AI for
Credit Risk Assessment

NTUC CapStone Project

VLC-SCAI012-24-0652

Contents

Introduction and Project Overview

- Credit risk assessment for personal loans evaluates an individual's creditworthiness, analyzing income, credit history, debt, and employment to predict repayment likelihood and minimize lender exposure to default.
- Effective credit risk assessment safeguards bank's capital, reputation, and customer trust, minimizing loan defaults and financial losses.
- This capstone project aims to design and implement an AI ML Credit Risk Assessment model, predicting loan application outcomes: approval or rejection.

Topic Selection Rationale

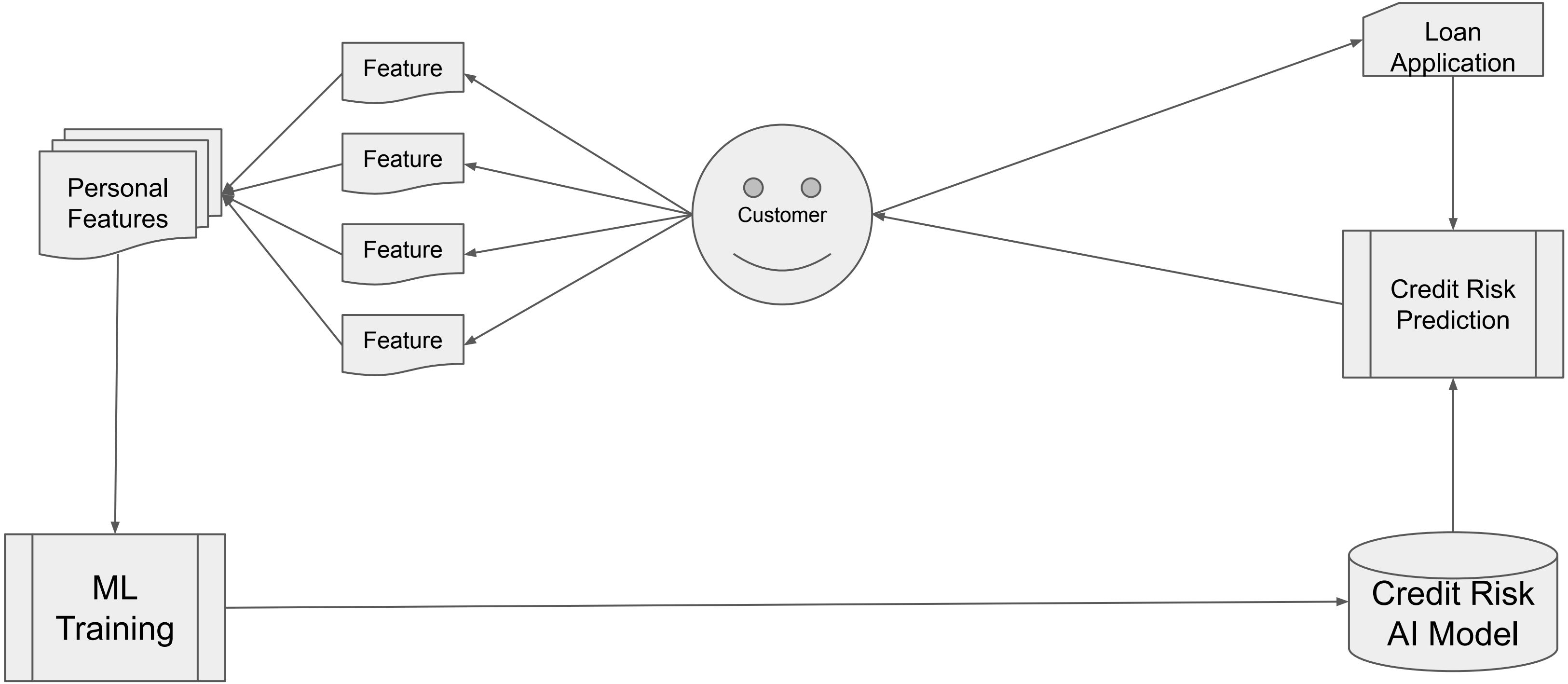
Although the current credit risk system is sophisticated and well-established, it has limitations:

- Periodic reassessments of borrowers' repayment capacity are costly and labor-intensive.
- Strict qualification criteria exclude potentially low-risk individuals from accessing loans.
- Some approved loans may exceed borrowers' actual needs, introducing unintended risks and potential misuse of funds.

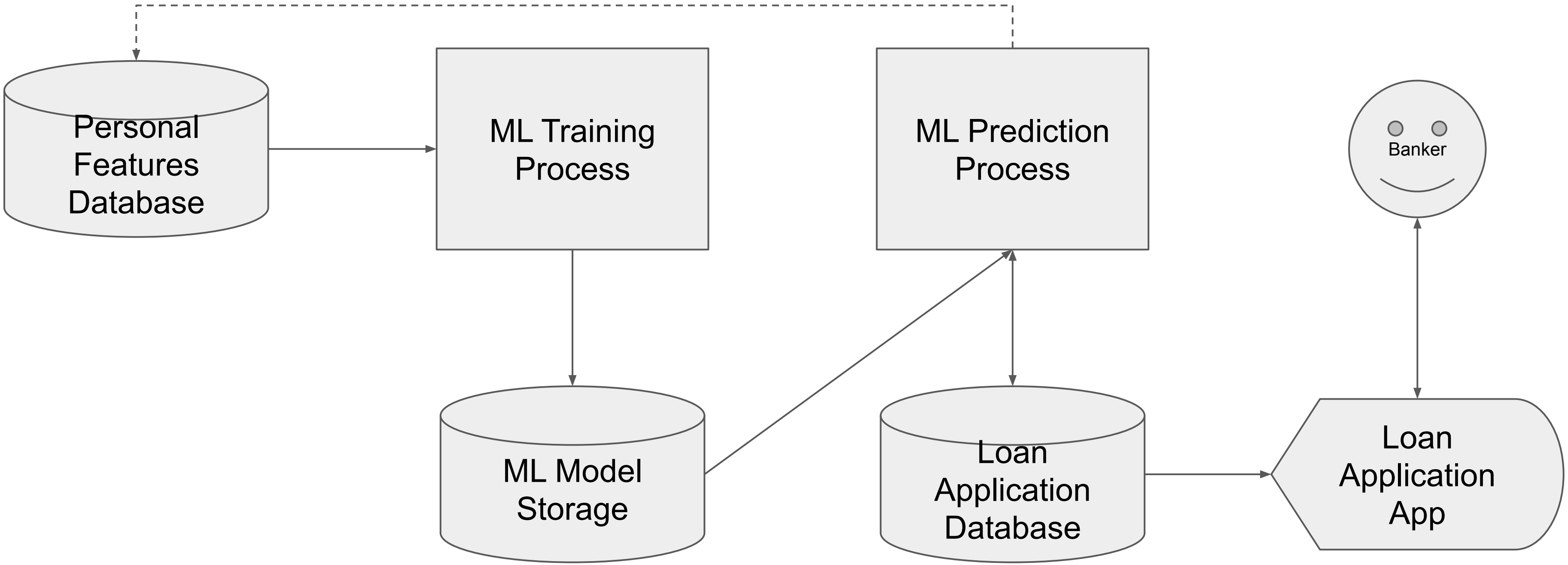
Exploring AI and Data driven approaches, leveraging customer information from various channels, can optimize credit risk assessment. This enhancement enables

- Expedited evaluation processes
- Reduced risk exposure
- Increased cost efficiency
- Broadened access to affordable banking services.

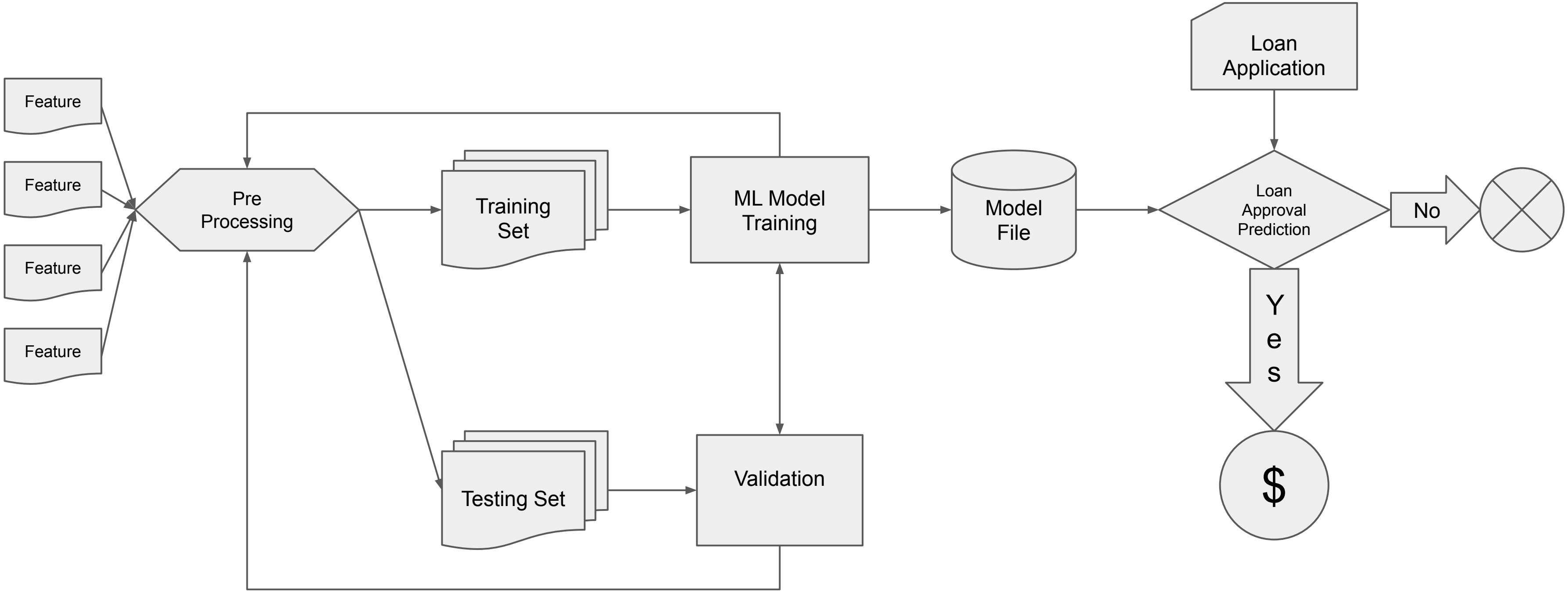
Data Processing Flow



System Architecture



Training Flow Design



Coding and Implementation Highlights

- PreProcessing
  - **One-Hot** vs Factorizing
  - Bin + One-Hot or **Not**
  - Fields may **not** affect the results.
  - **Based Fields** and Calculated Field
- Training
  - Supervised Training or **Unsupervised** Training, Seed assignment
  - GaussianMixture vs **KMeans**
  - Metrics Validation
  - Hyper-Parameter
- Model Evaluation
- Prediction

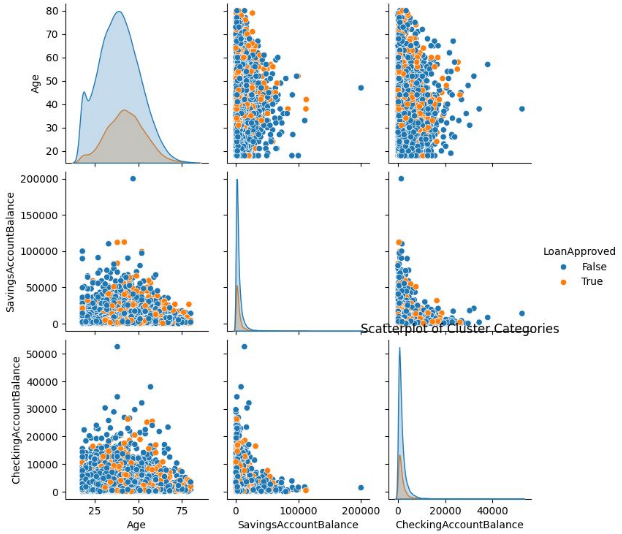
Features Engineering

A Pair Plot analysis examined the impact of Feature Fields on model training,

e.g. SavingsAccountBalance and CheckingAccountBalance.

No significant correlations or clusters were found, and subsequent testing confirmed that excluding these features has no noticeable impact on model accuracy.

Description	Score	Accuracy	Weighted
Include	0.49930	0.68585	0.70025
Exclude	0.49935	0.68590	0.70013



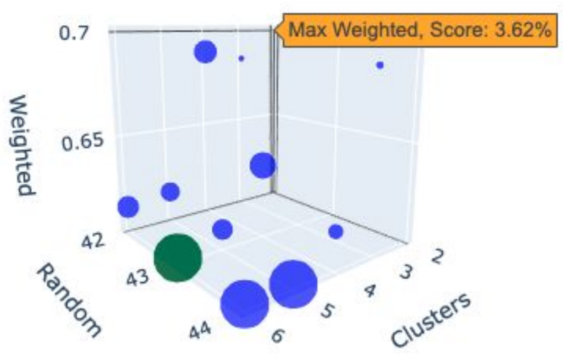
Testing and Validation Results

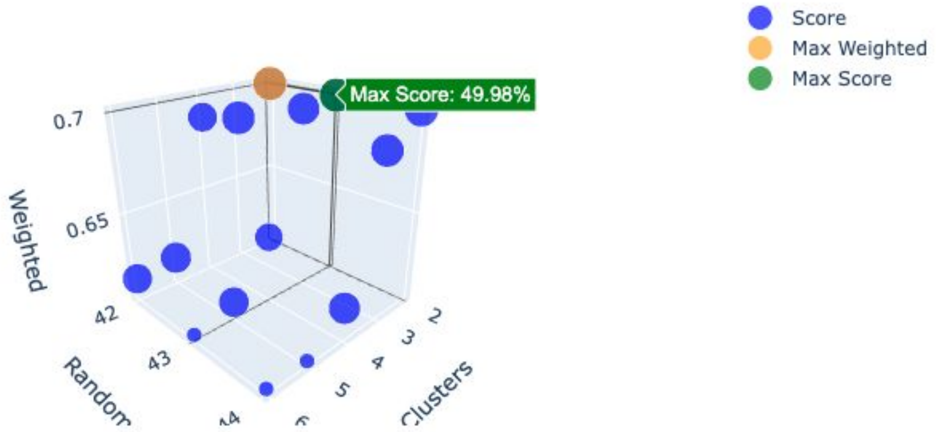
- Metrics Validation and Model Evaluation are two separate processes.
- **Metrics Validation**: Calculates training **Score** based on feature fields to compare models.
- **Model Evaluation**: Assesses training **Accuracy** by comparing predicted vs. manual labels.
- Both processes involve iterative steps to enhance score and accuracy.

Metric Validation

The goal is to evaluate and select the most effective metric function that measures training performance for model optimization.

- 15 testing, **Cluster**=2-6(x axis), **Random**=42-44 (y axis), **Weighted** (z axis) and **Score**(size of dot).
- **Balance Ratio** (Left) = abs(True - False) / (True + False). it can be far from Score in some cases
- **Gini Coefficient** (Right) = 1 - (True / (True + False))**2 - (False / (True + False))**2. Best
- **SilHouett**. it close to Gini Coefficient. But it is slow





Model Evaluation

The goal is to evaluate and select the optimal function to maximize model accuracy and predictive performance.

- Basic Accuracy
  - Simple
  - Not Customizable
- Weighted Accuracy
  - Customizable to fit individual preferences.
  - Aligning with Metric Validation Score ensures more consistent training outcomes.

```
# Custom weights
W_TT = 1.0   # TT (True Positives), Actual Value is same as Predicted Value, both are True
W_FF = 0.7   # FF (True Negatives), Actual Value is same as Predicted Value, both are False
W_TF = 0.5   # TF (False Negatives), Actual Value is False, Predicted Value is True
W_FT = 0.3   # FT (False Positives), Actual Value is True, Predicted Value is False

TP = la_tf.loc[(True,True)]   # True Positive (TT)
TN = la_tf.loc[(False,False)] # True Negative (FF)
FN = la_tf.loc[(True,False)]  # False Negative (TF)
FP = la_tf.loc[(False,True)]  # False Positive (FT)
total = TP + TN + FN + FP

accuracy = (TP + TN)/total
weighted = ((W_TT * TP) + (W_FF * TN) + (W_TF * FN) + (W_FT * FP)) / total
```

No	Clusters	Random	Score	AT	AF	Accuracy	Weighted	TT	FF	TF	FT
0	2	42	0.49935	10362	9638	0.68590	0.70013	4430	9288	5932	350
1	2	43	0.49980	10198	9802	0.69030	0.70006	4392	9414	5806	388
2	2	44	0.49952	10309	9691	0.68725	0.70008	4417	9328	5892	363
3	3	42	0.49555	9057	10943	0.68845	0.68497	3803	9966	5254	977
4	3	43	0.47990	7995	12005	0.74555	0.69739	3843	11068	4152	937
5	3	44	0.49249	8774	11226	0.70090	0.68703	3786	10232	4988	994
6	4	42	0.43692	6448	13552	0.77130	0.68964	3327	12099	3121	1453
7	4	43	0.41673	5919	14081	0.67065	0.63773	2056	11357	3863	2724
8	4	44	0.47255	7657	12343	0.58175	0.61945	2036	9599	5621	2744
9	5	42	0.45649	7050	12950	0.59320	0.61702	1847	10017	5203	2933
10	5	43	0.44854	6792	13208	0.58760	0.61127	1662	10090	5130	3118
11	5	44	0.22561	2592	17408	0.68820	0.60404	568	13196	2024	4212
12	6	42	0.44274	6616	13384	0.59900	0.61420	1688	10292	4928	3092
13	6	43	0.22286	2555	17445	0.69075	0.60473	575	13240	1980	4205
14	6	44	0.22420	2573	17427	0.68945	0.60437	571	13218	2002	4209

Hyper Parameters Tuning

- 15 round of testing:
- **Cluster** is from 2 to 6
- **Random** is from 42 to 44
- Calculate **Score** and **Weighted**
- Best Result has **Highest Score** and **third Weighted** (very small margin comparing with first Weighted)

No	Clusters	Random	Score	AT	AF	Accuracy	Weighted	TT	FF	TF	FT
0	2	42	0.49935	10362	9638	0.68590	0.70013	4430	9288	5932	350
1	2	43	0.49980	10198	9802	0.69030	0.70006	4392	9414	5806	388
2	2	44	0.49952	10309	9691	0.68725	0.70008	4417	9328	5892	363
3	3	42	0.49555	9057	10943	0.68845	0.68497	3803	9966	5254	977
4	3	43	0.47990	7995	12005	0.74555	0.69739	3843	11068	4152	937
5	3	44	0.49249	8774	11226	0.70090	0.68703	3786	10232	4988	994
6	4	42	0.43692	6448	13552	0.77130	0.68964	3327	12099	3121	1453
7	4	43	0.41673	5919	14081	0.67065	0.63773	2056	11357	3863	2724
8	4	44	0.47255	7657	12343	0.58175	0.61945	2036	9599	5621	2744
9	5	42	0.45649	7050	12950	0.59320	0.61702	1847	10017	5203	2933
10	5	43	0.44854	6792	13208	0.58760	0.61127	1662	10090	5130	3118
11	5	44	0.22561	2592	17408	0.68820	0.60404	568	13196	2024	4212
12	6	42	0.44274	6616	13384	0.59900	0.61420	1688	10292	4928	3092
13	6	43	0.22286	2555	17445	0.69075	0.60473	575	13240	1980	4205
14	6	44	0.22420	2573	17427	0.68945	0.60437	571	13218	2002	4209

Journey to Reach the Best Result

No	Clusters	Random	Score	AT	AF	Accuracy	Weighted	TT	FF	TF	FT	Description	Action
1	2	42	0.49965	10263	9737	0.68585	0.69887	4380	9337	5883	400	Factorize Encoding, FeatureFieldOF	
2	2	42	0.49930	10373	9627	0.68585	0.70025	4435	9282	5938	345	One-Hot Encoding, FeatureFieldOH	Choose it. Weighted is better.
3	2	42	0.49930	10373	9627	0.68585	0.70025	4435	9282	5938	345	No Encoding, FeatureFieldN	Choose it. Both Score and Weighted are better.
4	2	42	0.48020	8010	11990	0.59920	0.63172	2387	9597	5623	2393	Bin + One-Hot Encoding, FeatureFieldNH	
5	2	42	0.49930	10373	9627	0.68585	0.70025	4435	9282	5938	345	A/C Balance Fields, Include	
6	2	42	0.49935	10362	9638	0.68590	0.70013	4430	9288	5932	350	Not Include	Choose it. Minor Impact, treat it as noise.
7	2	42	0.49935	10362	9638	0.68590	0.70013	4430	9288	5932	350	Keep Both Base and Calculated Field	Choose it. Better Weighted to include all these fields
8	2	42	0.49856	10536	9464	0.63740	0.68048	4032	8716	6504	748	Remove Base Fields to calc DebtToIncomeRatio	
9	2	42	0.49982	10188	9812	0.65710	0.68500	4055	9087	6133	725	Remove DebtToIncomeRatio	
10	2	42	0.49935	10362	9638	0.68590	0.70013	4430	9288	5932	350	Use KMeans ++ Model	Choose it. Weighted is better.
11	2	42	0.16634	1831	18169	0.69975	0.59973	303	13692	1528	4477	Use GaussianMixture Model	Best Record has low Accuracy.
12	2	42	0.49935	10362	9638	0.68590	0.70013	4430	9288	5932	350	Use Gini Coefficient Evaluation	Choose it. Best Record is same, training speed is faster.
13	2	42	0.03620	10362	9638	0.68590	0.70013	4430	9288	5932	350	Use Balance Ratio Evaluation	Best Record based on Score is too far from Weighted.
14	2	42	0.02453	10362	9638	0.68590	0.70013	4430	9288	5932	350	Use SilHouette Evaluation	Best Record is same, but speed is slower.
15	2	42	0.49935	10362	9638	0.68590	0.70013	4430	9288	5932	350	Use 2 Clusters and Random State 42	
16	2	43	0.49980	10198	9802	0.69030	0.70006	4392	9414	5806	388	Use 2 Clusters and Random State 43	Best Record

Challenges Faced and Problem-Solving Approach

- Understanding of Dataset
  - Project Proposal > Datasets and its sample codes
  - Project Proposal > Dataset Study
  - Found a problem in Dataset DebtToIncomeRatio, TotalDebtToIncomeRatio
- How to tune the performance of model effectively
  - Structure the code block, make it reusable and can be run repeatable.
  - Tune one place at one time.
  - If multiple places, pack it in loop and analysis the results.
- Evaluating the Evaluation Method
  - Model Choosing
  - Metrics Validation
  - Model Evaluation

Conclusion and Future Work

This project develops an AI-powered Credit Risk Assessment solution using Machine Learning to approve/reject loan applications. Leveraging user-specific features, it offers a promising alternative to traditional Credit and Risk scoring systems. With more data and refinement, accuracy will improve, potentially outperforming existing methods.

- Future Work
  - To expand feature set to optimize model performance.
  - To predict loan interest rates tailored to individual creditworthiness.

Q&A and Interaction

References