

# **UNDERSTANDING AUDIENCE:A MACHINE LEARNING APPROACH TO CUSTOMER SEGMENTATION**

# **1. INTRODUCTION**

## **1.1 OVERVIEW**

These days, you can personalize everything. There's no one-size-fits-all approach. But, for business, this is actually a great thing. It creates a lot of space for healthy competition and opportunities for companies to get creative about how they acquire and retain customers.

One of the fundamental steps towards better personalization is customer segmentation. This is where personalization starts, and proper segmentation will help you make decisions regarding new features, new products, pricing, marketing strategies, even things like in-app recommendations.

But, doing segmentation manually can be exhausting. Why not employ machine learning to do it for us? In this article, I'll tell you how to do just that.

## **1.2 Purpose**

The goal of this project is to decide how to relate to customers in each segment in order to predict the financial status of customers. We can identify the most active users/customers, and optimize your application/offer towards their needs.

# **2. LITERATURE SURVEY**

## **2.1 EXISTING SYSTEM**

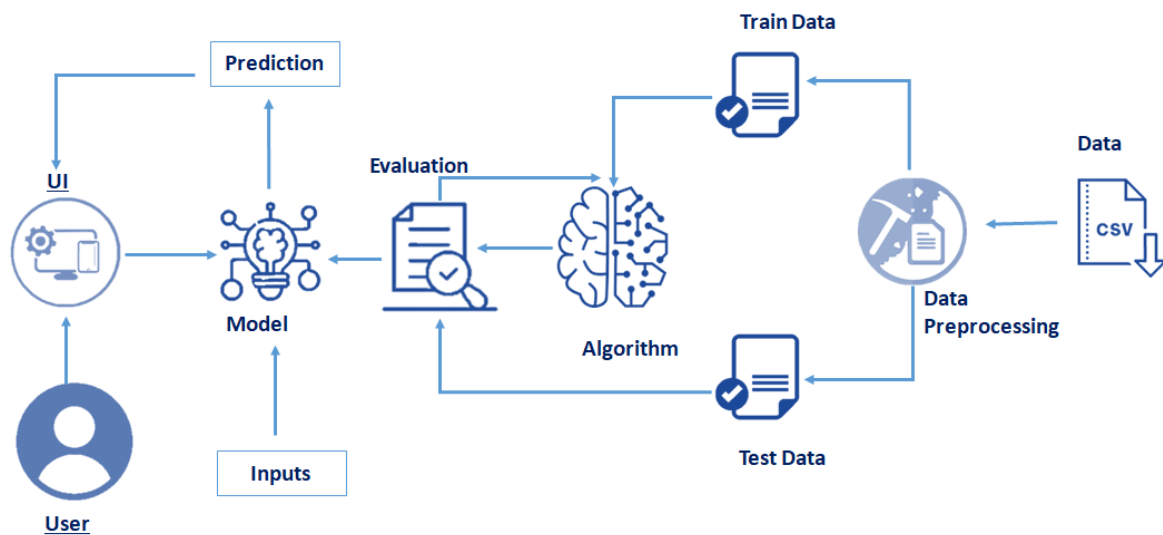
The problem of existing system is that it is not able to predict the financial status of a customer. And firm will not be able to develop or innovate product according to customer desire. So that firm will not be able to target right customer with right product.

## **2.2 PROPOSED SYSTEM**

The proposed work predicts the financial status of a customer. Here the entered data are fed into a model which predicts whether the customer is potential or not. We can identify the most active users/customers, and optimize our application/offer towards their needs.

### 3. THEORATICAL ANALYSIS

#### 3.1 BLOCK DIAGRAMS



#### 3.2 HARDWARE AND SOFTWARE DESIGNING

##### Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum, and first released on February 20, 1991. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

##### Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform, package management system. Anaconda comes with so many nice tools like JupyterLab, Jupyter Notebook,

QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

### Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

### Spyder

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community. Spyder is extensible with first-party and third-party plugins. It includes support for interactive tools for data inspection and embeds Python-specific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

### Flask

Web framework used for building. It is a web application framework written in Python which will be running in local browser with a user interface. In this application, whenever the user interacts with UI and selects emoji, it will suggest the best and top movies of that genre to the user. Hardware Requirements: Operating system: Windows 7 and above with 64bit

Processor Type -Intel

Core i3-3220

RAM: 4Gb and above

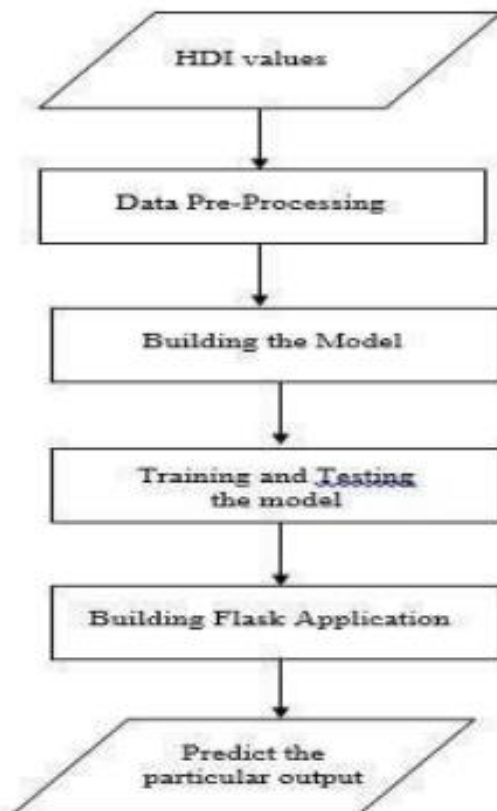
Hard disk: min 100GB

## **4. EXPERIMENTAL INVESTIGATION**

The text data need to be organized before proceeding with the project. The original dataset has a single folder. We will be using the .csv file to fetch the text data of training data. The data need to be unique and all fields need to be filled. The data set images are to be pre-processed before giving to the model. We will create a function that uses the pre-trained

model for predicting custom outputs. Then we have to test and train the model. After the model is build, we will be integrating it to a web application

## 5. FLOW CHART



## 6. RESULT

Customer Segmentation

Please enter the following details

The interface features a world map background with three circular callouts. The first callout on the left shows a family of four. The second callout in the middle shows a family of five. The third callout on the right shows a woman pushing a stroller. The background is a light blue world map with dashed lines connecting the callouts.

Customer Segmentation

Please enter the following details

Highly potential customer

The interface is identical to the one above, but with different input values. The background world map and callouts remain the same. The input fields are now: Id: 101, Sex: Female, Marital status: single, Age: Age, Education: Educ, Income: Income, Occupation: Not Working, Settlement size: 1. The Predict button is still present, and the output "Highly potential customer" is displayed below the form.



## 7. ADVANTAGES

Easy to use

Cost efficient

Time efficient

## 8. CONCLUSION

While this guide provides a step-by-step process for identifying, prioritizing, and targeting your best current customer segments, simply following it does not guarantee success. To be effective, you must prepare and plan for the various challenges and hurdles that each step may present, and always make sure to adapt your process to any new information or feedback that might change its output. Additionally, you cannot force feed this process on your business. If the key stakeholders that will be impacted by the best current customers segmentation process do not fully buy-in, then the outputs produced from it will be relatively meaningless.

## 9. FUTURE SCOPE

If you properly manage the best current customer segmentation process, however, the impact it can have on every part of your organization — sales, marketing, product development, customer service, etc. — is immense. Your business will possess stronger customer focus and market clarity, allowing it

to scale in a far more predictable and efficient manner. Ultimately, that means no longer needing to take on every customer that is willing to pay for your product or service, which will allow you to instead hone in on a specific subset of customers that present the most profitable opportunities and efficient use of resources. That is critical for every business, of course, but at the expansion stage, it can often be the difference between incredible success and certain failure.

## 10. BIBLIOGRAPHY

[https://www.daitm.org.in/wp-content/uploads/2019/04/15499016029\\_Abhijit-Bag.pdf](https://www.daitm.org.in/wp-content/uploads/2019/04/15499016029_Abhijit-Bag.pdf)

## 11. SOURCE CODE

### App.py

```
import numpy as np
import pickle
import pandas
import os
import joblib
from flask import Flask, request, jsonify, render_template

app = Flask(__name__)
model = pickle.load(open(r'F:/notebook/customer/Flask/xgbmodel1.pkl', 'rb'))
#scale = pickle.load(open(r'C:/Users/SmartbridgePC/Desktop/AI ML/Guided
projects/rainfall_prediction/IBM flask push/Rainfall IBM deploy/scale.pkl','rb'))

@app.route('/')# route to display the home page
def home():
    return render_template('index.html') #rendering the home page

@app.route('/predict',methods=["POST","GET"])# route to show the predictions in a web UI
def predict():
    # reading the inputs given by the user
    input_feature=[float(x) for x in request.form.values() ]
    features_values=[np.array(input_feature)]
    names = [['Id','Sex', 'Marital status', 'Age', 'Education', 'Income', 'Occupation','Settlement
size']]
    data = pandas.DataFrame(features_values,columns=names)
    #data = scale.fit_transform(features_values)
```



```

# predictions using the loaded model file
prediction=model.predict(data)
print(prediction)

if (prediction == 0):
    return render_template("index.html",prediction_text="Not a potential customer")
elif (prediction == 1):
    return render_template("index.html",prediction_text="Potential customer")
else:
    return render_template("index.html",prediction_text="Highly potential customer")
# showing the prediction results in a UI
if __name__=="__main__":

    # app.run(host='0.0.0.0', port=8000,debug=True)  # running the app
    port=int(os.environ.get('PORT',5000))
    app.run(port=port,debug=True,use_reloader=False)

```

## index.html

```

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Customer Segmentation</title>

</head>

<body
background="https://www.imf.org/external/pubs/ft/fandd/2020/03/images/032020/picture-
1600.jpg" text="black">

    <div class="login">

        <center>

            <h1>Customer Segmentation</h1>

        </center>

```

```
<!-- Main Input For Receiving Query to our ML -->

<form action="{{ url_for('predict')}}" method="post">

    <h1>Please enter the following details</h1>

    </style>

    </head>

    <label>Id:</label>

    <input type="number" name="Id" placeholder="Id" required="required" /><br>

    <label for="Sex">Sex:</label>

    <select id="Sex" name="Sex">

        <option value=0>Female</option>

        <option value=1>Male</option>

    </select> &nbsp;&nbsp;&nbsp;<br>

    <br><label for="Marital status">Marital status:</label> <select id="Marital status"
name="Marital status">

        <br>

        <br>

        <option value=0>single</option>

        <option value=1>Married</option>

    </select> &nbsp;&nbsp;&nbsp;<br>

    <label>Age:</label>

    <input type="number" min="20" max="80" name="Age" placeholder="Age"
required="required" /><br>

    <label>Education:</label>

    <input type="number" min="0" max="3" name="Education"
placeholder="Education" required="required" /><br>

    <label>Income:</label>
```

```

    <input type="number" min="5000" name="Income" placeholder="Income"
required="required" /><br>

    <br><label for="Occupation">Occupation:</label>

    <select id="Occupation" name="Occupation">

        <option value=0>Not Working</option>

        <option value=1>Working</option>

        <option value=1>Business</option>

    </select> &nbsp;&nbsp;&nbsp;<br>

    <br>

    <label for="Settlement size">Settlement size:</label>

    <br><br>

    <select id="Settlement size" name="Settlement size">

        <option value=1>1</option>

        <option value=0>0</option>

        <option value=2>2</option>

    </select> &nbsp;&nbsp;&nbsp;<br>

    <button type="submit" class="btn btn-primary btn-block btn-large"
style="height:30px;width:200px">Predict</button>

    <br>

    <br>

</form>

{{ prediction_text }}

<br> <br>

    
<br>

<br>

