A

Mini Project

On

# A MODEL FOR PREDICTION OF CONSUMER CONDUCT USING MACHINE LEARNING

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

CH.PIYUSH (217R1A05L9)

B.RAVI TEJA (217R1A05L0)

T.VINAY (217R1A05R1)

Under the Guidance of

**K. RANJITH REDDY**

(Assistant Professor)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

**2021-2025**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# CERTIFICATE

This is to certify that the project entitled "**A MODEL FOR PREDICTION OF CONSUMER CONDUCT USING MACHINE LEARNING"** being submitted by **CH.PIYUSH (217R1A05L9), B.RAVI TEJA (217R1A05L0) and T.VINAY (217R1A05R1)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-25.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

**K. Ranjith Reddy**
(Assistant Professor)
INTERNAL GUIDE

**Dr. A. Raji Reddy**
DIRECTOR

**Dr. N. Bhaskar**
HOD

**EXTERNAL EXAMINER**

**Submitted for viva voice Examination held on** ________________

# ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express our profound gratitude and deep regard to our guide **Mr. K. Ranjith Reddy** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators **Dr. J. NarasimhaRao, Mr. K. Ranjith Reddy, Dr. K. Maheswari, Mrs. K. Shilpa** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. N. Bhaskar,** Head of the Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**CH. PIYUSH (217R1A05L9)**

**B. RAVI TEJA (217R1A05L0)**

**T. VINAY (217R1A05R1)**

# ABSTRACT

The increasing availability of consumer data has created new opportunities for businesses to understand and influence consumer conduct. However, the complexity and variability of consumer behavior pose significant challenges for traditional analytical approaches. This study addresses these challenges by developing a machine learning framework for analyzing and predicting consumer conduct. Using a large-scale dataset of consumer transactions, we train and evaluate a range of machine learning algorithms like Clustering algorithms (k-means, hierarchical clustering), Decision tree algorithms (CART, random forest), Naive Bayes.

Our results demonstrate the effectiveness of machine learning in predicting consumer conduct, with significant improvements in accuracy and precision over traditional statistical approaches. The proposed framework provides businesses with a powerful tool for analyzing and influencing consumer behavior, enabling more effective marketing strategies, improved customer engagement, and increased revenue growth with high efficiency and high accuracy.

**KEYWORDS**: Consumer conduct, Machine learning, Behavioral patterns, Predictive analytics.

# LIST OF FIGURES

# LIST OF SCREEN SHOTS

# TABLE OF CONTENTS

# 1. INTRODUCTION

# 1. INTRODUCTION

## 1.1 PROJECT SCOPE

The project scope of Prediction of Consumer Conduct Using Machine Learning is to develop a machine learning model to predict consumer conduct by analyzing historical data and behavior patterns to anticipate consumer actions such click add or not, enabling better marketing and personalized recommendations. Various algorithms (e.g., classification, regression, or clustering) will be evaluated for optimal performance. The model will be assessed using accuracy, precision, recall, and other relevant metrics to ensure predictive reliability.

## 1.2 PROJECT PURPOSE

The purpose of this project is to develop a machine learning model that predicts consumer behavior, helping businesses make informed decisions based on data-driven insights. By analyzing historical data, behavior patterns, the model will enable businesses to anticipate customer actions, such as product purchases, engagement trends, or potential churn. This will allow companies to proactively retain at-risk customers, optimize marketing efforts, and deliver personalized recommendations, enhancing customer satisfaction and loyalty. The model will also help increase sales by identifying upselling opportunities and recommending products that align with consumer preferences. Ultimately, the goal is to transform consumer data into actionable insights that drive engagement, increase revenue, and support sustainable business growth.

## 1.3 PROJECT FEATURES

The project will include several key features to ensure accurate consumer behavior prediction. First, a robust data pipeline will be established to clean, preprocess, and transform consumer data for machine learning applications. The model will use advanced feature engineering to extract meaningful insights from various data sources, such as clicking add or not. Lastly, multiple machine learning algorithms will be evaluated, with the best-performing model integrated to provide actionable predictions with evaluation metrics.

# 2. SYSTEM ANALYSIS

# 2. SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

## 2.1 PROBLEM DEFINITION

The problem this project aims to solve is the challenge businesses face in predicting consumer behavior and making proactive decisions to enhance customer retention, engagement, and sales. With vast amounts of consumer data available, companies struggle to analyze it effectively and derive actionable insights that can improve marketing strategies and personalized offerings. Traditional methods often fail to capture the complexity of consumer decision-making, leading to missed opportunities in targeting, upselling, and customer relationship management. Unpredictable customer churn can result in lost revenue, while untargeted campaigns can increase marketing costs without delivering significant returns. Moreover, understanding what drives consumer engagement and how to anticipate future actions is essential for building long-term customer loyalty. The inability to predict these behaviors accurately prevents businesses from fully optimizing their customer experience and overall performance. Thus, developing a machine learning model to predict consumer conduct will address this gap, enabling companies to act on real-time insights, improve decision-making, and achieve sustainable growth. This solution will empower businesses to turn data into a competitive advantage in a rapidly evolving market.

## 2.2 EXISTING SYSTEM

Existing systems for predicting consumer conduct use algorithms like Naive Bayes, Random Forest, and Regression models to analyze historical data and detect behavior patterns. While these algorithms can provide correct predictions, they often consume significant computational time and may not be the most efficient. Among them, the Random Forest algorithm stands out for its accuracy, but there are newer algorithms that offer greater precision and efficiency. The existing systems also exhibit limited preprocessing capabilities, which can impact the quality of predictions. Moreover, these models are highly dependent on the attributes selected, which can lead to issues like overfitting. As a result, there is a need to explore more advanced algorithms that address these shortcomings for improved performance.

### 2.2.1 LIMITATIONS OF EXISTING SYSTEM

- High Complexity
- Limited Flexibility
- Scalability Issues
- Overfitting Risk
- Limited Preprocessing
- Implementation and Maintenance

## 2.3 PROPOSED SYSTEM

The proposed system introduces an ensemble learning method, specifically the Bagging (Bootstrap Aggregating) Classifier, to improve the accuracy and robustness of consumer conduct predictions. Bagging enhances traditional models like Logistic Regression and Random Forest by training multiple models on different data subsets and combining their predictions, thus reducing variance and improving overall performance. This approach aims to outperform individual classifiers by creating a more stable and accurate predictive model. In particular, the proposed system focuses on predicting whether a consumer will click on an advertisement, a critical behavior for optimizing marketing efforts. By aggregating the results of various models, Bagging increases prediction accuracy and mitigates the risk of overfitting, which is a common issue with traditional models. This ensemble method ensures that the system can better handle the complexities of consumer behavior. Ultimately, the system offers a more reliable and effective solution for businesses looking to optimize engagement and advertising strategies.

### 2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Enhanced Accuracy
- Scalability
- Predictive Analytics
- Flexibility and Adaptability
- Reduced Variance and Overfitting
- Performance Metrics

## 2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

### 2.4.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### 2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### 2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system instead, they must accept it as a necessity. The level of acceptance by the users solely depends on the methods employed to educate them about the system and to make them familiar with it.

## 2.5 HARDWARE AND SOFTWARE REQUIREMENTS

### 2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : i3 or above
- Hard disk : 64GB or above
- Memory : 4GB RAM or above

### 2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements.

- Operating system : Windows 8 or above
- Languages : Python (Version 3.8)

# 3. ARCHITECTURE

# 3. ARCHITECTURE

## 3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for consumer behavior prediction using machine learning, starting from input to final prediction.

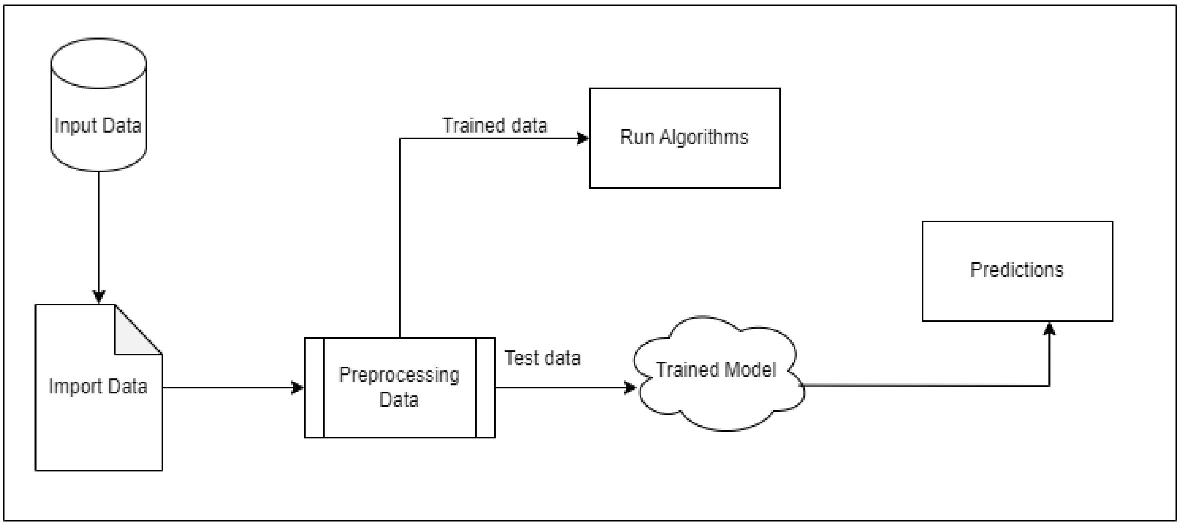



Figure 3.1: Project Architecture of Prediction of Consumer Conduct using Machine learning

## 3.2 DESCRIPTION

- **Input Data:** Input data is generally in .csv file format where the data is fetched and mapped in the data framed from the source columns.
- **Importing Data:** Convert the dataset into a usable format and preview it. After uploading the dataset, this module reads the CSV file into a pandas Dataframe, which is a tabular data structure suited for data manipulation and analysis.
- **Preprocess:** Prepare and clean the data, encode categorical variables, and visualize data distribution. To achieve the efficiency in computation we are going to normalize and clean the data values.
- **Training and test data:** Training data is passed to the Bagging Classifier to train the model. Test data is used to test the trained model whether it is making correct predictions or not.
- **Run Algorithms:** The purpose of choosing the Bagging classifier for this project the efficiency and accuracy that we have observed when compared to other classifiers. We also run different algorithms to compare their accuracy.

## 3.3 USE CASE DIAGRAM

In the use case diagram we have basically two actors who are the user and the system. The user has the rights to upload, preprocess and test the data and to view the results. Whereas all the process done within the system so results are stored or displayed by the system.

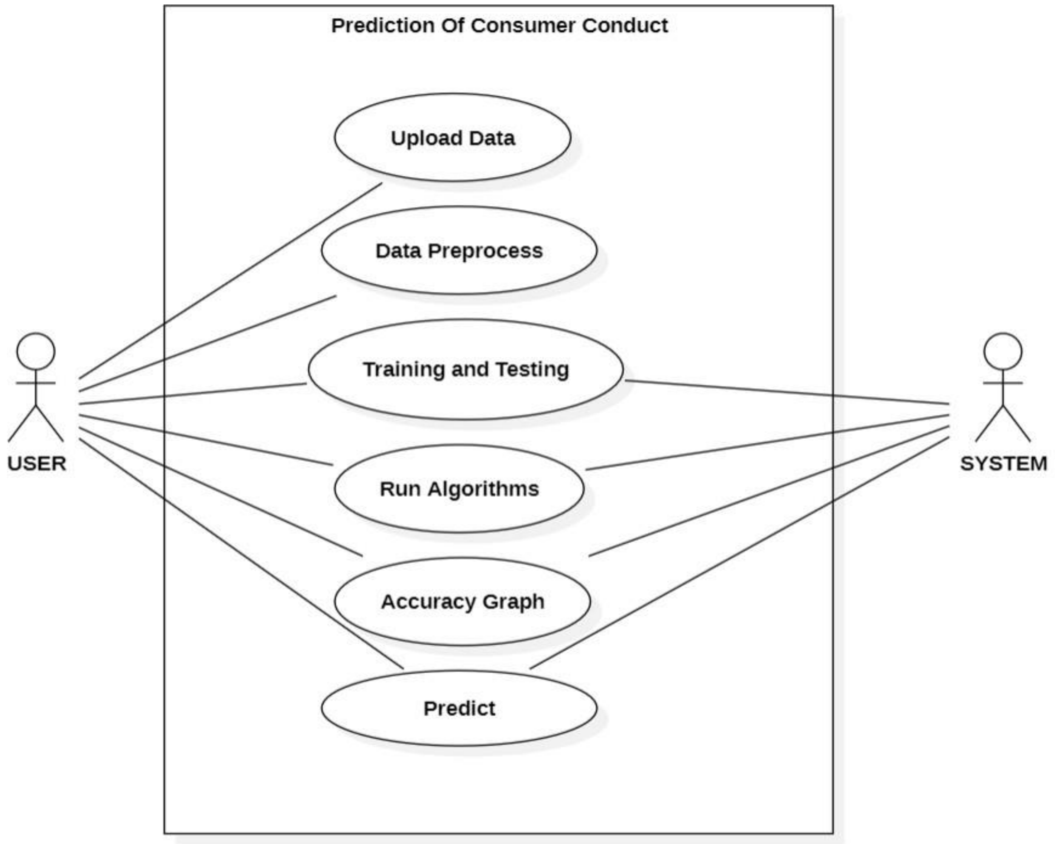



Figure 3.3: Use Case diagram for Prediction of Consumer Conduct using Machine Learning

## 3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects. It is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

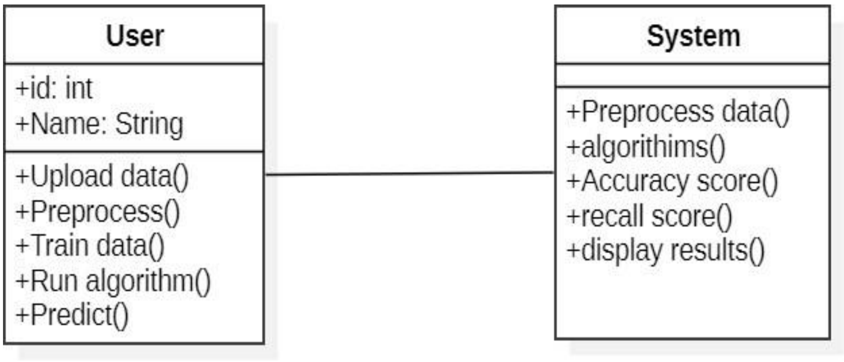



Figure 3.4: Class Diagram for Prediction of Consumer Conduct using Machine Learning

## 3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of system under development.

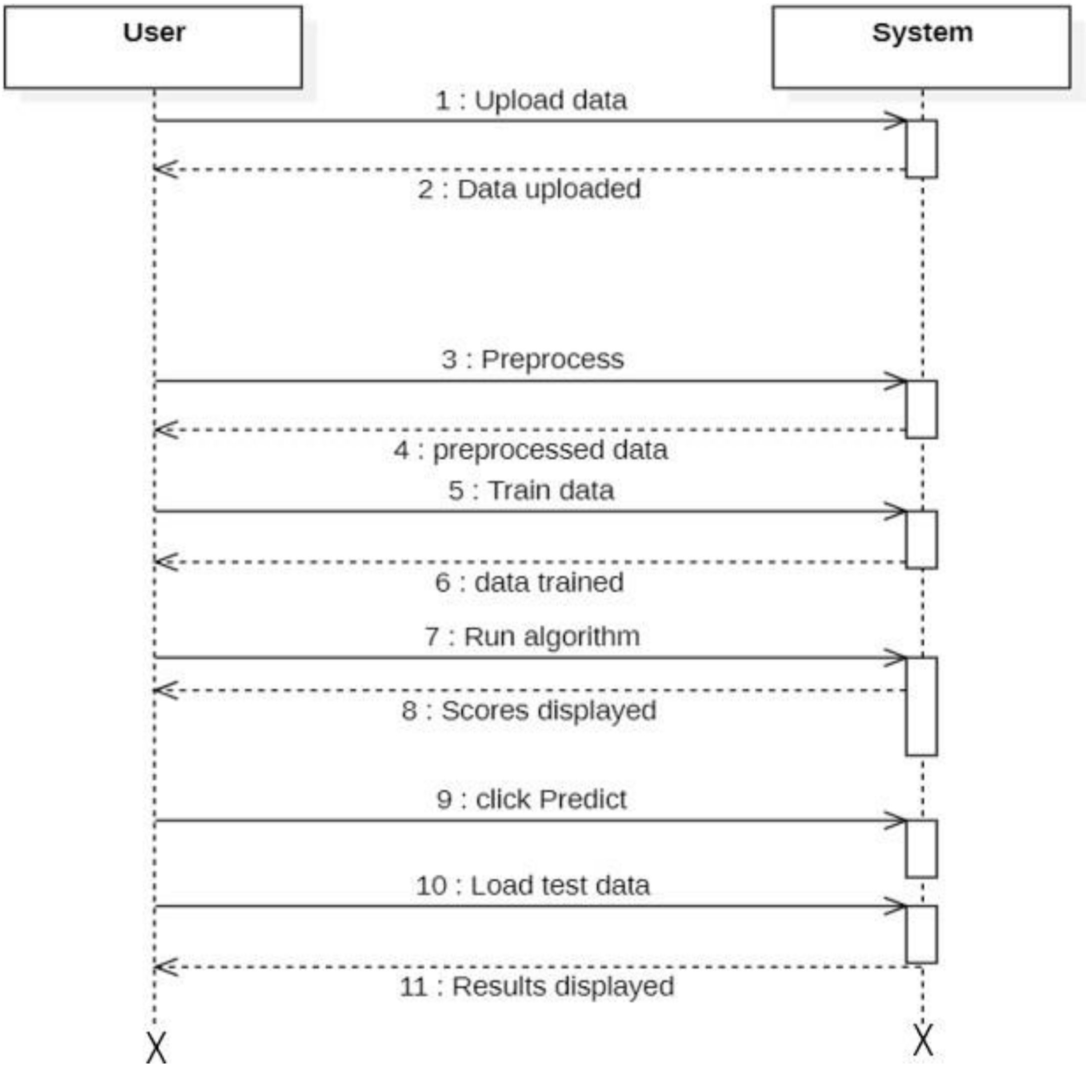



Figure 3.5: Sequence Diagram for Prediction of Consumer Conduct using Machine Learning

## 3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data.

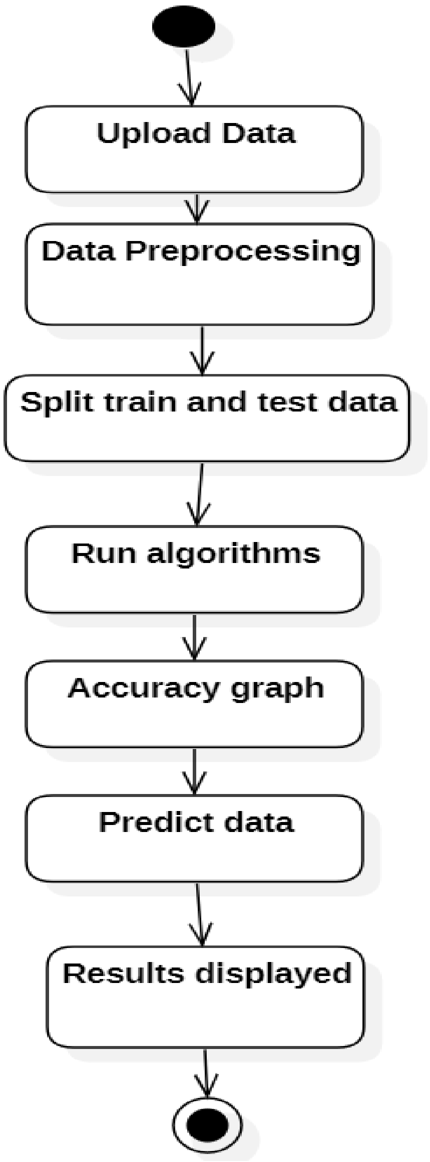



Figure 3.6: Activity Diagram for Prediction of Consumer Conduct using Machine Learning

# 4. IMPLEMENTATION

# 4. IMPLEMENTATION

## 4.1 SAMPLE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import LinearSVC
from sklearn.metrics import classification_report,f1_score,precision_score,recall_score
from sklearn.linear_model import LogisticRegression
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import BaggingClassifier

main = tkinter.Tk()
main.title("Consumer COnduct")
main.geometry("1300x1200")
global le1, le2, le3, le4, cls, extension_acc
def upload():
    global  filename
    global data
    text.delete('1.0', END)
```

```
    filename = askopenfilename(initialdir = "Dataset")
    pathlabel.config(text=filename)
    text.insert(END,"Dataset loaded\n\n")

def importdata():
    global filename
    global df
    df = pd.read_csv(filename,encoding = 'latin1')
    text.insert(END,"Data Information:\n"+str(df.head())+"\n")
    text.insert(END,"Columns Information:\n"+str(df.columns)+"\n")

def preprocess():
    global df
    global x,y
    X = df.drop(columns=['Clicked on Ad'])
    label_names = np.array(['No','Yes'])
    y = df['Clicked on Ad'].values
    feature_names  = np.array(list(X))
    x = np.array(X)
    sns.countplot(df["Clicked on Ad"])
    plt.show()
def plotCorrelationMatrix(df, graphWidth):
    #filename = df.dataframeName
    df = df.dropna('columns') # drop columns with NaN
    df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where there are
more than 1 unique values
    if df.shape[1] < 2:
        text.insert(END,f'No correlation plots shown: The number of non-NaN or constant
columns ({df.shape[1]}) is less than 2')
        return
    corr = df.corr()
    plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80, facecolor='w',
edgecolor='k')
    corrMat = plt.matshow(corr, fignum = 1)
    plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
    plt.yticks(range(len(corr.columns)), corr.columns)
```

```
    plt.gca().xaxis.tick_bottom()
    plt.colorbar(corrMat)
    plt.show()

def ttmodel():
    global le1, le2, le3, le4
    global x,y
    global df
    global X_train,X_test,y_train,y_test
    X_train,X_test,y_train,y_test = train_test_split(x,y,test_size=0.33,random_state=5)
    le1 = LabelEncoder()
    le2 = LabelEncoder()
    le3 = LabelEncoder()
    le4 = LabelEncoder()

    x[:,4] = le1.fit_transform(x[:,4])
    x[:,5] = le2.fit_transform(x[:,5])
    x[:,7] = le3.fit_transform(x[:,7])
    x[:,8] = le4.fit_transform(x[:,8])

    X_train[:,4] = le1.fit_transform(X_train[:,4])
    X_train[:,5] = le2.fit_transform(X_train[:,5])
    X_train[:,7] = le3.fit_transform(X_train[:,7])
    X_train[:,8] = le4.fit_transform(X_train[:,8])

    X_test[:,4] = le1.fit_transform(X_test[:,4])
    X_test[:,5] = le2.fit_transform(X_test[:,5])
    X_test[:,7] = le3.fit_transform(X_test[:,7])

    X_test[:,8] = le4.fit_transform(X_test[:,8])

    text.insert(END,"Train Shape: "+str(X_train.shape)+"\n")
    text.insert(END,"Test Shape: "+str(X_test.shape)+"\n")

    plotCorrelationMatrix(df, len(df.columns))
```

```
def mlmodels():
    global x,y
    global X_train,X_test,y_train,y_test,cls, extension_acc
    global lr_acc,svc_acc,rfc_acc,gnb_acc,dtc_acc
    clf_lr = LogisticRegression(random_state=0)
    clf_lr.fit(X_train,y_train)
    pred = clf_lr.predict(X_test)
    lr_acc=clf_lr.score(X_test, y_test)
    text.insert(END,"LOGIT Accuracy: "+str(clf_lr.score(X_test, y_test))+"\n")
    text.insert(END,"LOGIT recall_score: "+str(recall_score(y_test,pred))+"\n")
    text.insert(END,"LOGIT precision_score: "+str(precision_score(y_test,pred))+"\n")
    text.insert(END,"LOGIT f1_score: "+str(f1_score(y_test,pred))+"\n\n")

    clf_svc = LinearSVC(random_state=0)
    clf_svc.fit(X_train,y_train)
    clf_svc.score(X_test,y_test)
    pred = clf_svc.predict(X_test)
    svc_acc=clf_svc.score(X_test, y_test)
    text.insert(END,"SVC Accuracy: "+str(clf_svc.score(X_test, y_test))+"\n")
    text.insert(END,"SVC recall_score: "+str(recall_score(y_test,pred))+"\n")
    text.insert(END,"SVC precision_score: "+str(precision_score(y_test,pred))+"\n")
    text.insert(END,"SVC f1_score: "+str(f1_score(y_test,pred))+"\n\n")
    clf_gnb = GaussianNB()
    clf_gnb.fit(X_train,y_train)
    clf_gnb.score(X_test,y_test)
    pred = clf_gnb.predict(X_test)
    gnb_acc=clf_gnb.score(X_test, y_test)
    text.insert(END,"Naive Bayes Accuracy: "+str(clf_gnb.score(X_test, y_test))+"\n")
    text.insert(END,"Naive Bayes recall_score: "+str(recall_score(y_test,pred))+"\n")
    text.insert(END,"Naive Bayes precision_score: 
"+str(precision_score(y_test,pred))+"\n")
    text.insert(END,"Naive Bayes f1_score: "+str(f1_score(y_test,pred))+"\n\n")
    clf_rfc = RandomForestClassifier(random_state=0)
    clf_rfc.fit(X_train,y_train)
    clf_rfc.score(X_test,y_test)
```

```
  rfc_acc=clf_rfc.score(X_test, y_test)
   text.insert(END,"Random Forest Accuracy: "+str(clf_gnb.score(X_test, y_test))+"\n")
   text.insert(END,"Random Forest recall_score: "+str(recall_score(y_test,pred))+"\n")
   text.insert(END,"Random Forest precision_score: 
"+str(precision_score(y_test,pred))+"\n")
   text.insert(END,"Random Forest f1_score: "+str(f1_score(y_test,pred))+"\n\n")

   clf_dtc = DecisionTreeClassifier(random_state=0)
   clf_dtc.fit(X_train,y_train)
   clf_dtc.score(X_test,y_test)
   pred = clf_dtc.predict(X_test)
   dtc_acc=clf_rfc.score(X_test, y_test)
   text.insert(END,"Decision Tree Accuracy: "+str(clf_dtc.score(X_test, y_test))+"\n")
   text.insert(END,"Decision Tree recall_score: "+str(recall_score(y_test,pred))+"\n")
   text.insert(END,"Decision Tree precision_score: 
"+str(precision_score(y_test,pred))+"\n")
   text.insert(END,"Decision Tree f1_score: "+str(f1_score(y_test,pred))+"\n\n")

   cls = BaggingClassifier()
   cls.fit(x,y)
   cls.score(X_test,y_test)
   pred = cls.predict(X_test)
   extension_acc =cls.score(X_test, y_test)
   text.insert(END,"Extension Bagging Classifier Accuracy: "+str(extension_acc)+"\n")
   text.insert(END,"Extension Bagging Classifier recall_score: 
"+str(recall_score(y_test,pred))+"\n")
   text.insert(END,"Extension Bagging Classifier precision_score: 
"+str(precision_score(y_test,pred))+"\n")
   text.insert(END,"Extension Bagging Classifier f1_score: 
"+str(f1_score(y_test,pred))+"\n\n")
def predict():
   global cls
   global le1, le2, le3, le4
   text.delete('1.0', END)
```

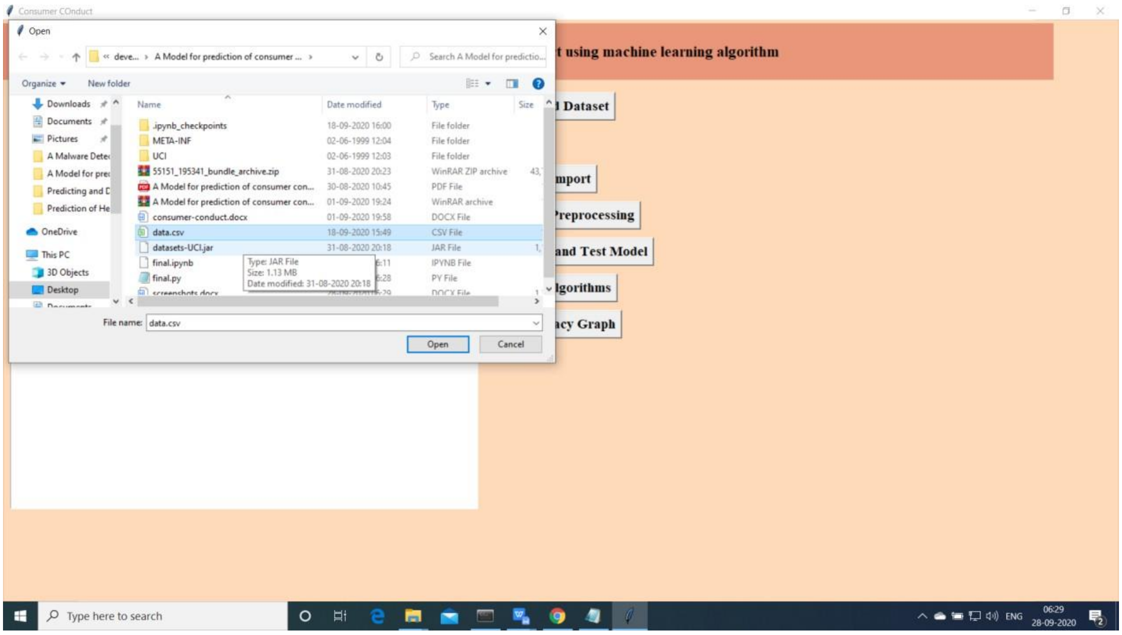
```
    dataset = pd.read_csv(filename,encoding='latin1')
    dataset.fillna(0, inplace = True)
    dataset = dataset.values
    XX = dataset[:,0:dataset.shape[1]]
    XX[:,4] = le1.fit_transform(XX[:,4])
    XX[:,5] = le2.fit_transform(XX[:,5])
    XX[:,7] = le3.fit_transform(XX[:,7])
    XX[:,8] = le4.fit_transform(XX[:,8])

    prediction = cls.predict(XX)
    print(prediction)
    for i in range(len(prediction)):
        if prediction[i] == 0:
            text.insert(END,"Test DATA : "+str(dataset[i])+" ===> PREDICTED AS
CONSUMER NOT CLICKED ON ADD\n\n")
        if prediction[i] == 1:
            text.insert(END,"Test DATA : "+str(dataset[i])+" ===> PREDICTED AS
CONSUMER CLICKED ON ADD\n\n")

def graph():
    global lr_acc,svc_acc,rfc_acc,gnb_acc,dtc_acc, extension_acc

    height  = [lr_acc,svc_acc,rfc_acc,gnb_acc,dtc_acc,  extension_acc]
    bars = ('Logit', 'SVC','RFC','GNB','DT', 'Extension BaggingClassifier')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos,  height)
    plt.xticks(y_pos, bars)
    plt.show()
font = ('times', 16, 'bold')
title = Label(main, text='A Model for prediction of consumer conduct using machine
learning algorithm')
title.config(bg='dark salmon', fg='black')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
```

```
font1 = ('times', 14, 'bold')
upload = Button(main, text="Upload Dataset", command=upload)
upload.place(x=900,y=100)
upload.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='dark orchid', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=900,y=150)

ip = Button(main, text="Data Import", command=importdata)
ip.place(x=900,y=200)
ip.config(font=font1)

pp = Button(main, text="Data Preprocessing", command=preprocess)
pp.place(x=900,y=250)
pp.config(font=font1)

tt = Button(main, text="Train and Test Model", command=ttmodel)
tt.place(x=900,y=300)
tt.config(font=font1)

ml = Button(main, text="Run Algorithms", command=mlmodels)
ml.place(x=900,y=350)
ml.config(font=font1)
gph = Button(main, text="Accuracy Graph", command=graph)
gph.place(x=900,y=400)
gph.config(font=font1)

predictButton = Button(main, text="Predict Consumer Conduct from Test Data",
command=predict)
predictButton.place(x=900,y=450)
predictButton.config(font=font1)
font1 = ('times', 12, 'bold')
text=Text(main,height=30,width=110)
```

```
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=100)
text.config(font=font1)

main.config(bg='peach puff')
main.mainloop()
```

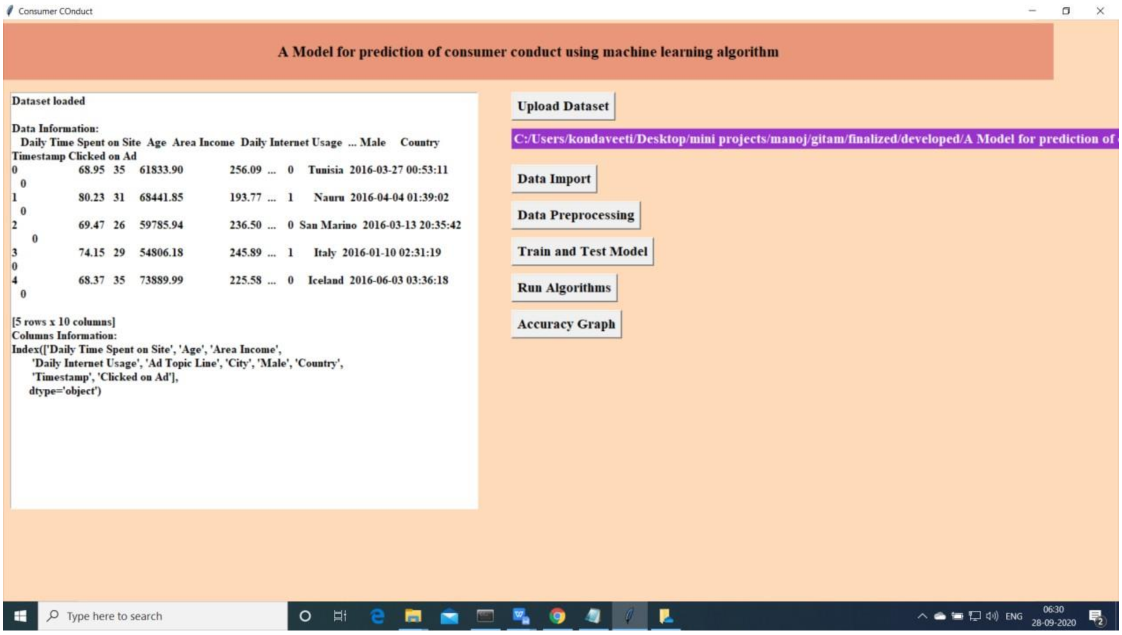
# 5. SCREENSHOTS

## 5.1 UPLOAD DATA RESULT



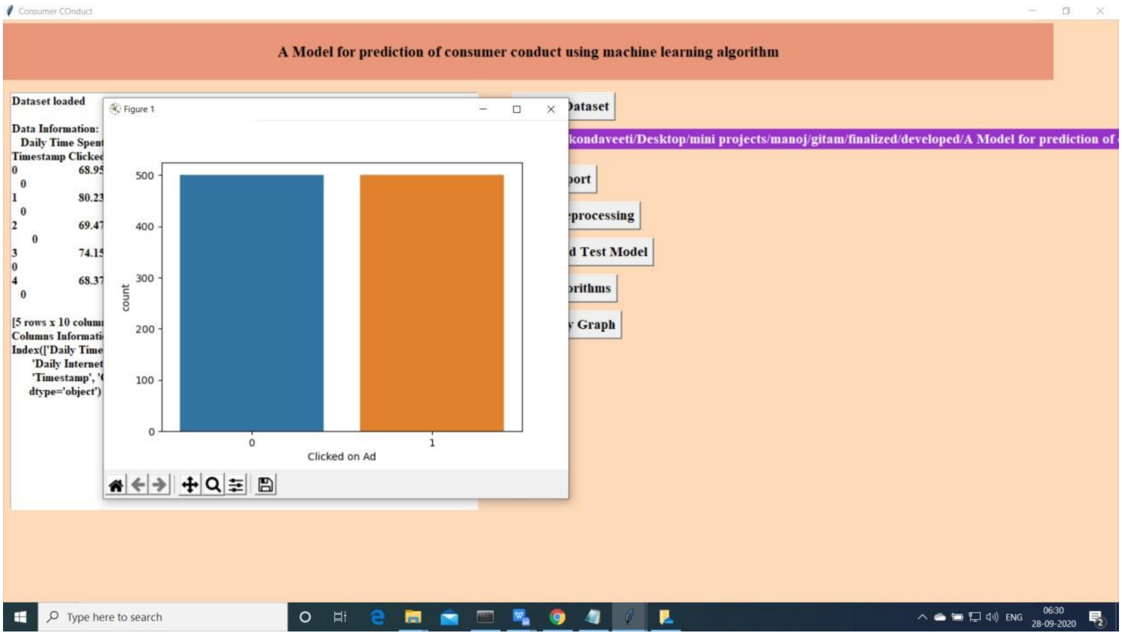

Screenshot 5.1: Upload data result of Prediction of Consumer Conduct using Machine Learning

## 5.2 IMPORT DATA RESULT



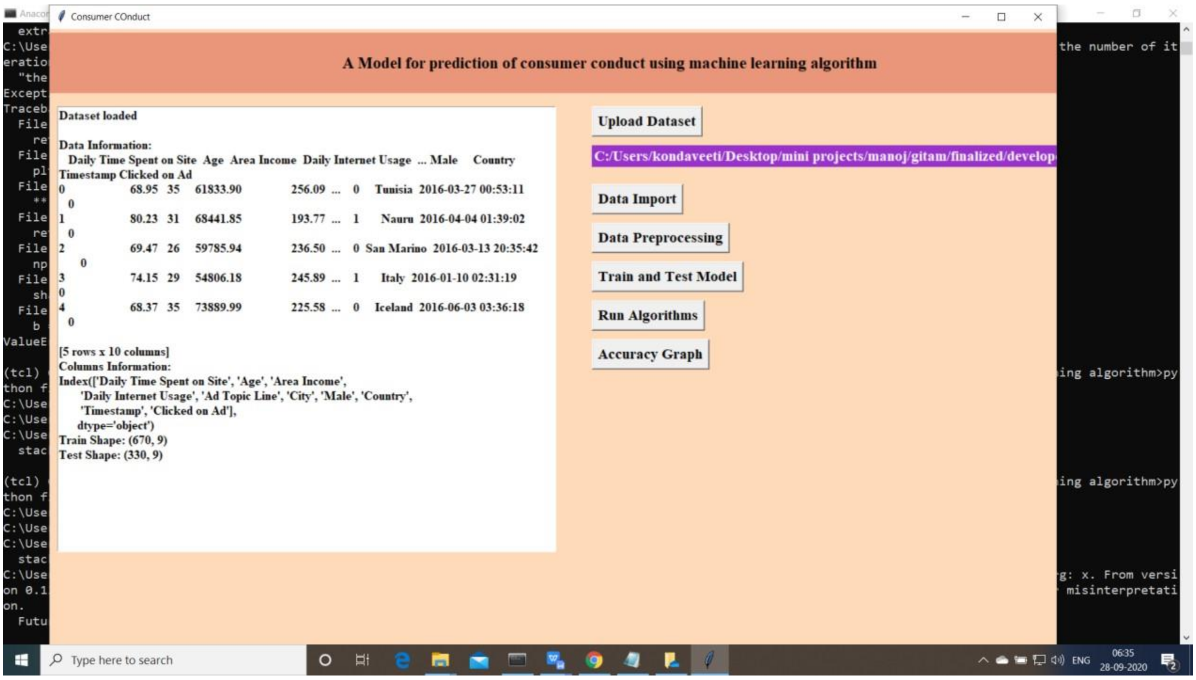

Screenshot 5.2: Import data result of Prediction of Consumer Conduct using Machine Learning

## 5.3 PREPROCESS DATA RESULT



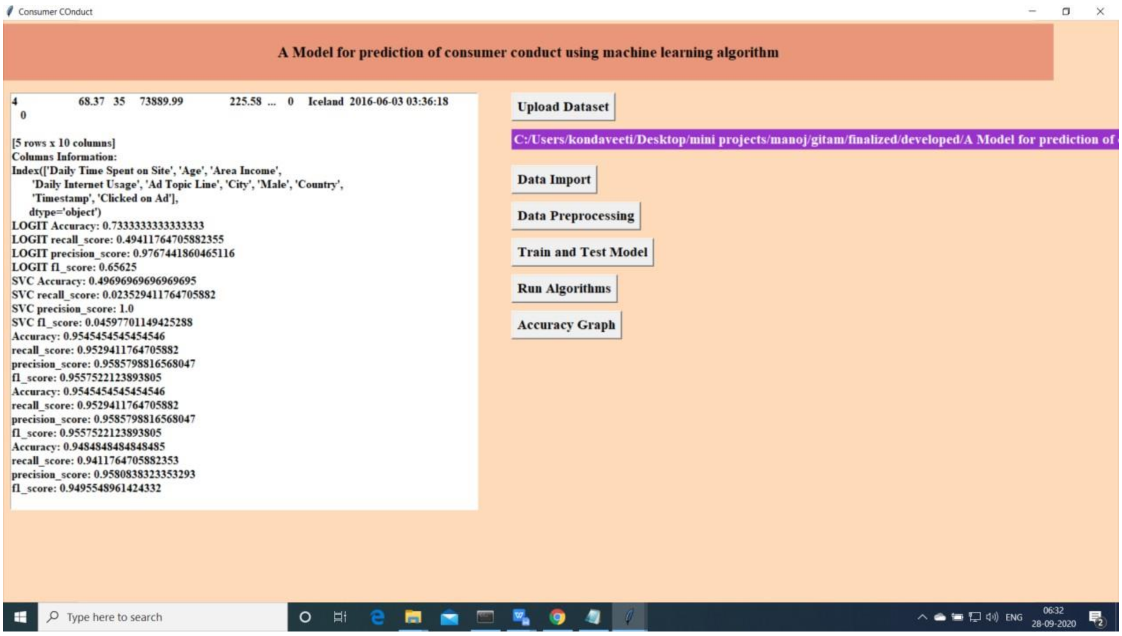

Screenshot 5.3: Preprocess data of Prediction of Consumer Conduct using Machine Learning

## 5.4 TRAIN AND TEST DATA RESULT



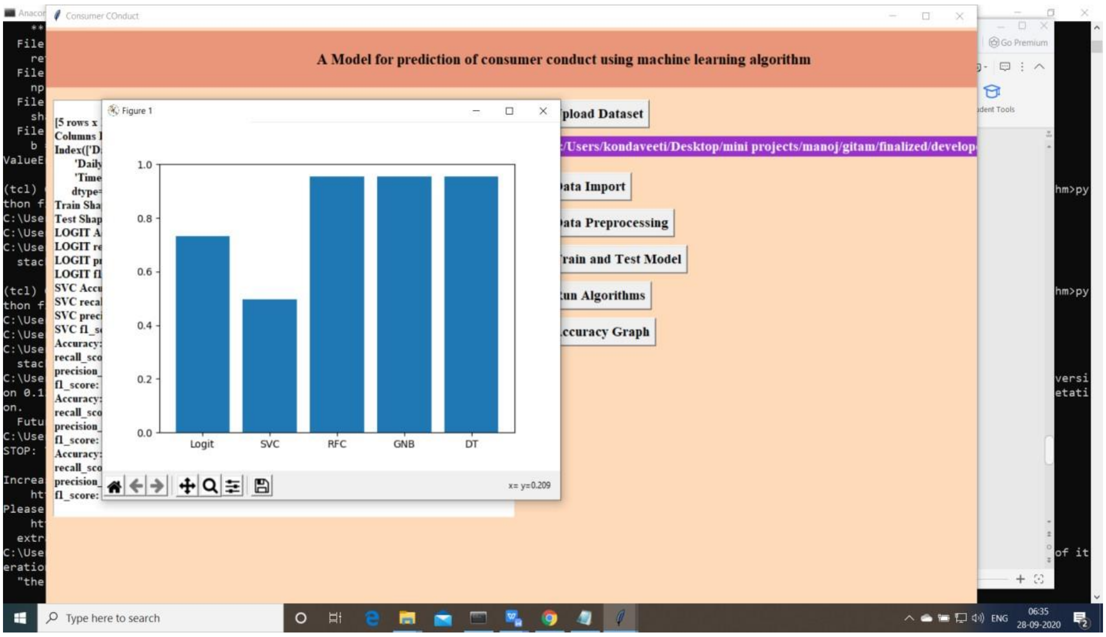

Screenshot 5.4: Train and Test data result of Prediction of Consumer Conduct using Machine Learning

## 5.5 RUN ALGORITHMS RESULT



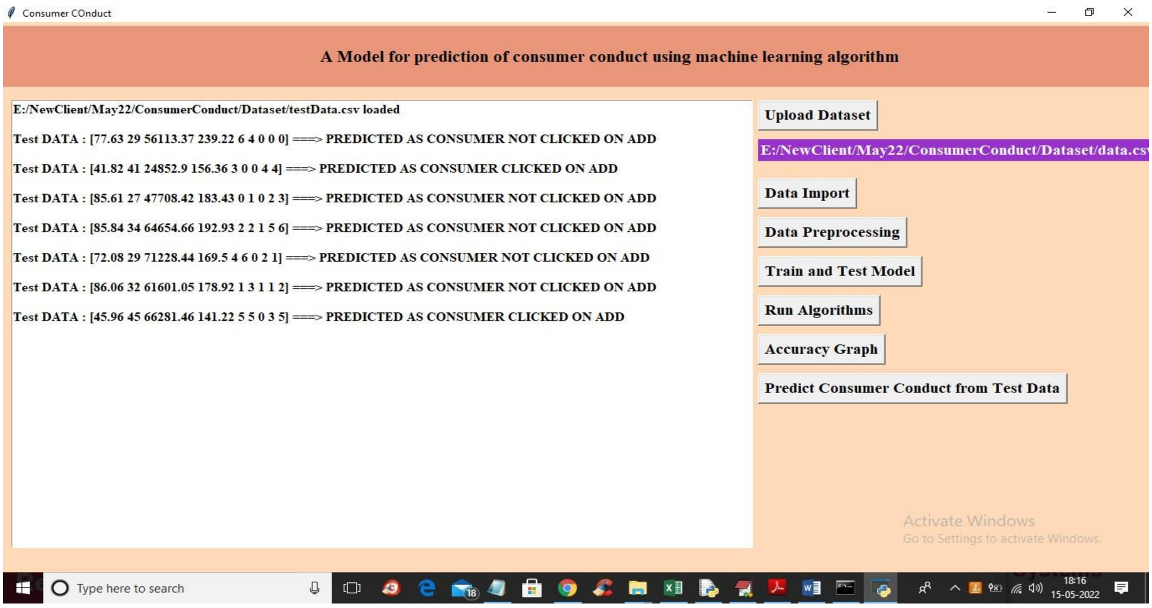

Screenshot 5.5: Run Algorithms result of Prediction of Consumer Conduct using Machine Learning

## 5.6 ACCURACY GRAPH RESULT




Screenshot 5.6: Accuracy graph result of Prediction of Consumer Conduct using Machine Learning

## 5.7 PREDICTION RESULTS




Screenshot 5.7: Prediction result of Prediction of Consumer Conduct using Machine Learning

# 6. TESTING

# 6. TESTING

## 6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## 6.2 TYPES OF TESTING

### 6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. Unit tests perform basic tests at component level and test a specific business process, application, and system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### 6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### 6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

| | |
|---|---|
| Valid Input | : identified classes of valid input must be accepted. |
| Invalid Input | : identified classes of invalid input must be rejected. |
| Functions | : identified functions must be exercised. |
| Output | : identified classes of application outputs must be exercised. |
| Systems/Procedures | : interfacing systems or procedures must be invoked. |

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows, data fields, predefined processes.

### 6.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

### 6.2.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### 6.2.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

## 6.3 TEST CASES

### 6.3.1 UPLOADING FILE

| Test case ID | Test case name | Purpose | Test Case | Output |
|---|---|---|---|---|
| 1 | User uploads valid data | Use it for predicting | The user uploads the consumer data as a csv file. | Uploaded successfully |
| 2 | User uploads invalid data | Use it for prediction | The user uploads the different consumers behavior data as a csv file | Not uploaded successfully |

### 6.3.2 PREDICTION

| Test case ID | Test case name | Purpose | Input | Output |
|---|---|---|---|---|
| 1 | Basic Functionality Test | To verify the model can make predictions on a valid, balanced dataset | A csv file is given | Model returns accurate predictions with no errors |
| 2 | Data Type Validation | To verify the model rejects invalid data types in the input | An another csv fileis given | Model throws a meaningful error for invalid data types |

# 7. CONCLUSION

# 7. CONCLUSION AND FUTURE SCOPE

## 7.1 PROJECT CONCLUSION

In conclusion, the proposed system utilizing the Bagging Classifier represents a significant advancement in predicting consumer conduct compared to traditional machine learning models. By employing an ensemble learning approach, this method enhances the robustness and accuracy of predictions, allowing businesses to make more informed decisions regarding consumer engagement. The ability to effectively predict whether a consumer will click on an advertisement is particularly valuable for optimizing marketing strategies and increasing conversion rates. Furthermore, the system addresses the limitations of existing models, such as overfitting and reliance on specific attributes, by aggregating the results of multiple classifiers. Additionally, the use of Bagging reduces variance and improves the overall performance of predictions, making it a suitable choice for dynamic market environments. Overall, the Bagging Classifier not only improves predictive accuracy but also enhances the potential for targeted marketing and personalized customer experiences. This innovative approach has the potential to transform how businesses approach consumer behavior analysis, paving the way for more effective and efficient marketing strategies in the future. Ultimately, by leveraging cutting-edge machine learning techniques, companies can unlock new opportunities for growth and customer engagement.

## 7.2 FUTURE SCOPE

The future scope of the proposed Bagging Classifier system for predicting consumer conduct is promising and multifaceted. As advancements in machine learning continue, integrating other ensemble techniques, such as Gradient Boosting or Stacking, could further enhance predictive performance and accuracy. Additionally, incorporating real-time data processing and analysis will allow businesses to adapt to changing consumer behaviors dynamically and improve response times for marketing campaigns. Ultimately, this evolution of the predictive model can lead to more personalized and effective marketing strategies, driving higher engagement and customer satisfaction in an increasingly competitive landscape.

# 8. BIBLIOGRAPHY

# 8. BIBLIOGRAPHY

## 8.1 REFERENCES

[1] Webb, G.I., Boughton, J., Wang, Z.: Not so naive bayes: Aggregating one-dependence estimators. Machine Learning 58, 5– 24 (2005) zbMATHCrossRef Google Scholar.

[2] Chen S., Martinez A.M., Webb G.I. (2014) Highly Scalable Attribute Selection for Averaged One-Dependence Estimators. In: Tseng V.S., Ho T.B., Zhou ZH., Chen A.L.P., Kao HY. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2014. Lecture Notes in Computer Science, vol 8444. Springer, Cham.

[3] Witten, I.H., Frank, E.: Data mining-Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann, San Francisco (2000), http://prdownloads.sourceforge.net/weka/datasets-UCI.jar Google Scholar

[4] Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988) Google Scholar

[5] Merz, C., Murphy, P., Aha, D.: UCI repository of machine learning databases. In Dept of ICS, University of California, Irvine(1997), http://www.ics.uci.edu/mlearn/MLRepository.html

[6] Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pp. 339–406 (1994) Google Scholar

## 8.2 WEBSITES

[1] https://github.com/chintalapudipiyush/ConsumerConduct