Task 2: Credit Risk Prediction Analysis

Machine Learning Project Report

Project Overview

This project demonstrates a complete machine learning pipeline for credit risk assessment using loan application data. The analysis implements data preprocessing, exploratory visualization, and classification modeling to predict loan approval status based on applicant characteristics and financial information.

Dataset Description

The loan prediction dataset contains 614 loan applications with 13 features:

- **Demographic**: Gender, Married status, Dependents, Education
- **Financial**: ApplicantIncome, CoapplicantIncome, LoanAmount, Credit_History
- **Property**: Property_Area (Urban/Semiurban/Rural)
- **Loan Details**: Loan_Amount_Term, Self_Employed status
- **Target**: Loan_Status (Y/N for approval/rejection)

Technical Implementation

Libraries Used

- **pandas & numpy**: Data manipulation and numerical operations
- **matplotlib & seaborn**: Statistical visualization

- **scikit-learn**: Machine learning pipeline including:
  - `train_test_split`: Data splitting
  - `LabelEncoder`: Categorical encoding
  - `StandardScaler`: Feature scaling
  - `LogisticRegression`: Classification algorithm
  - `accuracy_score, confusion_matrix`: Model evaluation

Data Loading and Inspection

```python
df = pd.read_csv('loan_prediction.csv')
```

Initial analysis reveals dataset structure (614 rows × 13 columns) and identifies missing values across multiple features, requiring comprehensive data cleaning.

Data Preprocessing Pipeline

1. Missing Value Treatment

Categorical Variables: Filled with mode (most frequent value)

- Gender, Married, Dependents, Self_Employed, Credit_History, Loan_Amount_Term

Numerical Variables: Filled with median

- LoanAmount (maintains distribution integrity)

Target Variable: Rows with missing Loan_Status dropped (ensures clean training data)

2. Data Cleaning Results

Post-cleaning verification ensures no missing values remain, creating a complete dataset ready for modeling.

Exploratory Data Analysis

1. Loan Amount Distribution

```python
sns.histplot(df['LoanAmount'], kde=True)
```

Purpose: Analyzes loan amount patterns and distribution shape

- Identifies typical loan ranges
- Reveals distribution skewness
- Helps understand lending patterns

2. Education vs Loan Status

```python
sns.countplot(x='Education', hue='Loan_Status', data=df)
```

Purpose: Examines education impact on loan approval

- Compares graduate vs non-graduate approval rates
- Identifies education as potential predictor

- Shows class distribution across education levels

3. Applicant Income Distribution

```python
sns.histplot(df['ApplicantIncome'], kde=True)
```

Purpose: Understanding income patterns among applicants

- Reveals income distribution characteristics
- Identifies potential outliers
- Assists in feature scaling decisions

Machine Learning Pipeline

1. Feature Encoding

```python
le = LabelEncoder()
for col in df.select_dtypes(include='object').columns:
    df[col] = le.fit_transform(df[col])
```

Process: Converts all categorical variables to numerical format

- Systematic encoding of all object-type columns
- Maintains data relationships while enabling ML processing
- Prepares data for mathematical operations

2. Feature Scaling and Splitting

```python
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

Feature Scaling: Standardizes all features to mean=0, std=1 **Data Splitting**: 80% training, 20% testing with fixed random state for reproducibility

3. Model Training

```python
model = LogisticRegression(max_iter=2000)
model.fit(X_train, y_train)
```

Algorithm Choice: Logistic Regression selected for:

- Binary classification suitability
- Interpretable coefficients
- Probabilistic output
- Fast training and prediction

Configuration: Increased max_iter=2000 to ensure convergence

Model Evaluation

1. Accuracy Assessment

```python
accuracy = accuracy_score(y_test, y_pred)
```

Metric: Overall prediction accuracy on test set

- Measures percentage of correct predictions
- Primary performance indicator
- Baseline for model comparison

2. Confusion Matrix Analysis

```python
cm = confusion_matrix(y_test, y_pred)
ConfusionMatrixDisplay(confusion_matrix=cm).plot()
```

Analysis Components:

- **True Positives**: Correctly predicted approvals
- **True Negatives**: Correctly predicted rejections
- **False Positives**: Incorrectly predicted approvals (Type I error)
- **False Negatives**: Incorrectly predicted rejections (Type II error)

Business Impact: False negatives represent missed opportunities, while false positives indicate potential bad loans.

Key Insights and Results

Data Characteristics

- **Missing Data**: Significant missing values required systematic imputation
- **Feature Distribution**: Income and loan amounts show right-skewed distributions
- **Class Balance**: Dataset shows natural loan approval patterns

Model Performance

- **Accuracy Score**: Quantifies overall prediction performance
- **Confusion Matrix**: Reveals model strengths and weaknesses in classification
- **Convergence**: Successful model training with adequate iterations

Feature Importance

- **Demographic Factors**: Education and marital status influence approvals
- **Financial Factors**: Income and credit history are key predictors
- **Property Location**: Geographic factors affect lending decisions

---

Business Applications

Risk Assessment

- **Automated Screening**: Reduces manual loan review workload
- **Consistent Evaluation**: Standardized approval criteria
- **Risk Quantification**: Probability-based decision making

Process Optimization

- **Faster Processing**: Immediate preliminary decisions
- **Resource Allocation**: Focus human reviewers on borderline cases
- **Data-Driven Policies**: Evidence-based lending criteria

Regulatory Compliance

- **Audit Trail**: Documented decision-making process
- **Fair Lending**: Consistent application of criteria
- **Performance Monitoring**: Trackable model performance

Technical Considerations

Model Limitations

- **Linear Assumptions**: Logistic regression assumes linear relationships
- **Feature Independence**: May miss complex feature interactions
- **Threshold Sensitivity**: Binary decisions require optimal cut-off points

Scalability

- **Fast Prediction**: Real-time scoring capability
- **Memory Efficient**: Lightweight model suitable for production
- **Easy Updates**: Simple retraining with new data

Future Enhancements

- **Feature Engineering**: Create derived variables (debt-to-income ratio)
- **Advanced Models**: Try ensemble methods (Random Forest, XGBoost)
- **Cross-Validation**: More robust performance estimation

Conclusion

This credit risk prediction project demonstrates a complete machine learning workflow from data preprocessing through model evaluation. The systematic approach to handling missing data, feature encoding, and model training creates a robust foundation for loan approval predictions. The logistic regression model provides interpretable results suitable for business decision-making while maintaining good predictive performance.

The project establishes essential practices for financial machine learning applications, including proper data handling, appropriate algorithm selection, and comprehensive evaluation metrics. This foundation supports both immediate business applications and future model enhancements.

Project Type: Binary Classification - Credit Risk Assessment
Dataset: 614 loan applications, 11 features, 1 target variable
Algorithm: Logistic Regression with Standard Scaling
Tools: Python, pandas, scikit-learn, matplotlib, seaborn