INFOSYS SPRINGBOARD
BATCH: 3 [DECEMBER 2024-FEBRURAY 2025]
AI PREDICTIVE MODEL FOR CREDIT UNDERWRITING

Abstract

Credit underwriting is a crucial process in financial institutions, determining loan eligibility based on a borrower's financial history and risk profile. This project introduces an AI-driven predictive model for credit underwriting, leveraging Streamlit, machine learning, and data analytics to automate and optimize credit risk assessment.

The model is trained on real-world credit datasets, incorporating key financial indicators such as income, loan amount, debt-to-income ratio, credit score, loan interest, and approval status. Advanced predictive techniques, including Random Forest, Gradient Boosting, and Support Vector Machines (SVM), are implemented to improve loan approval accuracy.

The interactive web application, built using Streamlit, enables financial analysts to input customer details and receive instant predictions on loan approval probability. The system also provides real-time insights, exploratory data analysis (EDA) visualizations, and automated PDF report generation, ensuring transparency and ease of decision-making.

Additionally, the project enhances efficiency, accuracy, and scalability in the underwriting process by eliminating manual assessments and minimizing human biases. By integrating AI-driven analytics, the application supports financial institutions in making data-driven lending decisions, reducing default risks, and improving overall credit evaluation practices. This project serves as a foundation for future enhancements, allowing for deeper financial analysis and more refined risk assessment models.

INFOSYS SPRINGBOARD
BATCH: 3 [DECEMBER 2024-FEBRURAY 2025]
AI PREDICTIVE MODEL FOR CREDIT UNDERWRITING

TASK-1: Dataset Collection from Kaggle and ChatGPT

Dataset Collection

1. **Identify Suitable Datasets:**
   - Search for credit risk or loan underwriting datasets on **Kaggle**.
   - Look for datasets related to **credit scoring, loan approvals, or risk assessment**.
   - Examples:
     - **Give Me Some Credit Dataset**
     - **Home Credit Default Risk**
     - **Lending Club Loan Data**
2. **Download & Explore the Data:**
   - Understand the dataset structure, including:
     - **Features**: Income, loan amount, debt-to-income ratio, loan status, loan interest, loan purpose, residence type, cibil score, bank asset value, etc.
     - **Target Variable**: Default or approval status (loan status).
   - Ensure the data is **comprehensive, diverse, and clean**.
3. **Use ChatGPT for Additional Insights:**
   - Ask ChatGPT to:
     - Suggest synthetic data generation methods if needed.
     - Identify additional variables that could improve the underwriting model.
     - Help with data preprocessing techniques.
     - Identify additional variables that could improve the underwriting model.

4. **Document the Dataset Collection Process:**

- Write about on Google:
  - **Dataset Source (Kaggle link, description).**
  - **Why this dataset is relevant?**
  - **File Name: credit_underwriting1.csv.**

5. **Conclusion:** The dataset collected forms the foundation for developing an AI model for credit underwriting. Further preprocessing and feature engineering will be performed in subsequent tasks.

TASK-2: Data Analysis

1. **Dataset Overview**: Display basic information such as column names, data types, and missing values.
2. **Statistical Summary**: Use the `describe ()` function to summarize numerical columns.
3. **Null Value Analysis**: Identify missing values and their distribution.
4. **Exploratory Data Analysis (EDA)**:
    - Boxplots (to visualize outliers)
    - Bar graphs (for categorical features)
    - Histograms (to show data distribution)

Then we process our file and generate insights.

Dataset Analysis Summary

1. Dataset Overview

- **Total Records**: 5312
- **Columns**: 19
- **No Missing Values**

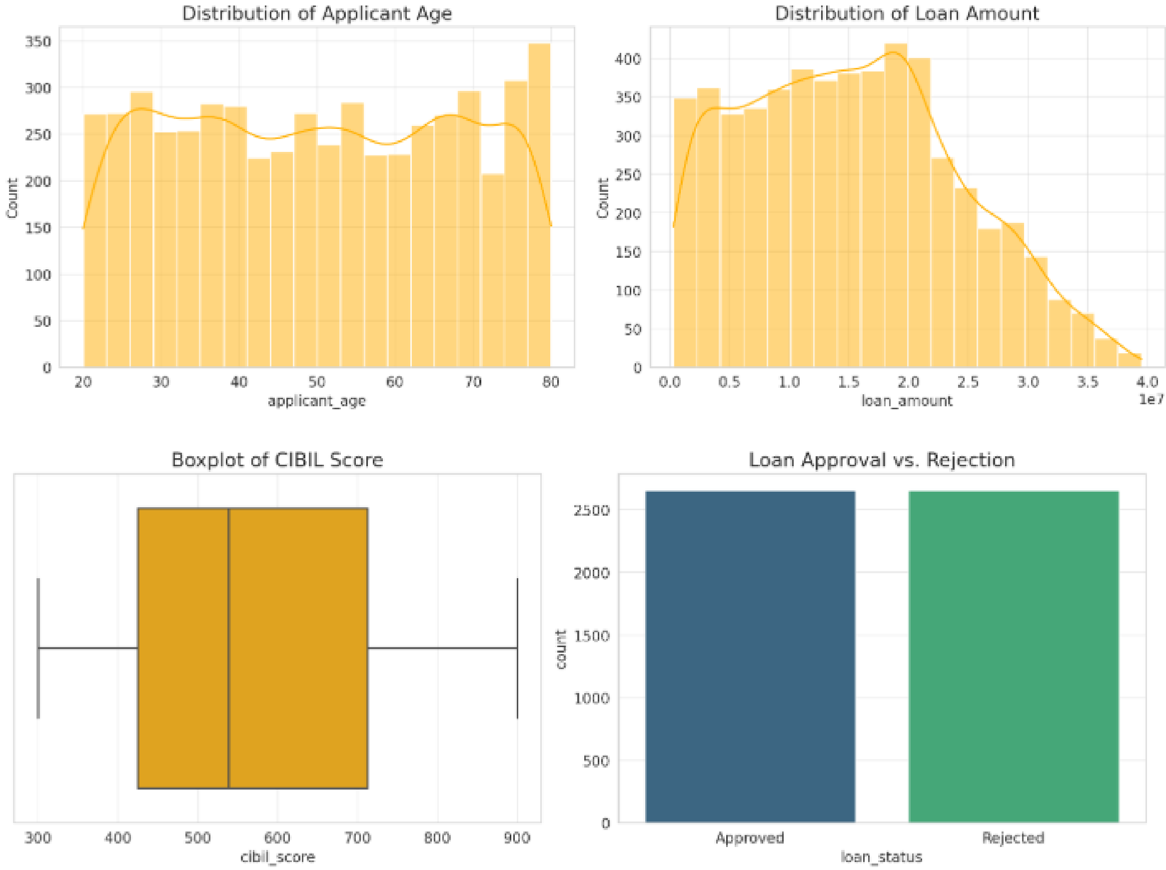
2. Key Features and Statistics

- **Applicant Age**: Ranges from 20 to 80 years (mean: ~50 years).
- **Income**: Varies between ₹200,000 and ₹9.9 million.
- **Loan Amount**: Up to ₹39.5 million (mean: ₹15.16 million).
- **CIBIL Score**: Ranges from 300 to 900.
- **Loan Term**: Between 2 and 20 years.
- **Loan Interest**: Varies between 0% and 10%.

3. Observations

- **Outliers**: Negative residential asset values indicate possible data errors.
- **Loan Status Distribution**: Needs visualization to check approval trends.

Next, we have generated boxplots, bar graphs, and histograms for in-depth analysis.



Here are key visual insights from the dataset:

1. **Applicant Age Distribution**: Most applicants are between 30-70 years old.
2. **Loan Amount Distribution**: Loans vary widely, with a few very high-value loans.
3. **CIBIL Score Boxplot**: Some applicants have very low CIBIL scores, indicating possible risk factors.
4. **Loan Status Bar Chart**: The dataset contains both approved and rejected loans, useful for predictive modelling.

TASK-3: Data Preprocessing & Model Building

1. Data Preprocessing

- Handle missing values (if any).
- Encode categorical variables.
- Normalize numerical features (if needed).
- Split data into training and testing sets.

2. Model Building & Evaluation

- Apply multiple machine learning algorithms:
  - **Logistic Regression**
  - **Decision Tree**
  - **Random Forest**
  - **Gradient Boosting (XGBoost)**
  - **SVM (Support Vector Machine) (for complex decision boundaries).**

3. Model Evaluation Metrics

- **Accuracy, Precision, Recall, F1-Score**
- **ROC-AUC Curve (for classification performance)**
- **Confusion Matrix (to analyse true/false positives and negatives)**

INFOSYS SPRINGBOARD
BATCH: 3 [DECEMBER 2024-FEBRURAY 2025]
AI PREDICTIVE MODEL FOR CREDIT UNDERWRITING

TASK-4: Basic Code And Structure Of Model

Based on the file initial model training. ipynb(in GitHub Repo) uploaded, the current model **loads data, preprocesses it, trains machine learning models, and evaluates their performance.**

Now, **TASK-4 will focus on structuring, optimizing, and making the code modular.** This ensures better organization, reusability, and ease of deployment.

Step 1: Creating a Modular Code Structure: -

Your current script contains everything in a **single file**, which makes it harder to manage. The first step is **splitting the code into modular scripts.**

Advantages of this structure:

- Code is modular → Easy to debug & update
- Reusable components → Can use parts independently
- Ready for deployment → Simplifies API integration

Step 2: Creating Modular Scripts: -

- **Data Preprocessing**
- ➢ **This script will:**
  1. Handle missing values.
  2. Encode categorical variables.
  3. Normalize numerical features.
  4. Split the data into training & test sets.

- **Model Training**
- ➢ **This script will:**
  1. Train **Random Forest & Gradient Boosting** models.
  2. Save the trained models.

- **Model Evaluation**
- **This script will:**
  1. Test the model.
  2. Calculate **accuracy, precision, recall, and F1-score.**
  3. Plot a **confusion matrix.**

- **Model Inference**
- **This script will:**

  1. Load the saved model.
  2. Predict on new data.

TASK-5: Streamlit Cloud

Streamlit is an open-source Python framework that makes it easy to create and deploy interactive web applications for data science and machine learning. It allows developers to build powerful and visually appealing web apps with minimal code by simply using Python scripts.

Key Features of Streamlit:

1. **Easy to Use** – No need for frontend experience; you write Python scripts, and Streamlit takes care of the UI.
2. **Interactive Widgets** – Add sliders, buttons, checkboxes, and text inputs with just one line of code.
3. **Real-Time Updates** – Changes to the Python script are reflected instantly in the web app.
4. **Seamless Integration** – Works well with libraries like Pandas, NumPy, Matplotlib, Scikit-learn, TensorFlow, and OpenAI.
5. **Deploy ability** – Easily deploy apps using Stream lit Community Cloud, AWS, GCP, or Heroku.
6. **Customization** – Allows adding custom CSS, JavaScript, and theming for better UI/UX.

Required Libraries for Streamlit-Based AI Predictive Methods for Credit Underwriting

To build and run a Streamlit-based AI Predictive Methods for Credit Underwriting application, you will need several Python libraries. Below is a categorized list of essential libraries along with installation commands**.**

---

1. Core Streamlit Libraries

- **streamlit** → The main framework for creating the web app.
- **streamlit_extras** (optional) → Provides additional UI enhancements.

Installation:

pip install streamlit streamlit-extras

2. Machine Learning & Data Processing

- **pandas** → Handles data manipulation and analysis**.**
- **numpy** → Supports numerical operations.
- **scikit-learn** → Used for machine learning models (classification, regression, etc.).
- **xgboost or lightgbm (**optional) → Boosting models for loan approval prediction**.**

Installation:

pip install pandas numpy scikit-learn xgboost lightgbm

---

3. Financial & Numerical Libraries

- **scipy** → Supports statistical functions (useful for risk assessment).
- **statsmodels** → For advanced statistical and econometric analysis.
- **numpy-financial** → For financial calculations like EMI.

Installation:

pip install scipy statsmodels numpy-financial

---

4. Model Deployment & Interaction

- **joblib** → For saving and loading ML models.
- **pickle** → Alternative for serializing ML models.
- **fastapi (optional)** → If you need an API backend.
- **openai** → If using AI for financial chatbot functionality.

Installation:

pip install joblib pickle-mixin openai fastapi

5. PDF Report Generation

- **`fpdf` and `fpdf2`** → Generate PDF reports.
- **pdfkit + wkhtmltopdf** → Convert HTML to PDF.

Installation:

pip install reportlab pdfkit

(For wkhtmltopdf, download it separately based on your OS.)

---

6. Visualization & UI Enhancements

- **matplotlib** → For basic plots and charts.
- **seaborn** → Advanced visualizations.
- **plotly** → Interactive graphs.
- **altair** → Lightweight visualization library.

Installation:

pip install matplotlib seaborn plotly altair

---

7. Custom Styling (Optional)

- **beautifulsoup4** → For modifying HTML/CSS within Streamlit.
- **streamlit-option-menu** → For creating sidebar navigation.

Installation:

pip install beautifulsoup4 streamlit-option-menu

Final Installation Command (All at Once)

pip install streamlit pandas numpy scikit-learn xgboost lightgbm scipy statsmodels numpy-financial joblib pickle-mixin openai reportlab pdfkit matplotlib seaborn plotly altair streamlit-extras streamlit-option-menu

TASK-6: Project Structure

Project Structure for AI Predictive Methods for Credit Underwriting

Below is the detailed structure of the project, including descriptions of each file and its purpose.

---

1. Project Directory Structure

```
├── streamlit_app.py  # Main application file
├── requirements.txt  # List of required Python libraries
├── best_features_model.pkl  # Pre-trained machine learning model file
├── FreeSerif.ttf  # Font for generating PDF reports (ensure this font is available)
```

---

2. File Descriptions

1. streamlit_app.py

- This is the **main application file** where the Streamlit web interface is created.
- It loads the machine learning model, handles user input, and generates predictions.
- It also integrates financial calculators and AI chat functionalities.
- Example content inside streamlit_app.py:

Code:

```
import streamlit as st

import pandas as pd

import joblib

# Load the pre-trained model

model = joblib.load("best_features_model.pkl")

st.title("AI Credit Underwriting System")
```

```
user_input = st.text_input("Enter customer details:")

if user_input:

    prediction = model.predict([user_input])

    st.write("Prediction:", prediction)
```

---

2. requirements.txt

- Contains the list of all Python libraries needed to run the project.
- Ensures easy installation of dependencies using:

```
pip install -r requirements.txt
```

- Example content inside requirements.txt:

```
streamlit

pandas

numpy

scikit-learn

joblib

fpdf2
```

---

3. best_features_model.pkl

- A **pre-trained machine learning model** saved in.pkl format using joblib.
- This file allows the app to make predictions without retraining the model.
- It should be generated beforehand using a model training script.
- **Example code for saving the model:**

```
import joblib

from sklearn.ensemble import Gradient boost classifier

model = Gradient boost classifier ()

joblib.dump(model, "best_features_model.pkl")
```

---

4. FreeSerif.ttf

- A **TrueType font file** used for generating PDF reports.
- Required for customizing fonts when exporting reports using fpdf2.
- Ensure the file is available in the directory before generating PDFs.
- **Example :**

```
from fpdf import FPDF

pdf = FPDF()

pdf.add_page()

pdf.set_font("FreeSerif", size=12)

pdf.cell(200, 10, "Credit Report", ln=True, align='C')

pdf.output("credit_report.pdf")
```

---

3. How to Run the Project

1. Install dependencies:

```
pip install -r requirements.txt
```

2. Run the Streamlit app:

```
streamlit run streamlit_app.py
```

3. Ensure best_features_model.pkl and FreeSerif.ttf are in the same directory for proper functionality.!

BENEFITS :-

1. Benefits of the Project

1.1 Improved Credit Risk Assessment

- The AI-powered model enhances the accuracy of credit risk assessment by analyzing financial data and predicting loan approval outcomes.
- Reduces human errors and subjectivity in decision-making.

1.2 Faster and Automated Decision-Making

- Automates the loan underwriting process, enabling financial institutions to process applications quickly.
- Streamlit's interactive interface allows for real-time predictions and adjustments.

1.3 Enhanced User Experience

- Provides a user-friendly web interface for loan officers, customers, or analysts to interact with the system.
- Custom styling and clear visualizations make the application intuitive.

1.4 Integration with Machine Learning Models

- Supports seamless integration with machine learning models stored as .pkl files, allowing easy updates and retraining.
- Models can be improved with additional training data to refine predictions over time.

1.5 Report Generation and Documentation

- Allows users to generate PDF reports of credit evaluations using fpdf2, making it easy to document underwriting decisions.
- Custom fonts ensure professional and readable reports.

1.6 Cost and Resource Efficiency

- Reduces the need for extensive manual reviews by automating credit underwriting workflows.
- Saves costs by streamlining the approval process while maintaining regulatory compliance.

1.7 Scalability and Deployment

- The application can be deployed using Streamlit Cloud, AWS, or on-premise, making it scalable for different financial institutions.
- New features and models can be integrated without significant modifications to the application.

CONCLUSIONS:-

The **AI Predictive Methods for Credit Underwriting** project is built upon a structured methodology encompassing multiple phases:

1. **Dataset Collection** - Identified and gathered relevant datasets from Kaggle, ensuring comprehensive credit underwriting features.

2. **Data Analysis** - Conducted statistical summarization, exploratory data analysis (EDA), and visualizations to understand patterns and trends.

3. **Data Preprocessing & Model Building** - Applied preprocessing techniques such as handling missing values, encoding categorical features, and training various machine learning models (Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and SVM).

4. **Model Evaluation** - Used accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrices to assess model performance.

5. **Application Development** - Integrated the trained model into a Streamlit-based web application for easy accessibility and real-time predictions.

6. **Deployment & Scalability** - Implemented a modular and deployable architecture that allows seamless integration with financial institutions.

With features such as **real-time predictions, automated loan risk assessment, report generation, and an intuitive UI**, this system empowers lenders to make **data-driven credit decisions** with greater confidence.

As AI and machine learning technologies continue to evolve, this project lays a strong foundation for future enhancements, ensuring **better risk management and financial inclusion**. This work serves as an innovative step towards an **intelligent and automated credit underwriting system**, benefiting both financial institutions and customers alike.