School of Computer Applications Union Christian College

Project Report

On

CUSTOMER SEGMENTATION USING IBM WATSON MACHINE LEARNING

Internal Guide:

AMRITHA MISS
Assoc. Professor

Submitted by:

DIYA NA

MEENAKSHY JEEVAN
MCA Department
UC College, Aluva.

Submitted on : 08/06/2022

Table of Contents

1. EXECUTIVE SUMMARY

The primal aim of any business is to grab potential customers who can generate profits for the organization. The primal task of Management is to identify potential customers from the rest. This will be simplified with the help of Machine Learning models to classify the customers into segments based on various attributes.
The Model we built will be able to classify the customer's potentiality in purchasing power.

This comparative study is conducted concentrating on the following aspects: modeling inputs, Visualising the data, modeling methods, and pre-processing techniques. The results provide a comparison of various evaluation metrics of these machine learning techniques and their reliability to predict rainfall by analyzing the weather data.

We will be using classification algorithms such as H-clustering, kmeans clustering 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. Once the model is saved, we integrate it with the flask application and also deploy the model in IBM.

2. PROJECT OVERVIEW

2.1. Objective of the project

- This project enables the learner to understand the business usecase of how and why to segment the customers.
- You will be understand the unsupervised lerning methods such as h-clustering and k-means clustering.
- You will be able to understand the problem to classify if it is a regression or a classification kind of problem
- You will be able to analyse or get insight into data through visualization.
- You will be able to know how to find the accuracy of the model.
- You will bw able to know how to build a web applivation using the flsk framework.ss

2.2. Stakeholders

- Shop owner

2.3. Scope of the Project

The future enhancement of this project can be an approach towards about how to reduce the percentage of errors present. Along with that one of the major enhancements will be to decrease the ratio for train data to test data, so that it will assist in improving the level of prediction within the available time and complexity. The accuracy of the algorithm can be additionally tested on increase in the complexity. Many other types of errors can be calculated in order to test the accuracy of any of the above algorithms. Henceforth, algorithm for testing daily basis dataset instead of accumulated dataset could be of paramount Importance for further research.

2.4. Feasibility Analysis

2.4.1. Technical Feasibility

We can strongly say that it is technically feasible since there will not be much difficulty in getting the required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same are available and we are utilizing the resources which are available already.

2.4.2. Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information systems. That will meet the organization’s operating requirements. Operational feasibility aspects of the projects are to be taken as an important part of the project implementation

Some of the important issues raised are to test the operational feasibility of a project include the following:

Is there sufficient support for the management from the users?

Will the system be used and work properly if it is being developed and implemented?

Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. The well-planned deigned would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

2.4.3. Schedule feasibility

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. You need to determine whether the deadlines are mandatory or desirable. The scheduled time for the online exam system was about 3 to 4 months and the project is completed in 4 months.

3. OVERALL PROJECT PLANNING

3.1. Development Environment

Front End: **HTML5, BOOTSTRAP** Backend: **Python 3.8**

Other Software Requirements:

- Anaconda Navigator
- Python Packages
- Flask
- IBM Watson Studio

Watson Studio accelerates the machine and deep learning workflows required to infuse AI into your business to drive innovation. It provides a suite of tools for data scientists, application developers, and subject matter experts, allowing them to connect to data collaboratively, wrangle that data and use it to build, train and deploy models at scale.
Successful AI projects require a combination of algorithms + data + a team and a very powerful computer infrastructure.

- IBM Watson Machine Learning
- IBM Cloud Object Storage

3.2. Constraints

- User Interface is only in English
- System is based on historical data.

3.3. Assumptions and Dependencies

Here, we will be creating and training our model for predicting the potential of a customer. Since there are multiple algorithms we can use to build our model, we will compare the accuracy scores after testing and pick the most accurate algorithm. From this list, we are using DecisionTree, RandomForest, and Kmeans to perform our predictions. We then see which algorithm produces the highest accuracy and select it as our algorithm of choice for future use.
On the results of the following algorithms, we have the conclusion that the Random Forest model is the most accurate out of all the models which we have tested.

3.4. Process Model

To solve actual problems in the industry, a software developer or a team of developers must integrate with a development strategy that includes the process, methods, tools layer, and generic phrases. This strategy is often referred to as a process model or a software developing paradigm.

Our project follows the Agile model.

The steps of the Agile Model are:

- Requirements gathering
- Design the requirements
- Construction/ iteration
- Testing/ Quality assurance
- Deployment
- Feedback

4. ITERATION PLANNING

4.1. Schedule

Project planning is the most important phase of a software development life cycle. The project plan discusses what should be done and how it should be done to reach the final goal. Creating such a plan vitally important because of the nature of software development

TASK	START DATE	DURATION(days)	END DATE
Data Collection	06/12/2021	02	07/12/2021
Visualizing And Analysing The Data	08/12/2021	04	11/12/2021
Data Pre-Processing	12/12/2021	10	21/12/2021
Model Building	22/12/2021	14	04/01/2022
Application Building	05/01/2022	14	18/01/2022
Train The Model On IBM	19/01/2022	07	25/01/2022
Documentation	27/01/2022	05	31/01/2022

5. DESIGN MODEL

5.1. Activity diagram

Activity diagram is describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

5.2. Class diagrams

Here class diagram describes about the structure of our system by showing the system’s classes, their attributes, operations, and the relationship among objects. In this project the main classes are application requests, admission. Each functionality require some input data. Those are represented as the attributes of the class. Class diagram helps to understand grouping of functionality.

5.3. Sequence Diagrams

In this project sequence diagram shows object interaction arranged in time sequence it is typically associated with use case realizations in the logical view of the system under devolvement.

5.4. UI design

User interface design usually the primary interface for human machine interaction. In this project use library that follows material design constraints.It is defined in Annexure 6.1.

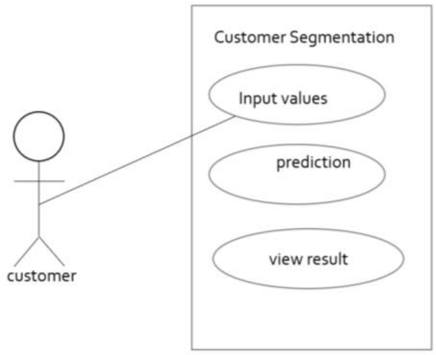
5.5. Theoretical Background

Hostel mess bill automation is an application that will be developed using latest industrial standard technologies like android and firebase back end. Android developed using java.

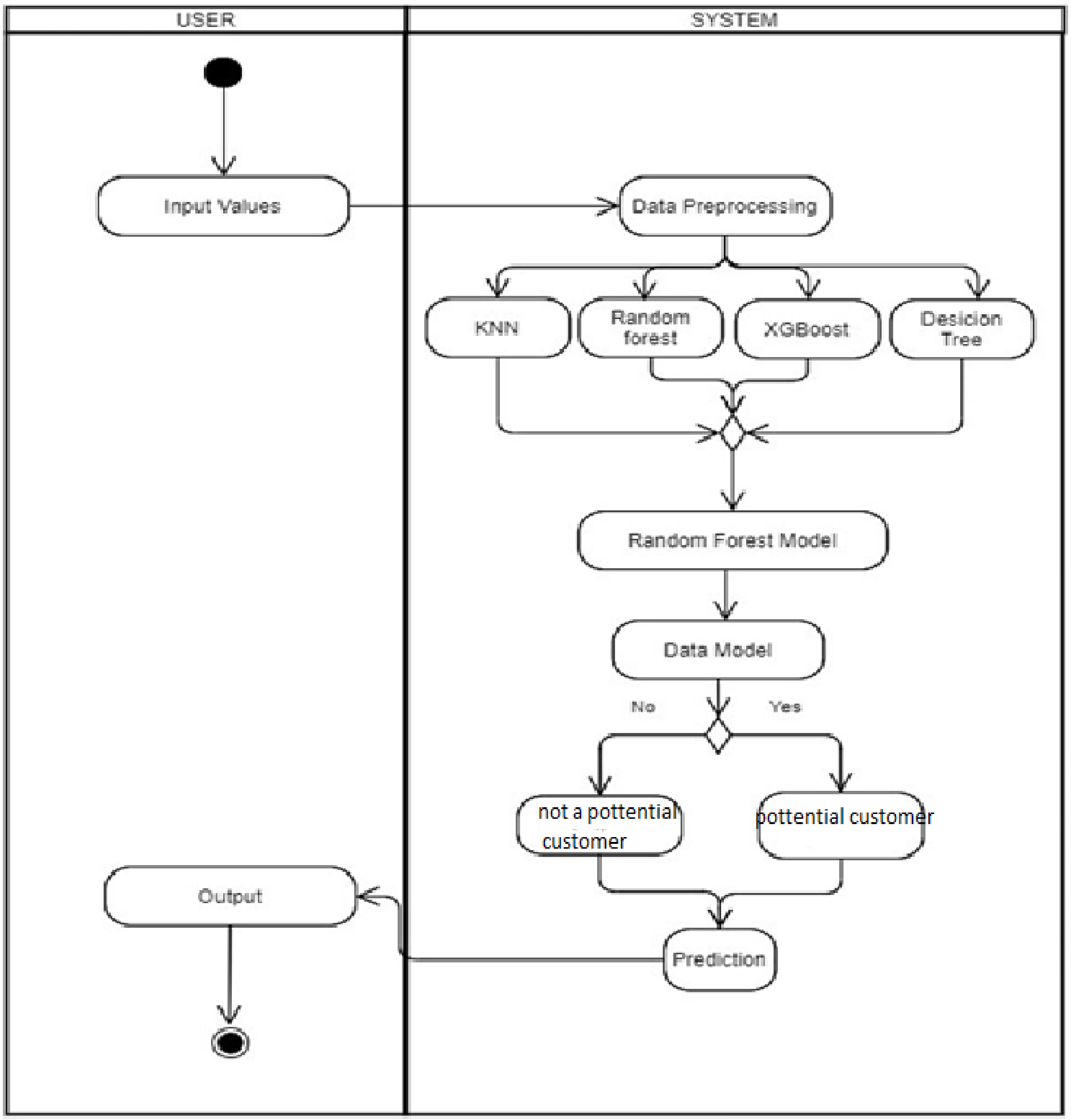
6. ANNEXURE: ss

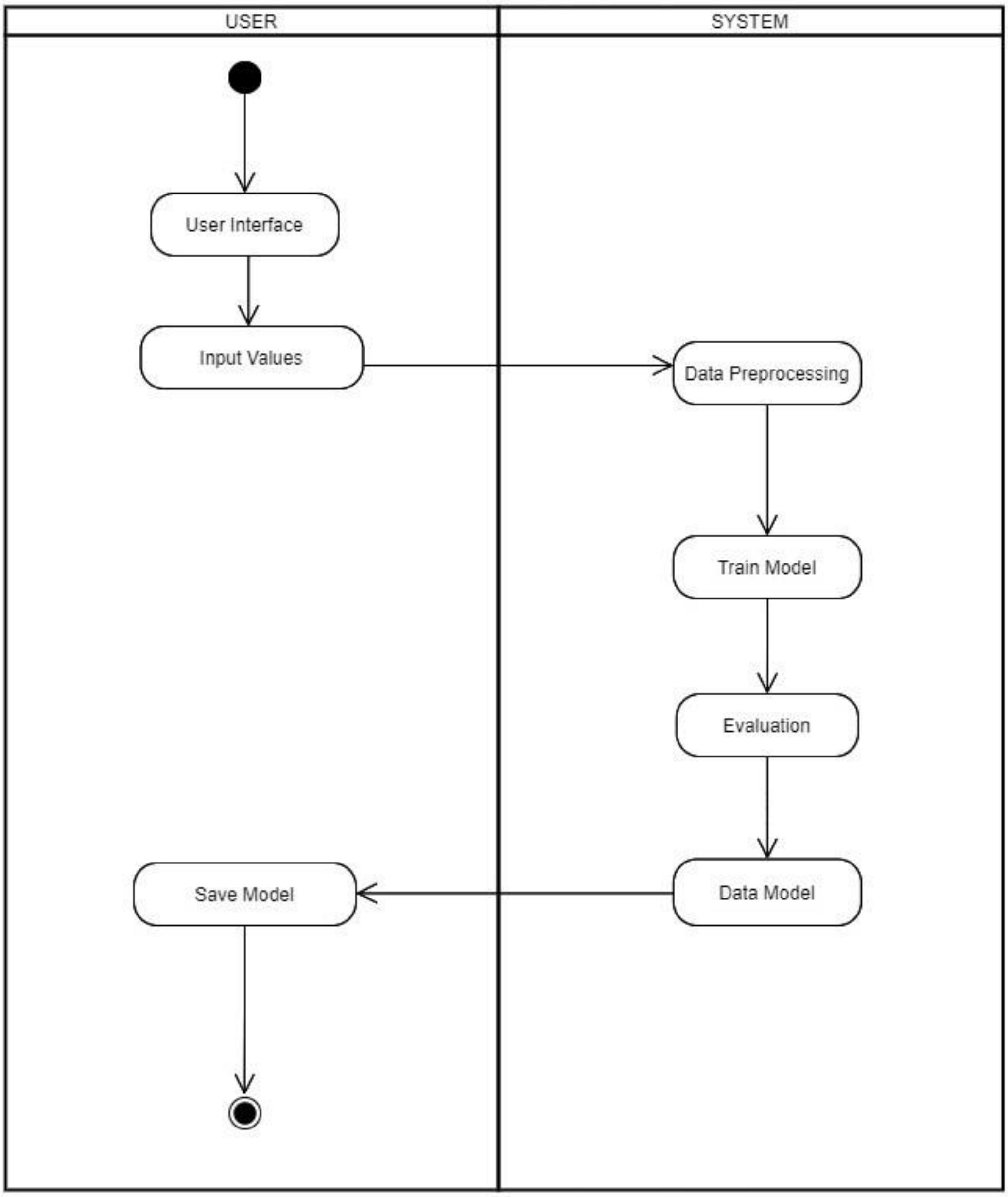
6.1. Architecture Diagrams

6.1.1. Use case diagram

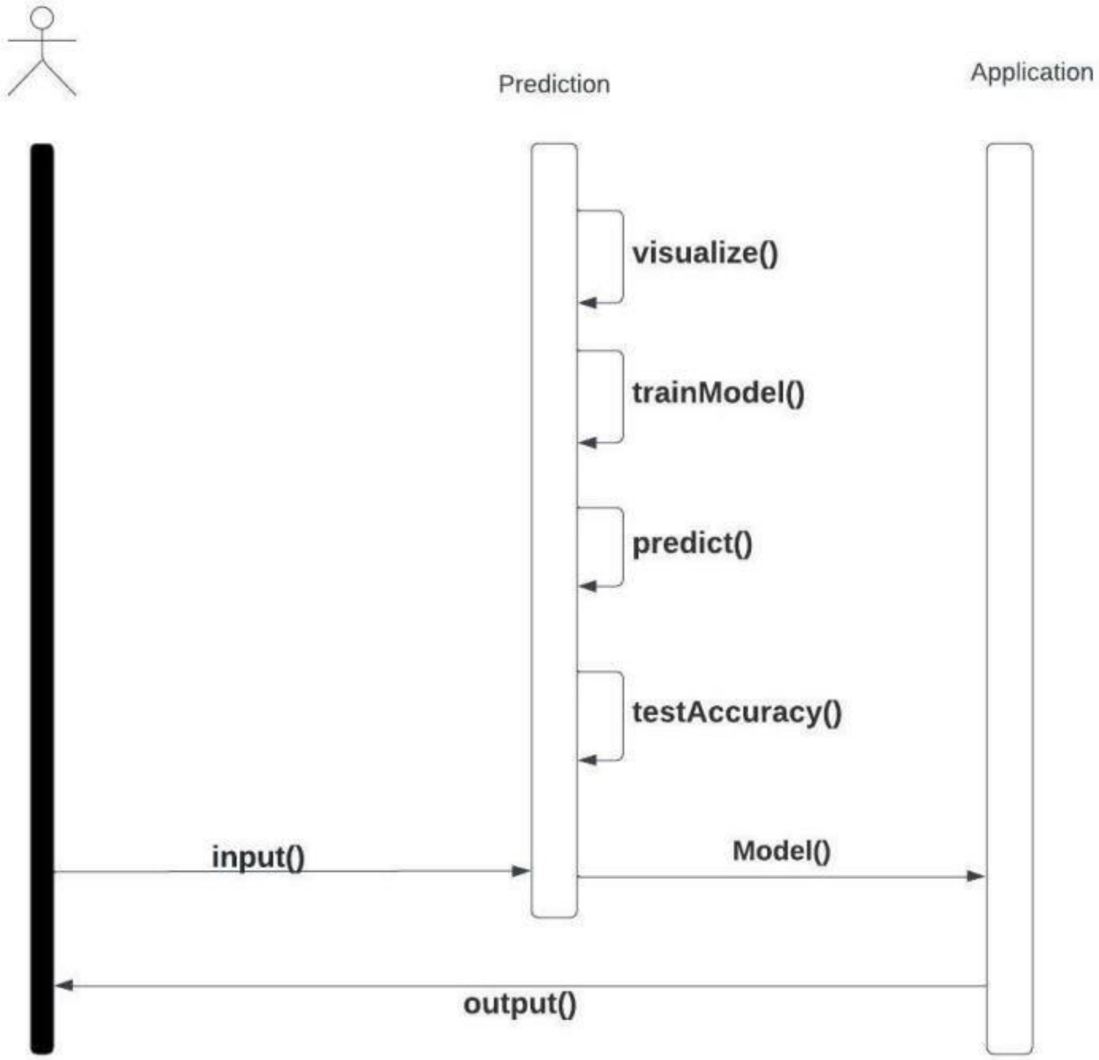


6.1.2. Activity Diagram

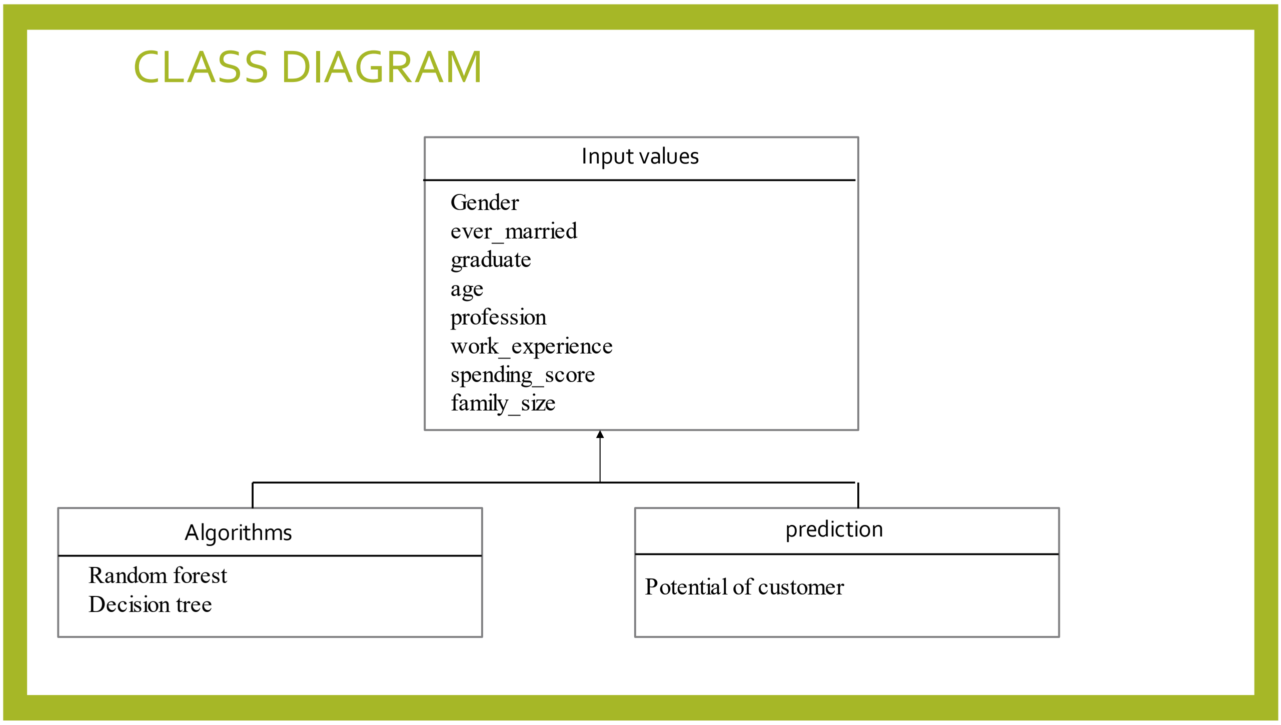

USER
SYSTEM
Input Values
Data Preprocessing
KNN
Random forest
XGBoost
Desicion Tree
Random Forest Model
Data Model
No
Yes
not a pottential customer
pottential customer
Output
Prediction



6.1.3. Sequence Diagram


Prediction
Application
visualize()
trainModel()
predict()
testAccuracy()
input()
Model()
output()

6.1.4. Class Diagram

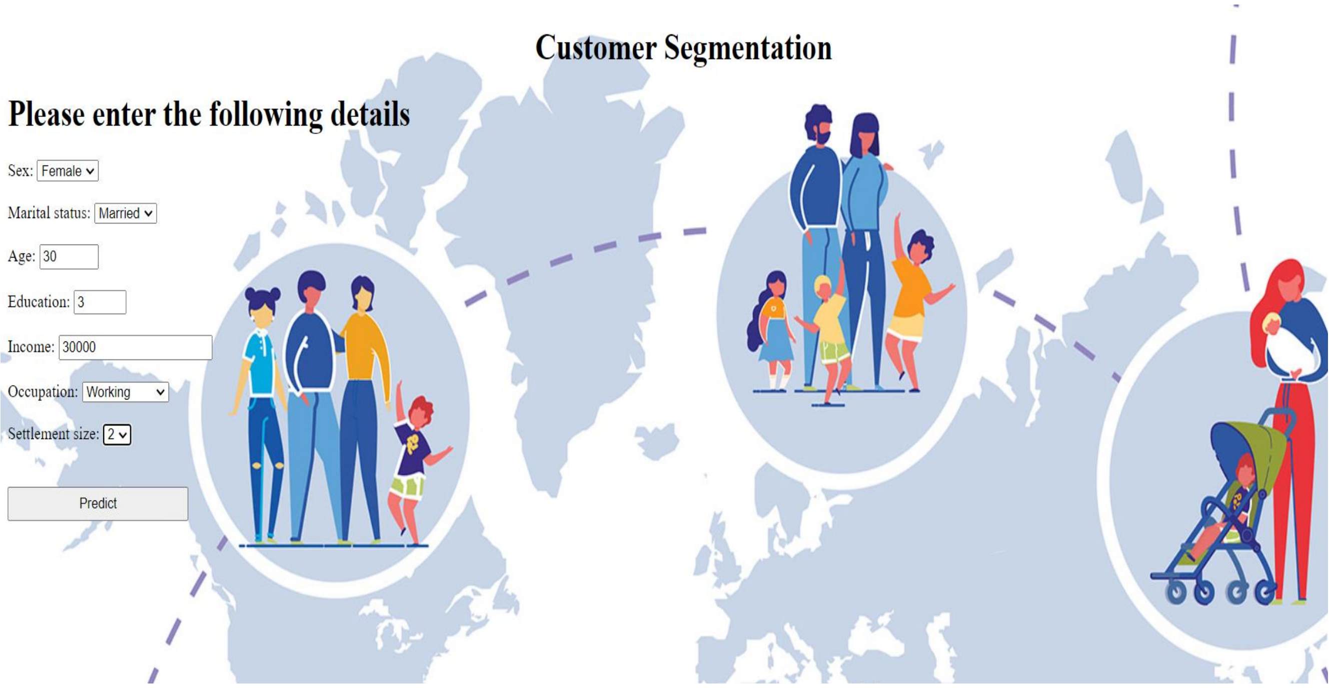


6.2. Dataset Design

sex	Varchar(20)
Marital status	Varchar(20)
age	number
education	Varchar(20)
income	number
occupation	Varchar(20)
Settlement size	Number

7. SCREEN IMAGES

The final result of the project Employee Promotion prediction . The output would be in this format:



Customer Segmentation

Please enter the following details

Sex: Female

Marital status: Married

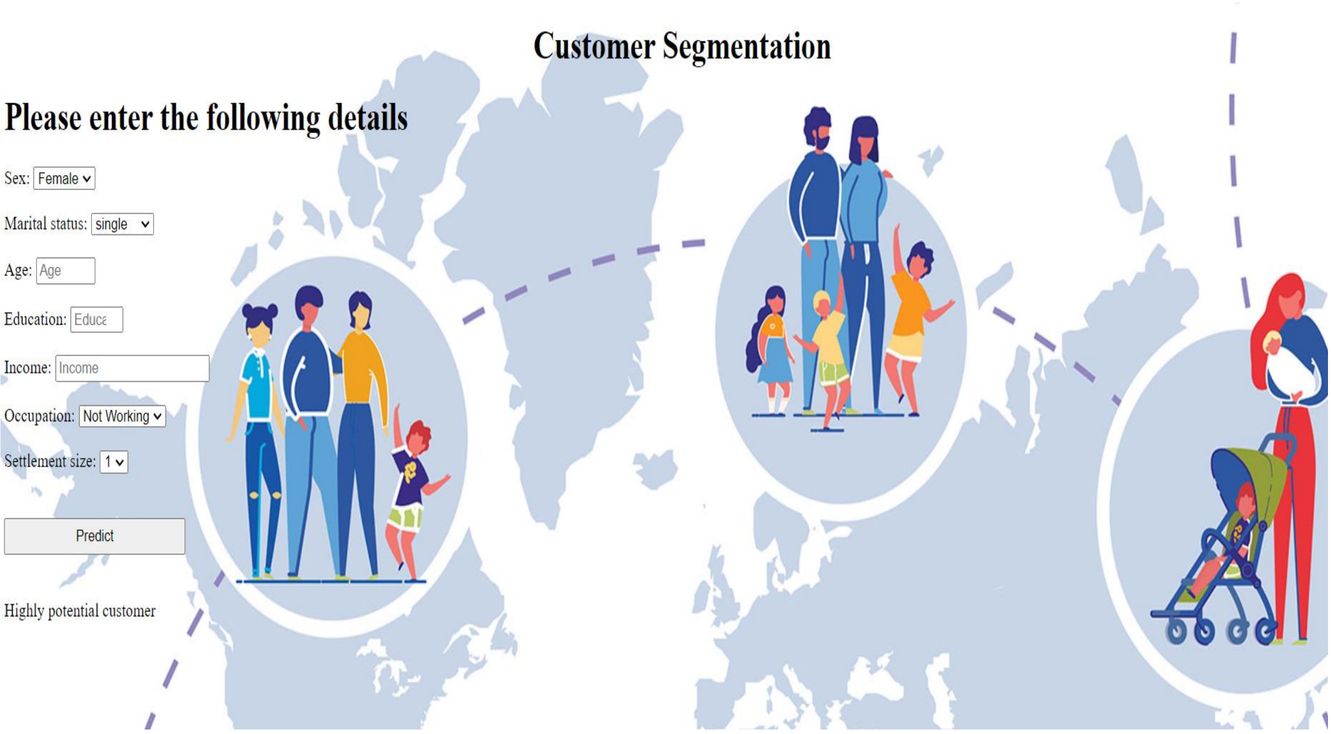
Age: 30

Education: 3

Income: 30000

Occupation: Working

Settlement size: 2



Customer Segmentation

Please enter the following details

Sex: Female

Marital status: single

Age: Age

Education: Educa

Income: Income

Occupation: Not Working

Settlement size: 1

Highly potential customer

ss

8. SAMPLE PROJECT CODE

ML Clustering.ipynb

```
#!/usr/bin/env python

# coding: utf-8


# # CLustering is the ML unsupervised methods,used to group the data based on similaries in the data - Hierarchial clustering
# # Hierarchial clustering

# In[1]:


# similar records will be clubbed together
# DO EDA process - not mandatory(few are mandatory)
# scale the data
# calculate the distance - Euclidean or manhatten
# cluster the records based on single/complete link ('least /farthest ' distance)
# divide in the clusters into 2 or 3 classes based on the requirement
# use dendogram to visualise the clustered data
#join the classes with main data


# In[6]:


import os


# In[7]:


os.chdir('G:\AI&ML\ML projects\cluster analysis')


# In[8]:
```

```
import pandas as pd


# In[9]:


# Reading the dataset
data = pd.read_csv(r'C:\Users\HP\Desktop\python\project_customer_segmentation\Customer segmentation model\Dataset\segmentation data.csv',header='infer')


# In[10]:


data.info()


# In[11]:


data.shape


# In[12]:


data.describe()


# In[13]:
```

```
cor = data.corr()
cor


# In[14]:


import seaborn as sns


# In[15]:


sns.heatmap(cor,annot=True)


# In[16]:


sns.pairplot(data)


# In[17]:


data.drop(columns=['ID'],axis=1,inplace=True)
```

```
# In[18]:


data.head()


# In[19]:


names = data.columns


# In[20]:


data.isna().sum()


# In[21]:


from sklearn import preprocessing


# In[22]:


data = preprocessing.minmax_scale(data,feature_range=(0,1))  #scaling the data
```

```
# In[23]:


data


# In[24]:


data = pd.DataFrame(data,columns=names) #scaled data will convert to array,so converting to dataframe


# In[25]:


data.head()


# In[26]:


# using dendogram to find optimal no of clusters
import scipy.cluster.hierarchy as sch
import matplotlib as plt


# In[27]:
```

```
dendogram = sch.dendrogram(sch.linkage(data,method="ward")) # calculates euclidean distance and
classfy by default

plt.pyplot.title('clustering of the data')

plt.pyplot.ylabel('distance')

plt.pyplot.xlabel('points')

plt.pyplot.show()



# In[28]:



from sklearn import cluster

import sklearn as sk



# In[29]:



clus = cluster.AgglomerativeClustering(n_clusters=3,affinity="euclidean",linkage='complete')

clus



# In[30]:



clus.fit(data)



# In[31]:
```

```
abc = clus.fit_predict(data)


# In[32]:


hclusdata = pd.DataFrame(data,pd.Series(abc))


# # Creating a labled data with the help of clustering model

# In[33]:


hclusdata['clus']= pd.Series(abc)


# In[34]:


hclusdata.head()


# In[35]:


y = hclusdata['clus']
x = hclusdata.drop(columns=['clus'],axis=1)
```

```
# # splitting the test and train data

# In[36]:


from sklearn.model_selection import train_test_split


# In[37]:


x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=0)


# In[38]:


print(x_train.shape)
print(x_test.shape)


# # Applying supervised learning on the data

# In[39]:


from sklearn.ensemble import RandomForestClassifier
from sklearn import tree
import xgboost
```

```
# In[40]:


rand_model = RandomForestClassifier()

tree_model =  tree.DecisionTreeClassifier()

xgb_model = xgboost.XGBClassifier()


# In[41]:


rand_model.fit(x_train,y_train)

tree_model.fit(x_train,y_train)

xgb_model.fit(x_train,y_train)


# In[42]:


pred = rand_model.predict(x_test)

pred1 = tree_model.predict(x_test)

pred2 = xgb_model.predict(x_test)


# In[43]:


from sklearn import metrics
```

```
# In[44]:


print(metrics.accuracy_score(pred,y_test))
print(metrics.accuracy_score(pred1,y_test))
print(metrics.accuracy_score(pred2,y_test))


# In[45]:


metrics.confusion_matrix(pred,y_train)


# # K-means clustering

# In[46]:


from scipy import spatial


# In[47]:


wcss = []
for i in range(1,11):
    kmeans = cluster.KMeans(n_clusters=i,init='k-means++',random_state=0)
    kmeans.fit(hclusdata)

    wcss.append(kmeans.inertia_)



# In[48]:



plt.pyplot.plot(range(1,11),wcss)

plt.pyplot.title('elbow method')

plt.pyplot.xlabel('no.of clus')

plt.pyplot.ylabel('wcss')

plt.pyplot.show()



# In[49]:



km_model = cluster.KMeans(n_clusters=3,init='k-means++',random_state=0)



# In[50]:



ykmeans = km_model.fit_predict(data)



# In[51]:



data.head()
```

```
# In[52]:


data['kclus'] = pd.Series(ykmeans)


# In[53]:


data.head()


# In[54]:


y = data['kclus']
x = data.drop(columns=['kclus'],axis=1)


# In[55]:


x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=0)


# In[56]:
```

```
rand_model = RandomForestClassifier()
tree_model =  tree.DecisionTreeClassifier()
xgb_model = xgboost.XGBClassifier()


# In[57]:


rand_model.fit(x_train,y_train)
tree_model.fit(x_train,y_train)
xgb_model.fit(x_train,y_train)


# In[58]:


pred = rand_model.predict(x_test)
pred1 = tree_model.predict(x_test)
pred2 = xgb_model.predict(x_test)


# In[59]:


print(metrics.accuracy_score(pred,y_test))
print(metrics.accuracy_score(pred1,y_test))
print(metrics.accuracy_score(pred2,y_test))


# # Saving the model
# In[60]:

import pickle

# In[61]:

pickle.dump(xgb_model,open("xgbmodel.pkl",'wb'))

# In[62]:

pwd

# In[ ]:
```

App.py

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


app = Flask(__name__)
model = pickle.load(open(r'C:\Users\HP\Desktop\python\project_customer_segmentation\Customer segmentation model\Flask\xgbmodel.pkl', 'rb'))
#scale = pickle.load(open(r'C:/Users/SmartbridgePC/Desktop/AIML/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 = [['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)
```

9. REFERENCES:

[1] Sukru Ozan, “A Case Study on Customer Segmentation by using Machine Learning Methods”. IEEF, Year: 2018.

[2] Jayant Tikmani, Sudhanshu Tiwari, Sujata Khedkar “Telecom customer segmentation based on Cluster analysis An Approach to Customer Classification using k-means”, UJIRCCE, Year: 2015.

[3] Chinedu Pascal Ezenkwu, Simeon Ozuomba, Constance kalu Electrical/Electronics and Computer Engineering Department.