SERSHAH ENGINEERING COLLEGE, SASARAM

(DEPT. OF SCIENCE AND TECHNOLOGY, BIHAR)

Barki Kharari, Sasaram, Bihar 821113

SUMMER ENTREPRENEURSHIP – II

(100510P)

ON

ARTIFICIAL INTELLIGENCE & GEN AI

An Internship Report submitted
in partial fulfilment of the requirements
for the award of the degree of

4 Year Full-Time Engineering

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

AASTHA SRIVASTAVA

REGISTRATION NUMBER: 22105124035

SEMESTER: VTH
SESSION: 2022-26

Trained under the Guidance of

YBI FOUNDATION

CERTIFICATE

CERTIFICATE

This is to certify that internship report entitled **"Artificial Intelligence & Gen AI"** which is submitted by **Aastha Srivastava**, in partial fulfilment of the requirements for the award of **Bachelor's degree in Technology (B.Tech.) in Computer Science Engineering** to **Sershah Engineering College**, affiliated from Bihar Engineering University, Patna is a bonafide record of the candidates' own work carried out by them under my supervision. The report has fulfilled standard requirements related to the degree. The matter embodied in this internship report, in full or in parts, is original and has not been submitted for the award of any other degree or diploma.

Mr. Om Prakash
Head of the Department – In-charge,
Computer Science Engineering,
Sershah Engineering College

DECLARATION

I hereby declare that this submission is my own work and that to the best of my knowledge and belief. I also declare that the work which is being presented in this in-plant training report titled "Artificial Intelligence & Gen AI Internship" by me, in partial fulfilment of the requirements for the award of Bachlerate degree in Technology (B.Tech.) in "Computer Science Engineering", is an authentic record of my own work carried out under the guidance of YBI Foundation and Mr. Om Prakash, Head of the Department – In-charge, Computer Science and Engg. at Sershah Engineering College.

This report has been made independently by me during my third year at Sershah Engineering College while pursuing an internship during the period of 19th May, 2025 to 2nd June 2025 (19/05/2025 – 02/06/2025). It contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institutes of higher learning, except where the acknowledgement has been made in the text.

Signature
Name: Aastha Srivastava
Registration No.: 22105124035
Sershah Engineering College

ACKNOWLEDGEMENT

It is my proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.

First and foremost, I wish to record my sincere gratitude to YBI Foundation, Mr. Om Prakash, and other faculty members for their constant support and encouragement in preparation of this report as well as the project.

Last but not the least, I would like to express my gratitude to my parents, family and all faculty members of our Computer Science and Engineering Department for providing academic inputs, guidance & encouragement throughout the training period. Their contributions and technical support in preparing this report are greatly acknowledged.

Name:Aastha Srivastava
Reg. no : 22105124035
Sershah Engineering College

.

Table of Contents

1. EXECUTIVE SUMMARY

This report documents my comprehensive internship experience in Artificial Intelligence and Generative AI. During this period, I gained hands-on experience with essential Python libraries, data visualization techniques, machine learning algorithms, and cutting-edge generative AI technologies. The internship covered fundamental concepts from basic data manipulation to advanced AI applications, providing a solid foundation for future AI development work.

The program was structured to build knowledge progressively, starting with Python fundamentals and culminating in practical generative AI applications. This systematic approach enabled me to understand both theoretical concepts and practical implementations across the AI spectrum.

---

2. INTRODUCTION

Artificial Intelligence has become a transformative force across industries, with Generative AI emerging as one of the most significant technological advances in recent years. This internship was designed to provide comprehensive exposure to both traditional machine learning techniques and modern generative AI applications.

The learning path encompassed essential data science tools, visualization techniques, various machine learning algorithms, and practical applications of generative AI technologies. Each module built upon previous knowledge, creating a cohesive understanding of the AI development pipeline from data preparation to model deployment.

---

3. UNDERSTANDING PYTHON LIBRARIES

3.1 Introduction to Pandas

Pandas serves as the backbone of data manipulation in Python. During this module, I learned to leverage pandas for efficient data handling and analysis.

Key Concepts Covered:

- Data Frame and Series objects
- Data indexing and selection
- Data cleaning and preprocessing
- Grouping and aggregation operations

Practical Example: python import pandas as pd

Creating a Data Frame

data = { 'Name': ['Alice', 'Bob', 'Charlie', 'Diana'], 'Age': [25, 30, 35, 28], 'Salary': [50000, 60000, 70000, 55000] } df = pd.DataFrame(data)

index	Name	Age	Salary
0	Alice	25	50000
1	Bob	30	60000
2	Charlie	35	70000
3	Diana	28	55000

Basic operations

print(df.describe()) print(df.groupby('Age').mean())

3.2 Pandas Read and Write External Files

Understanding file I/O operations is crucial for real-world data science applications. I mastered various file formats and their handling techniques.

File Formats Covered:

- CSV files
- Excel spreadsheets
- JSON files
- Database connections

Implementation Example: python

Reading different file formats

csv_data = pd.read_csv('data.csv') excel_data = pd.read_excel('data.xlsx') json_data = pd.read_json('data.json')

Writing data

df.to_csv('output.csv', index=False) df.to_excel('output.xlsx', index=False)

3.3 Pandas Data Summary

Data summarization techniques are essential for understanding dataset characteristics and identifying patterns before model development.

Summary Techniques:

- Descriptive statistics
- Data type analysis
- Missing value assessment
- Distribution analysis

---

4. DATA VISUALIZATION

4.1 Introduction to Matplotlib

Matplotlib provides the foundation for data visualization in Python. I learned to create various types of plots and customize them for professional presentations.

Basic Plotting Example: python import matplotlib.pyplot as plt import numpy as np

Sample data

x = np.linspace(0, 10, 100) y = np.sin(x)

Creating a basic plot

plt.plot(x, y)

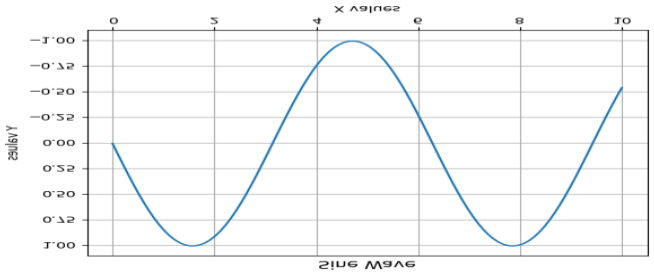
plt.title('Sine Wave')

plt.xlabel('X values')

plt.ylabel('Y values')

plt.grid(True)

plt.show()



4.2 Line Chart

Line charts are essential for visualizing trends over time and continuous data relationships.

4.3 Bar Chart

Bar charts effectively display categorical data comparisons and frequency distributions.

4.4 Histogram

Histograms reveal data distribution patterns and help identify outliers and data skewness.

4.5 Scatter Plot

Scatter plots visualize relationships between two continuous variables and help identify correlations.

Comprehensive Visualization Example: python

Creating subplots for multiple visualizations

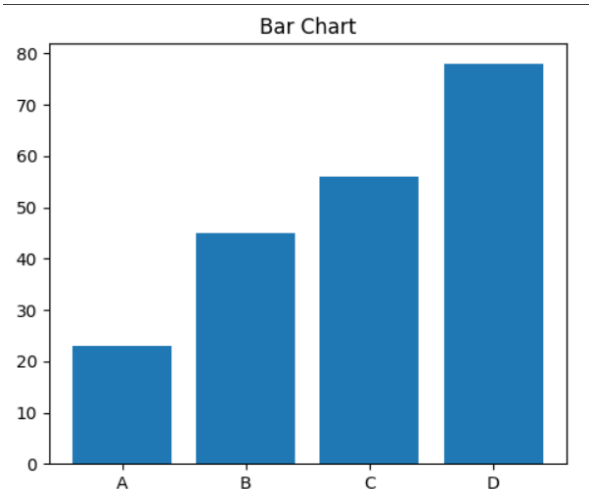
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

Line plot

axes[0,0].plot(x, y) axes[0,0].set_title('Line Plot')

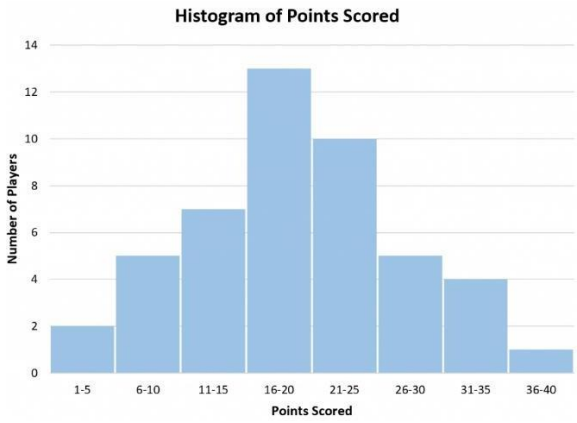
Bar chart

categories = ['A', 'B', 'C', 'D'] values = [23, 45, 56, 78] axes[0,1].bar(categories, values) axes[0,1].set_title('Bar Chart')



Histogram

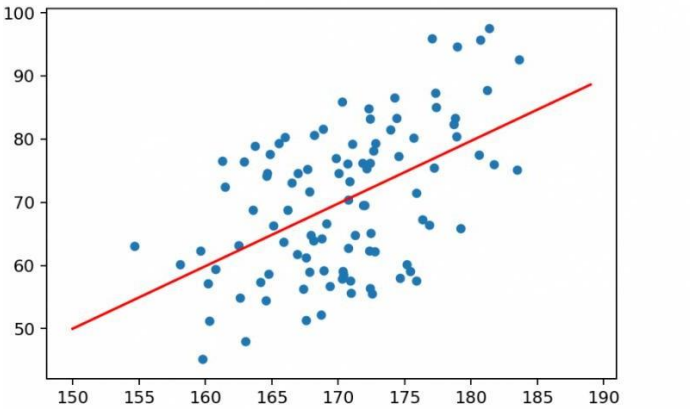
data = np.random.normal(100, 15, 1000) axes[1,0].hist(data, bins=30) axes[1,0].set_title('Histogram')



Scatter plot

x_scatter = np.random.randn(100) y_scatter = 2 * x_scatter + np.random.randn(100) axes[1,1].scatter(x_scatter, y_scatter) axes[1,1].set_title('Scatter Plot')

plt.tight_layout() plt.show()



---

5. UNDERSTANDING MACHINE LEARNING

5.1 What is Machine Learning?

Machine Learning is a subset of artificial intelligence that enables computers to learn and make decisions from data without being explicitly programmed. I learned about the three main types of machine learning:

- *Supervised Learning:* Learning with labeled data
- *Unsupervised Learning:* Finding patterns in unlabeled data
- *Reinforcement Learning:* Learning through interaction and feedback

5.2 Supervised vs Unsupervised Learning (with Examples)

Supervised Learning Applications:

- Email spam detection
- Medical diagnosis
- Price prediction
- Image classification

Unsupervised Learning Applications:

- Customer segmentation
- Anomaly detection
- Data compression
- Market basket analysis

5.3 Basics of Linear Regression Model

Linear regression models the relationship between a dependent variable and independent variables using a linear equation.

Implementation Example: python from sklearn.linear_model import LinearRegression from sklearn.model_selection import train_test_split import numpy as np

Generate sample data

X = np.random.randn(100, 1) y = 2 * X.ravel() + np.random.randn(100)

Split data

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

Train model

model = LinearRegression() model.fit(X_train, y_train)

Make predictions

predictions = model.predict(X_test) print(f"Model coefficient: {model.coef_[0]:.2f}") print(f"Model intercept: {model.intercept_:.2f}")

5.4 Basics of Logistic Regression Model

Logistic regression is used for binary and multi-class classification problems, using the logistic function to model probabilities.

5.5 Basics of KNN Model

K-Nearest Neighbors is a simple, instance-based learning algorithm that classifies data points based on the majority class of their nearest neighbors.

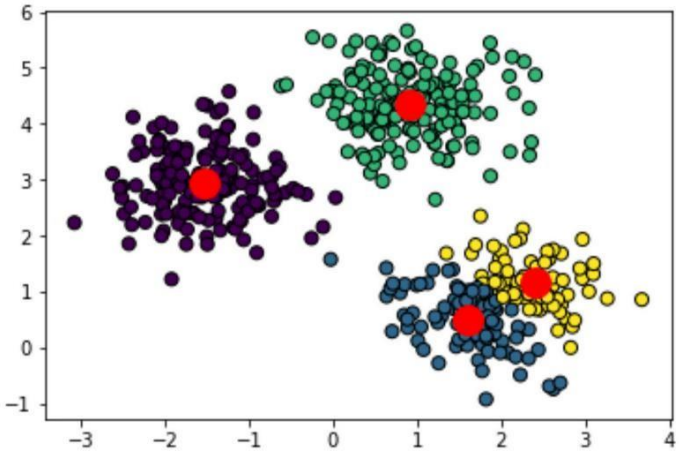
5.6 Basics of Decision Tree Model

Decision trees create a model that predicts target values by learning simple decision rules inferred from data features.

5.7 Basics of Random Forest Model

Random Forest combines multiple decision trees to create a more robust and accurate model while reducing overfitting.

5.8 Basics of K-means Clustering



K-means clustering partitions data into k clusters by minimizing within-cluster sum of squares.

K-means Implementation: python from sklearn.cluster import KMeans from sklearn.datasets import make_blobs

Generate sample data

X, _ = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=0)

Apply K-means

kmeans = KMeans(n_clusters=4, random_state=42) clusters = kmeans.fit_predict(X)

Visualize results

plt.scatter(X[:, 0], X[:, 1], c=clusters, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], c='red', marker='x', s=200) plt.title('K-means Clustering Results') plt.show()

---

6. UNDERSTANDING AI

6.1 What is AI?

Artificial Intelligence refers to the simulation of human intelligence in machines programmed to think and learn like humans. AI encompasses various subfields including machine learning, natural language processing, computer vision, and robotics.

6.2 History and Evolution of AI

The evolution of AI spans several decades:

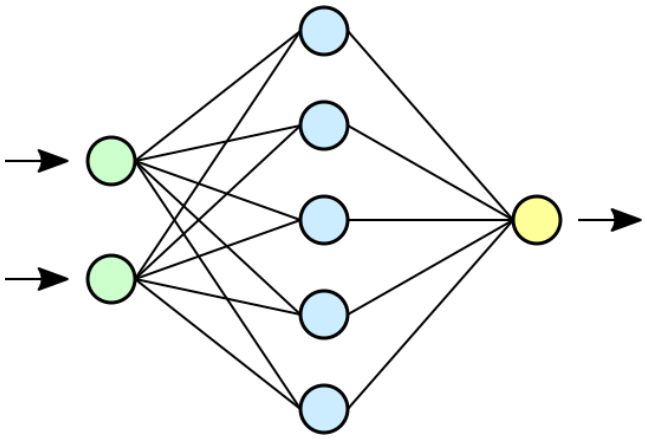
- *1S50s:* Alan Turing's foundational work
- *1Sc0s-70s:* Expert systems development
- *1S80s:* Neural networks resurgence
- *1SS0s-2000s:* Statistical approaches and big data
- *2010s-Present:* Deep learning revolution and generative AI

6.3 AI vs Machine Learning vs Deep Learning

Understanding the relationship between these concepts:

- *AI:* Broadest concept of machine intelligence
- *Machine Learning:* Subset of AI using algorithms to learn from data
- *Deep Learning:* Subset of ML using neural networks with multiple layers

6.4 ANN Regression



Artificial Neural Networks can be used for regression tasks, learning complex non-linear relationships between inputs and continuous outputs.

6.5 ANN Classification

Neural networks excel at classification tasks, learning to distinguish between different categories through pattern recognition.

Simple Neural Network Example: python from sklearn.neural_network import MLPClassifier from sklearn.datasets import make_classification

Generate dataset

X, y = make_classification(n_samples=1000, n_features=20, n_classes=2, random_state=42)

Split data

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

Create and train neural network

nn = MLPClassifier(hidden_layer_sizes=(100, 50), max_iter=1000, random_state=42) nn.fit(X_train, y_train)

Evaluate

accuracy = nn.score(X_test, y_test) print(f"Neural Network Accuracy: {accuracy:.3f}")

---

7. UNDERSTANDING GENERATIVE AI

7.1 What is Generative AI?

Generative AI refers to artificial intelligence systems capable of creating new content, including text, images, audio, and code. These systems learn patterns from training data to generate novel outputs that resemble human-created content.

Key Characteristics:

- Content creation capabilities
- Pattern learning from large datasets
- Probabilistic output generation
- Multi-modal applications

7.2 Tools and Platforms in GenAI (OpenAI, HuggingFace, etc.)

Major Platforms Explored:

OpenAI:

- GPT models for text generation
- DALL-E for image creation

- Codex for code generation

HuggingFace:

- Model hub with thousands of pre-trained models
- Transformers library for easy model integration
- Datasets library for training data access

Other Platforms:

- Google's PaLM and Gemini
- Anthropic's Claude
- Stability AI's Stable Diffusion

7.3 Prompt Engineering Basics

Prompt engineering is the art of crafting effective inputs to get desired outputs from generative AI models.

Key Principles:

- Clarity and specificity
- Context provision
- Role-based prompting
- Output format specification

Example Prompts:

Basic Prompt: "Write a summary of machine learning"

Engineered Prompt: "As a technical writer, create a 200-word summary of machine learning that explains the concept to business executives, focusing on practical applications and benefits."

7.4 Creating Your First Prompt

Learning to structure effective prompts involves understanding model capabilities and limitations while providing clear instructions and context.

7.5 Prompt Engineering Techniques (Zero-shot, Few-shot, Chain-of-Thought)

Zero-shot Learning: Asking the model to perform tasks without examples.

Few-shot Learning: Providing examples to guide the model's responses.

Chain-of-Thought: Encouraging step-by-step reasoning for complex problems.

Example Implementation: python

Few-shot learning example

prompt = """ Classify the sentiment of these reviews:

Review: "This product exceeded my expectations!" Sentiment: Positive

Review: "Terrible quality, waste of money." Sentiment: Negative

Review: "It's okay, nothing special." Sentiment: Neutral

Review: "Amazing features and great value!" Sentiment: """

7.6 Applications of GenAI in Content, Code, and Design

Content Applications:

- Article and blog post generation
- Marketing copy creation
- Technical documentation
- Creative writing assistance

Code Applications:

- Code completion and generation
- Bug detection and fixing
- Code explanation and documentation
- Algorithm optimization

Design Applications:

- Logo and graphic design
- UI/UX mockups
- Image and art generation
- Design iteration and variation

7.7 Ethical Use and Limitations of GenAI

Ethical Considerations:

- Bias in generated content
- Intellectual property concerns
- Misinformation risks

- Privacy and data protection

Limitations:

- Factual accuracy issues
- Context understanding limits
- Inconsistent outputs
- Training data dependencies

7.8 Check Your Understanding

Regular assessment of generative AI concepts through practical exercises and real-world applications ensured comprehensive understanding of the technology's capabilities and constraints.

---

8. PROJECT WORK

During the internship, I completed several practical projects that integrated the learned concepts:

Project 1: Customer Sentiment Analysis

- Analyzed customer reviews using NLP techniques
- Implemented classification models to predict sentiment
- Created visualizations for business insights

Project 2: Sales Forecasting Model

- Built time series forecasting models using historical sales data
- Compared linear regression, decision trees, and neural networks
- Developed interactive dashboards for stakeholder presentation

Project 3: Generative AI Content Assistant

- Developed a prompt engineering framework
- Created templates for various content types
- Implemented few-shot learning techniques for domain-specific applications

---

G. KEY LEARNINGS AND SKILLS ACQUIRED

Technical Skills:

- Proficiency in Python libraries (Pandas, Matplotlib, Scikit-learn)
- Machine learning algorithm implementation and evaluation
- Data visualization and storytelling
- Generative AI prompt engineering
- Neural network architecture understanding

Soft Skills:

- Problem-solving approach to AI challenges
- Critical thinking about AI limitations and ethics
- Project management and documentation
- Presentation and communication of technical concepts

Industry Knowledge:

- Current AI landscape and trends
- Practical applications across industries
- Ethical considerations in AI development
- Best practices for AI project lifecycle

---

10. CHALLENGES AND SOLUTIONS

Challenge 1: Data Quality Issues

- *Problem:* Inconsistent and missing data in real-world datasets
- *Solution:* Implemented comprehensive data cleaning pipelines and validation checks

Challenge 2: Model Overfitting

- *Problem:* Models performing well on training data but poorly on test data
- *Solution:* Applied regularization techniques and cross-validation methods

Challenge 3: Prompt Engineering Complexity

- *Problem:* Getting consistent and reliable outputs from generative AI models
- *Solution:* Developed systematic prompt testing and iteration frameworks

Challenge 4: Ethical AI Implementation

- *Problem:* Ensuring fair and unbiased AI applications
- *Solution:* Implemented bias detection methods and diverse testing scenarios

---

11. CONCLUSION AND FUTURE RECOMMENDATIONS

This internship provided a comprehensive foundation in both traditional machine learning and cutting-edge generative AI technologies. The structured approach from basic Python libraries to advanced AI applications created a solid understanding of the entire AI development pipeline.

Key Achievements:

- Mastered essential data science and machine learning tools
- Developed practical skills in generative AI applications
- Completed multiple real-world projects demonstrating learned concepts
- Gained understanding of ethical AI development practices

Future Recommendations:

For Personal Development:

- Pursue advanced deep learning specializations
- Explore emerging AI technologies like multimodal models
- Develop expertise in AI ethics and governance
- Build portfolio of diverse AI projects

For Organization:

- Invest in advanced AI infrastructure and tools
- Develop comprehensive AI ethics guidelines
- Create cross-functional AI teams
- Establish continuous learning programs for emerging technologies

Industry Trends to Watch:

- Multimodal AI systems combining text, image, and audio
- AI agent frameworks for autonomous task completion
- Edge AI deployment for real-time applications
- Regulatory developments in AI governance