A PROJECT REPORT

PREDICTING PERSONAL LOAN APPROVAL USING MACHINE LEARNING

Team ID	:	NM2023TMID17318
Team Leader	:	SANTHIYA C
Team members	:	RAMKUMAR T
	:	RATHINI R
	:	SABARISH N

INDEX

PREDICTING PERSONAL LOAN APPROVAL USING MACHINE LEARNING

1. INTRODUCTION

1.1 OVERVIEW

Predicting personal loan approval using machine learning is an application of data analytics that helps financial institutions automate their lending process by utilizing algorithms to predict the probability of loan approval based on various factors related to the applicant's creditworthiness. The use of machine learning in this context involves training a model on a historical dataset of loan applications and outcomes, which is used to predict the likelihood of future loan applications being approved or rejected.

The model is typically trained on a variety of factors, such as credit score, employment history, income, debt-to-income ratio, loan amount, loan purpose, and other demographic and financial information. The model then uses this data to assign a probability score to each loan application, which can be used to make lending decisions.

The benefits of using machine learning for personal loan approval include increased efficiency, improved accuracy, and reduced bias.

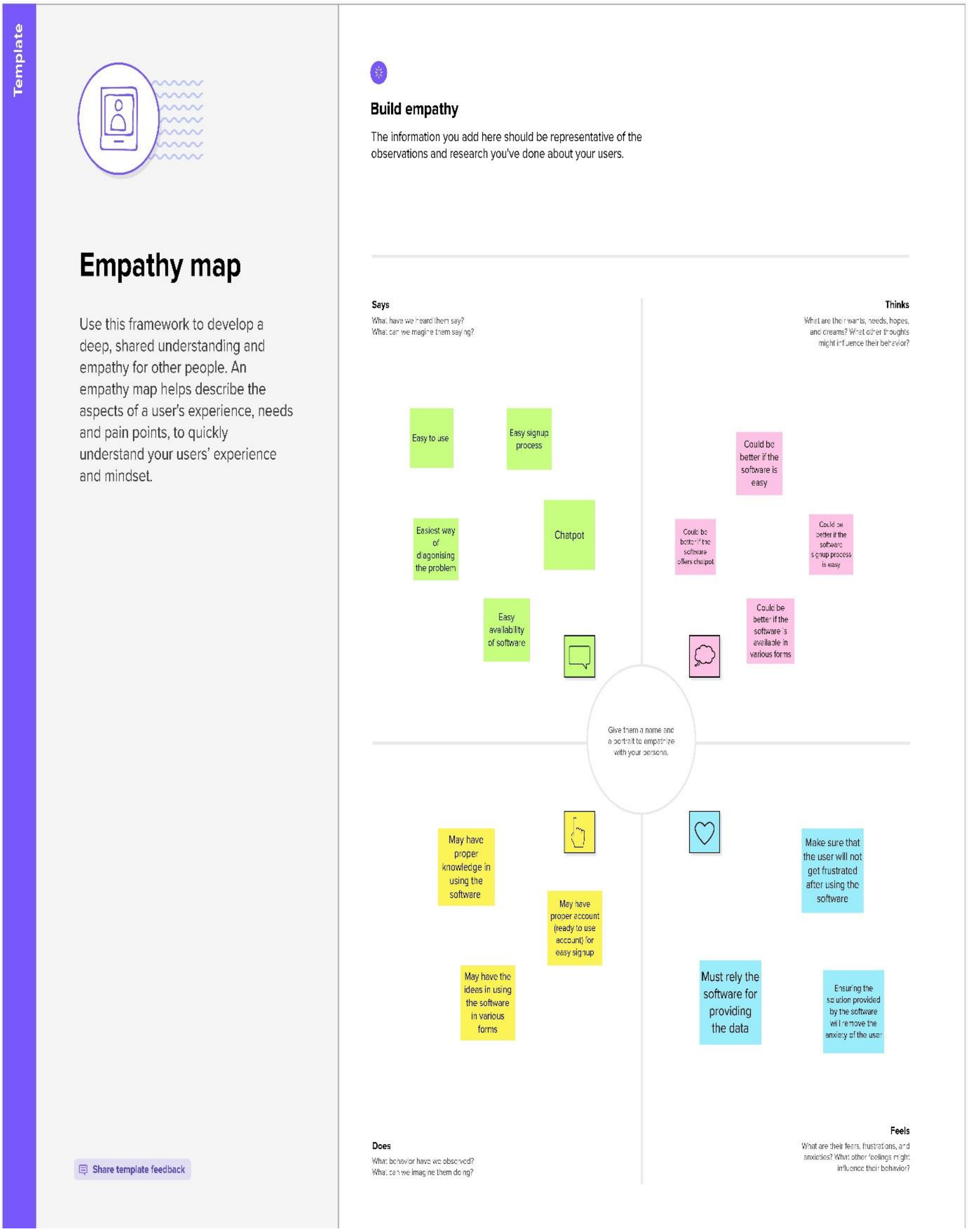
1.2 PURPOSE

The purpose of predicting personal loan approval using machine learning is to help financial institutions automate their lending process and make more informed and accurate decisions about loan approval. The use of machine learning algorithms can help financial institutions to:

1. Improve efficiency: Automating the lending process can help financial institutions save time and reduce costs associated with manually reviewing loan applications.
2. Reduce bias: By using objective data-driven models, machine learning can help to reduce human biases that may be present in the lending process.
3. Improve accuracy: Machine learning models can analyze vast amounts of data and identify patterns that may not be easily discernible to humans, which can lead to more accurate predictions about loan approval.
4. Mitigate risk: By using predictive models to evaluate loan applications, financial institutions can better assess the creditworthiness of applicants and reduce the risk of default.
5. Increase profitability: By making more accurate lending decisions, financial institutions can reduce losses associated with defaults and increase profitability

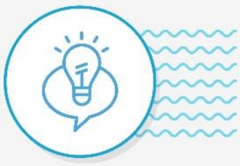
2.Problem Definition & Design Thinking

2.1 Empathy Map



2.2 IDEATION AND BRAINSTORMING

Template



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Person 1 (SANTHIYA C)

- Problem requirement analysis and planning
- Defining the problem
- Analysing the problem
- Study about the existing system and define the proposed system
- Gattering requirement for the proposed system
- Define project modules
- Aallotment of various modules to all the team members
- Planing the timeline
- Final group discussen about the invidual modules

Person 2 (RAMKUMAR T)

- project design
- designing module 1
- designing module 2
- designing module 3
- designing module 4
- designing module 5
- designing module 6
- designing module 7
- compined all the design

Person 3 (RATHINI R)

- Implementing the software
- implementation of module 1
- implementation of module 2
- implementation of module 3
- implementation of module 4
- implementation of module 5
- impeimentation of module 6
- implementation of module 7
- combining the implementing modules

Person 4 (SABARISH N)

- Testing the software
- perform unit testing
- make correction happen in testing unit testing
- perform integration testing
- make correction happen in testing integration testing
- perform system testing
- make correction happen in system testing
- perform acceptance testing
- make correction in happend inacceptance testing

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Predicting personal loan approval using machine learning involves developing a model that can analyze a set of data related to loan applicants can determine the likehood of approval.

One of the biuggest challenges is collecting the right data.You will need a variety of data points,such as credit score,income,employment status,loan amount,and purpose of the loan

There is a risk of bias in the data,which can lead to biased predictions. For example,if the data is biased towards certain demographics or geographic regions,the model may not be able to accurately predict loan approval for other groups.

Overall ,predicting personal loan approval using machine learning is a complex task that requires careful consideration of data,models and ethical consideration.

The lender can use the debt-to-income ration to analyze how much debt the applicant has in comparison to their income.This can help predict whether they will be able to afford the loan repayment.

Certain demographic factors,such as age,gender and location can also be used to predict loan approval.

The lender can use the debt-to-income ration to analyze how much debt the applicant has in comparison to their income.This can help predict whether they will be able to afford the loan repayment.

The lender can also assess the applicants income and employment status to determine if they have a stable source of income and can afford to repay the loan.

Develop a credit scoring model that uses an applicant credit score, payment history and other financial data to predict loan approval.The model can be trained on historical loan data to improve accuracy.

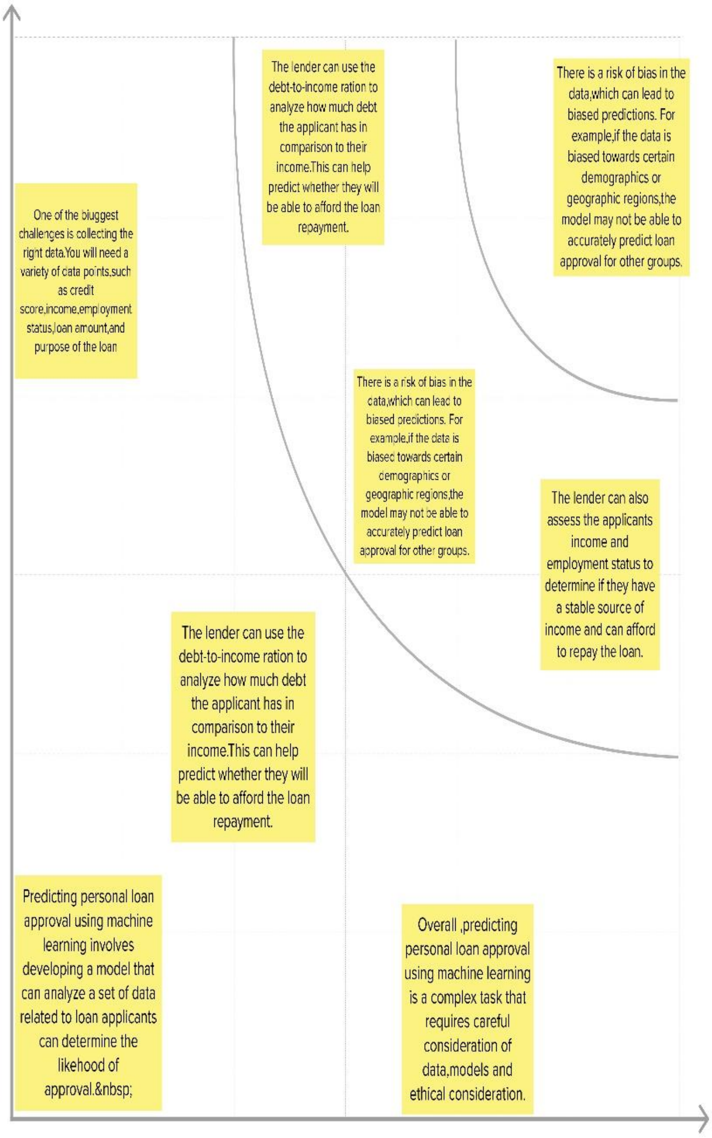
Prioritize

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

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.



Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

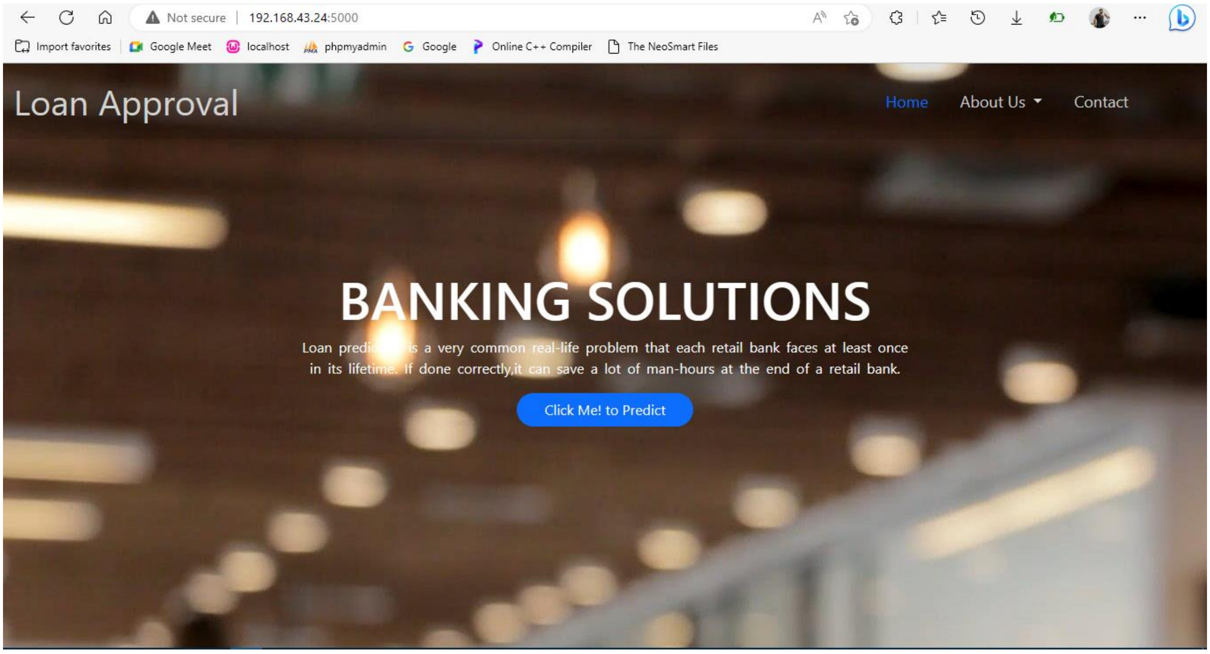
Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

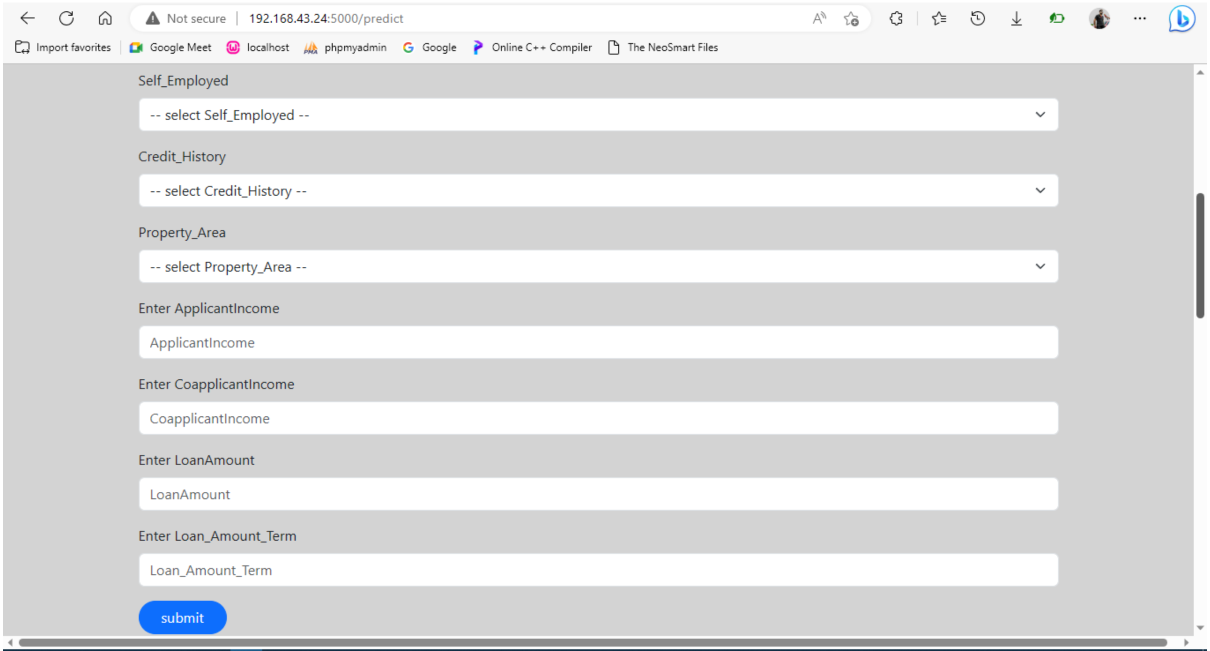
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

Result

Output:





ADVANTAGES & DISADVANTAGES

Advantages:

- Faster Processing: Machine learning algorithms can process large amounts of data quickly and accurately. This can significantly reduce the time taken to process loan applications, making the process faster and more efficient.
- Consistency: Machine learning algorithms are not influenced by emotions or biases, unlike human loan officers. This ensures that all loan applications are evaluated based on the same set of criteria, which can lead to more consistent and objective lending decisions.
- Improved Accuracy: Machine learning algorithms can analyze large amounts of data to identify patterns and trends that may not be visible to humans. This can lead to more accurate predictions of loan default and better risk assessment.
- Increased Access: Machine learning algorithms can be used to evaluate loan applications from anywhere in the world, which can increase access to credit for people in remote areas.

Disadvantages:

- Data Quality: The accuracy of machine learning algorithms depends on the quality of the data used to train them. If the data is incomplete, inaccurate, or biased, the algorithms may not make accurate predictions.
- Complexity: Machine learning algorithms can be complex and difficult to understand. This can make it challenging for loan officers and borrowers to understand why a loan application was approved or rejected.
- Overreliance on Data: Machine learning algorithms are only as good as the data they are trained on. If the data does not reflect the current economic or social conditions, the algorithms may not make accurate predictions.
- Lack of Transparency: Machine learning algorithms are often considered "black boxes" because it can be difficult to understand how they arrive at their predictions. This lack of transparency can lead to mistrust and suspicion, especially if the predictions are inaccurate or biased.

4. APPLICATION

Application of Predicting Personal Loan:

Predicting the likelihood of a borrower's ability to repay a personal loan is an important task for lenders to minimize risk and maximize profits. Machine learning techniques can be applied to this task, as they can analyze vast amounts of data and identify patterns that traditional statistical methods might miss.

Here are some steps involved in predicting personal loan using machine learning:

- Data Collection: Collecting data related to personal loan borrowers, including their demographic information, credit scores, employment history, income, and other relevant financial data.
- Data Preprocessing: Cleaning and preprocessing the data to eliminate missing or irrelevant data, transforming the data into numerical format, and scaling it if necessary.
- Feature Selection: Identifying the most relevant features that impact the borrower's ability to repay the loan.
- Model Selection: Selecting a machine learning model that can best predict whether a borrower will repay the loan or not. Some popular models used for this task include logistic regression, decision trees, random forest, and gradient boosting.
- Training the Model: Using the preprocessed data to train the machine learning model.
- Testing the Model: Evaluating the performance of the trained model using a testing dataset that was not used during training.
- Deployment: Finally, deploying the model in the production environment for use in predicting the likelihood of repayment for new loan applicants.
- Overall, the use of machine learning techniques for predicting personal loans can help lenders minimize their risk and make more informed decisions.

5.CONCLUSION

Conclusion

In conclusion, the use of machine learning models in predicting personal loan approval is a promising approach that can help lenders make better and faster decisions. By leveraging job prediction models and other relevant features such as income and credit score, machine learning models can accurately predict the likelihood of loan approval for new applicants. This can help lenders reduce the risk of default and improve their overall lending performance.

APPENDIX

Source Code

1. app.py(Source Code)

```
import os

from flask import Flask, render_template, request

import pickle

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


@app.route('/')
def home():
    return render_template('home.html')

@app.route('/predict')
def predict():
    return render_template('predict.html')

@app.route('/submit',methods=["POST", "GET"])
def submit():
    input_feature=[int(x) for x in request.form.values() ]
    input_feature = [np.array(input_feature)]
    print(input_feature)
    names = ['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'ApplicantIncome',
'CoapplicantIncome' 'LoanAmount', 'Loan_Amount_Term', 'Credit_History', 'Property_Area']
```

```
    data = pandas.DataFrame(input_feature,columns=names)
    print(data)

    prediction=model.predict(data)
    print(prediction)
    prediction = int(prediction)

    if(prediction == 0):
        return render_template("output.html",result = "Loan will not be Approved")
    else:
        return render_template("output.html", result="Loan will be Approved")

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000,debug=True)
    port=int(os.environ.get('PORT',5000))
    #app.run(dubug=True)
```

2.index.html

```
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title></title>

          <!-- offline bootstrap-->

  <link rel="stylesheet" type="text/css" href="bootstrap\css\bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="style.css">
```

```
<!-- online cdn bootsrap-->


 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">

<style type="text/css">

	.navbar{

	background-color: #0000001f;

}


.navbar-nav .nav-link.active, .navbar-nav .show>.nav-link {

   color: white;

}


.texts-primary, .nav-link{

	color: lightgray;

}


.menu{

	color: black;

}


.home-middle{

	color: white;

	text-align: center;

}


.home-middle h1{

	font-size: 60px;

}


.home-middle p{

	word-spacing:3px;

}
```

```css
.btn {
        width: 200px;
        border-radius: 100px;
}


body{
  background-repeat: no-repeat;
}

.navbar-nav{
  margin-left: 700px;
}

.nav-item{
  padding:0 10px 0 10px;
  font-size: 18px;
}

.navbar-brand{
  font-size: 40px;
}

.navbar-nav .nav-link.active, .navbar-nav .show>.nav-link {
    color: #0d6efd;
}


.frm{
  background-color: lightgrey;
}

.left img{
```

```
  width: 550px;
  padding: 80px 0 0 40px;
}

.txt{
  padding: 100px;
}

.blue{
  color: blue;
}

section#contacts{
  background-color: lightgrey;
  padding: 30px 0 30px 0;
}

.contacts div{
  padding: 30px 10px 10px 10px ;
}

#gallery img{
  width: 200px;
  padding: 5px;
}

</style>

</head>
<body background="https://www.walldevil.co/wallpapers/p14/indoors-people-office-lights-blurred-corporate-blurry.jpg">

<nav class="navbar navbar-expand-lg ">
  <div class="container-fluid">
```

```
    <a class="texts-primary navbar-brand " href="#">Loan Approval</a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">

      <ul class="navbar-nav me-auto mb-2 mb-lg-0">

        <li class="nav-item">

          <a class="nav-link active" aria-current="page" href="/">Home</a>

        </li>


        <li class="nav-item dropdown">

          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-
expanded="false">

            About Us

          </a>

          <ul class="dropdown-menu ">

            <li><a class="dropdown-item" href="/predict">About</a></li>

            <li><a class="dropdown-item" href="/predict ">Gallery</a></li>

          </ul>

        </li>

        <li class="nav-item">

          <a class="nav-link " href="/predict">Contact</a>

        </li>

      </ul>


    </div>

  </div>

</nav>




<div class="home-middle">

             <br><br><br><br><br><br>

<h1>BANKING SOLUTIONS</h1>
```

```
<p>Loan prediction is a very common real-life problem that each retail bank faces at least once

<br> in its lifetime. If done correctly,it can save a lot of man-hours at the end of a retail bank.</p>

<a href="/predict"><button type="predict" class="btn btn-primary">Click Me! to
Predict</button></a>

</div>

 <!-- script files-->



  <script type="text/javascript" src="bootstrap\js\bootstrap.bundle.min.js"></script>


  <!-- online cdn bootsrap script -->


 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
w76AqPfDkMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwxhTBTkF7CXvN"
crossorigin="anonymous"></script>


</body>

</html>
```

3.predict.html

```
 <!DOCTYPE html>
 <html>
 <head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <title></title>

     <!-- offline bootstrap-->

   <link rel="stylesheet" type="text/css" href="bootstrap\css\bootstrap.min.css">
   <link rel="stylesheet" type="text/css" href="style.css">

 <!-- online cdn bootsrap-->

 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
  rel="stylesheet" integrity="sha384-
  GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD"
  crossorigin="anonymous">
```

```
<style type="text/css">

  body{
  background-repeat: no-repeat;
}

.navbar-nav{
  margin-left: 700px;
}

.nav-item{
  padding:0 10px 0 10px;
  font-size: 18px;
}

.navbar-brand{
  font-size: 40px;
}

.navbar-nav .nav-link.active, .navbar-nav .show>.nav-link {
    color: #0d6efd;
}
.btn {
  width: 100px;
  border-radius: 100px;
}

.navbar{
  background-color: white;
}

.frm{
  background-color: lightgrey;
}

.left img{
  width: 550px;
  padding: 80px 0 0 40px;
}

.txt{
  padding: 100px;
}

.blue{
  color: blue;
}

section#contacts{
  background-color: lightgrey;
  padding: 30px 0 30px 0;
}
```

```
.contacts div{
  padding: 30px 10px 10px 10px ;
}

#gallery img{
  width: 200px;
  padding: 5px;
}

</style>

</head>
<body>

<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="text-primary navbar-brand " href="#">Loan Approval</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link " aria-current="page" href="/">Home</a>
        </li>

        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-
expanded="false">
            About Us
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="/predict">About</a></li>
            <li><a class="dropdown-item" href="/predict">Gallery</a></li>
          </ul>
        </li>
        <li class="nav-item">
          <a class="nav-link " href="/predict">Contact</a>
        </li>
      </ul>

    </div>
  </div>
</nav>


<section class="text-gray-600 body-font frm">
  <div class="container px-5 py-24 mx-auto">
    <div class="flex flex-col text-center w-full mb-20">
```

```html
    <h1 class="sm:text-3xl text-2xl font-medium title-font mb-4 text-gray-900 text-primary">Loan
Approval Prediction Form</h1>
    <p class="lg:w-2/3 mx-auto leading-relaxed text-base">fill the form for prediction</p>
    <br><br>
   </div>
   <div>

   </div>
  <!-- <a class="btn btn-primary" href="./" role="button">Back</a> -->
<form action='/predict' method='POST'>
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label"> gender</label>
  <select class="form-select" id="gender" name="gender" aria-label="Default select example">
  <option selected>-- select gender --</option>
  <option value="Male">Male</option>
  <option value="Female">Female</option>
</select>
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label"> married status</label>
  <select class="form-select" id="married" name="married" aria-label="Default select example">
  <option selected>-- select married status --</option>
  <option value="Yes">Yes</option>
  <option value="No">No</option>
</select>
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Dependents</label>
  <select class="form-select" id="dependents" name="dependents" aria-label="Default select
example">
  <option selected>-- select dependents --</option>
  <option value="0">0</option>
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="3+">3+</option>
</select>
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Education</label>
  <select class="form-select" id="education" name="education" aria-label="Default select example">
  <option selected>-- select education --</option>
  <option value="Graduate">Graduate</option>
  <option value="Not Graduate">Not Graduate</option>
</select>
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Self_Employed</label>
  <select class="form-select" id="employed" name="employed" aria-label="Default select
example">
```

```
  <option selected>-- select Self_Employed --</option>
  <option value="Yes">Yes</option>
  <option value="No">No</option>
</select>
</div>
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Credit_History</label>
  <select class="form-select" id="credit" name="credit" aria-label="Default select example">
  <option selected>-- select Credit_History --</option>
  <option value="1.000000">1.000000</option>
  <option value="0.000000">0.000000</option>
  <option value="0.842199">0.842199</option>
</select>
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Property_Area</label>
  <select class="form-select" id="area" name="area" aria-label="Default select example">
  <option selected>-- select Property_Area --</option>
  <option value="Semiurban">Semiurban</option>
  <option value="Urban">Urban</option>
  <option value="Rural">Rural</option>
</select>
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Enter ApplicantIncome</label>
  <input type="text" class="form-control" id="ApplicantIncome" name="ApplicantIncome"
 placeholder="ApplicantIncome">
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Enter CoapplicantIncome</label>
  <input type="text" class="form-control" id="CoapplicantIncome" name="CoapplicantIncome"
 placeholder="CoapplicantIncome">
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Enter LoanAmount</label>
  <input type="text" class="form-control" id="LoanAmount" name="LoanAmount"
 placeholder="LoanAmount">
</div>

<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">Enter Loan_Amount_Term</label>
  <input type="text" class="form-control" id="Loan_Amount_Term" name="Loan_Amount_Term"
 placeholder="Loan_Amount_Term">
</div>


    <button type="submit" class="btn btn-primary" value="predict">submit</button><br><br><br>
    </form>
```

```
  </div>
</section>

<br><br>

<section class="Abouts" id="Abouts">
  <h1  class="blue" align="center">About Us</h1>
<div class="row">
  <div class="col-lg-6 left ">
    <img src="https://alphacon-group.com/wp-content/uploads/2021/02/CRE_3-768x512.jpg">
  </div>

  <div class="col-lg-6 right">
    <div class="txt">
    <h4>Loan Approval How it works ?</h4>
    <p>Credit Information Bureau India Limited (CIBIL) score plays a critical role in the loan
 approval process for Indian banking industry. An individual customer's credit score provides loan
 providers with an indication of how likely it is that they will pay back a loan based on their
 respective credit history. This article is an attempt to discuss basics Loan Approval Process and
 working principles of CIBIL score in indian finance industry keeping a view of individual customer
 benefits. </p>
    <br>
    <a href="#" class="text-primary">Learn More</a>
    </div>
  </div>
</div>
</section>


<br><br>


<section class="contacts" id="contacts">

<div class="col-lg-12">
    <h1  class="blue" align="center">Contact Us</h1>
  </div>

<div class="container">
 <div class="row">

  <div class="col-lg-4">
    <p class="blue">6th Floor, Technical Block, Madthava Reddy Colony, Gachibowli, Hydrabad,
 Telangana, Warangal - 500032</p>
  </div>

  <div class="col-lg-4">
    <p align="center" class="blue">+91 6304320044</p>
  </div>

  <div class="col-lg-4">
    <p align="center" class="blue">info@thesmartbridge.com</p>
```

```
  </div>

 </div>
</div>

</section>

<br><br><br><br>

<section class="gallery" id="gallery">

  <div class="col-lg-12">
    <h1  class="blue" align="center">Gallery</h1>
  </div>

  <br><br>

  <div class="container">
 <div class="row">

  <div class="col-lg-3">


    <img
 src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&ri
 u=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-
 content%2fuploads%2f2020%2f04%2fcreative-office-
 environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl
 =&pid=ImgRaw&r=0">


    <img
 src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&ri
 u=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-
 content%2fuploads%2f2020%2f04%2fcreative-office-
 environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl
 =&pid=ImgRaw&r=0">


    <img
 src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&ri
 u=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-
 content%2fuploads%2f2020%2f04%2fcreative-office-
 environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl
 =&pid=ImgRaw&r=0">
  </div>

  <div class="col-lg-3">

    <img
 src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&ri
 u=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-
 content%2fuploads%2f2020%2f04%2fcreative-office-
```

environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl=&pid=ImgRaw&r=0">

<img src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&riu=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-content%2fuploads%2f2020%2f04%2fcreative-office-environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl=&pid=ImgRaw&r=0">

<img src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&riu=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-content%2fuploads%2f2020%2f04%2fcreative-office-environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl=&pid=ImgRaw&r=0">
</div>

<div class="col-lg-3">
<img src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&riu=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-content%2fuploads%2f2020%2f04%2fcreative-office-environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl=&pid=ImgRaw&r=0">

<img src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&riu=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-content%2fuploads%2f2020%2f04%2fcreative-office-environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl=&pid=ImgRaw&r=0">

<img src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&riu=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-content%2fuploads%2f2020%2f04%2fcreative-office-environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl=&pid=ImgRaw&r=0">
</div>

<div class="col-lg-3">

<img src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&riu=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-

```
 content%2fuploads%2f2020%2f04%2fcreative-office-
 environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl
 =&pid=ImgRaw&r=0">


<img
 src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&ri
 u=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-
 content%2fuploads%2f2020%2f04%2fcreative-office-
 environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl
 =&pid=ImgRaw&r=0">


  <img
 src="https://th.bing.com/th/id/R.b5b3483c3f28cc4cd25eb1173d7a75a8?rik=Fr62sWdbCZvO7A&ri
 u=http%3a%2f%2fwww.workspacesolutions.com%2fblog%2fwp-
 content%2fuploads%2f2020%2f04%2fcreative-office-
 environments.jpeg&ehk=InVbo%2bYHl35sPGXV36qI9rme2xcQY%2f%2bs9cQesa2jG5I%3d&risl
 =&pid=ImgRaw&r=0 ">
  </div>



 </div>
</div>
</section>

 <!-- script files-->


  <script type="text/javascript" src="bootstrap\js\bootstrap.bundle.min.js"></script>

  <!-- online cdn bootsrap script -->

 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"
 integrity="sha384-
 w76AqPfDkMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwxhTBTkF7CXvN"
 crossorigin="anonymous"></script>

</body>
</html>
```

5.PLACEMENT_PREDICTION.ipynb


```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier
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
```

```
data = pd.read_csv('loan_prediction.csv')
data
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmou
0	LP001002	Male	No	0	Graduate	No	5849	0.0	Na
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141
...	...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133

614 rows × 13 columns

```
print(data.isnull().sum())
```

```
data.info()
```

```
Loan_ID              0
Gender              13
Married              3
Dependents          15
Education            0
Self_Employed       32
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount          22
Loan_Amount_Term    14
Credit_History      50
Property_Area        0
Loan_Status          0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
```

```
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
data.isnull().sum()
```

```
Loan_ID               0
Gender               13
Married               3
Dependents           15
Education             0
Self_Employed        32
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Property_Area         0
Loan_Status           0
dtype: int64
```

```
# replace missing values in Gender column with mode
data['Loan_ID'] = data['Loan_ID'].fillna(data['Loan_ID'].mode()[0])

# replace missing values in Gender column with mode
data['Gender'] = data['Gender'].fillna(data['Gender'].mode()[0])

# replace missing values in Married column with mode
data['Married'] = data['Married'].fillna(data['Married'].mode()[0])

# replace missing values in Dependents column with mode
data['Dependents'] = data['Dependents'].fillna(data['Dependents'].mode()[0])

# replace missing values in Self_Employed column with mode
data['Self_Employed'] = data['Self_Employed'].fillna(data['Self_Employed'].mode()[0])

# replace missing values in LoanAmount column with mean
data['LoanAmount'] = data['LoanAmount'].fillna(data['LoanAmount'].mean())

# replace missing values in Loan_Amount_Term column with mode
data['Loan_Amount_Term'] = data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].mode()[0])

# replace missing values in Credit_History column with mode
data['Credit_History'] = data['Credit_History'].fillna(data['Credit_History'].mode()[0])
```

```
from sklearn.preprocessing import LabelEncoder

# Create an instance of LabelEncoder
label_encoder = LabelEncoder()

# Iterate through each categorical column in the dataframe
for col in data.columns:
    if data[col].dtype == 'object':  # Check if column is of object (string) data type
        data[col] = label_encoder.fit_transform(data[col].astype(str))  # Convert column to string and apply label encod

# Note: You may need to convert columns to string data type before applying label encoding to avoid potential issues wit
```

```
from imblearn.over_sampling import SMOTE

y = data['Loan_Status']
X = data.drop(columns=['Loan_Status'], axis=1)
```

```
# Perform SMOTE for oversampling
smote = SMOTE(random_state=42)
x_bal, y_bal = smote.fit_resample(X, y)
```

```
print(y.value_counts())
y_bal = pd.Series(y_bal)
print(y_bal.value_counts())
```

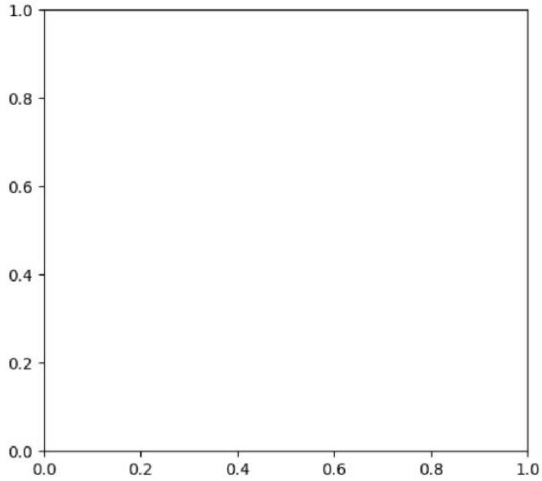
```
1    422
0    192
Name: Loan_Status, dtype: int64
1    422
0    422
Name: Loan_Status, dtype: int64
```

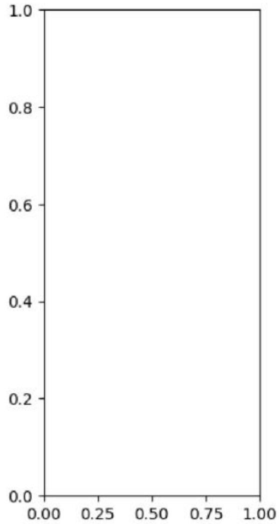
```
data.describe()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
count	614.000000	614.000000	614.000000	614.000000	614.000000	614.000000	614.000000	614.000000
mean	306.500000	0.817590	0.653094	0.744300	0.218241	0.133550	5403.459283	1621.245798
std	177.390811	0.386497	0.476373	1.009623	0.413389	0.340446	6109.041673	2926.248369
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	150.000000	0.000000
25%	153.250000	1.000000	0.000000	0.000000	0.000000	0.000000	2877.500000	0.000000
50%	306.500000	1.000000	1.000000	0.000000	0.000000	0.000000	3812.500000	1188.500000
75%	459.750000	1.000000	1.000000	1.000000	0.000000	0.000000	5795.000000	2297.250000
max	613.000000	1.000000	1.000000	3.000000	1.000000	1.000000	81000.000000	41667.000000

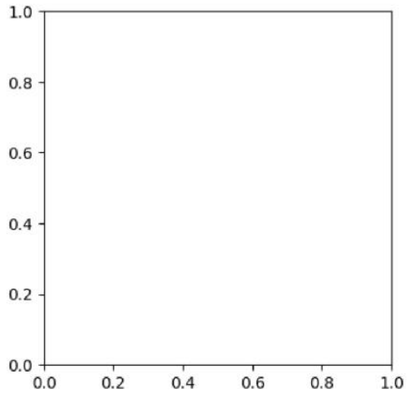
```
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.displot(data['ApplicantIncome'], color='r')
plt.subplot(122)
sns.displot(data['Credit_History'])
plt.show()
```

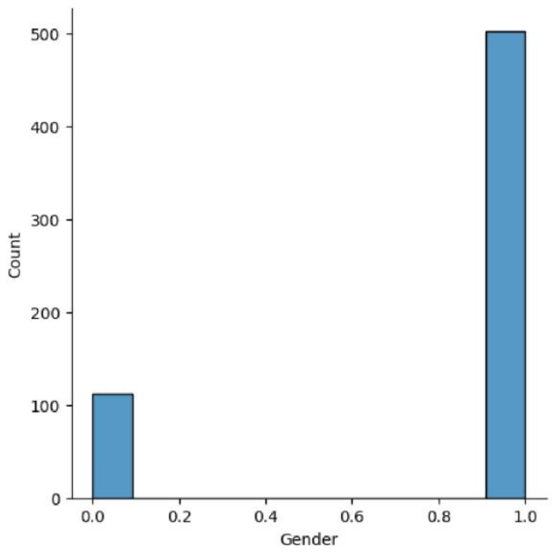
```
<ipython-input-10-15d1d8a644dc>:4: MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated si
  plt.subplot(122)
```

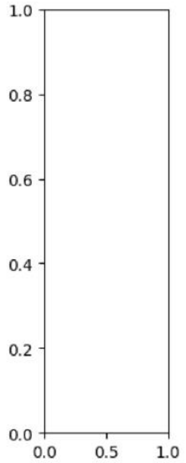


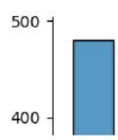



```
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.displot(data['Gender'])
plt.show()
plt.subplot(1,4,2)
sns.displot(data['Education'])
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.2 0.4 0.6 0.8 1.0


500
400
300
200
100
0
Count
0.0 0.2 0.4 0.6 0.8 1.0
Gender


1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.5 1.0

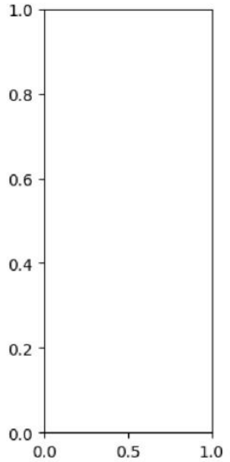

500
400

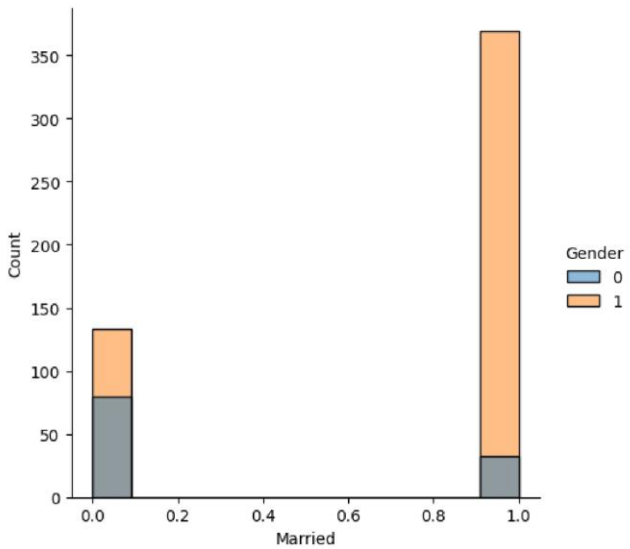
```
plt.figure(figsize=(20,5))
```

```
<Figure size 2000x500 with 0 Axes>
<Figure size 2000x500 with 0 Axes>
```

```
plt.subplot(131)
sns.displot(data, x='Married', hue='Gender')
```

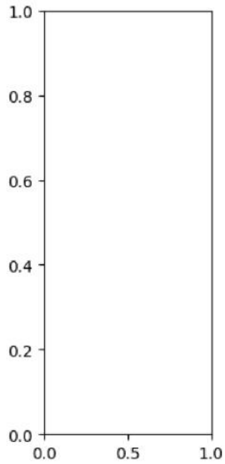
```
<seaborn.axisgrid.FacetGrid at 0x7f58b7be83a0>
```



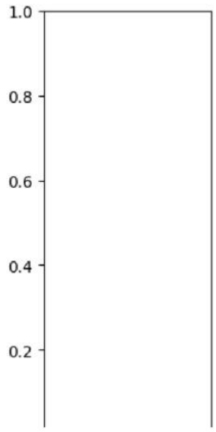



```
plt.subplot(132)
sns.displot(data, x='Self_Employed', hue='Education')
```

```
<seaborn.axisgrid.FacetGrid at 0x7f58b763e700>
```

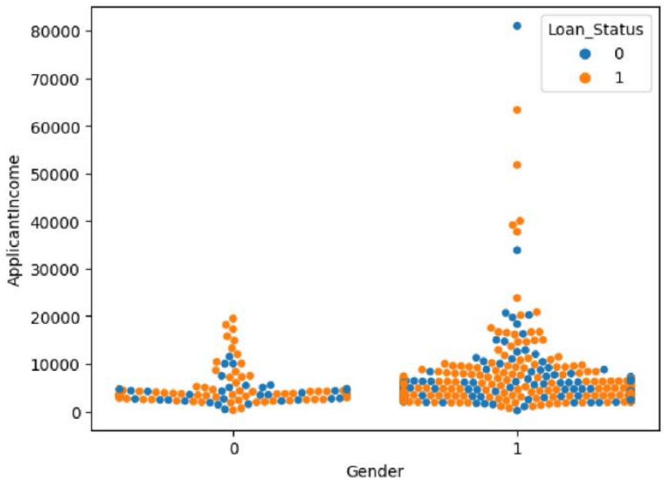



```
plt.subplot(133)
sns.displot(data, x='Property_Area', y='Loan_Amount_Term', hue='Loan_Status')
plt.show()
```




```
sns.swarmplot(data=data, x='Gender', y='ApplicantIncome', hue='Loan_Status')
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 45.8% of the points cannot be pla
  warnings.warn(msg, UserWarning)
<Axes: xlabel='Gender',
ylabel='ApplicantIncome'>/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 25.0% of
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 61.6% of the points cannot be pla
  warnings.warn(msg, UserWarning)
```




```
sc = StandardScaler()
x_bal = sc.fit_transform(x_bal)
x_bal = pd.DataFrame(x_bal)
```

```
X_train, X_test, y_train, y_test = train_test_split(x_bal, y_bal, test_size=0.33, random_state=42)
```

```
def KNN(x_train, x_test, y_train, y_test):
  knn = KNeighborsClassifier()
  knn.fit(x_train, y_train)
  ypred = knn.predict(x_test)
  print('***KNN Classifier***')
  print('Confusion matrix')
  print(confusion_matrix(y_test, y_pred))
  print('Classification report')
```

```
print(classification_report(y_test, y_pred))
```

```
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
from keras.models import Sequential
from keras.layers import Dense

classifier = Sequential()

# Update input_dim to 12 to expect 12 input features
classifier.add(Dense(units=100, activation='relu', input_dim=12))

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=32, epochs=100)
```

```
18/18 [==============================] - 0s 4ms/step - loss: 0.0920 - accuracy: 0.9752
Epoch 99/100
18/18 [==============================] - 0s 6ms/step - loss: 0.0933 - accuracy: 0.9664
Epoch 100/100
18/18 [==============================] - 0s 5ms/step - loss: 0.0872 - accuracy: 0.9770
<keras.callbacks.History at 0x7f5860333d30>
```

```
classifier.save("loan.h5")
```

```
y_pred = classifier.predict(X_test)
```

```
9/9 [==============================] - 0s 2ms/step
```

```
y_pred
```

```
       [7.43480682e-01],
       [1.82301155e-05],
       [9.63257253e-01],
       [9.95724857e-01],
       [3.78096700e-02],
       [3.58981924e-04],
       [9.98902977e-01],
       [3.58979672e-01],
       [2.91031860e-02],
       [9.99994040e-01],
       [8.96185756e-01],
       [4.62966323e-01],
       [8.69399011e-01],
       [6.86318502e-02],
       [9.99997437e-01],
       [9.92046535e-01],
       [1.44284073e-04],
       [9.75293100e-01],
       [8.15071523e-01],
       [4.51442413e-02],
       [9.90855694e-01],
       [2.49901623e-01],
       [3.64646927e-04],
       [9.99936283e-01],
       [8.35731387e-01],
       [9.25241709e-01],
       [9.96858060e-01],
       [9.91680801e-01],
       [5.36739062e-05],
       [8.72127354e-01],
       [1.25658844e-05],
       [7.09186867e-02],
       [8.37926865e-01],
       [9.97405410e-01],
       [1.19291376e-02],
       [9.82887864e-01],
       [9.46551204e-01],
       [1.58787254e-04],
       [2.70773835e-05],
       [9.88424301e-01],
       [9.79519844e-01],
       [9.51316237e-01],
       [7.56291486e-03],
       [4.98386294e-01],
       [9.90489900e-01],
       [9.99999166e-01],
       [4.03515220e-01],
       [7.65159726e-02],
       [9.89293814e-01],
       [5.90396821e-01],
       [1.03970878e-02],
       [9.74326022e-03],
       [1.68020859e-01],
       [2.20357440e-02],
       [9.22578812e-01],
       [4.04977143e-01],
       [6.10763252e-01],
       [2.29980916e-01]], dtype=float32)
```

```
y_pred = (y_pred > 0.5)
y_pred
```

```
           [ True],
           [False],
           [ True],
           [ True],
           [False],
           [False],
           [ True],
           [False],
           [False],
           [ True],
           [ True],
           [False],
           [ True],
           [False],
           [ True],
           [ True],
           [False],
           [ True],
           [ True],
           [False],
           [ True],
           [False],
           [False],
           [ True],
           [ True],
           [ True],
           [ True],
           [ True],
           [False],
           [ True],
           [False],
           [False],
           [ True],
           [ True],
           [False],
           [ True],
           [ True],
           [False],
           [False],
           [ True],
           [ True],
           [ True],
           [False],
           [False],
           [ True],
           [ True],
           [False],
           [False],
           [ True],
           [ True],
           [False],
           [False],
           [False],
           [False],
           [ True],
           [False],
           [ True],
           [False]])
```

```
# Scale the test data using the same scaler used for training data
X_test_scaled = sc.transform(X_test)

# Make predictions
predictions = classifier.predict(X_test_scaled)

# Convert probabilities to binary labels
predictions_binary = (predictions > 0.5).astype(int)

# Print the predictions
print("Predictions:", predictions_binary)
```

```
 [0]
 [1]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [1]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [1]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [1]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]
 [0]]
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, bu
```

```
def predict_exit(sample_value):
  sample_value = np.array(sample_value)
  sample_value = sample_value.reshape(1, -1)
  sample_value = sc.transform(sample_value)

  return classifier.predict(sample_value)

sample_value = [[1,1, 0, 1, 1, 4276, 1542,145, 240, 0,1,0]]
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, 1,1,0]]
if predict_exit(sample_value)>0.5:
    print('prediction: High chance of Loan Approval')
else:
    print('prediction: Low chance of Loan Approval')
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but
  warnings.warn(
1/1 [==============================] - 0s 23ms/step
prediction: High chance of Loan Approval
1/1 [==============================] - 0s 22ms/step
prediction: High chance of Loan Approval
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but
  warnings.warn(
```

```
def compareModel(X_train,X_test,y_train,y_test):
  KNN(X_train,X_test,y_train,y_test)
  print('-'*100)
```

```
compareModel(X_train,X_test,y_train,y_test)
```

```
***KNN Classifier***
Confusion matrix
[[103  45]
 [ 27 104]]
Classification report
              precision    recall  f1-score   support

           0       0.79      0.70      0.74       148
           1       0.70      0.79      0.74       131

    accuracy                           0.74       279
   macro avg       0.75      0.74      0.74       279
weighted avg       0.75      0.74      0.74       279

----------------------------------------------------------------------------------------------------
```

```
yPred = classifier.predict(X_test)
print(accuracy_score(y_pred,y_pred))
print("ANN Model")
print("Confusion_Matrix")
print(confusion_matrix(y_test,y_pred))
print("Classification Report")
print(classification_report(y_test,y_pred))
```

```
9/9 [==============================] - 0s 2ms/step
1.0
ANN Model
Confusion_Matrix
[[103  45]
 [ 27 104]]
Classification Report
              precision    recall  f1-score   support

           0       0.79      0.70      0.74       148
           1       0.70      0.79      0.74       131

    accuracy                           0.74       279
   macro avg       0.75      0.74      0.74       279
weighted avg       0.75      0.74      0.74       279
```

```
from sklearn.model_selection import cross_val_score

rf = RandomForestClassifier()
rf.fit(X_train,y_train)
yPred = rf.predict(X_test)
f1_score(yPred,y_test,average='weighted')
0.9679166666666668
cv = cross_val_score(rf,X,y,cv=5)
np.mean(cv)
0.985
```

```
0.985
```

```
pickle.dump(classifier,open('rdf.pkl','wb'))
```