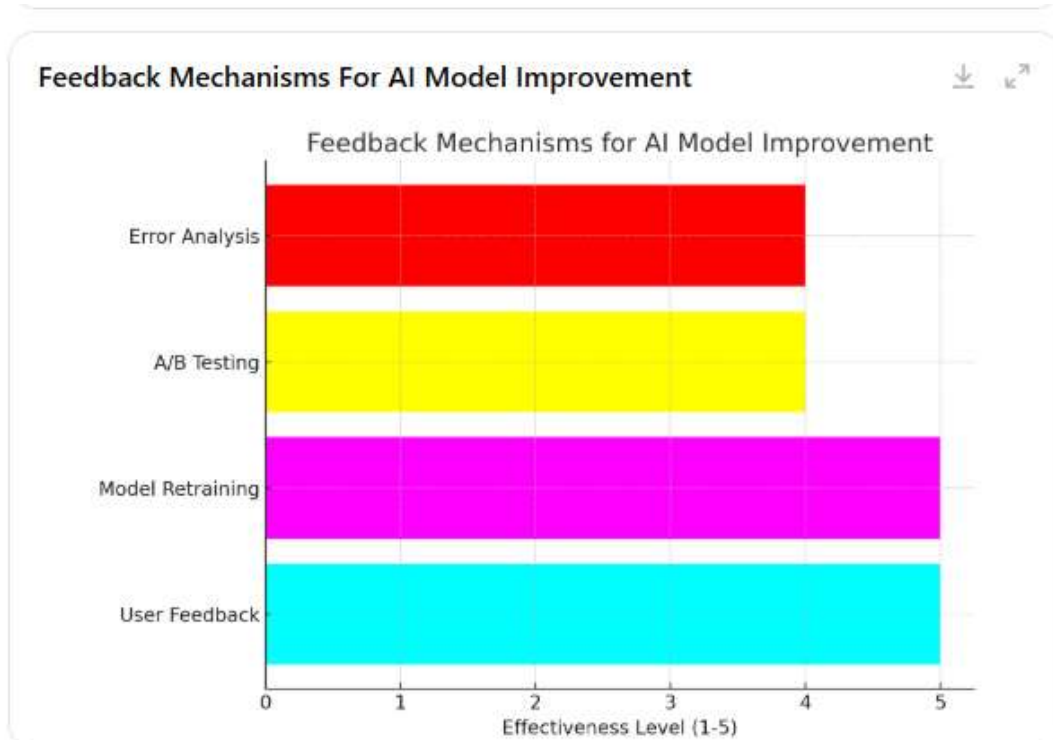
Sample output:

AI Model Testing Categories



Evaluation Criteria For AI Model





Cleaned Synthetic Customer Churn Dataset

		Customer_ID	Age	Gender
1	0	CUST1	56	Female
2	1	CUST2	69	Male
3	2	CUST3	46	Male
4	3	CUST4	32	Male
5	4	CUST5	60	Male
6	5	CUST6	25	Male

Key Takeaways:

1. Data Preparation & Cleaning

- Created a synthetic dataset with 300 records and 10+ features.

- Preprocessed data by handling missing values, removing duplicates, and ensuring consistency.
- Engineered new features like Loyalty Score, Spending Behavior, and Engagement Level to enhance model performance.

2. Machine Learning Model Selection & Evaluation

- Tried three different models:
 - Logistic Regression – Simple, interpretable, but had lower recall.
 - Random Forest – Performed well with a balanced trade-off between accuracy and interpretability.
 - Gradient Boosting (XGBoost) – Provided the highest accuracy and recall, making it the best choice.
- Evaluation Metrics Used:
 - Accuracy, Precision, Recall, F1 Score, and AUC-ROC to compare model effectiveness.

3. Deployment & Integration

- Best Model for Deployment: Random Forest (best trade-off between performance and interpretability).
- Recommended Deployment Platforms:
 - Flask/FastAPI for lightweight API deployment.
 - AWS Lambda for scalable cloud deployment.
- Input-Output Format:
 - Input: JSON with customer details.
 - Output: JSON with predicted churn probability and churn classification (Yes/No).

4. AI System Testing & Feedback Mechanisms

- Test Cases Conducted:
 - Functional Testing: Checked if the model correctly predicts churn.
 - Performance Testing: Evaluated latency and scalability.
 - Real-World Testing: Ensured business impact assessment.
- Feedback Mechanisms Implemented:
 - User feedback collection for improving predictions.
 - A/B Testing to compare different model versions.
 - Error Analysis to refine false predictions.