



AI Feature Mapping Validator

Compare functionalities between a Whitepaper and its Codebase

 Upload Whitepaper File



Drag and drop file here


Limit 200MB per file • TXT, MD, PDF

Browse files



Load Prediction Whitepaper.pdf 0.9MB



 Upload Code File



Drag and drop file here

Limit 200MB per file • PY, TXT, IPYNB

Browse files



loan-approval-prediction_v2.ipynb 419.8KB



Click to Process Files

Getting relevant chunks from vector database for white paper

Refining extracted chunks from vector database for white paper

Getting relevant chunks from vector database for model

Refining extracted chunks from vector database for model

Summarizing report findings

Objective of the White Paper

The white paper outlines a comprehensive machine learning-based loan approval system designed to enhance decision-making in financial services. It emphasizes the use of predictive modeling, real-time

processing, explainable AI, fairness, compliance, privacy, and human oversight to ensure ethical and efficient loan processing.

Report

1. Model Overview

- **Model Architecture or Type:**
 - **White Paper:** Specifies the use of Logistic Regression for its transparency and explainability.
 - **Code:** Implements XGBoost Classifier, which is a significant deviation from the white paper.
- **Feature Selection and Feature Engineering:**
 - **White Paper:** Mentions the use of categorical, numerical, and binary features but does not specify them.
 - **Code:** Uses features like `ApplicantIncome` , `CoapplicantIncome` , `LoanAmount` , `Loan_Amount_Term` , `Credit_History` , and `Property_Area` .
- **Data Splitting Strategy:**
 - **White Paper:** Recommends a 70% training, 15% validation, and 15% testing split.
 - **Code:** Uses `train_test_split` and `kFold` but does not specify exact split ratios.
- **Baseline Metrics:**
 - **White Paper:** Emphasizes high precision, recall, F1-score, and PR-AUC.
 - **Code:** Uses accuracy, recall, precision, and F1 score for baseline evaluation.
- **Fallback Mechanisms:**
 - **White Paper:** Suggests human oversight for high-impact or low-confidence decisions.
 - **Code:** Does not mention any fallback mechanisms.
- **Data Import and Preprocessing Steps:**
 - **White Paper:** Recommends imputation for missing values and data type consistency.
 - **Code:** Handles missing values in `Dependents` and identifies missing values in `Self_Employed` .
- **Handling of Imbalanced Data:**
 - **White Paper:** Mentions `class_weight='balanced'` and SMOTE for handling class imbalance.
 - **Code:** Does not address class imbalance.

2. Validation Metrics

- **White Paper:** Highlights PR-AUC, precision, recall, and F1-score.
- **Code:** Implements accuracy, recall, precision, and F1 score but does not mention PR-AUC.
- **Discrepancies:** The code does not align with the white paper's emphasis on PR-AUC and balanced accuracy.

3. Hyperparameter Configuration

- **White Paper:** Lists hyperparameters for Logistic Regression, including penalty, C, solver, and max_iter.
- **Code:** Uses hyperparameters for XGBoost, such as `eta` , `min_child_weight` , `gamma` , `subsample` , and `colsample_bytree` .
- **Mismatches:** The hyperparameters differ due to the change in model from Logistic Regression to XGBoost.

4. Final Summary

The code implementation does not align with the current white paper. The model type, hyperparameters, and some validation metrics differ significantly. Therefore, the conclusion is:

"White paper is not aligned with the code. Please update the white paper accordingly."