ABDELMALEK ESSAADI UNIVERSITY
NATIONAL SCHOOL OF APPLIED SCIENCES AL HOCEIMA

Data mining on Microsoft azure cloud

Prepared by:

Loubna Boukayoua

Najma El boutaheri

Framed by:

Professor Hayat Routaib

OVERVIEW

Introduction

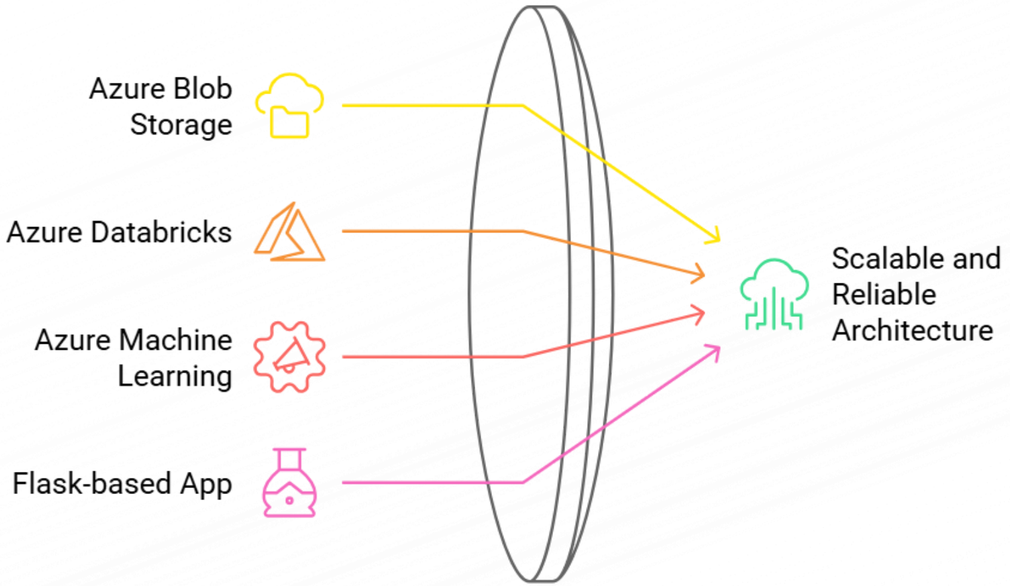
The financial sector faces challenges in managing loan portfolios, particularly in predicting borrower behaviors like delinquency and prepayment. Delinquency risks lenders by disrupting cash flow, while prepayment impacts revenue forecasts. Accurate predictions are vital for risk mitigation, financial planning, and operational efficiency.

PROJECT OVERVIEW

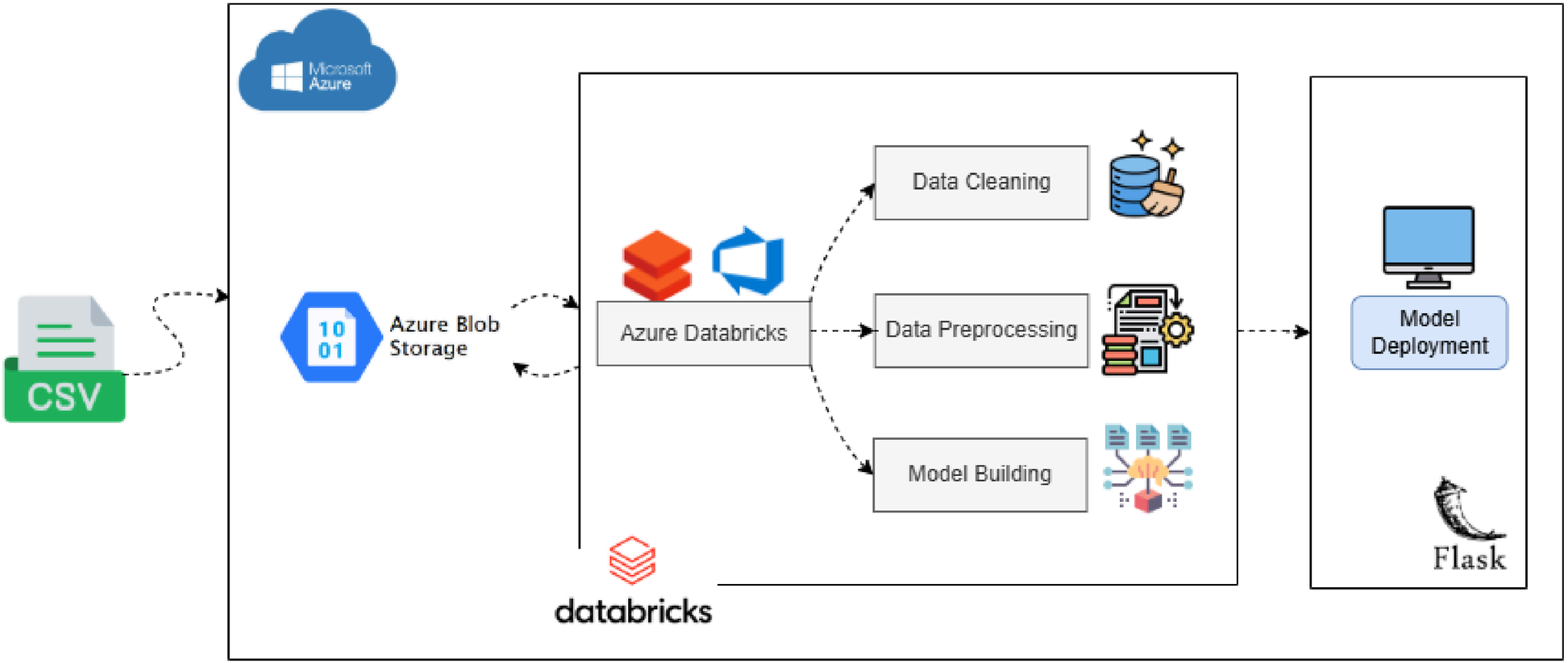
This project leverages data mining techniques to build predictive models for delinquency and prepayment behaviors using historical loan data. Key steps include data preprocessing, exploratory data analysis, feature engineering, and deploying machine learning models like logistic regression and gradient boosting.

PROJECT OVERVIEW

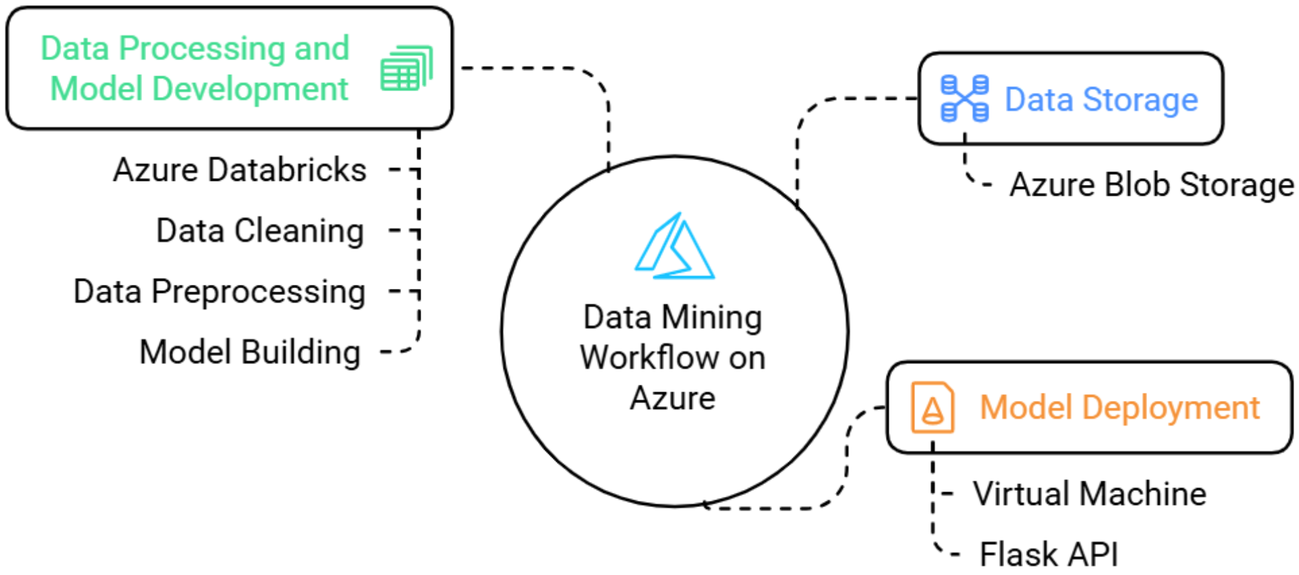
The architecture integrates Microsoft Azure services for scalability and reliability, utilizing Azure Blob Storage for data management, Azure Databricks for processing, and Azure Machine Learning for automation and monitoring. A Flask-based app enables real-time predictions, while Power BI dashboards provide actionable insights.



ARCHITECTURE



PROJECT COMPONENTS



PROJECT COMPONENTS

Azure Blob Storage is used to store raw data files for the machine learning workflow.

DATA STORAGE

Azure Databricks facilitates:

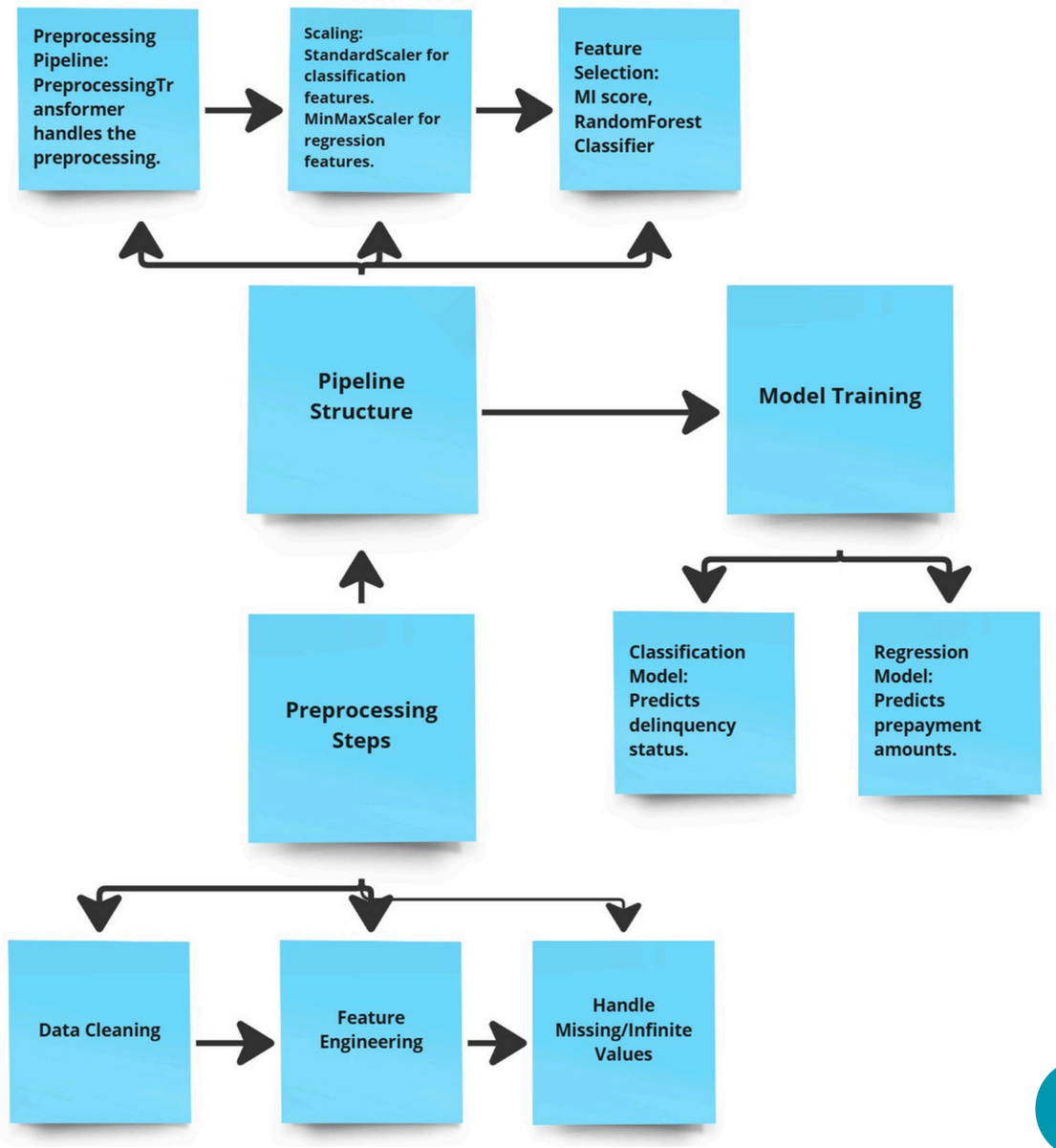
- Data cleaning.
- Preprocessing.
- Model Building: Training and validating machine learning models.

DATA PROCESSING AND MODEL DEVELOPMENT

Deploying the trained model to a live environment using a virtual machine provided by Microsoft Azure. Flask will be used for hosting the model as an API.

MODEL DEPLOYMENT

MACHINE LEARNING MODEL BUILDING STEPS



MACHINE LEARNING MODEL BUILDING STEPS

Data cleaning, feature engineering, and handling missing or infinite values prepare the data for modeling.

PREPROCESSING STEPS

A preprocessing pipeline, scaling, and feature selection are implemented for classification and regression tasks.

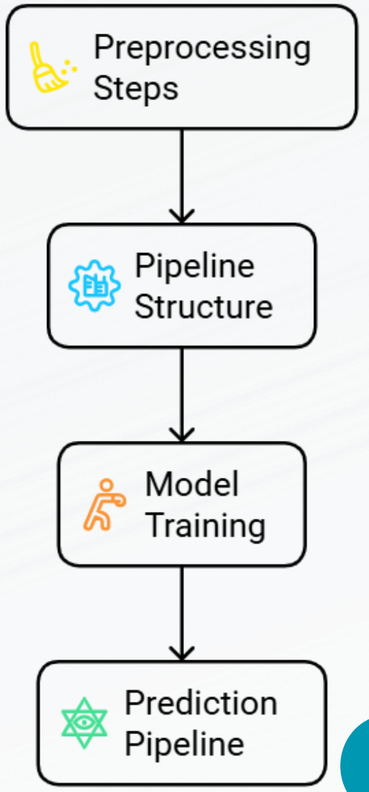
PIPELINE STRUCTURE

Classification models predict delinquency status, and regression models estimate prepayment amounts using relevant metrics.

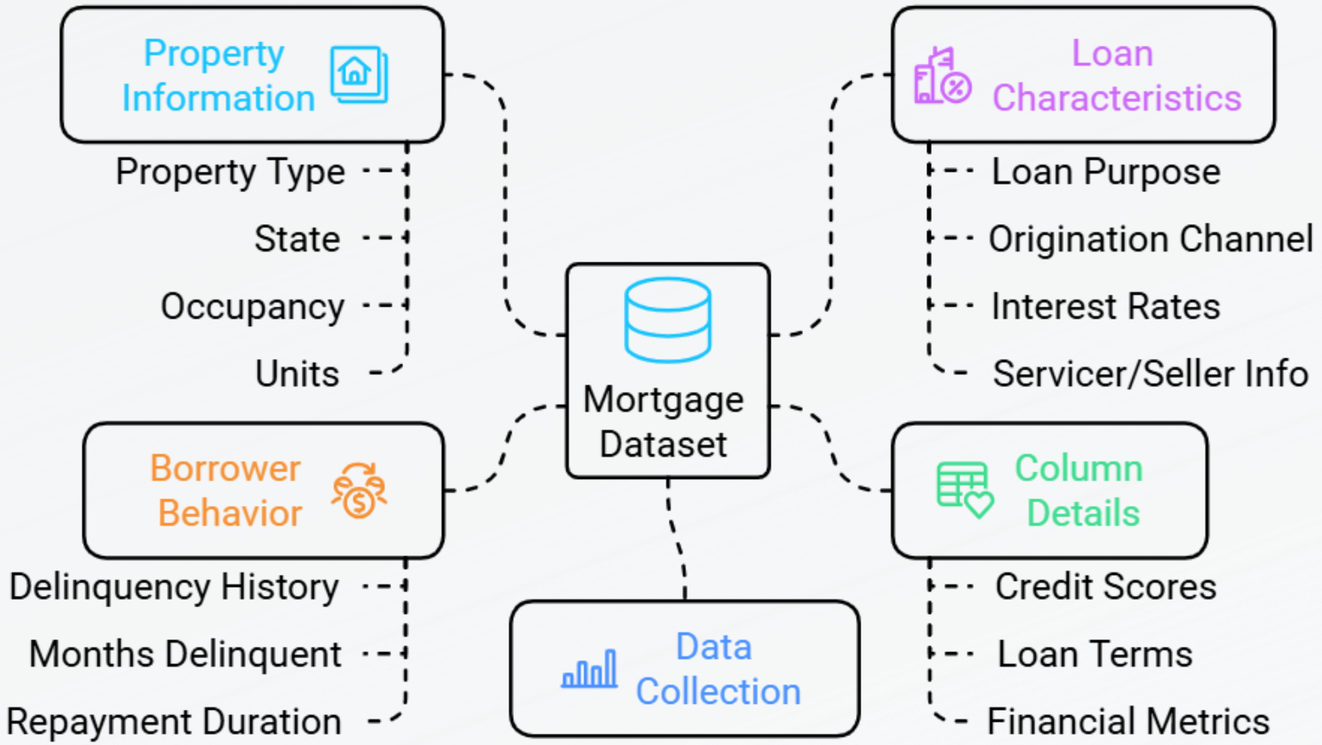
MODEL TRAINING

Classification predicts delinquency, and regression estimates prepayment for non-delinquent cases.

PREDICTION PIPELINE



ABOUT THE DATASET



ABOUT THE DATASET

The dataset was sourced from a GitHub repository and includes information on prepayment penalty mortgages (PPM).

DATA COLLECTION

Key columns include borrower credit scores, loan terms (e.g., FirstPaymentDate, MaturityDate), and financial metrics like DTI, LTV, and OrigUPB.

COLUMN DETAILS

Details about property type, state, occupancy, and units are included.

PROPERTY INFORMATION

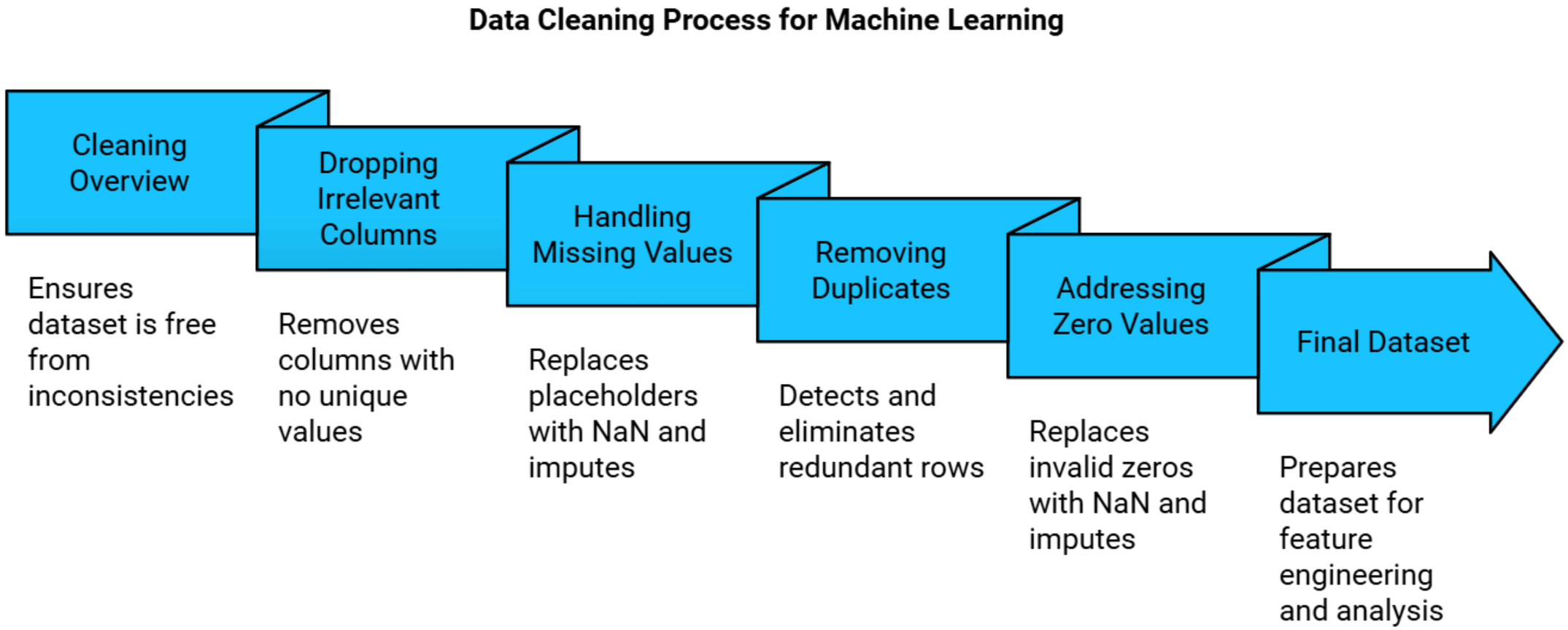
Data covers loan purpose, origination channel, interest rates, and servicer/seller information.

LOAN CHARACTERISTICS:

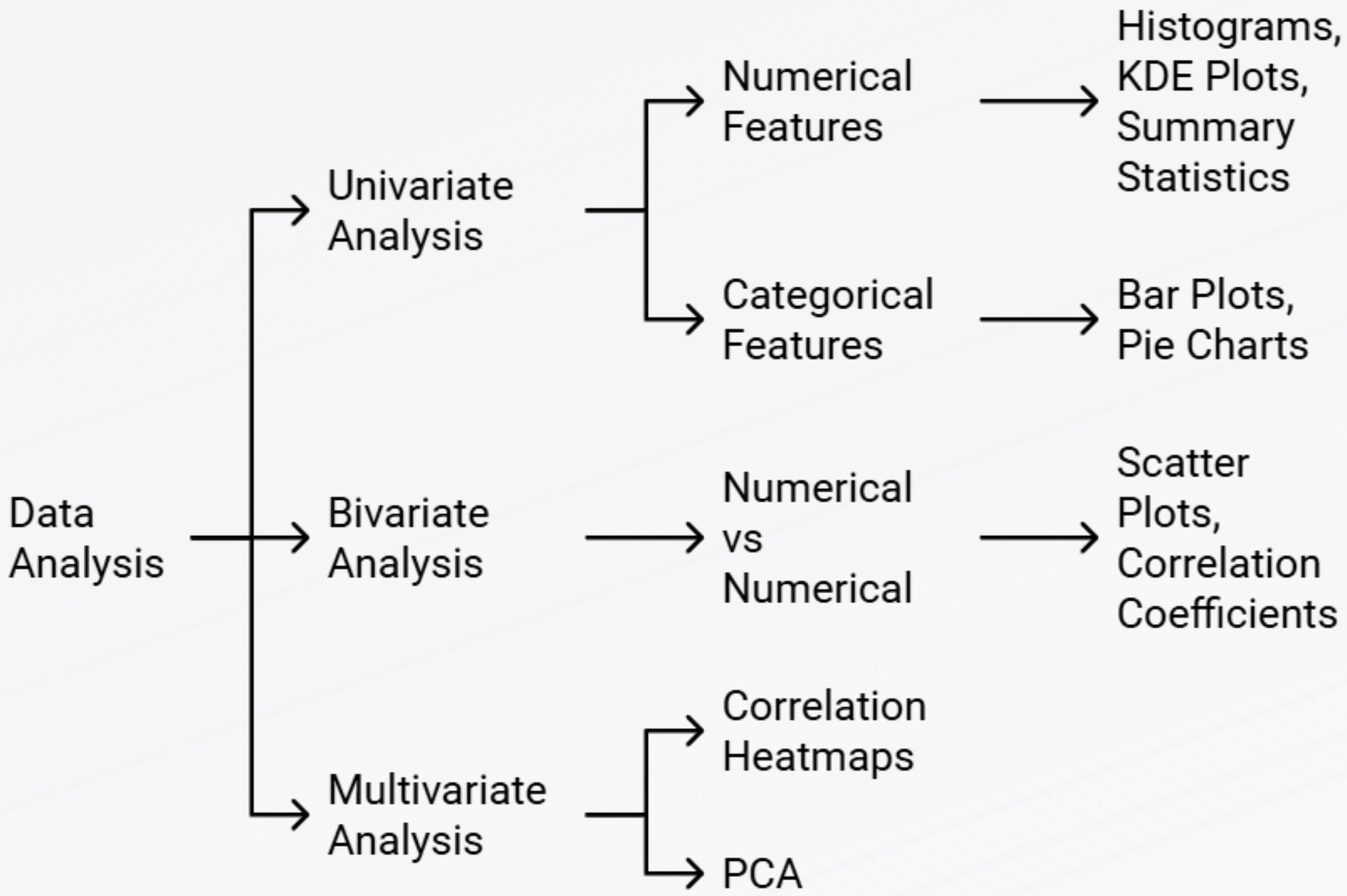
Metrics track delinquency history, months delinquent, and repayment duration.

BORROWER BEHAVIOR:

DATA PREPROCESSING



EXPLORATORY DATA ANALYSIS (EDA)



EXPLORATORY DATA ANALYSIS (EDA)

Univariate Analysis - Numerical Features

Distribution, skewness, and outliers in numerical features were analyzed using histograms, KDE plots, and summary statistics.

Univariate Analysis - Categorical Features

Frequency distributions of categorical variables were visualized with bar plots and pie charts to understand category proportions.

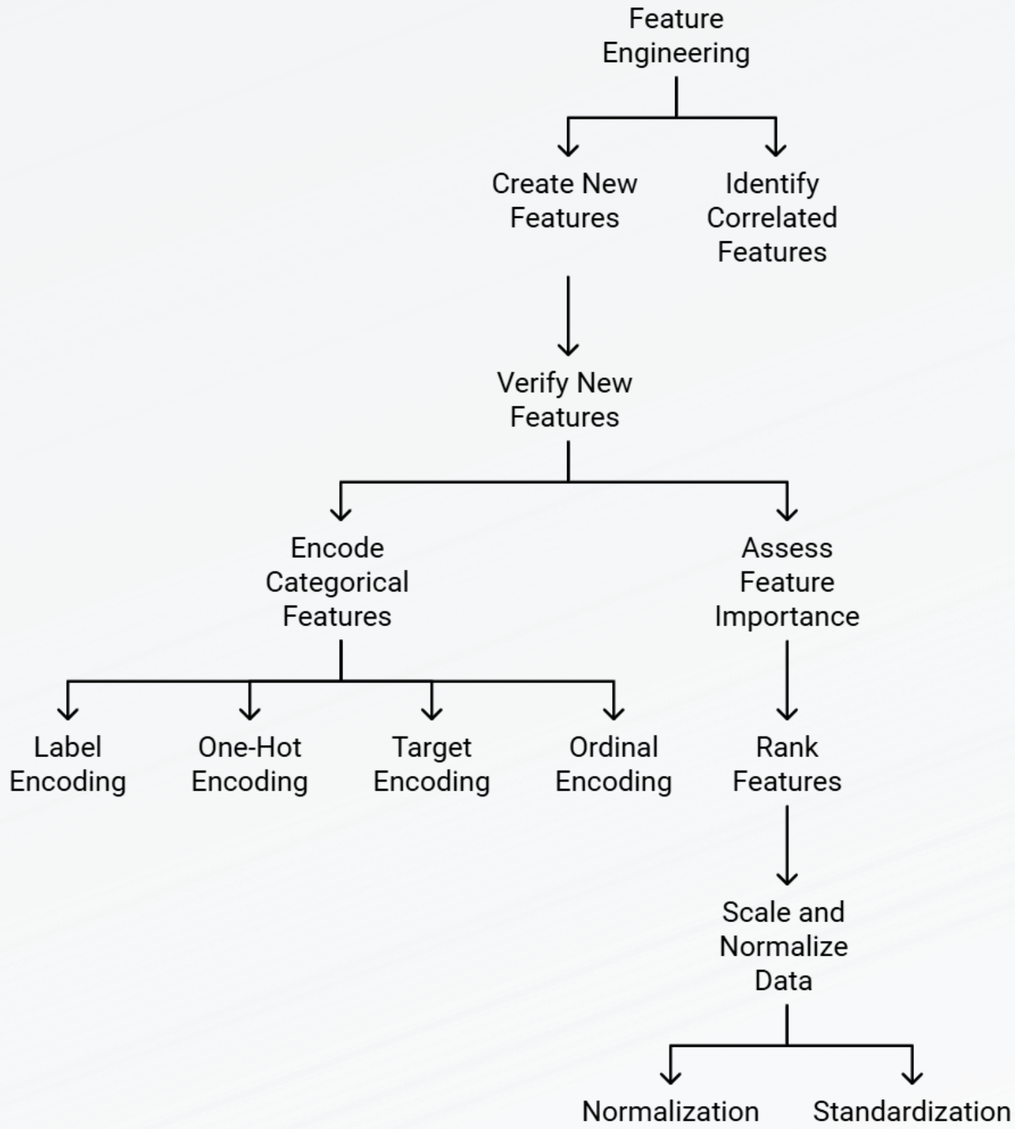
Bivariate Analysis - Numerical vs Numerical

Relationships between numerical features were explored using scatter plots and correlation coefficients to detect patterns or trends.

Multivariate Analysis

Explores interactions among multiple features using correlation heatmaps for relationships and PCA for dimensionality reduction.

PREPARING DATA FOR MODELING



DATA MODELING

Classification Models:

- Logistic Regression: Linear model predicting probabilities.
- Naive Bayes: Probabilistic model for categorical data.
- Decision Tree: Splits data into subsets for majority-vote predictions.
- Final Model: Random Forest for its robustness and handling imbalanced data.

Regression Models:

- Linear Regression: Models target-feature relationship linearly.
- Ridge Regression: Adds regularization to prevent overfitting.
- Final Model: Linear Regression for its simplicity and better RMSE

Pipeline for Classification and Regression

Pipeline Overview:

- Classification Stage: Random Forest predicts loan delinquency (EverDelinquent); results filter data for regression.
- Regression Stage: Linear Regression predicts prepayment for delinquent loans.

Data Splitting and Training:

Train/test split for classification first, followed by regression on filtered data.

Evaluation:

- Classification: Metrics like accuracy, precision, recall, and F1-score validate Random Forest.
- Regression: MAE, MSE, and R^2 evaluate Linear Regression reliability.

Algorithm	CV_Roc_Auc_Score	Training_accuracy	Testing_accuracy	Roc_Auc_score	f1_score
Logistic Regression	1.000000	1.000000	1.000000	1.000000	1.000000
Decision Tree	1.000000	1.000000	1.000000	1.000000	1.000000
GaussianNB	1.000000	0.988566	0.988425	1.000000	0.971297
BernoulliNB	0.845195	0.904684	0.904568	0.848416	0.717914

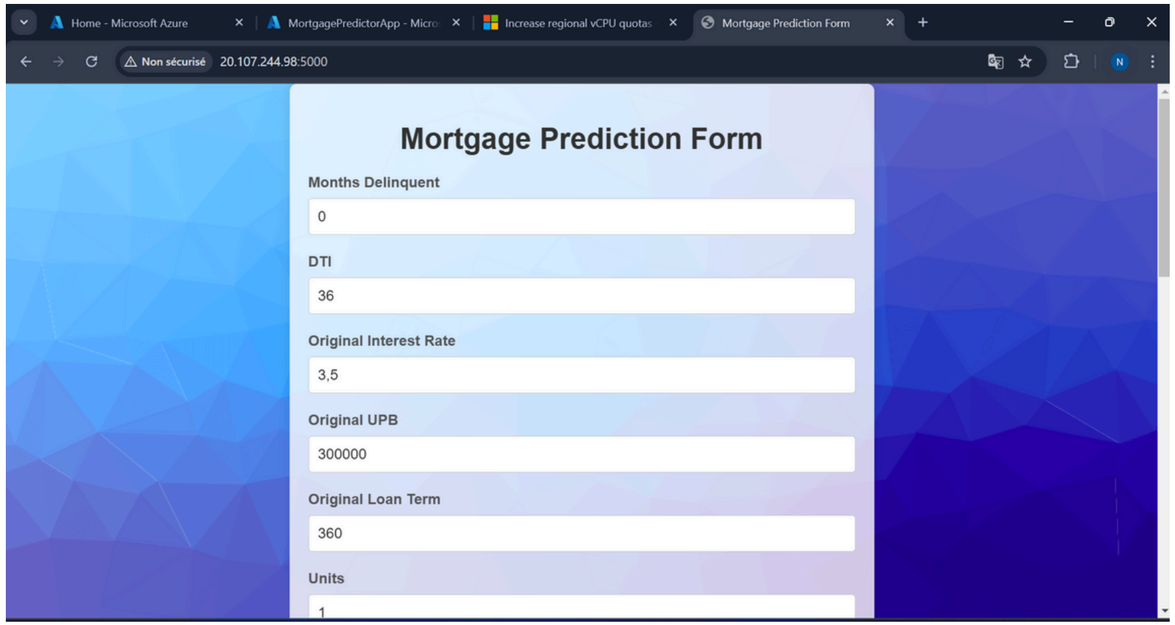
Serialization:

Pipeline saved with joblib for reusability and system integration.

Deployment

Azure VM Deployment:

- Flask app deployed on Microsoft Azure VM for remote access.
- VM ensures scalability, robust monitoring, and reliable performance.



Conclusion

This project developed a Flask-based machine learning app with classification and regression models, offering accurate predictions through a RESTful API. Future work will enhance model performance, scalability, deployment, and API security.

Thanks for your attention

References

1. **Scikit-learn Documentation. (n.d.). Retrieved from:** https://scikit-learn.org
1. **Microsoft Azure Documentation. (n.d.). Retrieved from** https://azure.microsoft.com
2. **Postman API Testing Tool. (n.d.). Retrieved from** https://www.postman.com
3. **Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.**