



# Loan Approval

**ME 781 Course Project**

**Prof. Asim Tiwari**

Project Team: Group I

Shiv Modi [19DI00011]

Jatin Choudhary [19DI70010]

Abhijeet Singh [215120039]

Pratik Khokle [19DI10012]



# Agenda

Introduction

Project Objective

Problem Definition

Technology Landscape Assessment

Project Outcome







# Introduction

There has been a growing demand for small loans from financial companies in recent years. This is because more and more people are finding themselves needing extra money to cover unexpected expenses or make ends meet. While banks and other traditional lenders have typically been the go-to source for loans, they are often unwilling to provide loans for smaller amounts. This is where financial companies specializing in small loans can come in handy.

# Project Objective

# Objective

- The current manual loan approving process is time-consuming and requires skilled staff, which may be in short supply. A well-functioning loan approval process is critical for any financial institution. A lengthy or complicated process can dissuade potential borrowers from applying for a loan, resulting in lost business opportunities.
- The primary problems to be addressed are those related to the manual loan application process: skills shortages on the loan team, longer loan approval times, and increased potential operational and control risk. Other issues that may need to be considered include gender and marital status factors potentially influencing the approval decision, the number of dependents an applicant has, and their education and income levels.
- In order to address these problems, it is essential to streamline the loan application process so that it is quicker and easier for both applicants and loan officers. It is also essential to consider other factors influencing the approval decision. Division of the high-risk and lower-risk groups is one of the main factors to be considered.



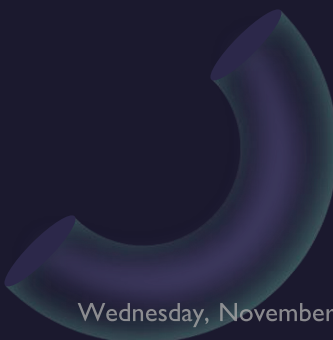
# Problem Definition

# Problem Definition



## WHAT?

An automated system that makes getting approved for a loan easy. The system checks your credit score and income to see if you qualify for a loan. If you do, you can get approved for a loan in as little as minutes.



## WHO?

Several financial companies provide small loans to consumers in need. Some of these companies include Cashnet USA, Prosper, etc. Each of these companies has different loan terms and requirements.

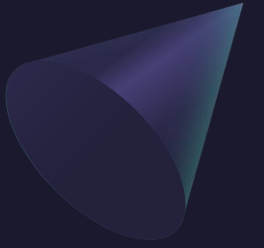
## WHY?

If an automated system were implemented to streamline the loan application process, it would positively impact both applicants and loan officers. The process would be quicker and easier for both groups, and the potential for errors would be significantly reduced. This would ultimately lead to more successful loan approvals and a better overall experience for everyone involved. The risk of missing critical information or making errors would be significantly reduced.

# Technology Landscape Assessment



# Technology Landscape Assessment



## PUBLISHED LITERATURE

Morduch, J., & Schneider, R. (2017). *The financial diaries: How American families cope in a world of uncertainty*. Princeton University Press.

Dinh, T. H. T., & Kleimeier, S. (2007). A credit scoring model for Vietnam's retail banking market. *International Review of Financial Analysis*, 16(5), 471–495.

Berry, L. L., & Parasuraman, A. (1997). Listening to the customer—the concept of a service-quality information system. *MIT Sloan Management Review*, 38(3), 65.

Farquhar, J., & Meidan, A. (2017). *Marketing financial services*. Bloomsbury Publishing.

Freedman, S., & Jin, G. Z. (2008). Do social networks solve information problems for peer-to-peer lending? *Evidence from Prosper. com*.

AREAS, B. (2018). Financial analysis. *Growth*, 30, 10.

## OPEN LIBRARIES

Pandas, Numpy, Matplotlib, Image (from Ipython), Seaborn, IO, Sklearn, Google Colab.

# Project Outcomes

# Interface Screenshots



## STEP 1

- Give the input of the required details as directed in the interface

```
In [*]: i1 = int(input("Applicant Income = "))
```

Applicant Income =

```
In [*]: i5 = int(input("Property Area (Enter 2 if 'Urban', 1 if 'Semi-Urban', 0 if 'Rural') = "))
```

Property Area (Enter 2 if 'Urban', 1 if 'Semi-Urban', 0 if 'Rural') =

## STEP 2

- After filling the all the details, Interface looks like this

```
In [31]: i8 = int(input("Number of Dependents (Enter 0 if 0 Dependents, 1 if 1 Dependent, 2 if 2 Dependents, 3 if 3 or more Dependents) = "))
```

Number of Dependents (Enter 0 if 0 Dependents, 1 if 1 Dependent, 2 if 2 Dependents, 3 if 3 or more Dependents) =

```
In [32]: i9 = int(input("Wether you are self Employed or not? (Enter 1 if 'Yes', 0 if 'No')"))
```

Wether you are self Employed or not? (Enter 1 if 'Yes', 0 if 'No') =

```
In [33]: i10 = int(input("Education (Enter 1 if 'Graduate', 0 otherwise) = "))
```

Education (Enter 1 if 'Graduate', 0 otherwise) =

```
In [34]: i11 = int(input("Credit History (Enter 1 if no pending loans, 0 otherwise) = "))
```

Credit History (Enter 1 if no pending loans, 0 otherwise) =

```
In [24]: i1 = int(input("Applicant Income = "))
```

Applicant Income = 2583

```
In [25]: i2 = int(input("Co-Applicant Income = "))
```

Co-Applicant Income = 2358

```
In [26]: i3 = int(input("Loan Amount = "))
```

Loan Amount = 120

```
In [27]: i4 = int(input("Loan Term = "))
```

Loan Term = 360

```
In [28]: i5 = int(input("Property Area (Enter 2 if 'Urban', 1 if 'Semi-Urban', 0 if 'Rural') = "))
```

Property Area (Enter 2 if 'Urban', 1 if 'Semi-Urban', 0 if 'Rural') = 2

```
In [29]: i6 = int(input("Married (Enter 1 if 'Yes', 0 if 'No') = "))
```

Married (Enter 1 if 'Yes', 0 if 'No') = 1

```
In [30]: i7 = int(input("Gender (Enter 1 if 'Male', 0 if 'Female') = "))
```

Gender (Enter 1 if 'Male', 0 if 'Female') = 1

## STEP 3

- The output of the interface looks like this

```
In [38]: result = grid.predict(df_test)
```

```
if result[0]==1:
```

display(Image(url= "Yes.jpg", width=400, height=400))

```
else:
```

display(Image(url= "No.jfif", width=400, height=400))



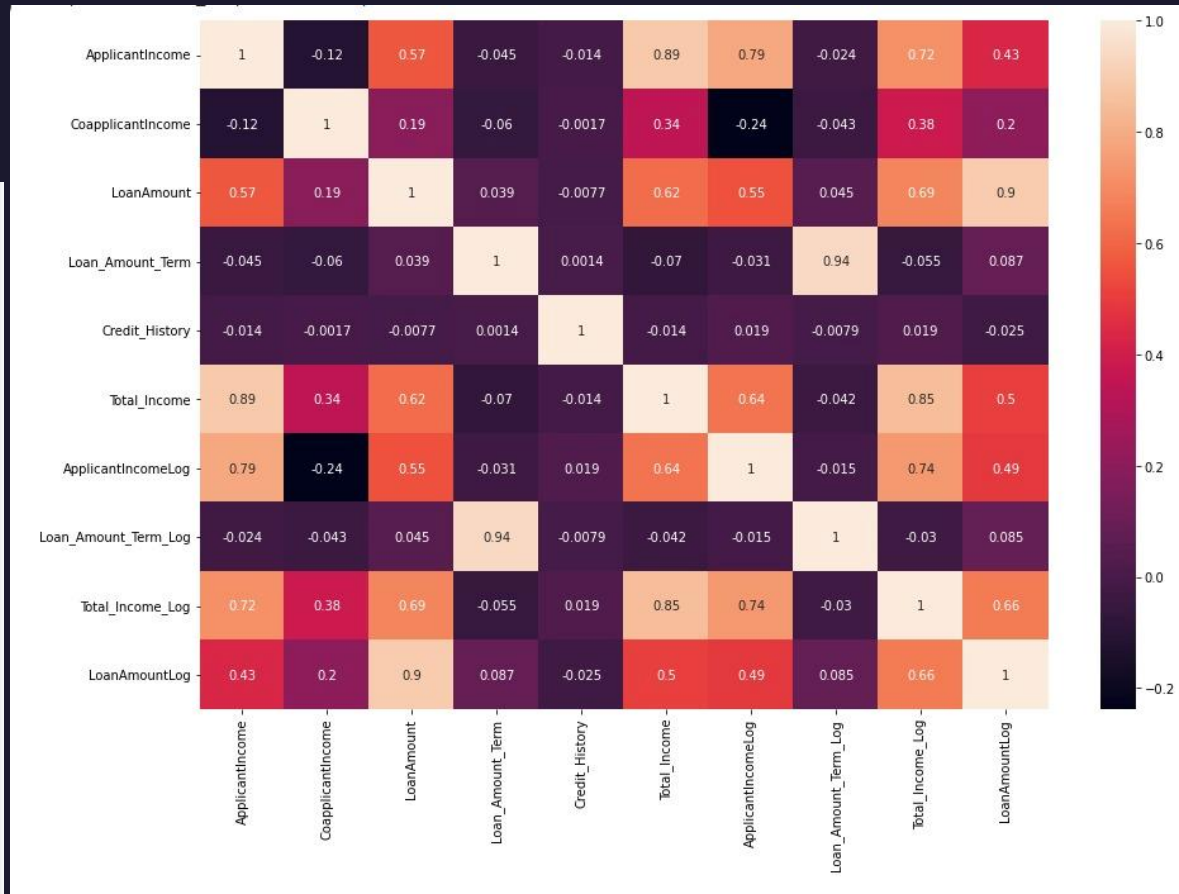
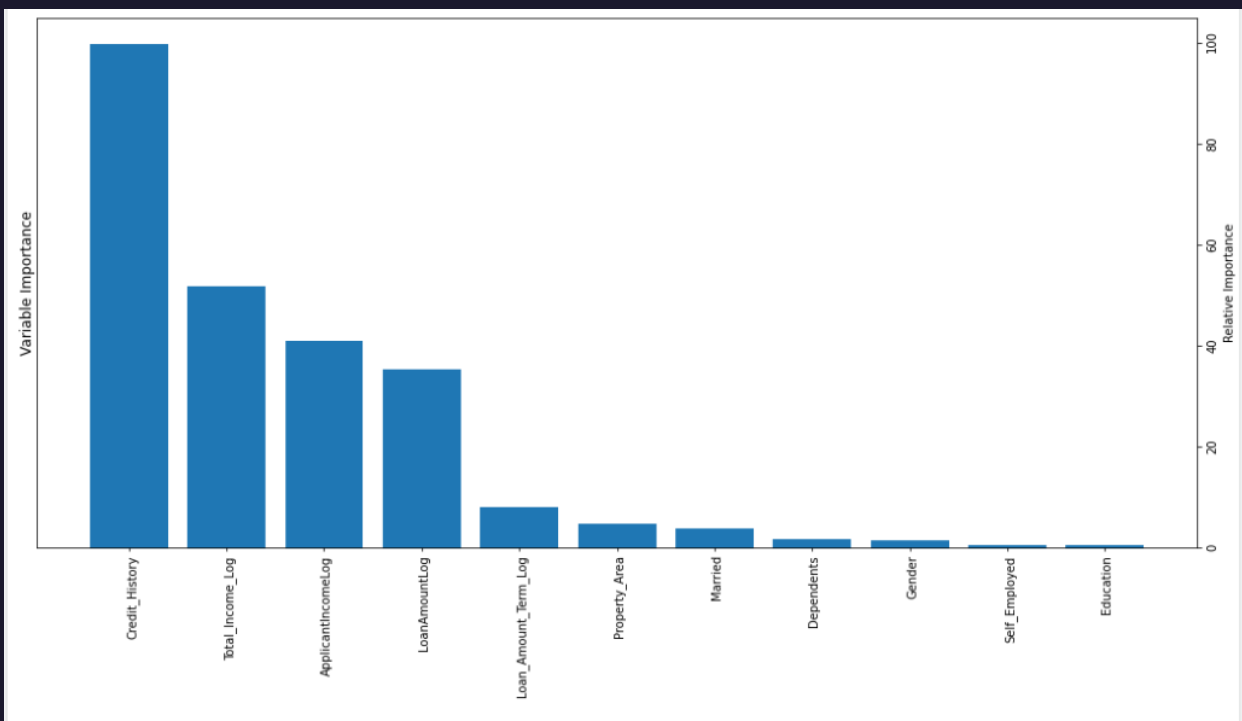
# Accuracy

We used a Multi-Model approach, in order to get the best model. Considering Binary classification problem Logistic Regression was our first go to model as it is easy to implement and fast as well. Since we need to decide on whether to provide a loan or not, our next models were Decision Tree and Random Forest. Also, our data was imbalanced and had missing values as well and so we thought to use Gradient Boosting.

	Training Accuracy Score	Testing Accuracy Score	Cross-Validation
Logistic Regression	81.47%	78.86%	80.95%
Decision Tree	100%	69.11%	71.50%
Random Forest	100%	76.42%	77.85%
Gradient Boosting	91.24%	94.30%	-

As we can see, the **Gradient Boosting** model has the highest accuracy and thus is the best model among the four.

# Covariance Matrix





# Additional Links

Conceptual Design:

<https://docs.google.com/document/d/1ooHrogneyQ7K2KvLIHJkUSiafSTkieawO2l85gyQRZs/edit?usp=sharing>

User Manual:

<https://docs.google.com/document/d/1YPIbdcnCIYD7wrTOA6h2JvEYIc7NEC-eC7NgOwqse6k/edit?usp=sharing>



# Thank You

Have a nice day!

