**Loan Approval Prediction - Data Analysis & Machine Learning**

**1. Introduction**

This project focuses on analyzing and predicting loan approval based on applicants' demographics, income, and financial history. The dataset is preprocessed to handle missing values, categorical encoding, and feature selection. Multiple classification models are trained and evaluated for performance.

**2. Data Preprocessing**

**2.1 Importing Libraries**

Essential libraries such as pandas, numpy, seaborn, sklearn, and matplotlib are used for data analysis and machine learning.

**2.2 Loading the Dataset**

The dataset loan\_approval\_data.csv is loaded, and the first few records are examined to understand the structure.

**2.3 Handling Missing Values**

* Dropped rows where more than two attributes were missing.
* Imputed missing categorical values randomly (gender, self\_employed, dependents).
* Filled missing numerical values using most frequent values (loan\_amount\_term, credit\_history).

**2.4 Feature Engineering**

* Combined applicantincome and coapplicantincome into a single total\_income feature.
* Dropped non-relevant columns (loan\_id, gender).

**3. Exploratory Data Analysis (EDA)**

**3.1 Visualizing Loan Approvals**

* Histograms and bar plots show trends in loan approvals across gender, married, education, self\_employed, and property\_area.
* Findings:
  + **Married applicants** had higher approval rates.
  + **Graduates** had a better chance of approval.
  + **Self-employed** individuals had lower approval rates.
  + **Urban property applicants** were more frequently approved.

**4. Model Training**

**4.1 Encoding Categorical Variables**

Converted categorical variables into numerical format using LabelEncoder.

**4.2 Splitting Data**

* Split dataset into **train (80%)** and **test (20%)** sets.
* Standardized numerical features using StandardScaler.

**4.3 Models Used**

* **Logistic Regression**
* **K-Nearest Neighbors (KNN)**
* **Naive Bayes**
* **Decision Tree**
* **Random Forest**
* **AdaBoost Classifier**
* **Gradient Boosting Classifier**
* **Stacking Classifier** (combination of multiple models)

**4.4 Model Evaluation**

* Evaluated using **accuracy score** and **confusion matrix**.
* Identified best-performing model based on test accuracy.

**5. Results & Conclusion**

* Random Forest and Gradient Boosting yielded the highest accuracy.
* Logistic Regression provided a baseline but was outperformed by ensemble models.
* Insights derived from EDA aligned with model predictions.

**6. Future Improvements**

* Hyperparameter tuning using GridSearchCV.
* Feature selection using SHAP values.
* Deploying the model using Flask or Django for real-time predictions.

**7. Code Repository**

The complete project, including dataset and Jupyter Notebook, is available on **GitHub Repository** (Provide link).

**Loan Approval Prediction System Front End**

**Overview**

The Loan Approval Prediction System is a machine learning-powered web application developed using Flask. It aims to predict loan approvals based on user-provided financial and demographic data. The system utilizes a trained machine learning model and integrates user authentication for secure access.

**Features**

* **User Authentication**: Secure login and registration using Flask-Login and Flask-WTF.
* **Loan Prediction Model**: Utilizes a pre-trained ML model to determine loan approval chances.
* **Interactive Web Interface**: Built using Flask with HTML, CSS, and Bootstrap.
* **Database Management**: Uses SQLAlchemy with MySQL for user management.
* **Real-Time Prediction**: Processes user inputs to provide instant loan approval feedback.

**Tech Stack**

* **Backend**: Flask, Flask-Login, Flask-WTF, Flask-Bcrypt
* **Frontend**: HTML, CSS, Bootstrap
* **Database**: MySQL, SQLAlchemy
* **Machine Learning**: Scikit-learn, NumPy, Pandas

**Project Structure**

LoanApprovalSystem/

│-- static/

│-- templates/

│ ├── index.html

│ ├── login.html

│ ├── register.html

│ ├── predict.html

│-- model.pkl # Pre-trained ML model

│-- app.py # Flask application

│-- requirements.txt

│-- README.md

**Implementation Details**

**User Authentication**

* Users can register and log in using a secure authentication system.
* Passwords are hashed using Flask-Bcrypt.
* Flask-Login handles session management.

**Loan Prediction Workflow**

1. Users input financial details such as income, loan amount, and credit history.
2. The system preprocesses the data and extracts relevant features.
3. The trained ML model predicts whether the loan will be approved or not.
4. The result is displayed on the prediction page.

**Model Training**

* The model was trained using historical loan application data.
* Features include applicant income, loan amount, credit history, and property area.
* Various algorithms were tested, with the final model chosen based on accuracy and performance.

**Deployment**

* The application is designed to be deployed on a cloud platform such as AWS, Heroku, or DigitalOcean.
* Uses Gunicorn for running in a production environment.
* Database hosted on a MySQL server.

**Future Enhancements**

* Implement role-based access control for admins and users.
* Enhance UI with better visualization tools.
* Train a more sophisticated model with deep learning techniques.

**Conclusion**

The Loan Approval Prediction System is a practical application that leverages machine learning to assist users in assessing their loan eligibility. With its secure authentication and real-time predictions, it provides an intuitive user experience for financial decision-making.

Github🡪 https://github.com/yashzob/loanPredictor/tree/master