Predicting Personal Loan Approval Using Machine Learning

1. INTRODUCTION

Now-a-days obtaining loans from banks have become a very common phenomenon. The banks gain profits from the loans lent to their customers in the form of interest. While approving a loan, the banks should consider many factors such as credit history and score, reputation of the person, the location of the property and the relationship with the bank.

Many people apply for loans in the name of home loan, car loan and many more. Everyone cannot be approved based on above mentioned conditions. There are so many cases where applicant's applications for loans are not approved by various finance companies. The right predictions whether to give a loan to a customer or not is very important for the banks to maximize the profits. The idea behind this project is to use Machine Learning to predict whether a customer can get a loan from a bank or not.

Overview

A loan is a sum of money that is borrowed and repaid over a period of time, typically with interest. There are various types of loans available to individuals and businesses, such as personal loans, mortgages, auto loans, student loans, business loans and many more. They are offered by banks, credit unions, and other financial institutions, and the terms of the loan, such as interest rate, repayment period, and fees, vary depending on the lender and the type of loan.

A personal loan is a type of unsecured loan that can be used for a variety of expenses such as home repairs, medical expenses, debt consolidation, and more. The loan amount, interest rate, and repayment period vary depending on the lender and the borrower's credit worthiness. To qualify for a personal loan, borrowers typically need to provide proof of income and have a good credit score.

Predicting personal loan approval using machine learning analyses a borrower's financial data and credit history to determine the likelihood of loan approval. This can help financial institutions to make more informed decisions about which loan applications to approve and which to deny.

Purpose:

The purpose of a personal loan approval project would be to develop a model that can predict whether an individual will be approved for a personal loan or not. This model would be trained on historical data of loan applications, where each application would be associated with a set of features such as the borrower's credit score, income, employment history, debt-to-income ratio, loan amount, and other relevant factors.

The aim of this project would be to build a predictive model that can accurately classify loan applications as either approved or rejected. The model could be used by lenders to make informed decisions about which loan applications to approve and which to reject, based on the likelihood of the borrower repaying the loan.

To develop this model, a machine learning approach would be used, where a dataset of historical loan applications would be split into a training set and a testing set. The training set would be used to train the model on the different features and their impact on loan approval, while the testing set would be used to evaluate the model's performance and accuracy.

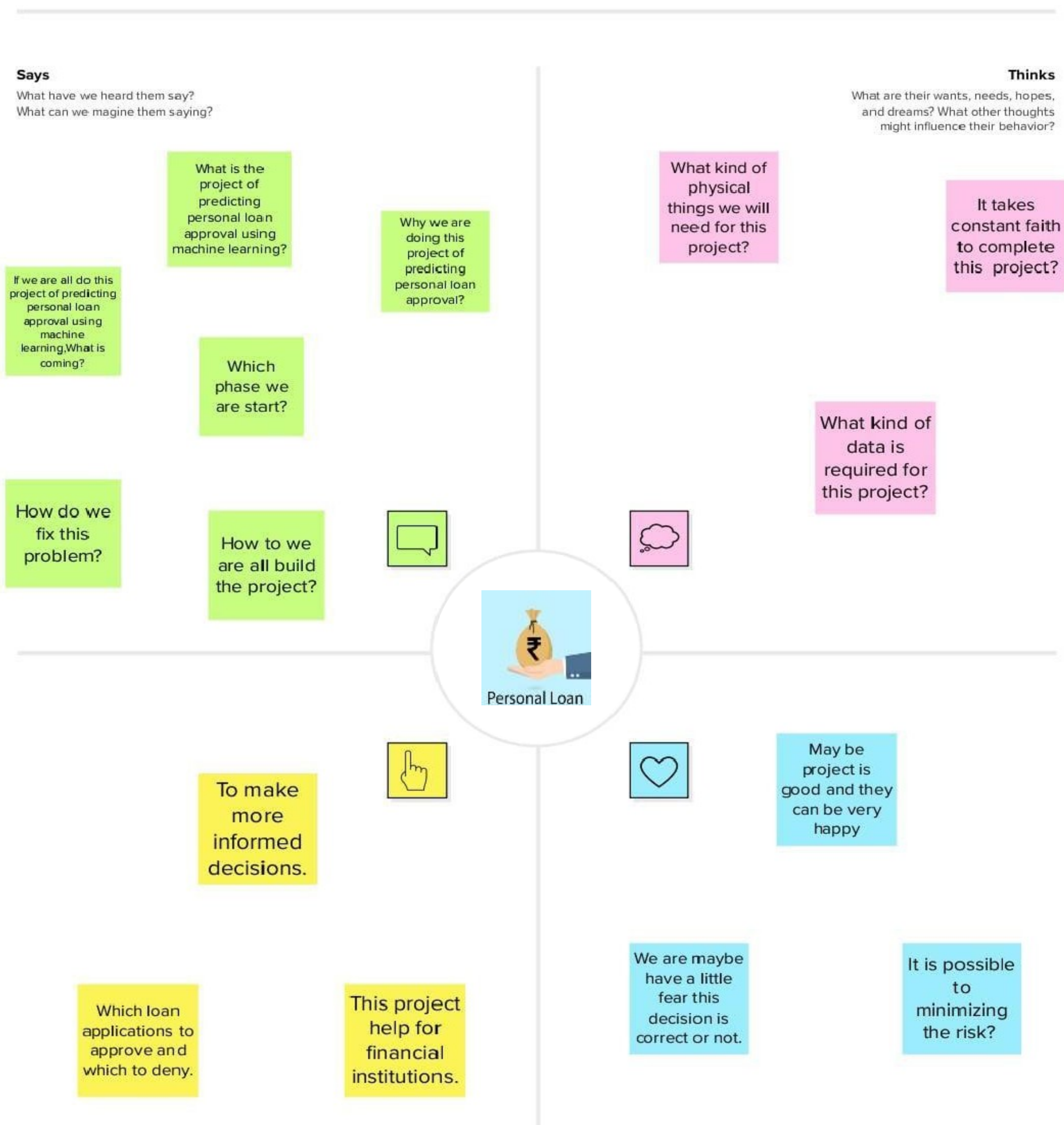
Some of the techniques that could be used to build this model include logistic regression, decision trees, random forests, or gradient boosting. Additionally, the model's performance could be improved by using techniques such as feature engineering, hyperparameter tuning, and cross-validation.

2. PROBLEM DEFINITION & DESIGN THINKING

PROBLEM DEFINITION:

The problem of predicting personal loan approval involves using data analysis and machine learning algorithms to predict whether a borrower will be approved for a personal loan or not. This problem is important for both lenders and borrowers. Lenders need to accurately assess the risk of lending money to borrowers, while borrowers need to understand their chances of getting approved and prepare accordingly. The problem of predicting personal loan approval involves analyzing various factors such as the borrower's credit history, income, employment status, debt-to-income ratio, loan amount, and loan purpose, among others. Machine learning algorithms such as logistic regression, decision trees, and neural networks can be used to analyze these factors and predict whether a borrower will be approved or not. The accuracy of the prediction model depends on the quality and quantity of data used for training and testing the model. Therefore, it is important to have a large and diverse dataset to ensure that the model is robust and can generalize well to new data. Additionally, feature engineering, data preprocessing, and model selection are important steps in building an accurate prediction model. Overall, the problem of predicting personal loan approval is an important and challenging task that requires careful analysis and modeling to ensure accurate predictions

Empathy Map

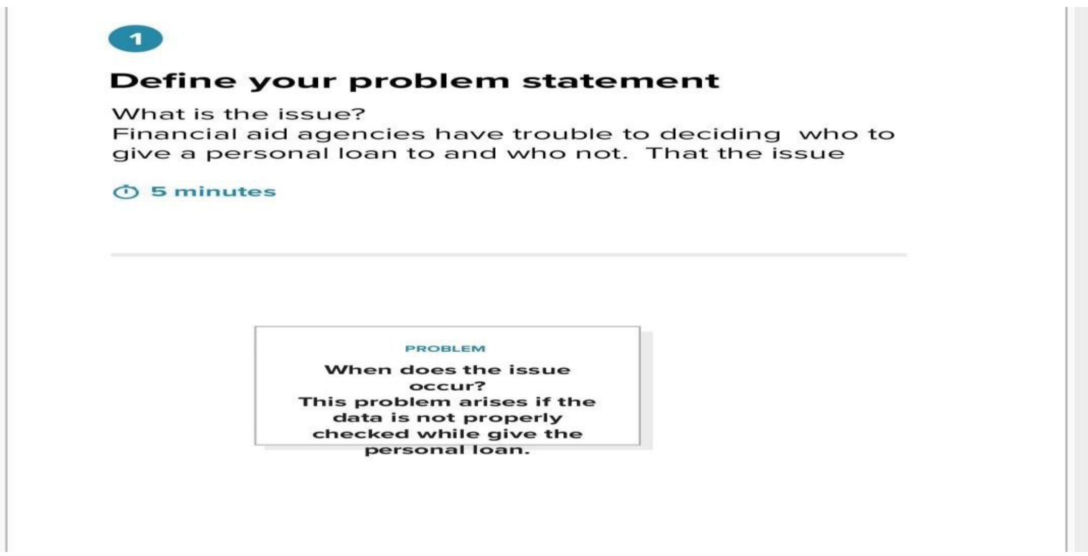


In the above Mural picture Our Group done the map with our ideas about Optimizing

Spam Filtering with Machine Learning. We talk about the topic from users side what they Says, Thinks, Feels and Does. We added the screenshot of our Mural map

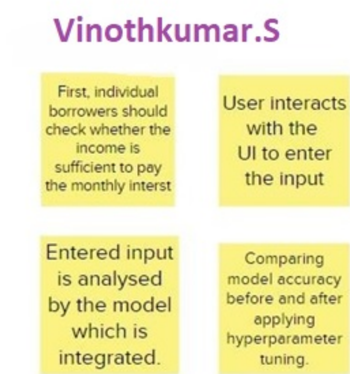
Ideation & Brainstroming

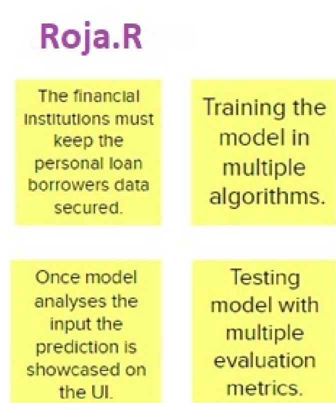
In the below we added Screenshot of our Brainstorm and Ideation. In this template we give our brainstorm ideas about our project.



BRAINSTROM

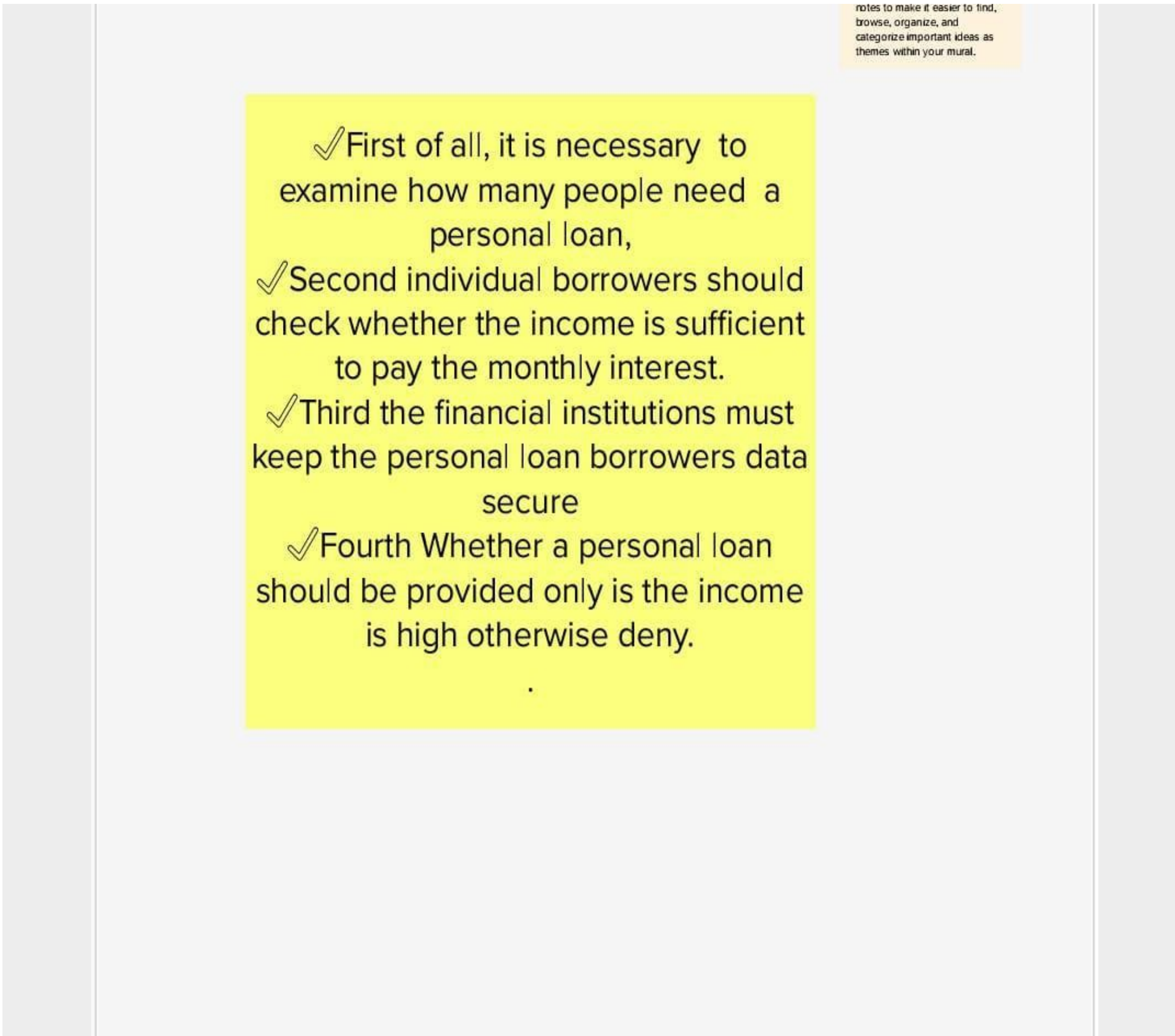

Sneka.k
First of all, it is necessary to examine how many people need a personal loan
A loan is some of money that is borrowed and repaid over period of time.
A personal loan is a type of unsecured loan that can be used for a variety of expences such as home repairs, medical expences,debt consolidation and more.
Predicting personal loan approval using machine learning analyses a borrower's financial data and credit history to determine the likehood of loan approval.


Vinothkumar.S
First, individual borrowers should check whether the income is sufficient to pay the monthly interst
User interacts with the UI to enter the input
Entered input is analysed by the model which is integrated.
Comparing model accuracy before and after applying hyperparameter tuning.

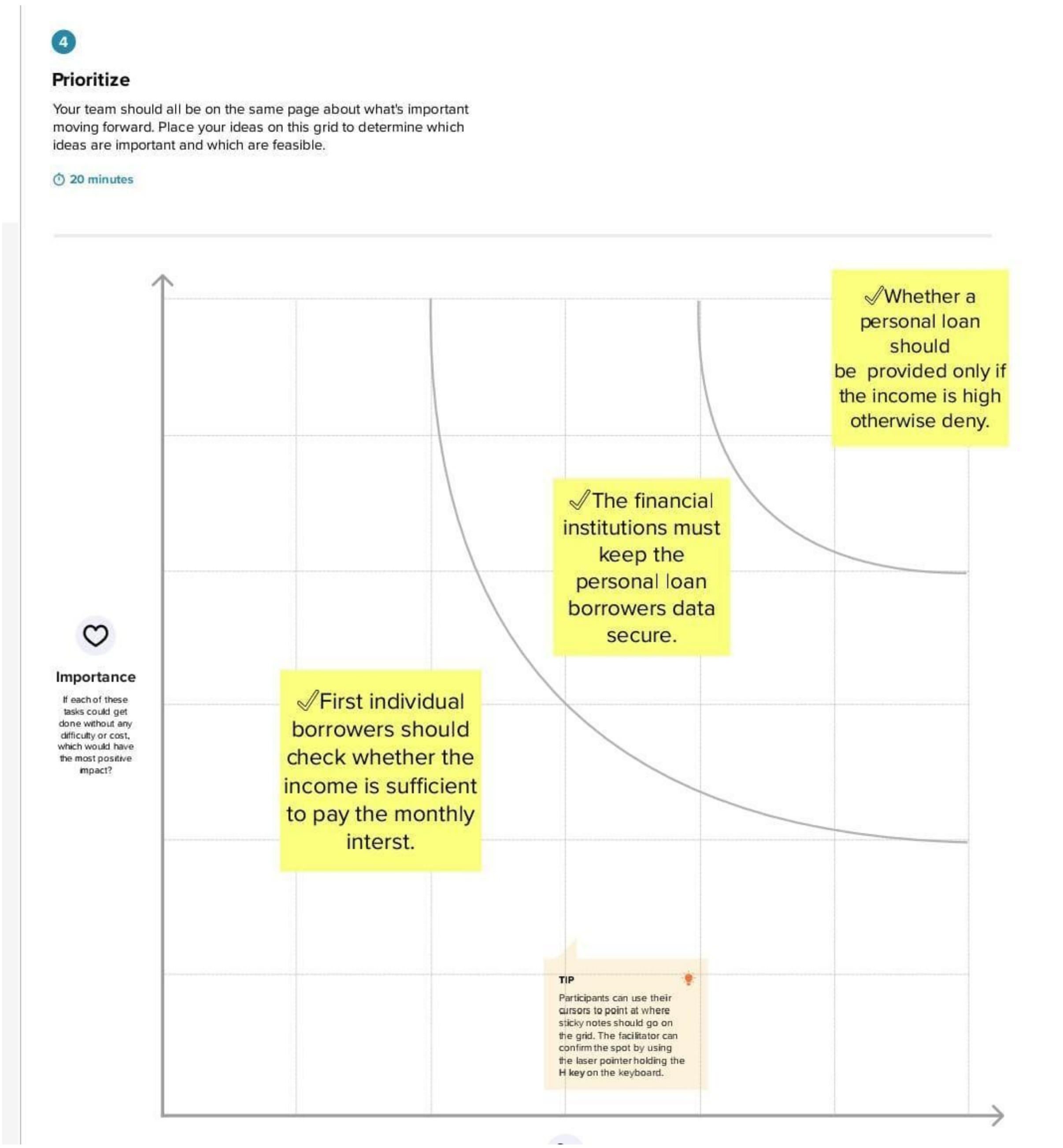

Roja.R
The financial institutions must keep the personal loan borrowers data secured.
Training the model in multiple algorithms.
Once model analyses the input the prediction is showcased on the UI.
Testing model with multiple evaluation metrics.


Ponmozhi.P
Whether a personal loan should be provided only if the income is high otherwise deny.
Specify the business problems
Refer the business requirements
Exploratory of the data analysis is descriptive statistical and visual analysis.

GROUP IDEAS

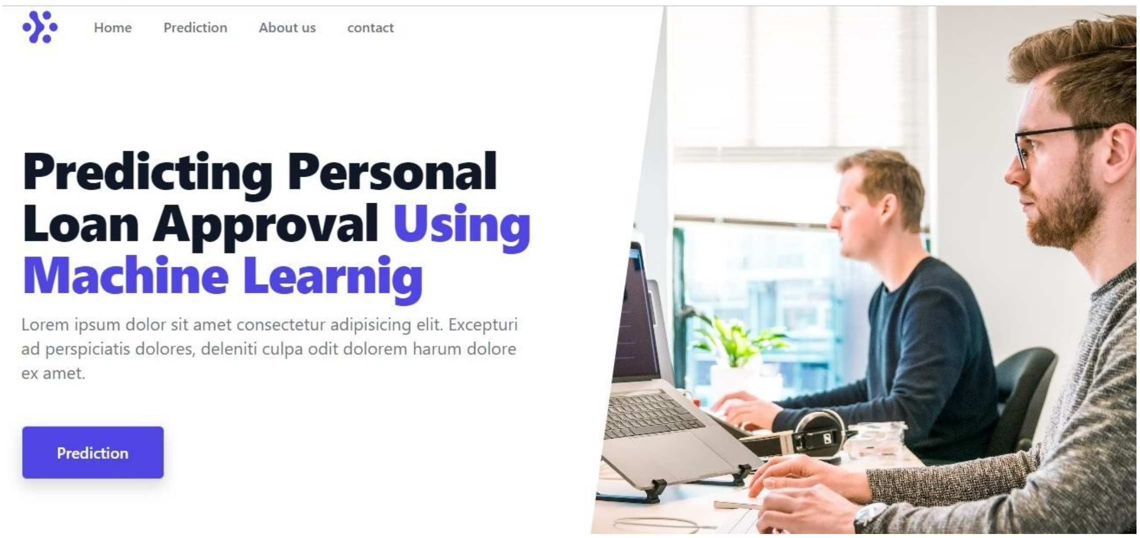

notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
✓First of all, it is necessary to examine how many people need a personal loan,
✓Second individual borrowers should check whether the income is sufficient to pay the monthly interest.
✓Third the financial institutions must keep the personal loan borrowers data secure
✓Fourth Whether a personal loan should be provided only is the income is high otherwise deny.
.

PRIORITIZE

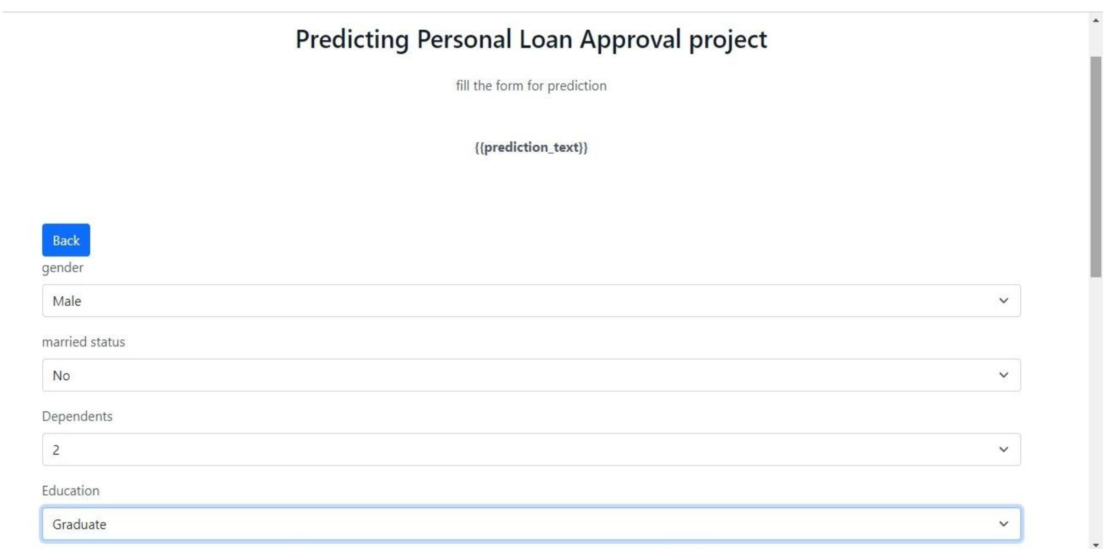

4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
✓Whether a personal loan should be provided only if the income is high otherwise deny.
✓The financial institutions must keep the personal loan borrowers data secure.
✓First individual borrowers should check whether the income is sufficient to pay the monthly interst.
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

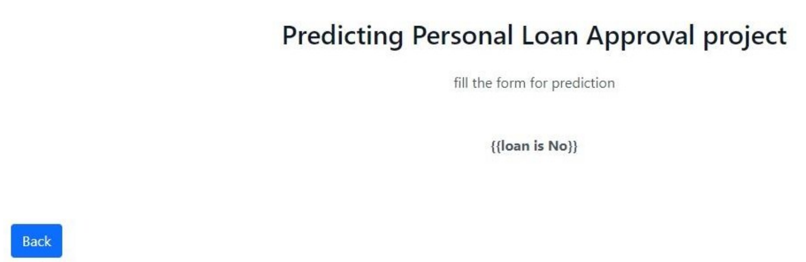
3.RESULT

Web page of Loan Approval



Web page of Loan Approval!!!





FINAL OUTPUT PREDICTED

4. ADVANTAGES

- ✓ Increased Efficiency: Machine learning algorithms can analyze large amounts of data quickly and accurately, allowing lenders to process loan applications faster and more efficiently. This can result in reduced manual effort, streamlined workflows, and quicker decision-making, saving both time and resources.
- ✓ Enhanced Accuracy: Machine learning models can analyze various data points, including credit scores, income levels, employment history, and other relevant factors, to predict the likelihood of loan approval with high accuracy. This can reduce the risk of human errors and bias in decision-making, resulting in more consistent and reliable loan approval predictions.
- ✓ Improved Risk Assessment: Machine learning algorithms can assess the creditworthiness of borrowers more accurately by analyzing historical data on loan defaults and other risk factors. This helps lenders make informed decisions about loan approvals, minimizing the risk of default and potential financial losses.
- ✓ Personalized Loan Offers: Machine learning models can analyze individual borrower profiles and generate personalized loan offers based on their financial situation, credit history, and other relevant factors. This can result in more customized loan terms and interest rates, increasing the chances of loan approval and borrower satisfaction.
- ✓ Enhanced Customer Experience: Faster loan processing, accurate predictions, and personalized loan offers can result in an improved customer experience. Borrowers can receive prompt feedback on their loan applications, have a better understanding of their loan eligibility, and receive loan offers that are tailored to their needs. This can result in increased customer satisfaction and loyalty.
- ✓ Cost Savings: By automating the loan approval process, machine learning can reduce the need for manual review and analysis, resulting in cost savings for lenders. It can also help lenders optimize their loan portfolio and reduce the risk of default, potentially saving money on potential losses.
- ✓ Compliance and Fairness: Machine learning models can be designed to adhere to regulatory requirements and fair lending practices, reducing the risk of bias and discrimination in loan approval decisions. This can help lenders ensure compliance with legal and ethical guidelines, promoting transparency and fairness in lending practices.

DISADVANTAGES

- ✓ Bias and Fairness Concerns: Machine learning models can inadvertently incorporate bias from historical data, resulting in biased loan approval decisions. This can perpetuate discriminatory lending practices, such as racial or gender bias, and result in unfair treatment of certain groups of borrowers.
- ✓ Lack of Interpretability: Some machine learning models, such as deep learning algorithms, may lack interpretability, making it challenging to understand how the model arrives at its predictions. This can make it difficult for lenders to explain loan approval decisions to borrowers or regulators, which may raise concerns about transparency and accountability.
- ✓ Overreliance on Data: Machine learning models rely heavily on data for training and prediction, and if the data used is incomplete, inaccurate, or biased, it can negatively impact the model's performance and the loan approval decisions. Ensuring high-quality, representative, and unbiased data can be a challenge.
- ✓ Changing Regulatory Landscape: Regulations governing lending practices and data privacy can evolve, and complying with these regulations can be complex. Machine learning models used for loan approval may need to be updated or modified to ensure compliance with changing regulatory requirements, which can be time-consuming and resource-intensive.
- ✓ Lack of Human Judgment: Machine learning models may not fully capture the nuances of human judgment, which can be important in assessing factors such as character and borrower intent. Over-reliance on automated predictions without human judgment may lead to inaccurate or incomplete loan approval decisions.
- ✓ Risk of Model Overfitting: Machine learning models can be susceptible to overfitting, where the model may perform well on the training data but may not generalize well to new, unseen data. This can result in inaccurate loan approval predictions and increased risk of loan defaults.

5. APPLICATIONS

Predicting personal loan approval using machine learning has a wide range of applications across the financial industry, including:

1. Banks and financial institutions: Banks and other financial institutions can use machine learning to automate the loan approval process and make more accurate loan approval decisions. This can lead to faster loan processing times, reduced costs, and improved customer satisfaction.
2. Peer-to-peer lending platforms: Peer-to-peer lending platforms can use machine learning to evaluate borrower creditworthiness and make loan approval decisions. This can help ensure that loans are being made to creditworthy borrowers and reduce the risk of default.
3. Credit scoring companies: Credit scoring companies can use machine learning to develop more accurate credit scoring models, which can be used by lenders to make loan approval decisions. This can help improve access to credit for underserved populations and reduce the risk of default.
4. Insurance companies: Insurance companies can use machine learning to assess the risk of lending to borrowers and make more accurate loan approval decisions. This can help reduce the risk of default and improve the profitability of insurance products.
5. Fintech startups: Fintech startups can use machine learning to develop innovative loan approval products and services, such as microloans and instant loan approvals. This can help improve access to credit for underserved populations and reduce the risk of default.
6. Government agencies: Government agencies can use machine learning to develop more effective loan programs and improve the efficiency of the loan approval process. This can help improve access to credit for individuals and small businesses and promote economic growth.

Overall, predicting personal loan approval using machine learning has the potential to transform the way that loans are approved and processed, leading to faster decision-making, reduced costs, and improved access to credit for individuals and small businesses.

6. CONCLUSION

Random Forest Classifier is giving the best accuracy with an accuracy score of 82% for the testing dataset. And to get much better results ensemble Learning techniques like Bagging and Boosting can also be used. Predicting personal loan approval using machine learning can be an effective tool for lenders to make informed decisions about loan applications. By analyzing various factors such as credit score, income, employment status, and loan history, machine learning algorithms can accurately predict whether an applicant is likely to be approved for a personal loan or not. Through the use of supervised learning algorithms such as logistic regression, decision trees, and neural networks, lenders can train their models on historical data to make accurate predictions about future loan applicants. These algorithms can help lenders identify risky loan applicants and reduce the risk of default. In conclusion, machine learning can be a powerful tool for predicting personal loan approval. Lenders can use machine learning algorithms to analyze a large amount of data and make informed decisions about loan approvals. However, it's important to ensure that the algorithms are trained on unbiased data and that ethical considerations are taken into account to avoid discrimination and maintain fairness in lending practices.

7. FUTURE SCOPE

The future scope of predicting personal loan approval using machine learning is very promising. As the financial industry continues to become more data-driven, machine learning models are expected to play an increasingly important role in the loan approval process. Here are some potential future developments:

1. Use of more advanced machine learning techniques: As machine learning techniques continue to evolve, more advanced algorithms and models may be developed that can improve loan approval predictions even further. For example, deep learning techniques may be used to analyze unstructured data, such as borrower social media activity, to improve loan approval predictions.
2. Integration with blockchain technology: Blockchain technology has the potential to improve the security and transparency of the loan approval process, and may be integrated with machine learning models to further improve loan approval predictions.
3. Collaboration between lenders: Lenders may collaborate to share data and develop more accurate loan approval models. This could lead to more consistent loan approval decisions across different lenders, and could help reduce the risk of default.
4. Increased use of alternative data sources: Machine learning models may be trained on alternative data sources, such as mobile phone usage data or utility bill payment history, to improve loan approval predictions. This could help improve access to credit for underserved populations who may not have traditional credit histories.
5. Expansion to other types of loans: The use of machine learning to predict loan approvals may expand to other types of loans, such as business loans, mortgage loans, and car loans.

Overall, the future scope of predicting personal loan approval using machine learning is very promising, and is likely to lead to continued improvements in the speed, efficiency, and accuracy of the loan approval process.

8. APPENDIX

```
import pandas as pd

import numpy as np
import pickle
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier, RandomFo
restClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import RandomizedSearchCV
import imblearn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report
, confusion_matrix, f1_score
import warnings
warnings.filterwarnings('ignore')
```

Importing data set

```
df = pd.read_csv('/content/train_u6lujuX_CVtuZ9i.csv')
df


df.drop(['Loan_ID'],axis=1,inplace=True)
df.head()

# Handling Categorical values
### Changing the data type of each float cloumn to int

df['Gender']=df['Gender'].map({'Female':1,'Male':0})
df['Property_Area']=df['Property_Area'].map({'Urban':2,'Semiurban
':1, 'Rural':0})
df['Married']=df['Married'].map({'Yes':1,'No':0})
df['Education']=df['Education'].map({'Graduate':1,'Not Graduate':
0})
df['Self_Employed']=df['Self_Employed'].map({'Yes':1,'No':0})
df['Loan_Status']=df['Loan_Status'].map({'Y':1,'N':0})
df.head()
```

Handling Missing Value

```
#sum of null values
df.isnull().sum()
```

```
df['Gender'] = df['Gender'].fillna(df['Gender'].mode()[0])
df['Married'] = df['Married'].fillna(df['Married'].mode()[0])
df['Dependents']=df['Dependents'].str.replace('+','')
df['Dependents']=df['Dependents'].fillna(df['Dependents'].mode()[
0])
df['Self_Employed'] = df['Self_Employed'].fillna(df['Self_Employe
d'].mode()[0])
df['LoanAmount'] = df['LoanAmount'].fillna(df['LoanAmount'].mode(
)[0])
df['Loan_Amount_Term'] = df['Loan_Amount_Term'].fillna(df['Loan_A
mount_Term'].mode()[0])
df['Credit_History'] = df['Credit_History'].fillna(df['Credit_His
tory'].mode()[0])
#sum of null values
df.isnull().sum()

df.info()

# Handling Categorical values
### Changing the data type of each float cloumn to int
df['Gender']=df['Gender'].astype('int64')
df['Married']=df['Married'].astype('int64')
df['Dependents']=df['Dependents'].astype('int64')
df['Self_Employed']=df['Self_Employed'].astype('int64')
df['CoapplicantIncome']=df['CoapplicantIncome'].astype('int64')
df['LoanAmount']=df['LoanAmount'].astype('int64')
df['Loan_Amount_Term']=df['Loan_Amount_Term'].astype('int64')
df['Credit_History']=df['Credit_History'].astype('int64')
```

Visual Analysis

Univariate Analysis

```
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['ApplicantIncome'], color='r')
plt.subplot(122)
sns.distplot(df['Credit_History'])
plt.show()
#gender andeducation Visual
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.countplot(df ['Gender'],color='r')
plt.xlabel('Gender')
```

```
plt.subplot(1,4,2)
sns.countplot(df['Education'])
plt.xlabel('Education')
plt.show()
plt.figure(figsize=(10,4))
plt.subplot(131)
sns.countplot(df['Married'])
```

Bivariate Analysis

```
plt.figure(figsize=(10,5))

sns.heatmap(df.corr(),cmap='BrBG',fmt='.2f',
      linewidths=2,annot=True)

plt.figure(figsize=(20,5))
plt.subplot(1,3,1)
sns.countplot(x = 'Married', hue = "Gender", data = df)
plt.subplot(1,3,2)
sns.countplot(x = 'Self_Employed', hue = "Education", data = df)
plt.subplot(1,3,3)
sns.countplot(x = 'Property_Area', hue = "Loan_Amount_Term", data
 = df)

pd.crosstab(df['Gender'],[df['Self_Employed']])
```

Multivariate Analysis

```
#visualized based gender and income what would be the application
 status
sns.swarmplot(x = 'Gender', y = 'ApplicantIncome', hue = "Loan_St
atus", data = df)


# Handling Imbalance Data
from imblearn.combine import SMOTETomek
smote = SMOTETomek()
y = df['Loan_Status']
x = df.drop(columns=['Loan_Status'],axis=1)
```

dividing the dataset into dependent and independent y and x respectively

```
x.shape
y.shape

x_bal,y_bal = smote.fit_resample(x,y)
print(y.value_counts())
print(y_bal.value_counts())
```

```
names = x_bal.columns
x_bal.head()
```

Scalling the dataset

```
#performing feature scaling operation using standard scaller an x
 part ofthe dataset because
#there different type of value in the columns
sc=StandardScaler()
x_bal=sc.fit_transform(x_bal)
x_bal = pd.DataFrame(x_bal,columns=names)
x_bal.head()
```

SPLITING DATA SET

```
#splitting the dataset in train and test on balnmced data set
x_train,x_test,y_train,y_test = train_test_split(x_bal,y_bal, tes
t_size=0.33, random_state=42)
x_train.shape

x_train.shape
y_train.shape, y_test.shape
```

Model bulding

```
#RandomForest model
from tensorflow.keras import Model
def RandomForest(x_train,x_test,y_train,y_test):
  model = RandomForestClassifier()
  model.fit(x_train,y_train)
  y_tr = model.predict(x_train)
  print(accuracy_score(y_tr,y_train))
  yPred = model.predict(x_test)
  print(accuracy_score(yPred,y_test))

RandomForest(x_train,x_test,y_train,y_test)

#decisionTree model
def decisionTree(x_train,x_test,y_train,y_test):
 model = DecisionTreeClassifier()
 model.fit(x_train,y_train)
 y_tr = model.predict(x_train)
 print(accuracy_score(y_tr,y_train))
 yPred = model.predict(x_test)
 print(accuracy_score(yPred,y_test))

decisionTree(x_train,x_test,y_train,y_test)

#KNN model
def KNN(x_train,x_test,y_train,y_test):
  model = KNeighborsClassifier()
```

```
  model.fit(x_train,y_train)
  y_tr = model.predict(x_train)
  print(accuracy_score(y_tr,y_train))
  yPred = model.predict(x_test)
  print(accuracy_score(yPred,y_test))

KNN(x_train,x_test,y_train,y_test)

#XGB model
def XGB(x_train,x_test,y_train,y_test):
  model = GradientBoostingClassifier()
  model.fit(x_train,y_train)
  y_tr = model.predict(x_train)
  print(accuracy_score(y_tr,y_train))
  yPred = model.predict(x_test)
  print(accuracy_score(yPred,y_test))

XGB(x_train,x_test,y_train,y_test)

import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
classifier = Sequential()
classifier.add(Dense(units=100, activation='relu', input_dim=11))
classifier.add(Dense(units=50, activation='relu'))
classifier.add(Dense(units=1, activation='sigmoid'))
classifier.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
classifier.fit(x_train,y_train,batch_size=100, validation_split=0
.2, epochs=100)

y_pred = classifier.predict(x_test)

y_pred
y_pred = y_pred.astype(int)
y_pred

dt = DecisionTreeClassifier()
dt.fit(x_train,y_train)

print(classification_report(y_test,dt.predict(x_test)))

confusion_matrix(y_test,dt.predict(x_test))
#checking the accuracy
print(accuracy_score(y_pred,y_test))
print("ANN Model")
```

```
print("confusion_matrix")
print(confusion_matrix(y_test, y_pred))
print("classification_report")
print(classification_report(y_test, y_pred))
dt.predict([[1,1, 0, 1, 1, 4276, 1542,145, 240, 0,1]])
rfr = RandomForestClassifier()
rfr.fit(x_train,y_train)
print(classification_report(y_test,dt.predict(x_test)))

rfr.predict([[1,1, 0, 1, 1, 4276, 1542,145, 240, 0,1]])
knn = KNeighborsClassifier()
knn.fit(x_train,y_train)
print(classification_report(y_test,dt.predict(x_test)))
knn.predict([[1,1,0,1,1,4276,1542,145,240,0,1]])
xgb = GradientBoostingClassifier()
xgb.fit(x_train,y_train)
print(classification_report(y_test,dt.predict(x_test)))

xgb.predict([[1,1, 0, 1, 1, 4276, 1542,145, 240, 0,1]])
classifier.save("loan.h5")
y_pred = classifier.predict(x_test)
y_pred
y_pred = (y_pred > 0.5)
y_pred
def predict_exit(sample_value):
  sample_value = np.array(sample_value)
  sample_value = sample_value.reshape(1, -1)
  sample_valu = sc.transform(sample_value)
  return classifier.predict(sample_value)
sample_value = [[1,1,0,1,1,4276,1542,145,240,0,1]]
if predict_exit(sample_value)>0.5:
  print('Prediction: High Chance of Loan Approval!')
else:
  print('Prediction: Low Chance of Loan Approval!')

sample_value = [[1,0, 1, 1, 1, 45, 14,45, 240, 1,1]]

if predict_exit(sample_value)>0.5:
  print('Prediction: High Chance of Loan Approval!')
else:
  print('Prediction: Low Chance of Loan Approval!')
```

saving the model

```
#saving the  model
pickle.dump(rfr,open('model.pkl','wb'))
pickle.dump(sc,open('scale.pkl','wb'))
```