Credit Scoring Model to assess Loan-worthiness

Ashmit Thawait, COE, athawait_be22@thapar.edu
Anchit Gupta, COE, agupta9_be22@thapar.edu

1. **Introduction:**
   A. Technical background of project
   B. Technical Concepts used
   C. Motivation
   D. Problem Statement
   E. Area of application
   F. Dataset and input format
2. **Objective**
   A. Main Objective
   B. Sub Objective (if any)
3. **Methodology**
   A. Steps
   B. Deliverable of each steps or phase
4. **Working Model**
   A. Technical Diagram
   B. Working Module
   C. Attained Deliverable
5. **Results**
   A. Outcome Graphs
   B. Comparative Studies
6. **Conclusion**
   A. Justification of Objectives
   B. Future Scope
7. **References**

Submitted by:	**Submitted to:**
Ashmit Thawait	Dr. Nitin Arora
102203790	Assistant Prof.
Anchit Gupta	Dept. of CSE
102203766	TIET, Patiala

Introduction:

Technical background of project:

Credit scoring is a critical aspect of financial decision-making, where applicants are evaluated for their ability to repay loans based on various factors. With advancements in data science and machine learning, automated systems for predicting loan-worthiness have become a practical and efficient alternative to traditional methods. This project utilizes machine learning algorithms to classify applicants into binary categories - loan-worthy or not, based on their responses to specific questions.

Technical Concepts used:

The project employs a Gradient Boosting Classifier, a robust ensemble learning technique known for its accuracy in handling classification problems. Hyperparameter tuning is incorporated to optimize the model's performance. The web application is built using the Streamlit framework, which provides an interactive and user-friendly interface for input and output. The model's deployment leverages Streamlit Community Cloud for ease of access and scalability.

Motivation:

Financial institutions often face delays and inconsistencies in loan assessment due to manual processes and subjective evaluations. This project aims to simplify and standardize the process by providing a reliable, quick, and automated tool for credit score prediction. It bridges the gap between advanced machine learning tools and real-world usability for financial entities.

Problem Statement:

How can financial institutions efficiently predict an applicant's creditworthiness to expedite loan approval processes while minimizing risk?

Area of application:

This solution is designed for financial institutions, including banks, microfinance organizations, and peer-to-peer lending platforms. It can also benefit fintech companies focused on automating lending services and risk assessment.

Dataset and input format:

The model was trained using a dataset sourced from Kaggle. The dataset includes relevant features such as income, age, employment status, housing status, and other personal or financial attributes. For predictions, the user provides their responses through an interactive questionnaire in the web app. These inputs are processed and fed into the machine learning model to determine their loan-worthiness.

Objective:

The primary objective of this project is to develop an efficient, accurate, and user-friendly system for predicting the credit score or loan-worthiness of applicants. By leveraging machine learning and an intuitive web interface, the project aims to streamline loan assessment processes for financial institutions, ensuring quick decision-making and reduced risk.

Methodology:

Step 1: Data Acquisition, Preprocessing, and Resampling

- Source the dataset from Kaggle, which includes features such as income, employment status, and credit history.
- Perform data cleaning to handle missing values, outliers, and inconsistencies.
- Conduct exploratory data analysis (EDA) to understand feature distributions, correlations, and patterns.
- Apply feature engineering, such as encoding categorical variables and scaling numerical features, to prepare the data for modeling.
- Address class imbalance using **SMOTE (Synthetic Minority Oversampling Technique)** to generate synthetic samples for the minority class, ensuring balanced training data for the model.

Step 2: Model Selection and Training

- Choose the Gradient Boosting Classifier for its high performance in classification tasks.
- Split the dataset into training and testing subsets.
- Apply hyperparameter tuning using techniques like Grid Search or Random Search to optimize model performance.

Step 3: Web Application Development

- Build an interactive web app using Streamlit to collect user input and display predictions.
- Create a user-friendly form to capture relevant applicant data.
- Integrate the trained model into the app for real-time predictions.

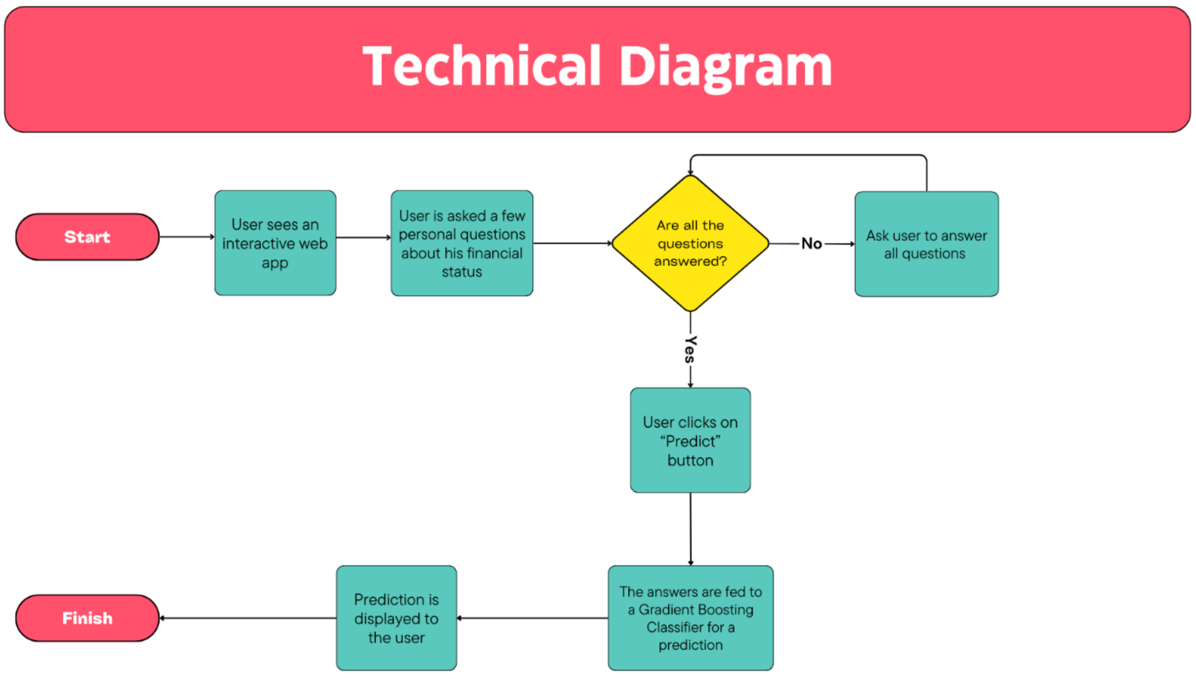
Step 4: Model Deployment

- Deploy the app on Streamlit Community Cloud for easy accessibility.
- Ensuring scalability and testing the app's functionality with diverse user inputs.

Deliverables of Each Step or Phase

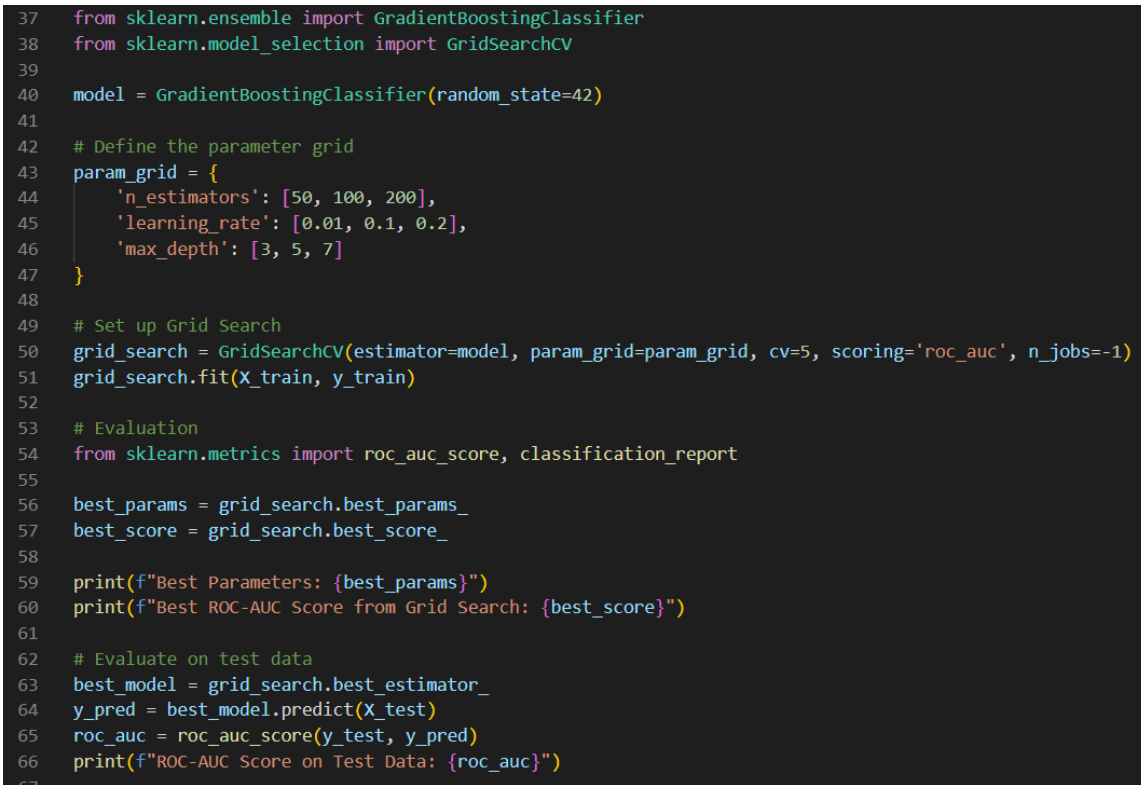
- **Step 1: Data Acquisition and Preprocessing**
  Deliverable: A cleaned and feature-engineered dataset ready for training.
- **Step 2: Model Selection and Training**
  Deliverable: A hyperparameter-optimized Gradient Boosting Classifier with high prediction accuracy, boasting a ROC-AUC score of 0.903.
- **Step 3: Web Application Development**
  Deliverable: An interactive Streamlit-based web application that collects user data and generates predictions.
- **Step 4: Model Deployment**
  Deliverable: A fully functional and publicly accessible web application hosted on Streamlit Community Cloud.

Working Model


Technical Diagram
Start
User sees an interactive web app
User is asked a few personal questions about his financial status
Are all the questions answered?
No
Ask user to answer all questions
Yes
User clicks on "Predict" button
The answers are fed to a Gradient Boosting Classifier for a prediction
Prediction is displayed to the user
Finish

Working Module

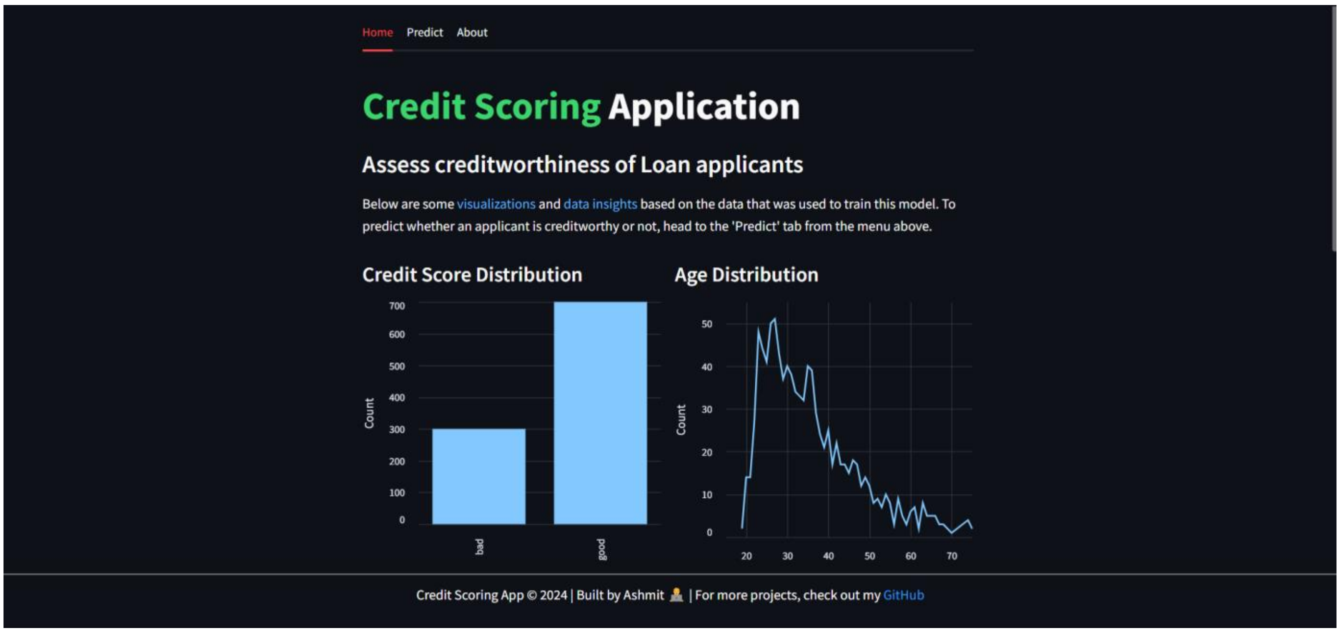
The hyperparameter tuning and evaluation of the model. It achieved a ROC-AUC score of 0.903, which makes it quite good at distinguishing between the positive and negative classes.



```python
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import GridSearchCV

model = GradientBoostingClassifier(random_state=42)

# Define the parameter grid
param_grid = {
    'n_estimators': [50, 100, 200],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7]
}

# Set up Grid Search
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5, scoring='roc_auc', n_jobs=-1)
grid_search.fit(X_train, y_train)

# Evaluation
from sklearn.metrics import roc_auc_score, classification_report

best_params = grid_search.best_params_
best_score = grid_search.best_score_

print(f"Best Parameters: {best_params}")
print(f"Best ROC-AUC Score from Grid Search: {best_score}")

# Evaluate on test data
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
roc_auc = roc_auc_score(y_test, y_pred)
print(f"ROC-AUC Score on Test Data: {roc_auc}")
```

Attained Deliverable

Home page of the web app.



Credit Scoring Application

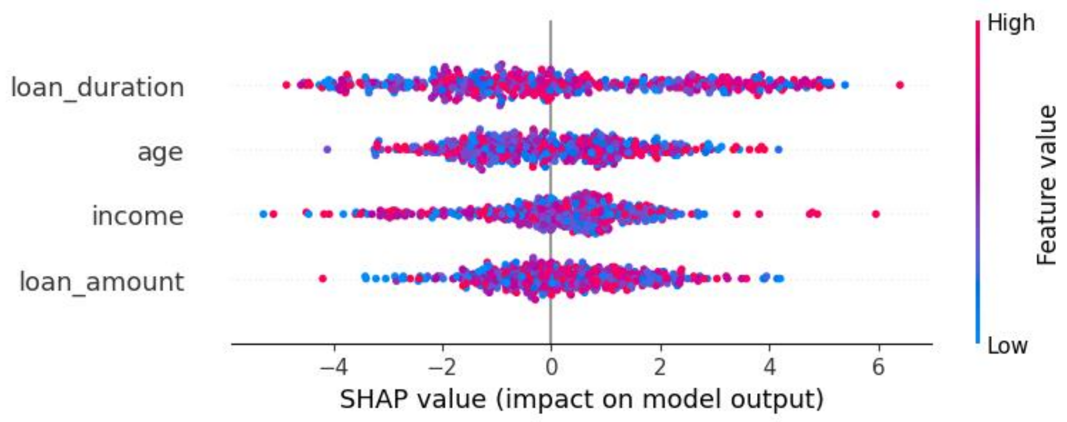
Assess creditworthiness of Loan applicants

Below are some visualizations and data insights based on the data that was used to train this model. To predict whether an applicant is creditworthy or not, head to the 'Predict' tab from the menu above.

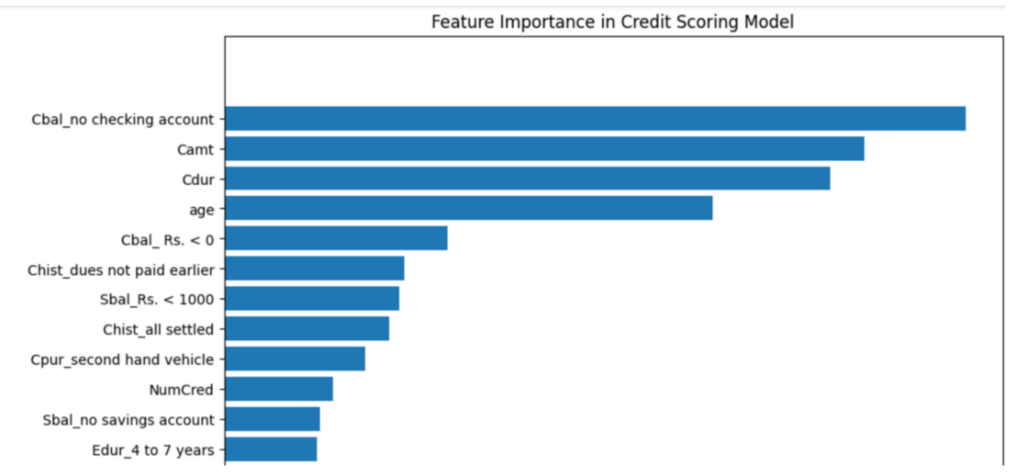
Credit Scoring App © 2024 | Built by Ashmit 👷 | For more projects, check out my GitHub

Results

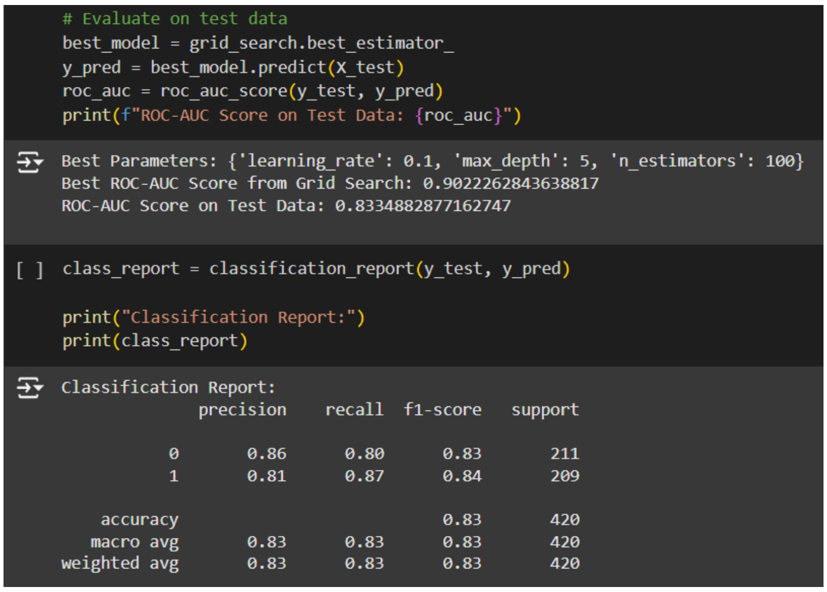
Shap values of some important features:



Some of the most important feature values:



ROC-AUC score and Classification Report of the model:



```
# Evaluate on test data
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)
roc_auc = roc_auc_score(y_test, y_pred)
print(f"ROC-AUC Score on Test Data: {roc_auc}")
```

```
Best Parameters: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100}
Best ROC-AUC Score from Grid Search: 0.9022262843638817
ROC-AUC Score on Test Data: 0.8334882877162747
```

```
[ ] class_report = classification_report(y_test, y_pred)

    print("Classification Report:")
    print(class_report)
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.80      0.83       211
           1       0.81      0.87      0.84       209

    accuracy                           0.83       420
   macro avg       0.83      0.83      0.83       420
weighted avg       0.83      0.83      0.83       420
```

Comparative Studies:

The evaluation of the Gradient Boosting Classifier in this project included a comparison with other widely used machine learning algorithms to ensure its suitability for credit score prediction. Comparative studies were conducted based on various performance metrics, such as accuracy, precision, recall, and F1-score, using the same dataset and preprocessing pipeline.

Algorithms Compared

1. **Logistic Regression**
    - Strength: Simple, interpretable, and effective for binary classification problems.
    - Limitation: Struggles with complex patterns and non-linear relationships in data.
2. **Decision Tree Classifier**
    - Strength: Easy to visualize and understand, capable of capturing non-linear patterns.
    - Limitation: Prone to overfitting, resulting in reduced generalizability.
3. **Random Forest Classifier**
    - Strength: Ensemble-based model with good generalization, reducing overfitting.
    - Limitation: Computationally intensive compared to simpler models.
4. **Gradient Boosting Classifier** (Chosen Model)
    - Strength: Combines weak learners iteratively to achieve superior accuracy and performance, making it well-suited for imbalanced datasets.
    - Limitation: Higher computational cost compared to simpler models.

Results of Comparative Studies

- The **Gradient Boosting Classifier** consistently outperformed the other algorithms in terms of accuracy and recall, which are critical for minimizing false negatives (classifying a loan-worthy applicant as non-loan-worthy).
- While **Random Forest** offered competitive results, Gradient Boosting provided better optimization during hyperparameter tuning, resulting in slightly higher F1-scores.
- Logistic Regression, although computationally efficient, fell short in capturing the complexities of the dataset, resulting in lower accuracy.

Conclusion:

Justification of Objectives

- **Efficiency**: Automating credit score prediction reduces the time spent on manual evaluations, enabling quicker decision-making.
- **Accuracy**: The use of Gradient Boosting Classifier ensures reliable results, minimizing false positives or negatives in creditworthiness evaluations.
- **User Accessibility**: With a Streamlit web app, both financial institutions and end-users can easily interact with the system.
- **Scalability**: Deployment on Streamlit Community Cloud allows the solution to be accessible anywhere, ensuring broader usability.

Future Scope

- **Enhanced Models**: Experimenting with other machine learning models, such as neural networks or ensemble techniques, to improve prediction accuracy.
- **Feature Expansion**: Incorporating additional features like social media behavior, geolocation, or transaction history to refine the model further.
- **Credit Risk Profiling**: Extending the application to provide detailed risk analysis reports alongside binary predictions.
- **Integration with Financial Systems**: Linking the application with financial institution databases for seamless workflow integration.

References:

https://www.kaggle.com/datasets/bbjadeja/predicting-creditworthiness

https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137

https://www.datacamp.com/tutorial/guide-to-the-gradient-boosting-algorithm

https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/

https://docs.streamlit.io/