**Project Report Format**

1. **INTRODUCTION** 
   1. Project Overview
   2. Purpose
2. **LITERATURE SURVEY**
   1. Existing problem
   2. References
   3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
   3. Proposed Solution
   4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
   3. Reports from JIRA
7. **CODING & SOLUTIONING** 
   1. GUI using Flask
   2. Loan Prediction model
8. **TESTING** 
   1. Test Cases
   2. User Acceptance Testing
9. **RESULTS**
   1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

1. **INTRODUCTION**
   1. **Project Overview**

One of the most important factors which affect our country’s economy and financial condition is the credit system governed by the banks. The process of bank credit risk evaluation is recognized at banks across the globe. “As we know credit risk evaluation is very crucial, there is a variety of techniques are used for risk level calculation. In addition, credit risk is one of the main functions of the banking community. The prediction of credit defaulters is one of the difficult tasks for any bank. But by forecasting the loan defaulters, the banks definitely may reduce their loss by reducing their non-profit assets, so that recovery of approved loans can take place without any loss and it can play as the contributing parameter of the bank statement. This makes the study of this loan approval prediction important. Machine Learning techniques are very crucial and useful in the prediction of these types of data. We will be using classification algorithms such as Decision tree, Random-forest, KNN, and XGboost. We will train and test the data with these algorithms. From this best model is selected and saved in pkl format. We will be doing flask integration, develop a web interface and IBM deployment.

* 1. **Purpose**

The purpose of Smart Lender System is to provide a comprehensive web based platform to manage different lenders & borrowers. Lenders can specify their loan criteria, lending terms, rate of interest, mandatory documentation and agreements etc. Borrowers can then apply for loans in the system. The loan origination software checks borrower eligibility and matches it with the lending criterion according to the algorithm. The loan is disbursed after approval of the lending terms. Smart Lending system finds algorithmic match for borrower eligibility, loan terms and conditions. It eliminates repetitive manual steps that are best executed digitally and allows human expertise to be applied where it works best. Integration with credit data sources and services such as LexisNexis or Experian lets lenders automatically and quickly verify applicant information. No lost or misplaced documents. Paper documents converted to digital images are immediately and securely accessible by the underwriter, so they can review applicant materials more quickly. The combination of decision rules and integration with credit data sources and services lets lenders automatically calculate optimum loan structures and terms.

1. **LITERATURE SURVEY**
   1. **Existing problem**

A bank is a financial institution licensed to receive deposits and make loans needs away to verify the customer details and their documents for getting loan because they need a trustable customer with proper documents who can repay the loan amount and interest on time. A lender is an individual or a financial institution that makes funds available to a person with expectation that the funds will be repaid who needs a way to easily and quickly approve the loan for a trustworthy person because manually loan approval is a time taking process. The lender needs a way to trust the borrower’s credentials so that he can give loan to the borrower with assured repayment of the loan. A lender is a party who loans out money needs a way to automate the loan prediction process because he cannot easily trust the person. A bank manager who needs a way to predict the loan approval of a person automatically because of the difficulty in manual loan prediction as he wants to hire highly professional individuals for approving loan and security issues. A bank is money lender who needs away to lend loans to its customers securely with proper interest and repayment because being impetuous might cause a lot of damage to itself.

* 1. **Reference**

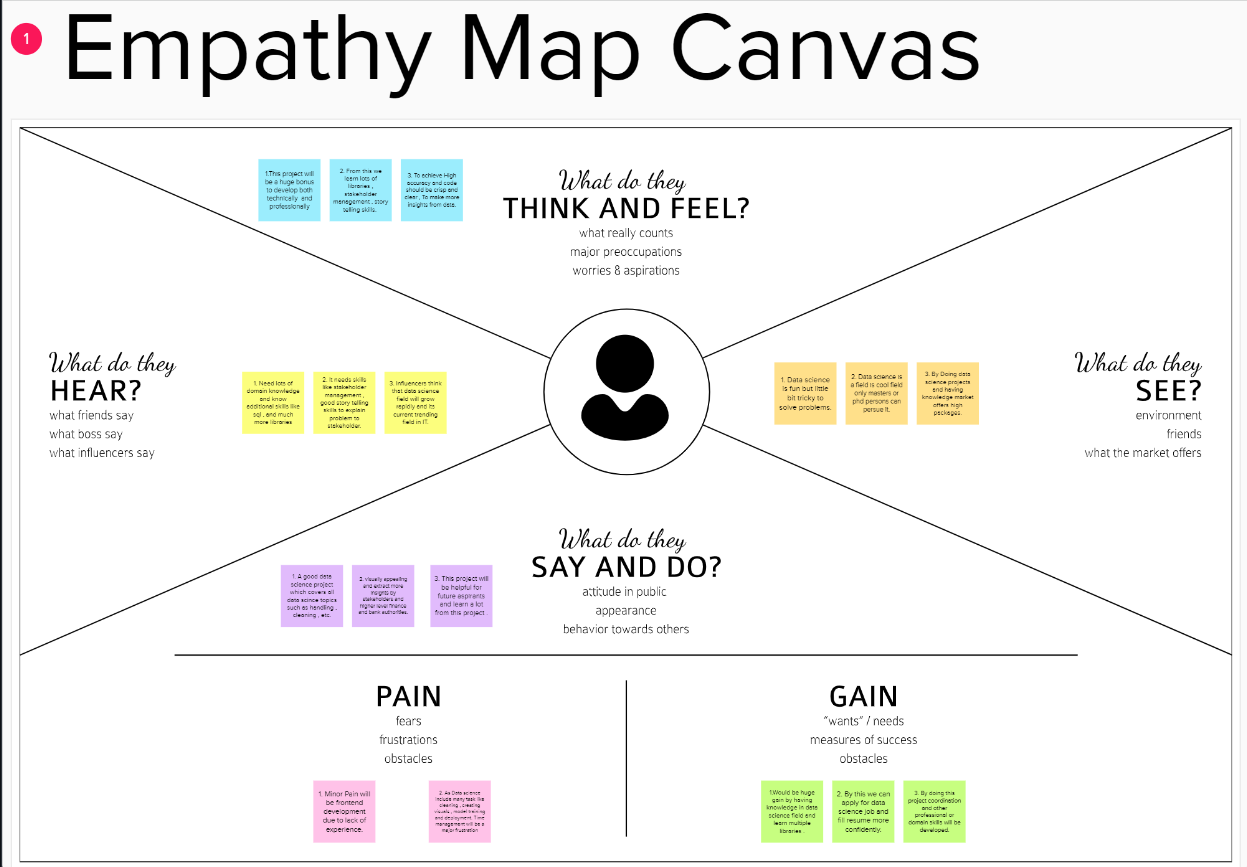
1. “Survey on Prediction of Loan Approval Using Machine Learning Techniques” - Ambika and Santosh Biradar/ Department of Computer Engineering, D. Y. Patil College of Engineering,Pune, India
2. “Process Evaluation and Improvement: A Case Study of The Loan Approval Process” - MAJA PUSNIK, KATJA KOUS, ANDREJ GODEC and BOASTJAN SUMAK, University of Maribor
3. “Loan Approval Prediction based on Machine Learning” - Kumar Arun, Garg Ishan, Kaur Sanmeet
4. “Loan Approval Prediction” - Shubham Nalawade, Suraj Andhe, Siddhesh Parab, Prof. Amruta Sankhe - Information Technology, Atharva College of Engineering, Mumbai
5. “Predict Loan Approval in Banking System Machine Learning Approach for Cooperative Banks Loan Approval” - Amruta S. Aphale ,Prof. Dr. Sandeep. R. Shinde Department of Computer Science and Engineering Savitribai Phule Pune University Vishwakarma Institute of Technology, Pune
   1. **Problem Statement Definition**

A bank is a financial institution licensed to receive deposits and make loans needs a way to verify and trust the customer details and their documents for getting loan because they need an trustable customer with proper assets, cash flow, documents and background who can repay the loan amount and interest on time. The prediction of credit defaulters is one of the difficult tasks for any bank. But by forecasting the loan defaulters, the banks definitely may reduce their loss by reducing their non-profit assets, so that recovery of approved loans can take place without any loss and it can play as the contributing parameter of the bank statement. This makes the study of this loan approval prediction important. Machine Learning techniques are very crucial and useful in the prediction of these types of data. We will be using classification algorithms such as Decision tree, Random-forest, KNN, and XGboost. We will train and test the data with these algorithms. From this best model is selected and saved in pkl format. We will be doing flask integration, develop a web interface and IBM deployment.

1. **IDEATION & PROPOSED SOLUTION**
   1. **Empathy Map Canvas**

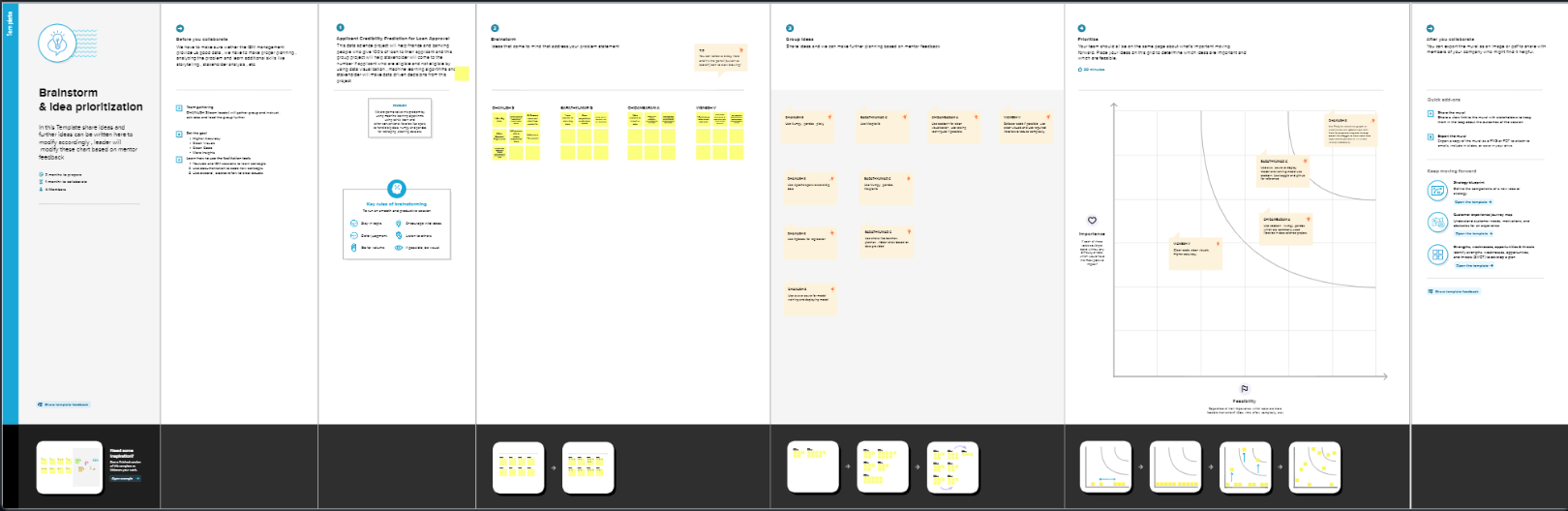
Empathy maps are a straightforward, effective technique for developing knowledge of your people. Empathy, the capacity to comprehend another person's feelings and thoughts, is the name's etymological source. When grounded in actual data and used in conjunction with other mapping techniques, they can:

* Eliminate bias from our designs and bring the team together around a single, shared knowledge of the user
* Find the gaps in our study's findings
* Find out what the user needs—needs that the user may not even be aware of
* Learn what motivates user action. Point us in the direction of genuine innovation



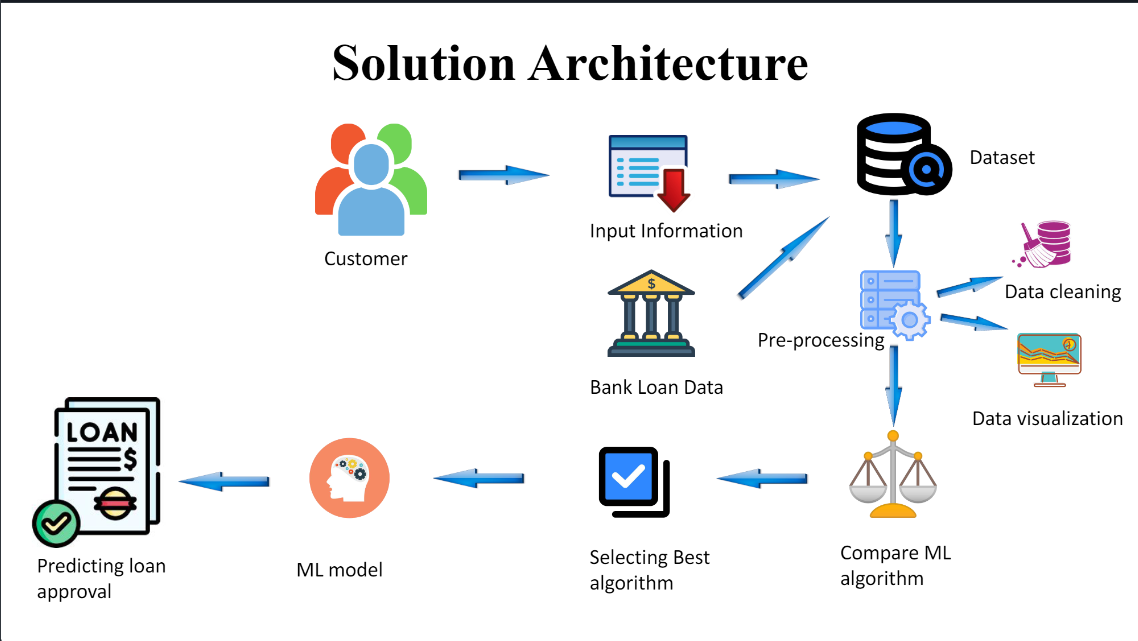
* 1. **Ideation & Brainstorming**

By posing a problem to a group of individuals or team members and engaging them in an open dialogue, the brainstorming approach allows for the generation of ideas. Agile Brainstorming is the name given to this method when it is used in agile projects since it may provide creative ideas. Our group speaks aloud each danger as it is identified. They can take notes so they won't forget a concept before their turn if an increased risk prompts a fresh thought for someone who is not yet in line.

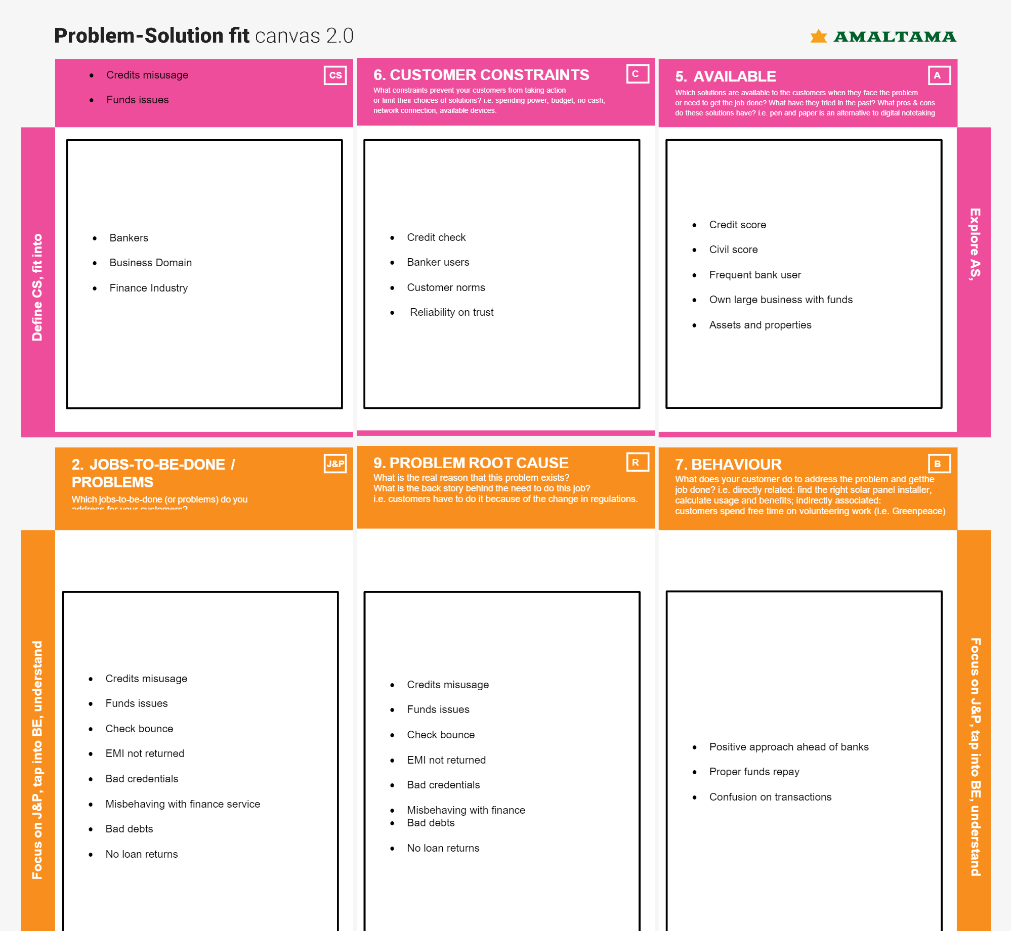
****

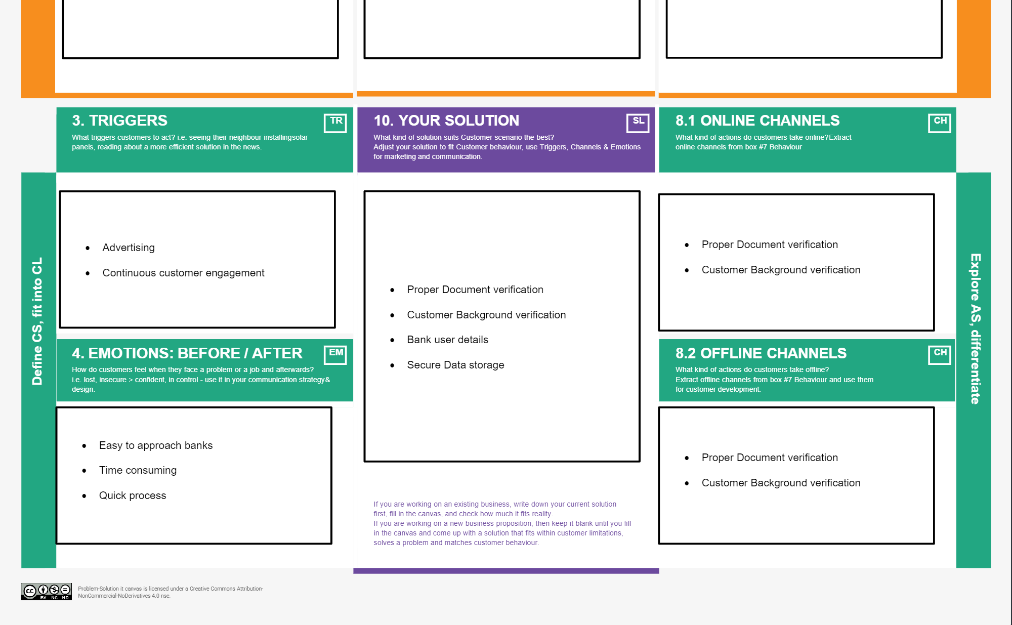
* 1. **Proposed Solution**

The prediction of credit defaulters is one of the difficult tasks for any bank. But by forecasting the loan defaulters, the banks definitely may reduce their loss by reducing their non-profit assets, so that recovery of approved loans can take place without any loss and it can play as the contributing parameter of the bank statement. This makes the study of this loan approval prediction important. Machine Learning techniques are very crucial and useful in the prediction of these types of data. We will be using classification algorithms such as Decision tree, Random-forest, KNN, and XGboost. We will train and test the data with these algorithms. From this best model is selected and saved in pkl format. We will be doing flask integration, develop a web interface and IBM deployment.

****

* The customer only needs to enter the details, the loan approval status is then predicted automatically and quickly. The property documents of the customer need to be submitted and the customer should agree to the terms and conditions of the bank. The loan approval will also depend on the CIBIL score of the customer. Provide captcha security.
* Automatic calculation of interest rate and repayment date based on loan amount. Varies efficient machine learning algorithms can be used to predict the loan eligibility of the customer. Provide customer ratings and reviews for understanding the customer. Adding digital signature of the customer on agreement of the terms and conditions.
* Provides data security. The customer details will not be shared to the third party. Instant Loan approval status. Easy and fast loan approval process for the customer. Approves Loan to a trustable person. Bank can find a genuine person to provide loan
  1. **Problem Solution Fit**

****

****

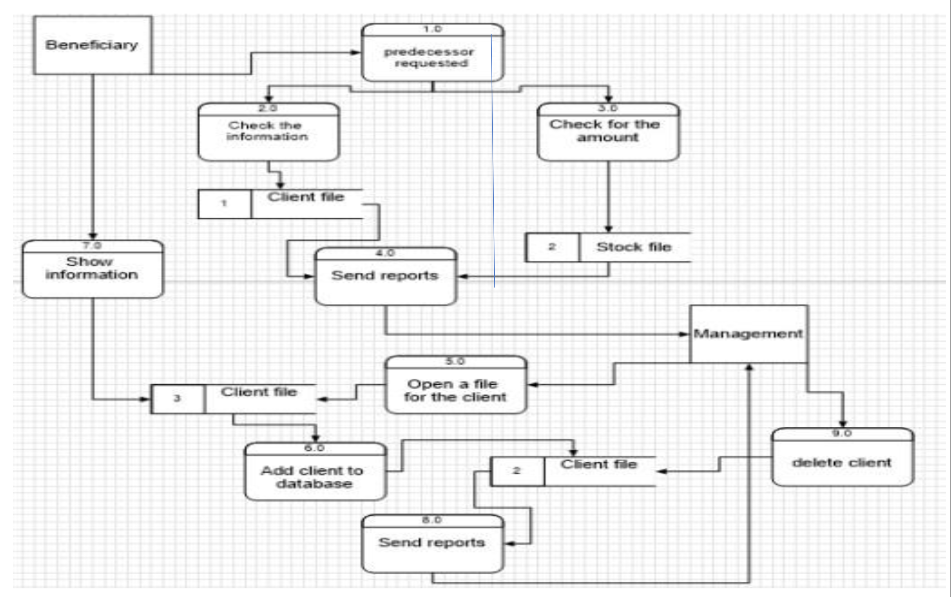
1. **REQUIREMENT ANALYSIS**
   1. **Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form  Registration through Gmail  Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email Confirmation via OTP |
| FR-3 | User Application | Filling of application  Modification of application  Verification of application |
| FR-4 | Loan Issuance | Checking status of loan  Loan Approval  Loan Rejection |
| FR-5 | Credit history analysis | Credit score auditing Income auditing |
| FR-6 | User management | Choosing appropriate loan program for users  Categorising users according to credit history. |

* 1. **Non-Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Simple and understandable UI.  Easy to navigate  Smooth and seamless  Easy to comprehend |
| NFR-2 | **Security** | Restricted access to data. Login  verification  Registration verification  Upholding privacy of user |
| NFR-3 | **Reliability** | Backup to prevent data loss  Negation of data loss due to lag. |
| NFR-4 | **Performance** | Web based application.  Requires minimum Intel Pentium 4 processor, 4 GB RAM, 1280x1024 screen with application window size 1024x680 |

1. **PROJECT DESIGN**
   1. **Data Flow Diagrams**



* 1. **Solution & Technical Architecture**

1. First the Model is trained with the obtained dataset. The data set is given by IBM.

2. Next the dataset will be pre-processed and then the data would be split to train and

test data.

3. Then then model would be saved as a PKL file

4. A website would be created for the interaction and Flask would be used to integrate

the model and website

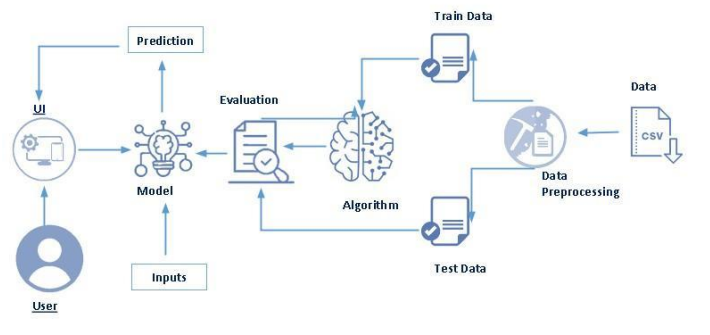
5. The User would give the input, the inputs would be processed and then the

prediction would be made.

6. After the prediction is made the output would be given as “Eligible” or “Not Eligible”

7. This can be scaled even more as an API and integrated into the Mobile banking

application, making it even more convenient for the customer to know the eligibility.



* 1. **User Stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| **Customer (Web user)** | **Forms** | USN – 1 | As a user, I can enter the data which I have and also the data which the website asks to me | Submit the required data for prediction | **High** | Sprint – 1 |
|  | **Prediction** | USN – 2 | As I have given the data into the webpage, now the data can be predicted for the loan avail | Pre-processing is done and data is scaled in  Backend and sent to the model for prediction | **High** | Sprint – 3 |
|  | **Deployment of the Webpage in Cloud** | USN – 3 | As a user, I require global access to the web page as a user | I can get to the Webpage using the provided Web  address | **Medium** | Sprint – 4 |
|  | **Deployment of AI model in the cloud** | USN – 4 | Model would be running on the Cloud | I can access the model through the web address where I typed my data  that’s been set up on the IBM cloud. | **Medium** | Sprint – 4 |
|  | **Model building** | USN – 5 | I require an ML model that can categorise  Credit defaulters | I can use the ML model to classify the Credit defaulters | **High** | Sprint – 2 |
|  | **User Interface building** | USN – 6 | As a User, I need a medium to enter my data | I can use the webpage which uses Flask at the  backend to integrate with the ML Model created | **Medium** | Sprint – 3 |

1. **PROJECT PLANNING & SCHEDULING**
   1. **Sprint Planning & Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team**  **Members** |
| Sprint-1 | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 3 | High | DHANUSH S  BARATHKUMAR G |
| Sprint-1 |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | 3 | High | CHIDAMBARAM A  VIGNESH V |
| Sprint-1 |  | USN-3 | As a user, I can register for the application through Facebook | 1 | Low | DHANUSH S  VIGNESH V |
| Sprint-1 |  | USN-4 | As a user, I can register for the application through Gmail | 2 | Medium | DHANUSH S  BARATHKUMAR G |

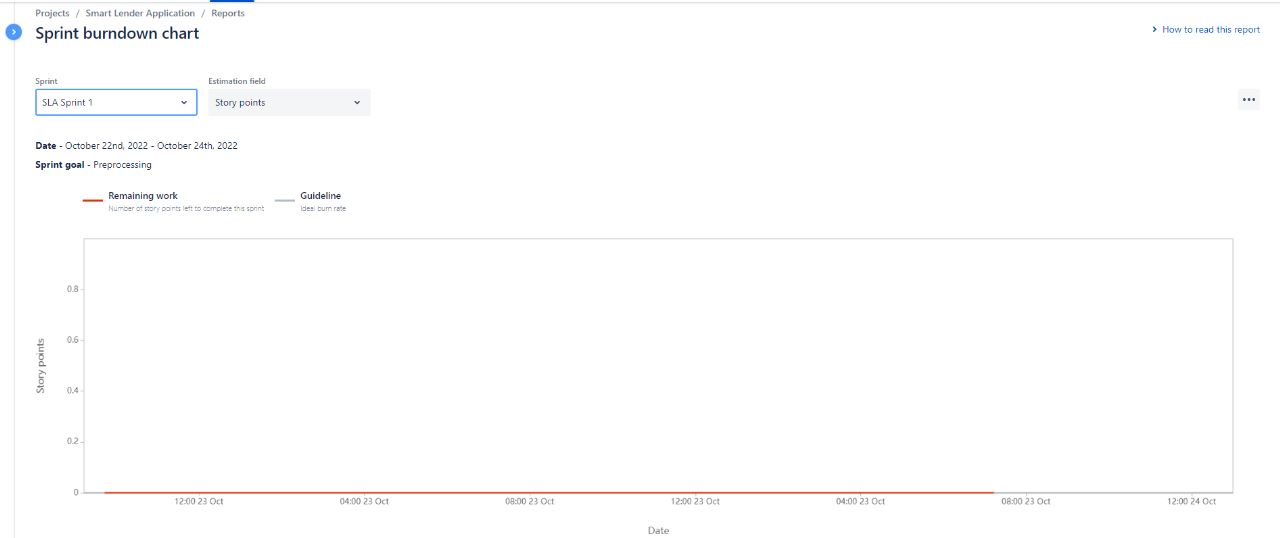
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team**  **Members** |
| Sprint-1 | Login | USN-5 | As a user, I can log into the application by entering email & password | 3 | High | CHIDAMBARAM A  VIGNESH V |
| Sprint-1 | Dashboard | USN-6 | As a user, I should be able to access the dashboard with everything I am allowed to use. | 2 | Medium | DHANUSH S  VIGNESH V |
| Sprint-1 | Registration | USN-7 | As a user, I can register for the application by entering my email, password, and confirming my password. | 3 | High | DHANUSH S  BARATHKUMAR G |
| Sprint-1 |  | USN-8 | As a user, I will receive confirmation email once I have registered for the application | 3 | High | DHANUSH S  BARATHKUMAR G |
| Sprint-1 |  | USN-9 | As a user, I can register for the application through Facebook | 1 | Low | CHIDAMBARAM A  VIGNESH V |
| Sprint-1 |  | USN-10 | As a user, I can register for the application through Gmail | 2 | Medium | DHANUSH S  BARATHKUMAR G |
| Sprint-1 | Login | USN-11 | As a user, I can log into the application by entering email & password | 3 | High | DHANUSH S  BARATHKUMAR G  CHIDAMBARAM A  VIGNESH V |
| Sprint-1 | Dashboard | USN-12 | As a user, I should be able to access the dashboard with everything I am allowed to use. | 2 | Medium | DHANUSH S  VIGNESH V |

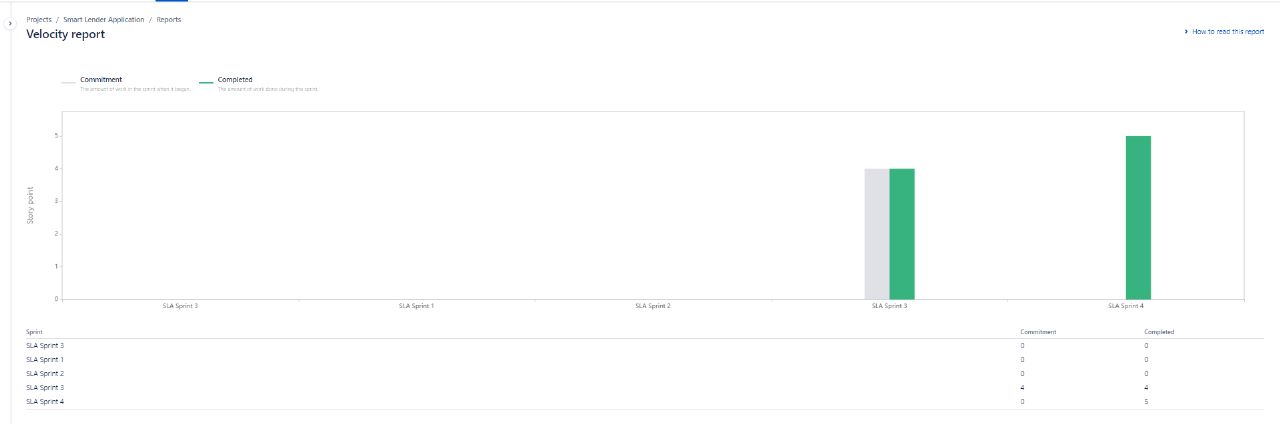
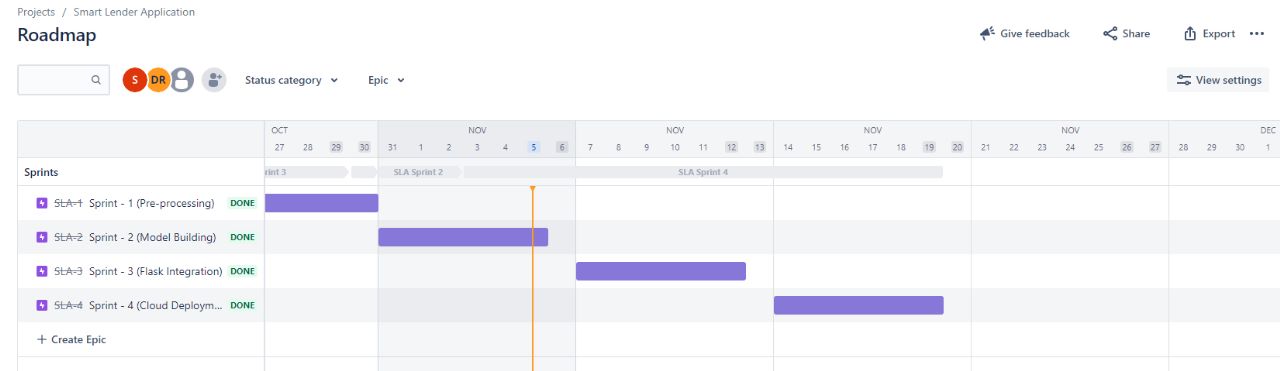
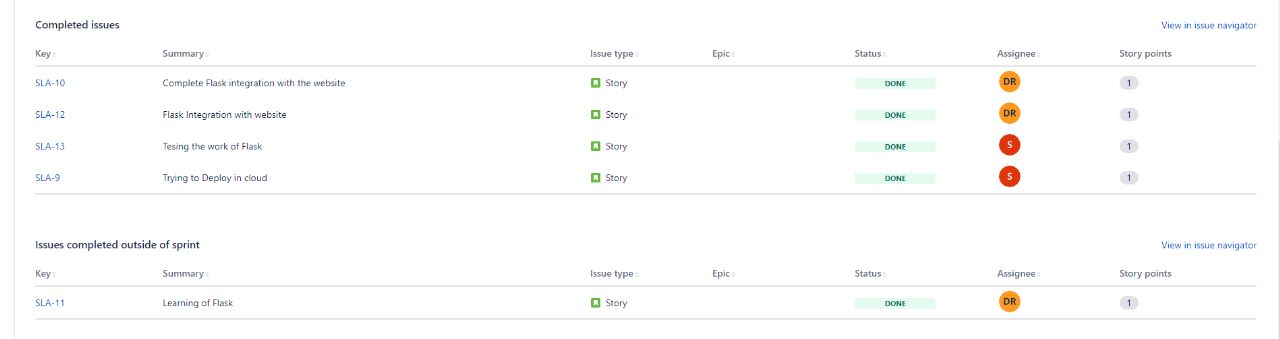
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Story Points** | **Priority** | **Team**  **Members** |
| Sprint-2 | Register | USN-13 | As a loan approval officer, I should be able to register myself as one using unique email and password. | 5 | Medium | DHANUSH S  BARATHKUMAR G |
| Sprint-2 | Login | USN-14 | As a loan approval officer I should be able to login myself as one using unique email and password. | 5 | Medium | DHANUSH S  BARATHKUMAR G |
| Sprint-3 | Automated analysis of credit history | USN-15 | As a loan approval officer, I can access the dashboard where I feed application for loan prediction. | 10 | High | DHANUSH S  BARATHKUMAR G |
| Sprint-3 |  | USN-16 | As a loan approval officer, I can get a decision followed by some details for the decision when I feed an application for loan prediction. | 15 | High | CHIDAMBARAM A  VIGNESH V |
| Sprint-4 | Register | USN-17 | As an admin, I should be able to register myself as one using unique email and password. | 2 | Medium | DHANUSH S  BARATHKUMAR G  CHIDAMBARAM A  VIGNESH V |
| Sprint-4 | Login | USN-18 | As an admin I should be able to login myself as one using unique email and password. | 2 | Medium | DHANUSH S  VIGNESH V |
| Sprint-4 | Dashboard | USN-19 | As a admin, I should be able to access the dashboard with everything I am allowed to use. | 2 | Medium | DHANUSH S  BARATHKUMAR G |

* 1. **Sprint Delivery Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points**  **Completed (as on**  **Planned End Date)** | **Sprint Release Date (Actual)** |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 28 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 10 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 25 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 6 | 19 Nov 2022 |

* 1. **Reports from JIRA**

****

****

1. **CODING & SOLUTIONING**
   1. **GUI using Flask**

from flask import Flask, render\_template,request

import numpy as np

import pandas

import pickle

import requests

# NOTE: you must manually set API\_KEY below using information retrieved from your IBM Cloud account.

API\_KEY = "gty1PYR\_T522sN6\_r51HL2g88kxNxhyQXGVp5uPGmGFC"

token\_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":

API\_KEY, "grant\_type": 'urn:ibm:params:oauth:grant-type:apikey'})

mltoken = token\_response.json()["access\_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

app = Flask(\_\_name\_\_)

model = pickle.load(open(r'rdf.pkl','rb'))

@app.route("/", methods=['GET', 'POST'])

def home():

return render\_template("home.html")

@app.route("/predict",methods=['POST','GET'])

def predict():

if request.method == 'POST':

project\_name=request.form['full-name']

print(project\_name)

return render\_template("predict.html",project\_name=project\_name)

@app.route("/success",methods=['POST','GET'])

def evaluate():

input\_feature = [int(x) for x in request.form.values()]

print(input\_feature)

# input\_feature=[np.array(input\_feature)]

print(input\_feature)

names = ['Gender', 'Married', 'Dependents', 'Education', 'Self Employed', 'Applicant Income', 'Coapplicant Income', 'Loan Amount', 'Loan\_Amount\_Term', 'Credit\_History', 'Property\_Area']

# NOTE: manually define and pass the array(s) of values to be scored in the next line

payload\_scoring = {"input\_data": [{"fields": [names],

"values": [input\_feature]}]}

response\_scoring = requests.post(

'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/a05131f3-dcb8-46cd-bf08-1c2ecf28cc86/predictions?version=2022-11-13',

json=payload\_scoring,

headers={'Authorization': 'Bearer ' + mltoken})

predictions = response\_scoring.json()

prediction = predictions['predictions'][0]['values'][0][0]

print("Scoring response")

print(response\_scoring.json())

print(prediction)

# data = pandas.DataFrame(input\_feature, columns=names)

# print(data)

# prediction=model.predict(data)

# print(prediction)

# prediction = int(prediction)

# print(type(prediction))

loan=1

if (prediction == 0):

loan=0

return render\_template("success.html",result = "Loan will Not be Approved",loan=loan)

else:

return render\_template("success.html",result = "Loan will be Approved",loan=loan)

return render\_template("success.html")

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

* 1. **Loan Prediction Model**

*# Authors: DHANUSH S, BARATHKUMAR G, CHIDAMBARAM, VIGNESH S*

**import** os**,** types

**import** pandas **as** pd

**from** botocore.client **import** Config

**import** ibm\_boto3

**def** \_\_iter\_\_(self): **return** 0

*# @hidden\_cell*

*# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.*

*# You might want to remove those credentials before you share the notebook.*

client\_5158bfd5065b40c4b6cf7e02a60cf879 **=** ibm\_boto3**.**client(service\_name**=**'s3',

ibm\_api\_key\_id**=**'Rob46tTNo97O\_Wdw9cPUe7whW\_akOBfAuD9qWugyZBTB',

ibm\_auth\_endpoint**=**"https://iam.cloud.ibm.com/oidc/token",

config**=**Config(signature\_version**=**'oauth'),

endpoint\_url**=**'https://s3.private.us.cloud-object-storage.appdomain.cloud')

body **=** client\_5158bfd5065b40c4b6cf7e02a60cf879**.**get\_object(Bucket**=**'ibmsmartlender-donotdelete-pr-fn1gcvrcmxp1mg',Key**=**'test.csv')['Body']

*# add missing \_\_iter\_\_ method, so pandas accepts body as file-like object*

**if** **not** hasattr(body, "\_\_iter\_\_"): body**.**\_\_iter\_\_ **=** types**.**MethodType( \_\_iter\_\_, body )

test **=** pd**.**read\_csv(body)

test**.**head()

body **=** client\_5158bfd5065b40c4b6cf7e02a60cf879**.**get\_object(Bucket**=**'ibmsmartlender-donotdelete-pr-fn1gcvrcmxp1mg',Key**=**'train.csv')['Body']

*# add missing \_\_iter\_\_ method, so pandas accepts body as file-like object*

**if** **not** hasattr(body, "\_\_iter\_\_"): body**.**\_\_iter\_\_ **=** types**.**MethodType( \_\_iter\_\_, body )

train**=** pd**.**read\_csv(body)

train**.**head()

**import** pandas **as** pd

**import** numpy **as** np

**from** sklearn.preprocessing **import** MaxAbsScaler

**from** sklearn.tree **import** DecisionTreeClassifier

**from** sklearn.ensemble **import** RandomForestClassifier

**from** sklearn.neighbors **import** KNeighborsClassifier

**from** sklearn.ensemble **import** GradientBoostingClassifier

**from** sklearn.metrics **import** confusion\_matrix

**from** sklearn.metrics **import** classification\_report

**from** sklearn.model\_selection **import** cross\_val\_score

**from** sklearn.metrics **import** f1\_score

**import** pickle

scaler **=** MaxAbsScaler()

train\_y **=** train**.**iloc[:,**-**1]

train\_x **=** train**.**drop('Loan\_Status',axis**=**1)

test\_y **=** test**.**iloc[:,**-**1]

test\_x **=** test**.**drop('Loan\_Status',axis**=**1)

x **=** pd**.**concat([train\_x,test\_x],axis**=**0)

y **=** pd**.**concat([train\_y,test\_y],axis**=**0)

train\_x **=** scaler**.**fit\_transform(train\_x)

test\_x **=** scaler**.**transform(test\_x)

dt **=** DecisionTreeClassifier()

dt**.**fit(train\_x,train\_y)

y\_pred **=** dt**.**predict(test\_x)

print("\*\*\*\* Decision Tree Classifier \*\*\*\*")

print('Confusion Matrix')

print(confusion\_matrix(test\_y,y\_pred))

print('Classification Report')

print(classification\_report(test\_y,y\_pred))

pip install ibm\_watson\_machine\_learning

**from** ibm\_watson\_machine\_learning **import** APIClient

**import** json

wml\_credentials **=** {

"apikey" : "gty1PYR\_T522sN6\_r51HL2g88kxNxhyQXGVp5uPGmGFC",

"url" : "https://us-south.ml.cloud.ibm.com"

}

wml\_client **=** APIClient(wml\_credentials)

wml\_client**.**spaces**.**list()

SPACE\_ID **=** "bfdc6002-40bf-49da-b75a-6a88c850d1b7"

wml\_client**.**set**.**default\_space(SPACE\_ID)

wml\_client**.**software\_specifications**.**list(100)

**import** sklearn

sklearn**.**\_\_version\_\_

MODEL\_NAME **=** 'Model\_building\_SL'

DEPLOYMENT\_NAME **=** 'loan-prediction'

DEMO\_MODEL **=** dt

software\_spec\_uid **=** wml\_client**.**software\_specifications**.**get\_id\_by\_name('runtime-22.1-py3.9')

model\_props **=** {

wml\_client**.**repository**.**ModelMetaNames**.**NAME: MODEL\_NAME,

wml\_client**.**repository**.**ModelMetaNames**.**TYPE: 'scikit-learn\_1.0',

wml\_client**.**repository**.**ModelMetaNames**.**SOFTWARE\_SPEC\_UID: software\_spec\_uid

}

model\_details **=** wml\_client**.**repository**.**store\_model(

model**=**DEMO\_MODEL,

meta\_props**=**model\_props,

training\_data**=**train\_x,

training\_target**=**train\_y

)

model\_details

model\_id **=** wml\_client**.**repository**.**get\_model\_id(model\_details)

model\_id

deployment\_props **=** {

wml\_client**.**deployments**.**ConfigurationMetaNames**.**NAME:DEPLOYMENT\_NAME,

wml\_client**.**deployments**.**ConfigurationMetaNames**.**ONLINE: {}

}

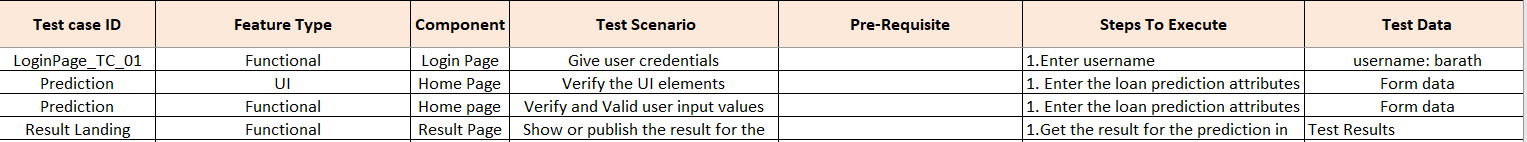
deployment **=** wml\_client**.**deployments**.**create(

artifact\_uid**=**model\_id,

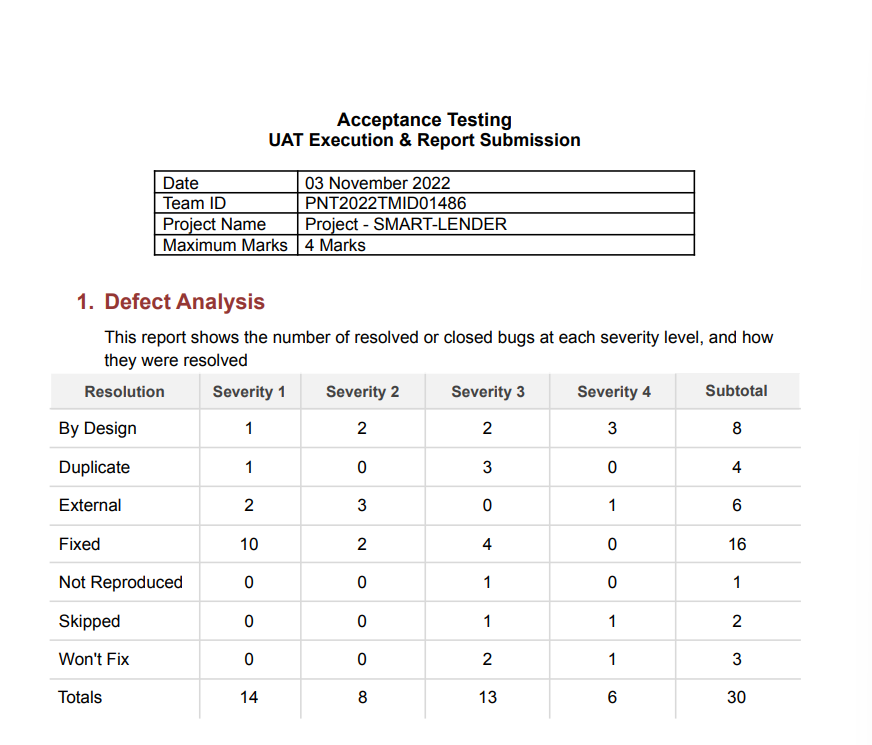
meta\_props**=**deployment\_props

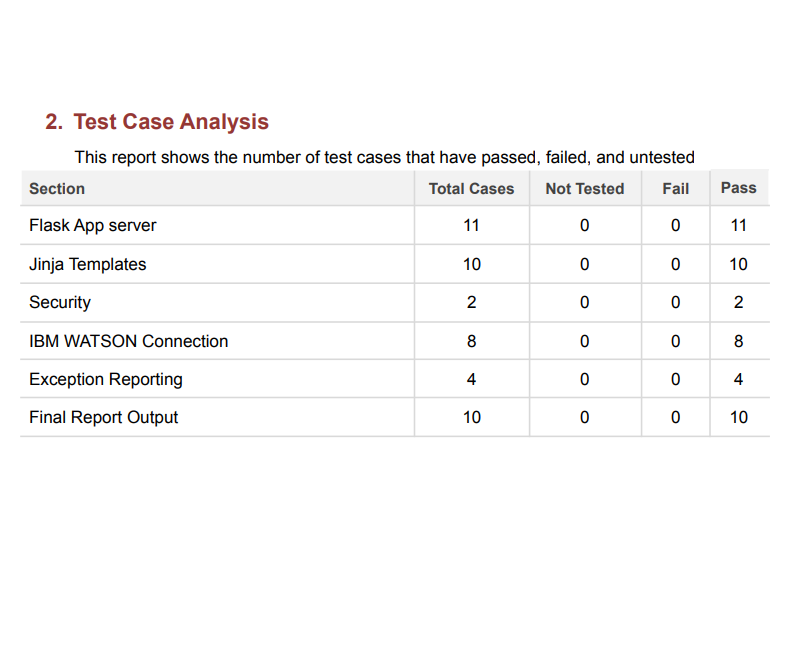
)

1. **TESTING** 
   1. Test Cases

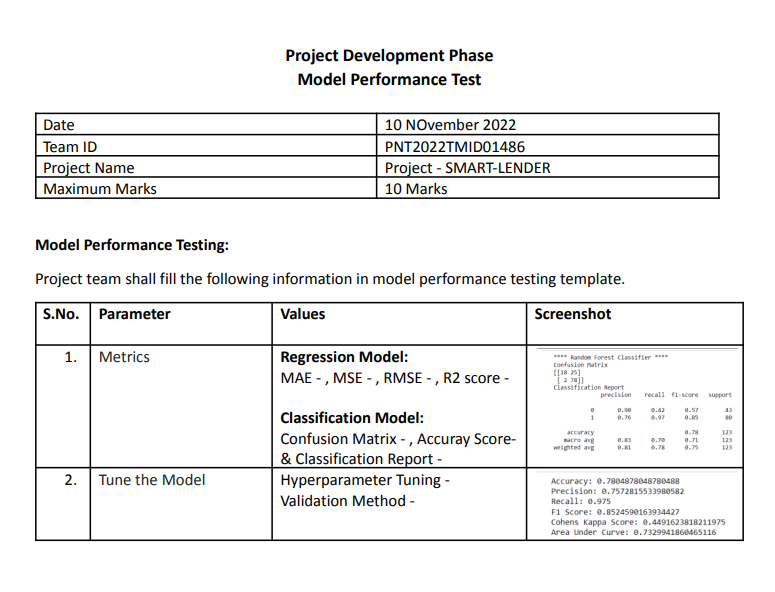


* 1. User Acceptance Testing





1. **RESULTS**
   1. Performance Metrics



1. **ADVANTAGES & DISADVANTAGES**

**Pros:-**

* Provide a better user experience to improve the speed and accuracy of loan applications
* Process a larger number of loan applications with existing resources
* Eliminate sources of human error for faster, better-quality evaluation decisions
* Establish predictable, repeatable, and auditable processes that support compliance
* Reduce delays and costs associated with paper processes
* Analyze process and loan performance with the goal of continually improving efficiency and profitability

1. **CONCLUSION**

Smart Lender offers all the operational efficient processes involved in the end-to-end Credit Loan Origination, Assessment and Management. Considering the labor-intensive task of Lending, Commercial Loan Origination is made easy with SmartLender driven by best market practice processes.

1. **FUTURE SCOPE**

The future seems to be highly promising. Few years down the lane, banks will be serving fewer clients physically while having a deeper relationship with them. Robots will serve as a means to store data and could work alongside humans which will help them to work more efficiently (Jeet, 2015). There will be robot advisors in the future banks which will help people to make correct financial decisions and prevent them from making unsound decisions. The future banking may be completely replaced by platforms run by robots. As mentioned in the article “Robotics in Banking, 2015”, the Robotic Process Automation or what Andrew Burgess, an outsourcing advisor, calls “robotic software agents”, does not depend on arms and legs but has a powerful impact when it comes to repetitive processes that are rules-based and frequent. Banks can entertain queries all through the day and night using this Robotic Process Automation, which will in effect prove to be considerably economic than remunerating an employee.

1. **APPENDIX**

GitHub Link : https://github.com/IBM-EPBL/IBM-Project-18475-1659685844

Project Demo Link : https://youtu.be/6VV9D7djwds