LOAN APPROVAL PREDICTION

Author: ABHIJITH P

Date: 01.07.2024

---

1. Overview of Problem Statement

This project aims to predict the loan approval status of applicants using various machine learning models. The dataset includes several features relevant to the loan applicants, such as their education level, employment status, income, loan amount, and asset values. The objective is to build, evaluate, and compare the performance of multiple models to identify the most accurate one for loan approval prediction.

---

2. Objective

Develop a model that accurately predicts loan approval status based on applicant information and financial details.

---

3. Data Overview

- **Data Source:** Historical loan approval data collected from Kaggle Loan-Approval-Prediction-Dataset.
- **Data Import:** The data is imported using Pandas library in Python.
- **Description:** The dataset includes features such as the number of dependents, education level, employment status, annual income, loan amount, loan term, CIBIL score, residential assets value, commercial assets value, luxury assets value, bank asset value, and loan status.
- **Size:** The dataset contains 4269 rows 13 columns.
- **Attributes**:
  - loan_id: Unique identifier for each loan application.
  - no_of_dependents: Number of dependents of the applicant.
  - education: Applicant's education level.
  - self_employed: Whether the applicant is self-employed.
  - income_annum: Annual income of the applicant.
  - loan_amount: Loan amount requested.
  - loan_term: Term of the loan.

- cibil_score: CIBIL score of the applicant.
- residential_assets_value: Value of the applicant's residential assets.
- commercial_assets_value: Value of the applicant's commercial assets.
- luxury_assets_value: Value of the applicant's luxury assets.
- bank_asset_value: Value of the applicant's bank assets.
- loan_status: Status of the loan (approved or rejected).

---

3. Data Preparation and Cleaning

- **Data Preparation:** The loan_id column was dropped as it is not relevant for the prediction. Categorical features such as education, self_employed, and loan_status were encoded using LabelEncoder.
- **Handling Missing Values:** Missing values were imputed using mean, median, or mode as appropriate.
- **Removing Duplicates:** Duplicate records were detected and removed.

```
# Check for missing values
print(data.isnull().sum())

#Remove leading/trailing spaces in column names
data.columns = data.columns.str.strip()

# Encode categorical variables
le = LabelEncoder()
data['education'] = le.fit_transform(data['education'])
data['self_employed'] = le.fit_transform(data['self_employed'])
data['loan_status'] = le.fit_transform(data['loan_status'])

# Drop the loan_id column as it's not needed for modeling
data = data.drop(['loan_id'], axis=1)
```

---

4. Exploratory Data Analysis

- **Missing Values**: The dataset was checked for missing values, and any rows containing missing values were dropped.
- **Data Statistics**: Basic statistics of the dataset were computed to understand the distribution of values.

---

5. Data Splitting

The cleaned dataset was split into training and testing sets.

```
# Split the data into features and target variable
X = data.drop(["loan_status"], axis=1)
y = data["loan_status"]

# Split the data into training and test sets
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

---

6. Model Building and Evaluation

Selected five models based on the problem nature and data:

1) Logistic Regression
2) Decision Tree
3) Random Forest Classifier
4) Support Vector Classifier (SVC)
5) Gradient Boosting Classifier

- **Logistic Regression**

```
reg = LogisticRegression()
reg.fit(x_train, y_train)
regpred = reg.predict(x_test)
```

- **Decision Tree**

```
dtree = DecisionTreeClassifier()
dtree.fit(x_train, y_train)
dtreepred = dtree.predict(x_test)
```

- **Random Forest**

```
rf = RandomForestClassifier(n_estimators=100)
rf.fit(x_train, y_train)
rfpred = rf.predict(x_test)
```

- **Gradient Boosting**

```
gb = GradientBoostingClassifier()
gb.fit(x_train, y_train)
gbpred = gb.predict(x_test)
```

- **Support Vector Classifier (SVC)**

```
svm = SVC()
svm.fit(x_train, y_train)
svmpred = svm.predict(x_test)
```

---

7. Model Performance Comparison

- **Logistic Regression**: Accuracy = 61.83 %,

Report:	precision	recall	f1-score	support
0	0.62	1.00	0.76	264
1	0.50	0.01	0.01	163
accuracy			0.62	427
macro avg	0.56	0.50	0.39	427
weighted avg	0.57	0.62	0.48	427

- **Decision Tree**: Accuracy = 97.89 %,

Report:	precision	recall	f1-score	support
0	0.98	0.98	0.98	264
1	0.97	0.98	0.97	163
accuracy			0.98	427
macro avg	0.98	0.98	0.98	427
weighted avg	0.98	0.98	0.98	427

- **Random Forest**: Accuracy = 97.89%,

Report:	precision	recall	f1-score	support
0	0.97	0.99	0.98	264
1	0.99	0.96	0.97	163
accuracy			0.98	427
macro avg	0.98	0.97	0.98	427
weighted avg	0.98	0.98	0.98	427

- **Gradient Boosting**: Accuracy = 97.42%,

Report:	precision	recall	f1-score	support
0	0.97	0.98	0.98	264
1	0.97	0.96	0.97	163
accuracy			0.97	427
macro avg	0.97	0.97	0.97	427
weighted avg	0.97	0.97	0.97	427

- **Support Vector Machine**: Accuracy = 61.82%,

Report:	precision	recall	f1-score	support
0	0.62	1.00	0.76	264
1	0.00	0.00	0.00	163
accuracy			0.62	427
macro avg	0.31	0.50	0.38	427
weighted avg	0.38	0.62	0.47	427

---

8. Feature Importance & Feature Selection:

Feature importance: It was computed for the Random Forest model to identify the most significant features influencing the loan approval prediction.

Feature selection: It is a crucial step in the machine learning pipeline. It helps in selecting the most important features that contribute to the prediction variable. This not only improves the performance of the model but also reduces overfitting and computational cost.

1. **SelectKBest**: This method selects the top k features based on the highest scores obtained from a specific statistical test (f_classif in this case).
   - score_func=f_classif: This function computes the ANOVA F-value for the provided features and target variable.
   - k=10: The number of top features to select.
2. **Selected Features**: The selected features based on the statistical test.The selected features are those that have the highest correlation with the target variable.

---

9. Conclusion

- The loan approval prediction project successfully analyzed the dataset and built five machine learning models to predict whether a loan application will be approved or rejected.
- The models evaluated include Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and Gradient Boosting Classifier.
- Each model's performance was assessed using various evaluation metrics, including accuracy, mean absolute error, mean squared error, root mean squared error, and R2 score.
- Based on the evaluation results, the Random Forest model demonstrated the highest accuracy of 97.66% in predicting loan approval outcomes.
- The feature importance analysis revealed that the CIBIL score is the most influential factor in predicting loan approval, followed by loan term, loan amount, and income.