AI PREDICTIVE MODEL FOR CREDIT UNDERWRITING

SYNOPSIS

- **Objective:** Develop an AI-Predictive Model for credit underwriting system to enhance loan approval predictions.
- **Data Collection & Preprocessing:** Handle missing values, feature encoding,outlier removal, normalization.
- **Results & Impact:** GBC improves accuracy, risk assessment, and decision-making, reducing false approvals and rejections.
- **Outcome:** A Streamlit-based AI app for loan predictions, EMI calculations, AI-driven financial advice, and report generation.

AI PREDICTIVE CREDIT UNDERWRITING VS. TRADITIONAL CREDIT UNDERWRITING

AI Predictive Credit Underwriting	Traditional Credit Underwriting
• Automated & Data-Driven	• Manual & Rule-Based
• Advanced Risk Analysis	• Limited Data Usage
• Reduced Bias	• Higher Risk of Bias
• Self-Learning Models	• Fixed Risk Assessment

ML MODELS

1. Logistic Regression
2. Decision Tree
3. Random Forest AdaBoost
4. Naive Bayes
5. Gradient Boosting

TESTING AND EVALUATION

Comparison of Machine Learning Models

#	Model	Training Accuracy	Testing Accuracy	Overall Accuracy
1	Logistic Regression	92.28	92.28	92.28
2	Decision Tree	100.0	98.68	99.38
3	Random Forest	100.0	98.77	99.57
4	AdaBoost	97.36	97.27	97.36
5	Naive Bayes	86.93	93.69	86.88
6	Gradient Boosting	99.67	98.77	99.57

GRADIENT BOOSTING

WHY GRADIENT BOOSTING IS THE BEST CHOICE?

- Higher Accuracy & Performance
- Handles Complex Relationships
- Feature Importance

RESULTS & PERFORMANCE METRICS

- **Accuracy:** 98.78%
- **Precision :** 98.90% (Class 0) , 98.65% (Class 1)
- **Recall :** 98.71% (Class 0) , 98.84% (Class 1)
- **F1-Score :** 98.80% (Class 0) , 98.75% (Class 1)
- **Confusion Matrix:**

  [[537, 7] , [6, 513]]

LIBRARIES

- **Programming Language:** Python
- Streamlit
- FPDF
- pandas
- matplotlib
- joblib
- transformers
- langdetect

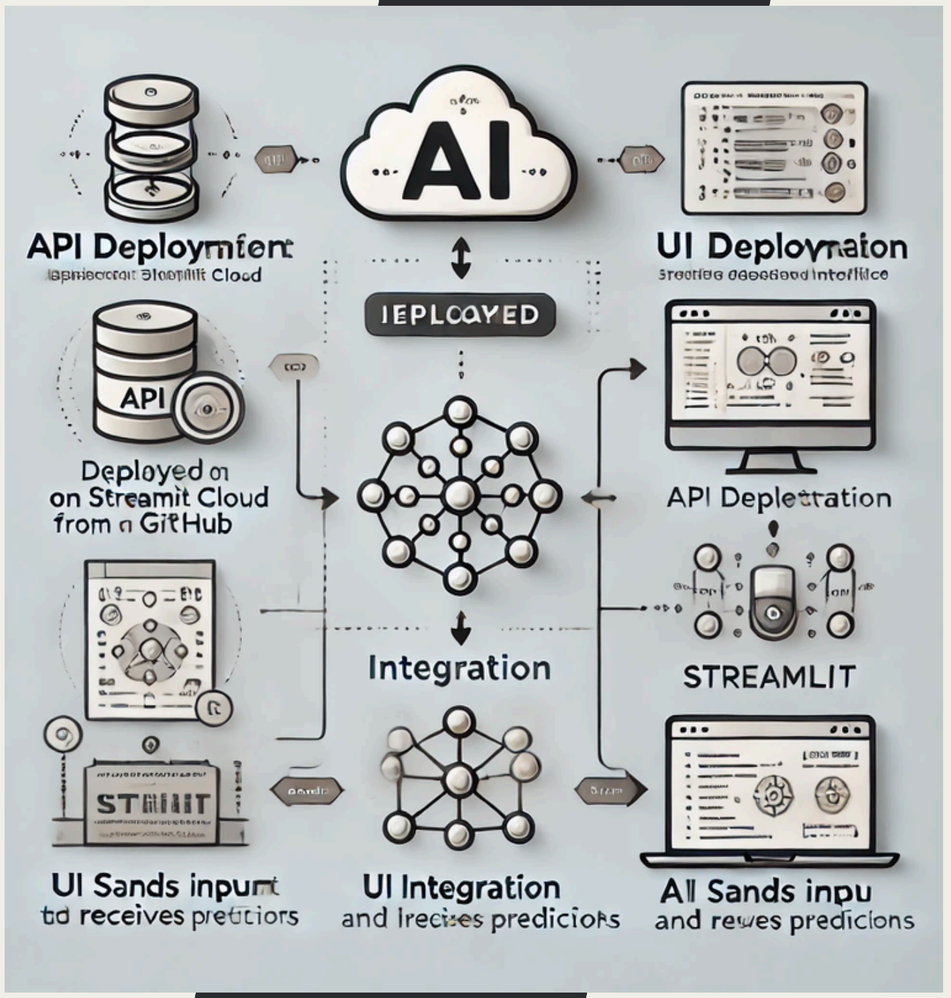
IMPLEMENTATION DEPLOYMENT

1. API Deployment

2. UI Deployment

3. Integration

4. Testing



BENEFITS OF MODEL

🚀 Faster Loan Processing

📊 Data-Driven Decision Making

💰 Reduced Risk of Loan Defaults

🎯 High Accuracy in Predictions

📉 Lower Operational Costs

👥Fair & Transparent Credit Scoring

CHALLENGES IN TRADITIONAL CREDIT UNDERWRITING

- ⏳ Time-Consuming Process
- 👥 Human Bias & Subjectivity
- 📄 Limited Data Utilization
- 💸 High Operational Costs
- ⚠️ Higher Risk of Loan Defaults
- 📉 Inconsistent Decision-Making

BUSINESS PROPOSITION & TARGET CUSTOMERS

- Faster Loan Approvals
- Lower Default Risks
- Cost Efficiency
- NBFCs (Non-Banking Financial Companies)
- Fintech Companies

CASE STUDY

Background:
A leading NBFC faced challenges in loan approval delays, high default rates, and biased decision-making do to manual underwriting.

Problem Statement

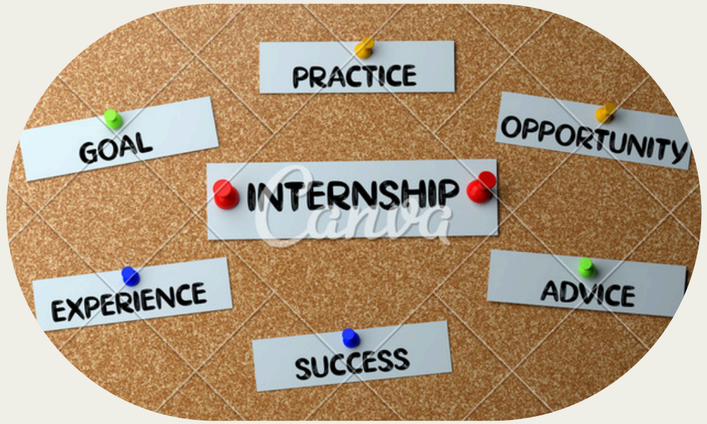
- Lengthy loan approvals (7–10 days).
- High default rates due to poor risk assessment.
- Manual underwriting caused bias & inconsistency in approvals.
- Limited scalability to process high loan application volumes.

CONCLUSION

- After evaluating multiple models, Decision Tree, Random Forest, and Gradient boosting achieved over 99% accuracy, proving their robustness.
- Logistic Regression performed poorly, highlighting its limitations in complex credit risk patterns.
- The final model was deployed using Streamlit for an interactive and user-friendly experience.

ACKNOWLEDGEMENT

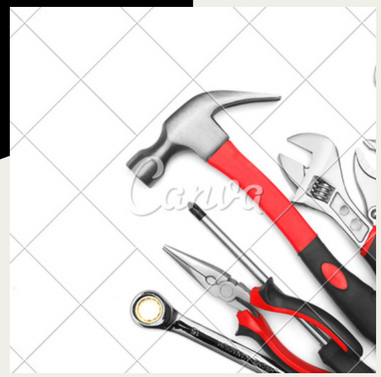
We sincerely thank **Mr. Vivek Gautam sir** for his invaluable guidance and support throughout this internship.



We also extend our gratitude to **Infosys Springboard** for providing this amazing opportunity to apply our academic knowledge to real-world challenges.

Through this project, we gained hands-on experience in: Machine Learning,Model Development, and UI Development.

REFERENCES & ONLINE RESOURCES



Online Resources:

- Scikit-learn Documentation: scikit-learn.org
- Pandas Documentation: pandas.pydata.org
- Kaggle: kaggle.com.

Articles & Learning Materials:

"A Comprehensive Guide to Regression in Machine Learning"

Streamlit Documentation:

https://streamlit.io

Tools for Development & Deployment:

- YouTube Tutorials (YT)
- Google Search & Google Colab (AI & ML Experiments)
- VS Code (Code Editing & Debugging)

THANK YOU!