

Project Documentation for Credit Card Approval Project

Credit Card Approval Project

Overview

This project focuses on analyzing and predicting credit card approval using machine learning techniques. By automating the approval process, financial institutions can reduce manual effort, eliminate bias and improve service delivery.

Business Value

- Helps reduce human error and bias in approval decisions
- Improves efficiency and customer experience
- Supports better targeting of credit worthy individual
- Seamless integration into FinTech solution

Dataset Files

credit_risk_dataset.csv: The raw dataset used for training and testing the models.

credit_risk.csv: The cleaned and preprocessed dataset.

model_accuracies.csv; model accuracy file saved from ml code

class_imbalance.csv: class imbalance file saved from ml code

report_after.csv: classification report file after handling class imbalance. Saved from ml code

confusion_matrix.csv: Confusion matrix report saved from ml code.

cross_validation_scores.csv: Cross Validation results saved from ml code.

Streamlit_credit_card_analysis-checkpoint: file for web application prototype

Credit_card_analysis-checkpoint: main file for ml model.

Project Workflow

Import Libraries:

- Import essential libraries for data manipulation, visualization, model building, and evaluation.

Load the Dataset:

- Load the dataset using pd.read_csv and display the first few rows to understand its structure.

Dataset Overview:

- Check the size and basic information of the dataset to identify data types and missing values.

Handle Missing Values:

- Replace missing values in 'person_emp_length' with 0.
- Replace missing values in 'loan_int_rate' with the median value.

Encoding Categorical Values:

- Convert categorical variables into numeric values using dictionaries for 'person_home_ownership', 'loan_intent', and 'cb_person_default_on_file'.

Correlation Matrix:

- Generate and visualize a correlation matrix to understand the relationships between features.

Save and Reload Transformed Dataset:

- Save the cleaned and transformed dataset to a CSV file and reload it for further processing.

Feature and Target Variables:

- Define the feature matrix X and target vector y.

Train-Test Split:

- Split the data into training and testing sets using `train_test_split`.

Data Standardization:

- Standardize the feature variables to have a mean of 0 and standard deviation of 1.

Initialize Classifiers:

- Initialize various classifiers including Logistic Regression, Decision Tree, K-Nearest Neighbors, XGBoost, and Random Forest.

Voting Classifier:

- Create a Voting Classifier that combines the predictions of the individual classifiers.

Model Training and Evaluation:

- Train the Voting Classifier and evaluate its performance.
- Identify and evaluate the best-performing individual classifier.

Handle Class Imbalance:

- Check for class imbalance in the target variable.
- Use SMOTE to generate synthetic samples for the minority class.

Train-Test Split for Resampled Data:

- Split the resampled data into training and testing sets.

Standardization and Model Training on Resampled Data:

- Standardize the resampled data and train the Voting Classifier.
- Evaluate the model's performance on the resampled data.

Cross-Validation:

- Perform cross-validation to assess the model's robustness and generalizability.

Results and insights.

- Logistic Regression performed best among the individual models
- Voting Classifier ensemble model selected the best model between individual models and improved performance generally.
- Data balancing improved model fairness.
- Majority of customers had a good credit behavior, implying a high potential for approval.

Tools and Technologies

- Python (pandas, scikit-learn, matplotlib, seaborn)
- SMOTE (Handle class imbalance)

- Jupyter Notebook
- Streamlit
- PowerBI (For additional story telling)

Future Improvements

- Deploy model as an API for real time predictions
- Integrate customer feedback for continuous learning

License

This project is an open source project.