ONLINE SHOPPERS INTENTION USING IBM WATSON

A UG PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

BOINI RUTHWIK	**19UK1A05D4**
GADE ROHAN REDDY	**19UK1A05C6**
KOUDAGANI SIDDHU	**19UK1A05D0**
BANDI DEVENDAR	**19UK1A05C0**

Under the esteemed guidance of

Ms. K. NAVYA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
(Affiliated to JNTUH, Hyderabad)
Bollikunta, Warangal – 506005

2019– 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005
2019 – 2023

CERTIFICATE OF COMPLETION
UG PROJECT PHASE-1

This is to certify that the UG Project Phase-1 entitled "**Online shoppers intention using IBM Watson"** is being submitted by **B.RUTHWIK(H.NO:19UK1A05D4),G.ROHAN REDDY(19UK1A05C6),K.SIDDHU(19UK1A05D0),B.DEVENDAR(19UK1A05C0)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** during the academic year **2022-23**, is a record of work carried out by them under the guidance and supervision.

Project Guide
Ms. K. NAVYA
(Assistant Professor)

Head of the Department
Dr. R. Naveen Kumar
(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr. R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **Ms.K.NAVYA** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

B. RUTHWIK (19UK1A05D4)
G. ROHAN REDDY (19UK1A05C6)
D. SIDDHU (19UK1A05D0)
B. DEVENDAR (19UK1A05D0)

ABSTRACT

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus the customer will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as flipkart or ebay. Since the application is available in the Smartphone it is easily accessible and always available

TABLE OF CONTENTS:-

LIST OF FIGURES PAGE NO

1. INTRODUCTION

MOTIVATION:

Retail shopping is continuing to shift to E-commerce shopping and as a result the dynamics of shopping is changing around the world. E-commerce has already become a major form of retail market. Online customers often browse pages of e-commerce sites before they place orders or abandon their browsing without purchase. This information can help businesses to better cater to customer preferences and help both the business and customers mutually by recommending products specific to each customer and therefore increasing sales for the businesses. However, most of the time customers visiting these online websites may not make any purchase at all. This could be for various reasons i.e., Price of product or window shopping. It is important to predict customers' purchasing intention so that retention measures like e.g., recommending suitable products can be taken to convert potential customers into purchasers. Currently, the closest existing solution to this problem has been the recommendation system. Where previous purchase information from a customer is processed to predict the types of products a customer would be interested in. There is evidence [12] to suggest that retention measures such as an apt recommendation system plays an extremely important part in converting sales. Here the prediction of customer purchase intention can help strategize different marketing strategies and could be added to the mechanism of the recommendation system[13] of an ecommerce retailer. An example could be that if the ML solution predicts a strong customer purchase intention, then maybe the recommend system could recommend a higher quality or a more expensive product as it can be inferred that the user would be willing to consider a better or more expensive product if their intention to purchase a type of product is very strong. If the solution predicts a low purchase intention, then recommendation could recommend products that are on discount or products with special offers . Later this historical data of customer intention changes with such recommendation can also be studied and be applied to improve the recommendation system itself further. However, this report focuses only to the extent of predicting customer purchase intention.

1.1. PROBLEM DEFINITION:

Machine learning models are identified and implemented for solving the task of classifying a web shop visitor as aborting or non-aborting, in the following referred to as no buying and buying sessions.

The models are applied to different data types, namely clickstream data generated by each visitor of the web shop as well as customer data if a visitor could be identified. This is done to establish which model and data is best-suited for the task of predicting the buying probability of an online shopping session, in terms of performance, latency, and comprehensibility. Latency is important since in the use case, predictions have to be conducted in real-time and model comprehensibility is considered since the demand for explaining decisions made by machine learning models is rising. The boosted tree model that is already being used by the German clothing retailer, will act as a baseline for the comparison of the different algorithms. To provide further insight and reasoning about the varying performances on different datasets, an exploratory data analysis on the clickstream and static customer data is conducted. An RNN, representing the class of stateful machine learning models, is addition- ally trained on datasets, which required fewer feature engineering, to show if stateful models provide good results while reducing the need for feature engineering.

After obtaining the results of the different models on all datasets, the models are tested for their robustness to different conditions. Examples of such conditions are the gender of the visitor or the device on which the web shop was visited. This analysis indicates how the models will perform under real conditions after deployment.

Deployment is out of scope for this research and can only be regarded as an implication if a tested model outperforms the baseline model.

1.2. PROJECT OBJECTIVE:

The objective of the project is to make an application in android platform to purchase items in an existing shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an android application with web view.

1.3. ORGANIZATION OF DOCUMENTATION:

The remainder of this report first discusses findings from the literature, analyzing model performance on similar tasks as the one at hand in Chapter 2. Then the methods, consisting of the research framework, the utilized algorithms and software as well as the evaluation metrics are described in Chapter 3. The methodology chapter is followed by the data Chapter 4 discussing data pre-processing, a first data analysis and the used features. Then the implementation including hyper-parameter tuning is described in Chapter 5. The results are presented in Chapter 6 and dis-cussed afterward in Chapter 7. The thesis ends in Chapter 8 with a conclusion, giving answers to all above-stated research questions

2.PROBLEM STATEMENT

People nowadays are more prefer on online purchasing than offline purchasing. We want to investigate what are the factor that influencing their intention to online purchase which are by the relationship between gender and intention to online purchase, relationship between trust and intention to online purchase, relationship between risk to online purchase and also the relationship between customer satisfaction and intention to online purchase.

Difference gender differences perceptions on the online purchase. Whether male or female are tend to online shopping might possible to our study. Basically, male is less buying in online rather than female. Female have their own reasons why they more tend to online purchase and might be there have much choices that can make they to online purchase. Other than that, male are likely to purchase their things at the shop because of their satisfaction to see the existing product in the shop and want to test the product itself.

The consumer that purchases online might have high in trust on online purchase. Is the online purchase can be believed or not? They can make their assumptions based on the online shop that their friends and family purchase it before. Consumer also can see through their feedback of the customer of their services and products whether in good condition or not. They can analyze by their own to make the response towards the online shop.

After the consumer trust to the online purchase, the consumer must bear the risk of the product whether in the late of delivery or actual product might be difference from display product at their websites and others. The consumer will take a risk on online purchase because the perception on the online is not same in their real perceptions.

The consumer satisfaction will lead to consumer repeat purchase. On online purchase, if the service that provided from seller is the services is good and in efficiently the consumer will try to make the repetition of purchase. Other than the easiness of purchase through online, they can make the consumer satisfied with the products and services that the seller gave to them.

3.LITERATURE SURVEY

INTRODUCTION

This chapter reviews the results and algorithms used by studies concerned with a similar binary classification problem as the one at hand. To create a broader under-standing of the problem and the different algorithms, binary classification and the algorithms used for solving it are each explained before reviewing their application and performance in literature. All reviewed literature is summarized in Table 2.1, showing details on the data and results obtained in each study. Looking at the table one has to keep in mind that comparing results across studies is difficult since different data types, evaluation metrics, evaluation thresholds, etc. were used. Finally, a gap analysis of the reviewed literature is conducted.

Understanding consumer’s internet purchase intention in Malaysia.

This study aims to explore the antecedents relating to the extent of both the attitude and the purchasing intention of online shopping developed by Narges Delafrooz, Laily H. Paim and Ali Khatibi. It examined the factors influencing consumers’ attitude toward online shopping and shopping intention from the Malaysian perspectives. From an e-commerce perspective, the understanding of the Theory of Reasoned Action (TRA), Theory of Planned Behavior (TPB), and Technology Acceptance Model (TAM) could provide a valid basis in explaining and predicting consumers’ intention towards adopting an online shopping behaviour. The data were examined using frequency and path analysis. Result of path analysis showed that trust and attitude had stronger direct effect on online shopping intention, whereas utilitarian orientation, convenience, prices wider selection, and income had stronger indirect effect on online shopping intention through the attitude towards online shopping as mediation. This paper outlined the key online shopping intention and events in Malaysia and pioneered the building of an integrated research framework to understand how consumers form their attitude and make purchase intentions toward online shopping.

Perceived Risk, Perceived Technology, Online Trust for the Online Purchase Intention.

The aim of this research is to evaluate and validate the impacts of perceived technology and perceived risk on the online trust and how online trust is related to online purchase intention developed by Kwek Choon Ling. Methods used are research design, Questionnaire Design, sampling, administration of survey and data analysis. This research finding concludes that there is a positive relationship between perceived technology and online trust. In other words, the finding of this study supports the argument from Koufaris and Hampton-Sosa (2004). The study of Koufaris and Hampton-Sosa (2004) describes
websites that are easy-to-use and useful (i.e., both important components of perceived technology) may lead to increase trust from new customers towards the company.

EXISTING SYSTEM:

The current system for shopping is to visit the shop manually and from the available product choose the item customer want and buying the item by payment of the price of the item .

1. It is less user-friendly.
2. User must go to shop and select products.
3. It is difficult to identify the required product.
4. Description of the product limited.
5. It is a time-consuming process
6. Not in reach of distant users

PROPOSED SYSTEM:

In the proposed system customer need not go to the shop for buying the products. He can order the product he wish to buy through the application in his Smartphone. The shop owner will be admin of the system. Shop owner can appoint moderators who will help owner in managing the customers and product orders. The system also recommends a home delivery system for the purchased products

Here we are building a model by applying various machine learning algorithms to find the best accurate model. Some of the machines learning algorithms are:

1. Linear Regression:

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog).

Steps to implement Linear regression model:

1. Initialize the parameters.
2. Predict the value of a dependent variable by given an independent variable.
3. Calculate the error in prediction for all data points.
4. Calculate partial derivative w.r.t a0 and a1.
5. Calculate the cost for each number and add them

2. Multiple Linear Regression:

Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable. Multiple regression is a broader class of regressions that encompasses linear and nonlinear regressions with multiple explanatory variables.

3. Random Forest:

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

Working of Random Forest Algorithm

Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result...

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

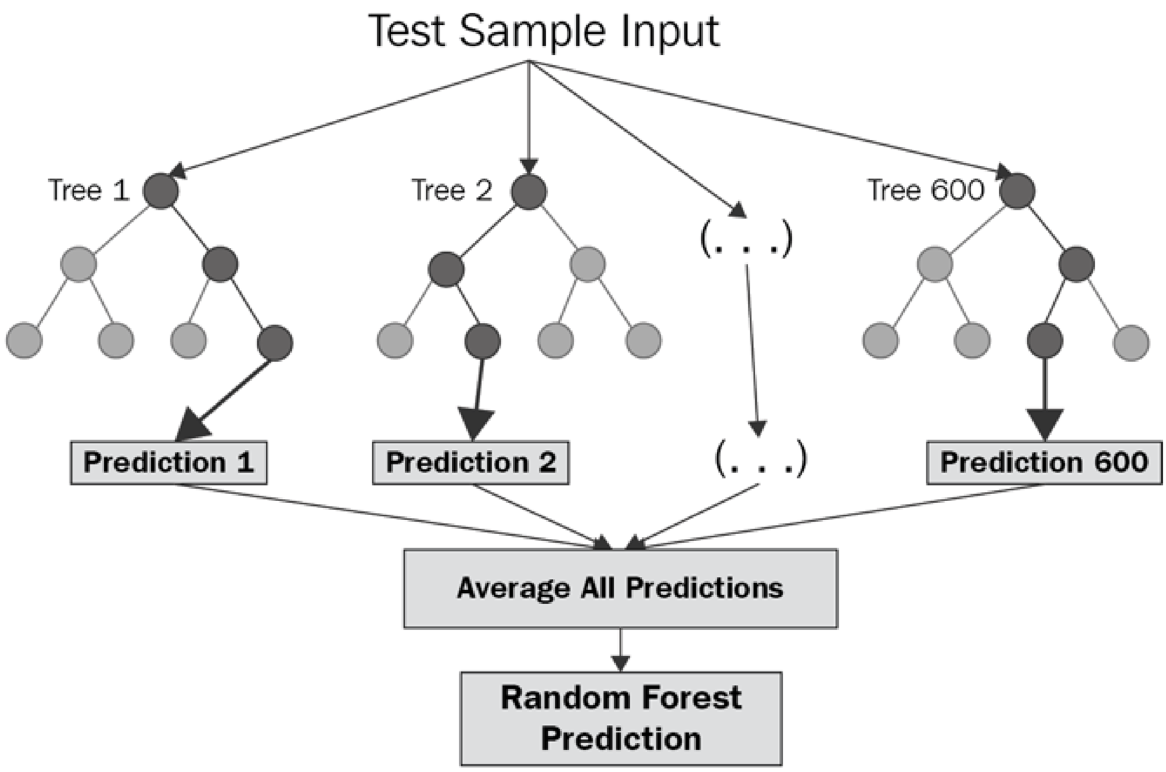


Figure 1 : Random Forest

4. Logistic regression:

- Logistic regression is **a supervised learning classification algorithm used to predict the probability of a target variable**. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes....

  Mathematically, a logistic regression model predicts P(Y=1) as a function of X.
- Logistic Regression is used when the dependent variable (target) is categorical. For example,
- To predict whether an email is a spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)

You will need to train the datasets to run smoothly and see an incremental improvement in the prediction rate.

5. k-nearest neighbor algorithm:

- It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbor's to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- K-Nearest Neighbors (KNN) is one of the simplest algorithms used in **Machine Learning for regression and classification problem**. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). The data is assigned to the class which has the nearest neighbors.
- It's also worth noting that the KNN algorithm is also part of a family of –lazy learning‖ models, meaning that it only stores a training dataset versus undergoing a training stage. This also means that all the computation occurs when a classificationor prediction is being made. Since it heavily relies on memory to store all its training data, it is also referred to as an instance-based or memory-based learning method.

- The K-NN working can be explained on the basis of the below algorithm:

  Step-1: Select the number K of the neighbors

  Step-2: Calculate the Euclidean distance of **K number of neighbors**

  Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

  Step-4: Among these k neighbors, count the number of the data points in each category.

  Step-5: Assign the new data points to that category for which the number of the neighboris maximum.

  Step-6: Our model is ready.

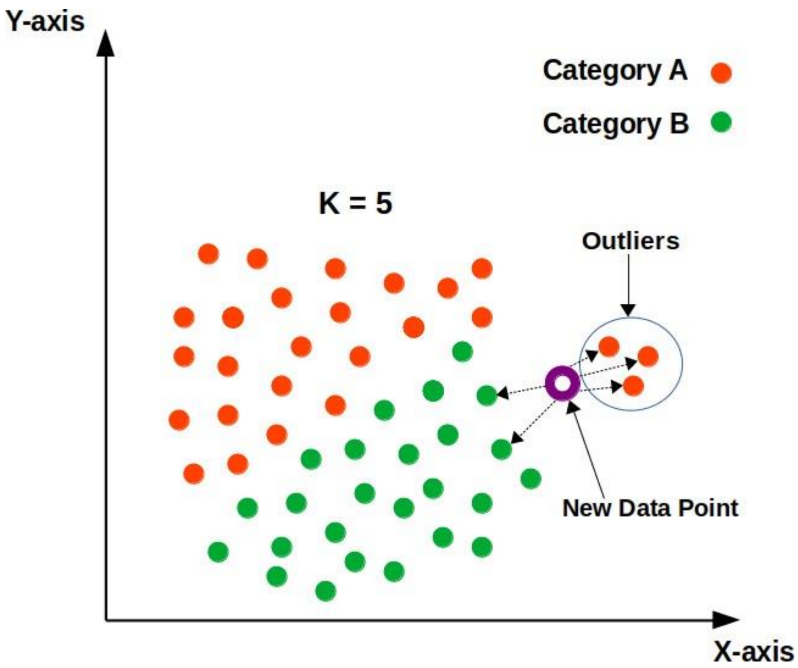
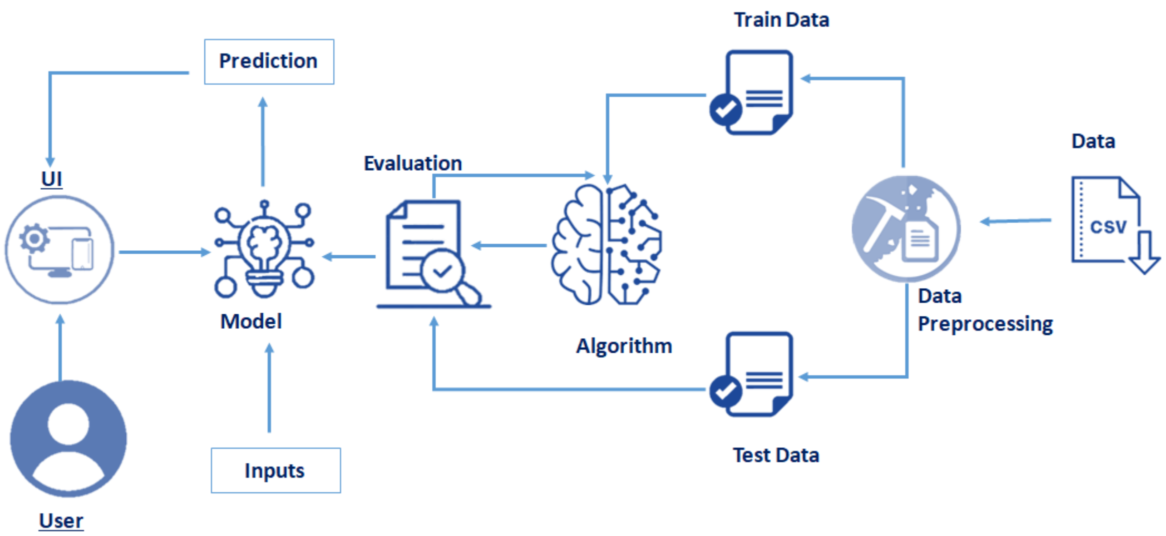


Figure 2 : k-nearest neighbor

4. EXPERIMENTAL ANALYSIS

Technical Architecture:



Pre requisites:

To complete this project, you must require following software's , concepts and packages

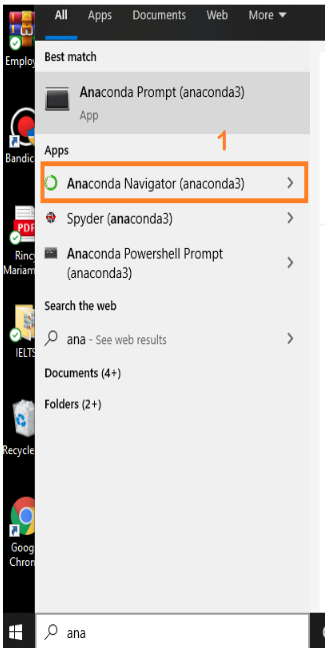
- **Anaconda Application:**
- Refer to the link below to download anaconda application

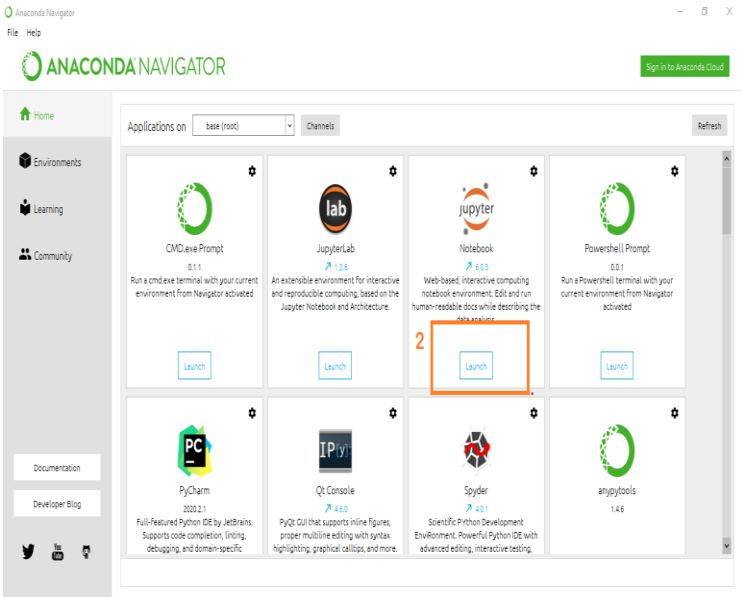
  Link : https://www.anaconda.com/products/individual
- **Python packages:**

  Open anaconda prompt as administrator.
- Type "pip install numpy" and click enter.
- Type "pip install pandas" and click enter.
- Type "pip install matplotlib" and click enter.
- Type "pip install scikit-learn" and click enter.
- Type "pip install Flask" and click enter.

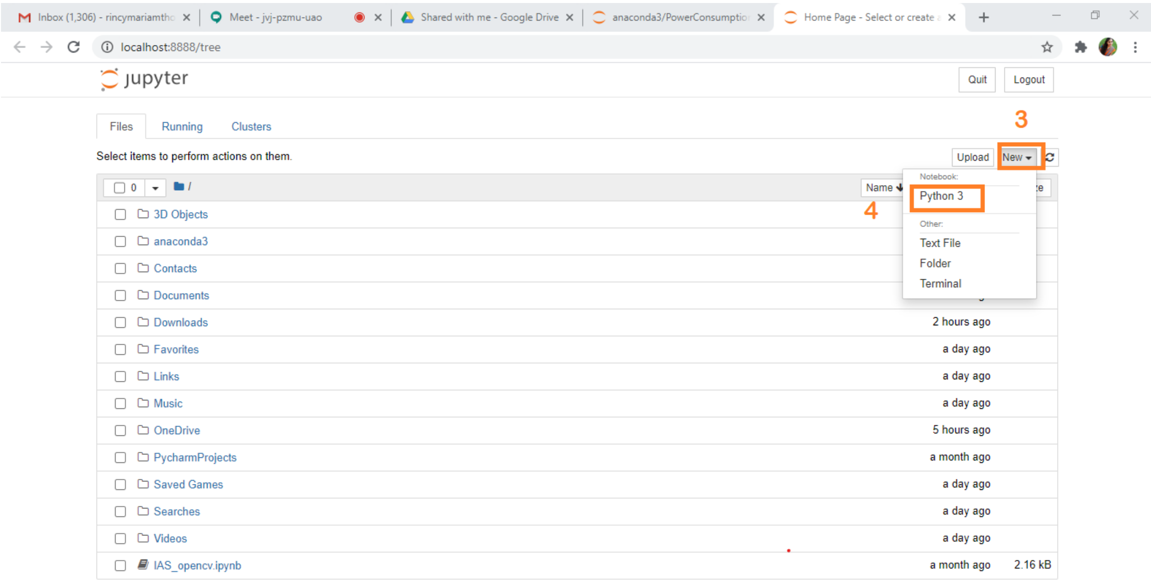
  The above steps allow you to install the packages in the anaconda environment

- **Launch Jupyter**
- Search for Anaconda Navigator and open Launch Jupyter notebook.





- Then you will be able to see that the jupyter notebook runs on local host:8888.
- To Create a new file Go to New →Python3.The file in jupyter notebook is saved with .ipynb extension.



- Flask Basics : https://www.youtube.com/watch?v=lj4I_CvBnt0

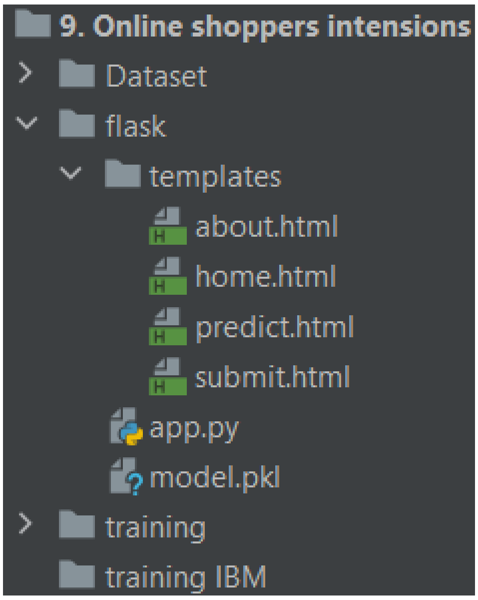
Project Flow:

- User interacts with the UI (User Interface) to enter the input values
- Entered input values are analysed by the model which is integrated
- Once model analyses the input the prediction is showcased on the UI

  To accomplish this, we have to complete all the activities and tasks listed below
- Data Collection.
  - Collect the dataset or Create the dataset
- Exploratory Data Analysis (***EDA***).
  - Import the Libraries.
  - Importing the dataset.
  - Data Cleaning.
  - Data Visualization.
  - Outlier Treatment.
  - Data Transformation.
  - Encoding
  - Splitting Data into Train and Test.
- Model Building
  - Import the model building Libraries
  - Initializing the model
  - Training and testing the model
  - Evaluation of Model
  - Save the Model
- Application Building
  - Create an HTML file
  - Build a Python Code

Project Structure:

Create a Project folder which contains files as shown below



- A python file called app.py for server side scipting.
- We need the model which is saved and the saved model in this content is **model.pkl**
- Templates folder which contains index.HTML file, chance.HTML file, noChance.HTML file.

Milestone 1: Data Collection:

ML depends heavily on data, without data, it is impossible for an "AI" to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training **data set.** It is the actual **data set** used to train the model for performing various actions.

Activity1: Download The dataset

Please refer to the link given below to download the data set and to know about the dataset
https://www.kaggle.com/henrysue/online-shoppers-intention

Milestone 2: Data Pre-processing / Exploratory Data Analysis (*EDA*)

Data Pre-processing includes the following main tasks

- Import the Libraries.
- Importing the dataset.
- Data Cleaning.
- Data Transformation.
- Encoding
- Splitting Data into Train and Test.

Activity 1: Import Necessary Libraries

- It is important to import all the necessary libraries such as pandas, numpy, matplotlib.
- **Numpy**- It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.
- **Pandas**- It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.
- **Seaborn**- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- **Matplotlib**- Visualisation with python. It is a comprehensive library for creating static,animated, and interactive visualizations in Python

- **Sklearn** – which contains all the modules required for model building

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('fivethirtyeight')
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.decomposition import PCA
from sklearn.model_selection import cross_val_score
import pickle
```

Activity 2: Importing the Dataset

- You might have your data in .csv files, .excel files

- Let's load a .csv data file into pandas using **read_csv() function.**We will need to locate the directory of the CSV file at first (it's more efficient to keep the dataset in the same directory as your program).

- If your dataset is in some other location ,Then

  Data=pd.read_csv(r"File_location/datasetname.csv")

  Note:r stands for "raw" and will cause backslashes in the string to be interpreted as actual backslashes rather than special characters.
- If the dataset in same directory of your program, you can directly read it, without giving raw as r.
- Our Dataset online_shoppers_intention.csv contains following Columns
- Administrative, Administrative_Duration, Informational, Informational_Duration
- ProductRelated, ProductRelated_Duration, BounceRates, ExitRates, PageValues
- SpecialDay, Month, OperatingSystems, Browser, Region, TrafficType
- VisitorType, Weekend, Revenue

The output column to be predicted is shoppers intention .Based on the input variables we predict the House rent. The predicted output gives them a fair idea about the rent of the house based on given cities.

Activity 3: Analyse the data

- head() method is used to return top n (5 by default) rows of a DataFrame or series.

```
df = pd.read_csv(r'D:\TheSmartBridge\Projects\9. Online shoppers intensions\Dataset\online_shoppers_intention.csv')
df.head()
```

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues
0	0	0.0	0	0.0	1	0.000000	0.20	0.20	0.0
1	0	0.0	0	0.0	2	64.000000	0.00	0.10	0.0
2	0	0.0	0	0.0	1	0.000000	0.20	0.20	0.0
3	0	0.0	0	0.0	2	2.666667	0.05	0.14	0.0
4	0	0.0	0	0.0	10	627.500000	0.02	0.05	0.0

- info() gives information about the data

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12330 entries, 0 to 12329
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Administrative           12330 non-null  int64  
 1   Administrative_Duration  12330 non-null  float64
 2   Informational            12330 non-null  int64  
 3   Informational_Duration   12330 non-null  float64
 4   ProductRelated           12330 non-null  int64  
 5   ProductRelated_Duration  12330 non-null  float64
 6   BounceRates              12330 non-null  float64
 7   ExitRates                12330 non-null  float64
 8   PageValues               12330 non-null  float64
 9   SpecialDay               12330 non-null  float64
 10  Month                    12330 non-null  object 
 11  OperatingSystems         12330 non-null  int64  
 12  Browser                  12330 non-null  int64  
 13  Region                   12330 non-null  int64  
 14  TrafficType              12330 non-null  int64  
 15  VisitorType              12330 non-null  object 
 16  Weekend                  12330 non-null  bool   
 17  Revenue                  12330 non-null  bool   
dtypes: bool(2), float64(7), int64(7), object(2)
memory usage: 1.5+ MB

df.shape

(12330, 18)
```

Activity 4: Data Visualisation

- Data visualization is where a given data set is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data.
- Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. In fact, even the most sophisticated machine learning models will perform poorly on data that wasn't visualized and understood properly.
- To visualize the dataset we need libraries called Matplotlib and Seaborn.
- The Matplotlib library is a Python 2D plotting library which allows you to generate plots, scatter plots, histograms, bar charts etc.

  Let's visualize our data using Matplotlib and searborn library.

  Univariate Analysis

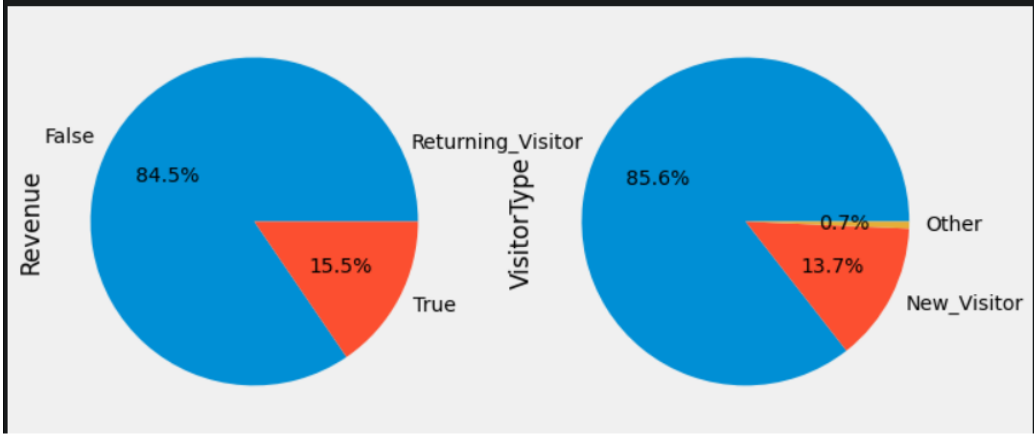
  In simple words, univariate analysis is understanding the data with single feature. Here we have used pieplot.

  Note: Different approaches can be used by the developer to analyze the data.

- The pie plot is plotted with categorical features value counts. With a pie plot, the importance of categories in categorical features is analyzed in form of a percentage. To perform this plot, plot() function is used. From the below image we can visualize most of the customers are return visitors. The probability of buying product chance is high if any discounts are allotted to products.

Output :-

```
plt.figure(figsize=(10,10))
plt.subplot(121)
df['Revenue'].value_counts().plot(kind='pie',autopct='%.1f%%')
plt.subplot(122)
df['VisitorType'].value_counts().plot(kind='pie',autopct='%.1f%%')

"""From the dataset, 84% of customers have not done shopping. And 86% of customers are returning visitors."""

'From the dataset, 84% of customers have not done shopping. And 86% of customers are returning visitors.'
```



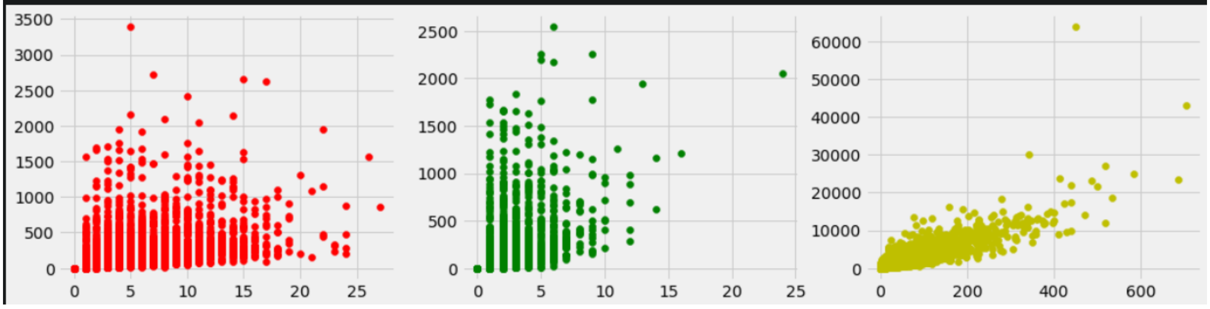
Bivariate Analysis

To find the relation between two features we use bivariate analysis.

- "Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and "Product Related Duration" represents the number of different types of pages visited by the visitor and total time spent in each of these page categories. To find the relationship between two features scatterplot() function is used. (Refer below image) Product related page and product related duration has high positive linearity.

```
"""Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and "Product
Related Duration" represent the number of different types of pages visited by the visitor and total time spent in each of
these page categories. """

plt.figure(figsize=(16,4))
plt.subplot(131)
plt.scatter(df['Administrative'],df['Administrative_Duration'],color='r')
plt.subplot(132)
plt.scatter(df['Informational'],df['Informational_Duration'],color='g')
plt.subplot(133)
plt.scatter(df['ProductRelated'],df['ProductRelated_Duration'],color='y')

"""Most of the visitors spending time on product related page"""
```

```
'Most of the visitors spending time on product related page'
```



Multivariate Analysis

In simple words, multivariate analysis is to find the relation between multiple features.

- To find the no. of sales based on month and visitor type crosstab() method is used. (refer below image) November month has the highest sales.

```
"""Based on month and visitor type features, count of revenue is analysed."""

pd.crosstab([df['Month'],df['VisitorType']],df['Revenue'])
```

June	Other	1	0
	Returning_Visitor	235	22
Mar	New_Visitor	196	36
	Returning_Visitor	1519	156
May	New_Visitor	231	88
	Returning_Visitor	2768	277
Nov	New_Visitor	291	128
	Other	19	3
	Returning_Visitor	1928	629
Oct	New_Visitor	96	28
	Returning_Visitor	338	87
Sep	New_Visitor	80	28
	Returning_Visitor	282	58

Descriptive Analysis

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.

```
df.describe(include='all')
```

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	P
count	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	123
unique	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
mean	2.315166	80.818611	0.503569	34.472398	31.731468	1194.746220	0.022191	0.043073	
std	3.321784	176.779107	1.270156	140.749294	44.475503	1913.669288	0.048488	0.048597	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	7.000000	184.137500	0.000000	0.014286	
50%	1.000000	7.500000	0.000000	0.000000	18.000000	598.936905	0.003112	0.025156	
75%	4.000000	93.256250	0.000000	0.000000	38.000000	1464.157214	0.016813	0.050000	
max	27.000000	3398.750000	24.000000	2549.375000	705.000000	63973.522230	0.200000	0.200000	3

Activity 5: Outliers Treatment

- In statistics, an outlier is **a data point that differs significantly from other observations**. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses.

Checking For Null Values

- For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function to it. From the below image we found that there are no null values present in our dataset. So we can skip handling of missing values step.

.

```
df.isnull().sum()

Administrative             0
Administrative_Duration    0
Informational              0
Informational_Duration     0
ProductRelated             0
ProductRelated_Duration    0
BounceRates                0
ExitRates                  0
PageValues                 0
SpecialDay                 0
Month                      0
OperatingSystems           0
Browser                    0
Region                     0
TrafficType                0
VisitorType                0
Weekend                    0
Revenue                    0
dtype: int64
```

Handling Categorical Values

As we can see our dataset has categorical data we must convert the categorical data to integer encoding or binary encoding.

To convert the categorical features into numerical features we use encoding techniques. There are several techniques but in our project we are using label encoder

```
# handling categorical feature

le = LabelEncoder()
df['Month'] = le.fit_transform(df['Month'])
df['VisitorType'] = le.fit_transform(df['VisitorType'])
df['Weekend'] = le.fit_transform(df['Weekend'])
df['Revenue'] = le.fit_transform(df['Revenue'])
```

Dropping Unwanted Features

Now let's create our 1st model with unsupervised ML algorithm- KMeans clustering. For implementing this algorithm target feature should be removed. With drop() method revenue column is dropped.

```
dfKmeans = df.drop('Revenue',axis=1)
```

Scaling The Features

Normalization technique Minmax scaler() is used on dfKmeans and the array values are converted into dataframe.

- Minmax scaler is initialized and fit_transform() method is used to scale the values. Now, those array values are created as a dataframe as shown in below image.

```
# Scalling values

scaler = MinMaxScaler()
scaled_df = scaler.fit_transform(dfKmeans)
```

```
dfKmeans = pd.DataFrame(scaled_df,columns=dfKmeans.columns)
dfKmeans.head()
```

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues
0	0.0	0.0	0.0	0.0	0.001418	0.000000	1.00	1.00	0.0
1	0.0	0.0	0.0	0.0	0.002837	0.001000	0.00	0.50	0.0
2	0.0	0.0	0.0	0.0	0.001418	0.000000	1.00	1.00	0.0
3	0.0	0.0	0.0	0.0	0.002837	0.000042	0.25	0.70	0.0
4	0.0	0.0	0.0	0.0	0.014184	0.009809	0.10	0.25	0.0

Elbow Method

To find the optimal number of clusters elbow method is used. The plot will be plotted with sse (sum of squared error). Inertia is used from kmeans algorithm to calculate sse.

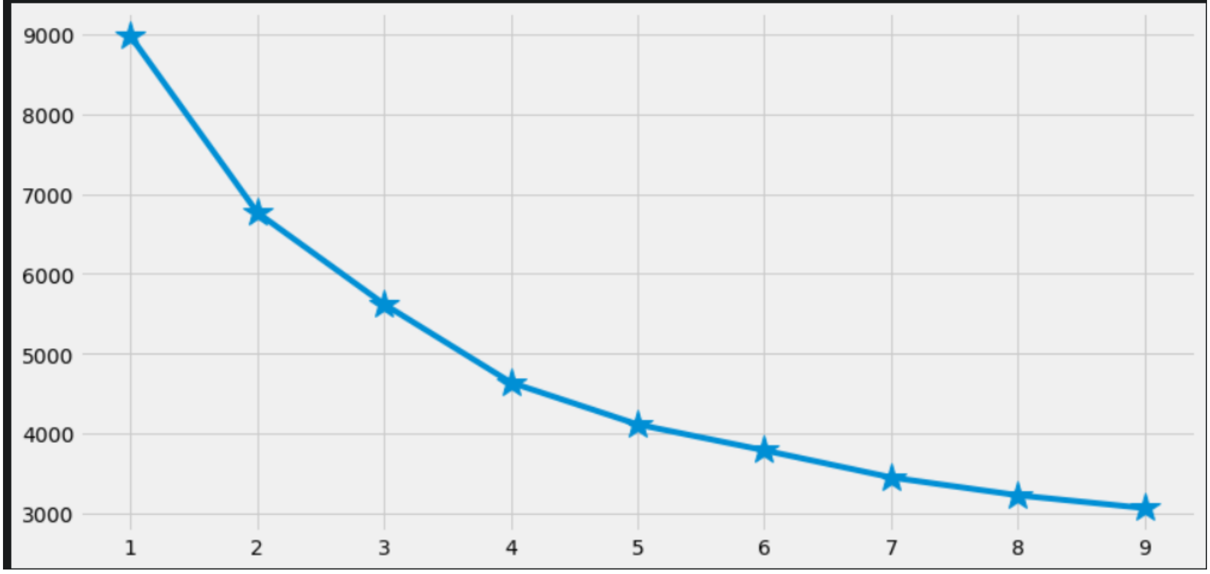
```
# Finding N clusters by elbow method.

n_cluster = range(1,10,1)
sse = []
for i in n_cluster:
    k = KMeans(n_clusters=i)
    ypred = k.fit(scaled_df)
    sse.append(k.inertia_)
```

```
sse
```

```
[8977.629530570664,
 6768.01393714652,
 5619.938835283883,
 4638.301369734072,
 4112.527344033329,
 3788.079018522391,
 3448.4291813087816,
 3224.6785631183748,
 3063.896789860229]
```

```
plt.figure(figsize=(12,6))
plt.plot(n_cluster,sse,marker='*',markersize=20)
```

```
[<matplotlib.lines.Line2D at 0x29a758d0d90>]
```



Dimensionality Reduction

PCA- Principal Component Analysis is used to reduce the dataset dimension. Scatterplot is used to visualize the clusters.

- PCA() is initialized to pca variable. We need 2 columns. So, let's pass 2 as a parameter. To transform the dataset into 2 columns fit_transform() is used.

```
#dfKmeans['Cluster']=ypred
```

```
pca = PCA(n_components=2)
dfPCA = pca.fit_transform(dfKmeans)
dfPCA
```

```
array([[-1197.63555578,   -40.8241322 ],
       [-1133.68455771,   -43.2912114 ],
       [-1197.63603661,   -40.82414532],
       ...,
       [-1013.48454092,   -47.94070163],
       [ -849.24642902,    18.23938657],
       [-1176.36824437,   -41.65359505]])
```

- New dataframe is created. Previous step(2d array) are passed as the parameters. To pass the cluster values, a new column is created as shown in the below figure.

```python
dfPCA = pd.DataFrame(dfPCA,columns=['PCA 1','PCA 2'])
dfPCA.head()
```

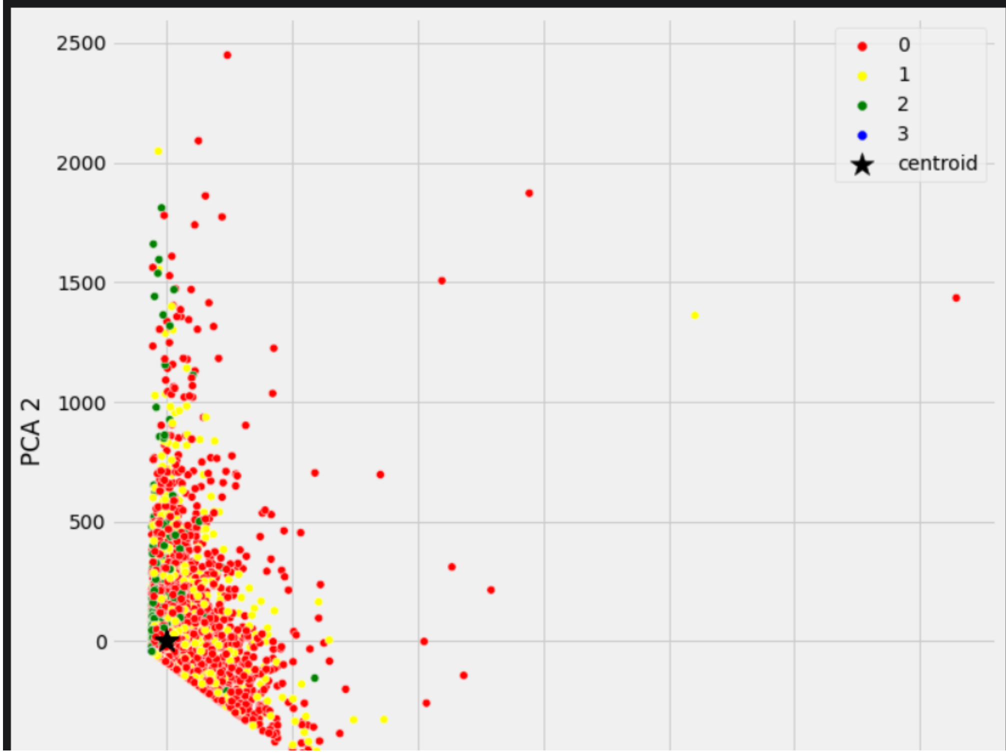
	PCA 1	PCA 2
0	-1197.635556	-40.824132
1	-1133.684558	-43.291211
2	-1197.636037	-40.824145
3	-1194.952046	-40.933714
4	-570.631539	-65.005398

```python
dfPCA['Cluster'] = ypred
```

- Scatterplot() from seaborn is used to visualize the data points based on clusters. Refer the below image.

```python
plt.figure(figsize=(10,10))
sns.scatterplot(dfPCA['PCA 1'], dfPCA['PCA 2'],hue = dfPCA['Cluster'],palette=['red','yellow','green','blue'])
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='black',s=300,marker='*',label='centroid')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x29a75938820>
```



Activity 6: Splitting the data into Train and Test

- When you are working on a model and you want to train it, you obviously have a dataset. But after training, we have to test the model on some test dataset. For this, you will a dataset which is different from the training set you used earlier. But it might not always be possible to have so much data during the development phase. In such cases, the solution is to split the dataset into two sets, one for training and the other for testing.

- But the question is, how do you split the data? You can't possibly manually split the dataset into two sets. And you also have to make sure you split the data in a random manner. To help us with this task, the Scikit-learn library provides a tool, called the Model Selection library. There is a class in the library which is,'**train_test_split**.' Using this we can easily split the dataset into the training and the testing datasets in various proportions.
- The train-test split is a technique for evaluating the performance of a machine learning algorithm.
- **Train Dataset**: Used to fit the machine learning model.
- **Test Dataset**: Used to evaluate the fit machine learning model.
- In general you can allocate 80% of the dataset to training set and the remaining 20% to test set.We will create 4 sets— X_train (training part of the matrix of features), X_test (test part of the matrix of features), Y_train (training part of the dependent variables associated with the X train sets, and therefore also the same indices), Y_test (test part of the dependent variables associated with the X test sets, and therefore also the same indices.
- There are a few other parameters that we need to understand before we use the class:
- **test_size** — this parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset
- **train_size** — you have to specify this parameter only if you're not specifying the test_size. This is the same as test_size, but instead you tell the class what percent of the dataset you want to split as the training set.
- **random_state** — here you pass an integer, which will act as the seed for the random number generator during the split. Or, you can also pass an instance of the Random_state class, which will become the number generator. If you don't pass anything, the Random_state instance used by np.random will be used instead.
- Now split our dataset into train set and test using train_test_split class from scikit learn library.
- **from sklearn import model_selection**

x_train,x_test,y_train,y_test=model_selection.train_test_split(x,y,test_size=0.2,random_state =0)

```
# Splitting dataset

x = df.drop('Revenue',axis=1)
y = df['Revenue']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=10)
```

Milestone 3: Model Building:

Model building includes the following main tasks

- Import the model building Libraries
- Initializing the model
- Training and testing the model
- Evaluation of Model
- Save the Model

Training and Testing the Model

- Once after splitting the data into train and test, the data should be fed to an algorithm to build a model.

- There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms are Regression algorithms.

For Regression kind of problem some examples models are:-

1. Linear Regression
2. Polynomial Regression
3. Ridge Regression
4. Lasso Regression
5. ElasticNet regression
6. Decision Tree Regressor
7. Random Forest Regressor

For Classification kind of problem some examples models are:-

i. Logistic Regression
ii. Decision Tree Classifier
iii. Random Forest Classifier
iv. KNN
v. svm
vi. xgboost

Steps in Building the model:-

- **Initialize the model**
- **Fit the models with x_train and y_train**
- **Predict the y_train values and calculate the accuracy**
- **Predict the y_test values and calculate the accuracy**

```
km = KMeans(n_clusters=4)
ypred = km.fit_predict(dfKmeans)
```

We're going to use x_train and y_train obtained above in train_test_split section to train our Random Forest regression model. We're using the fit method and passing the parameters as shown below.

We are using the algorithm from Scikit learn library to build the model as shown below,
Once the model is trained, it's ready to make predictions. We can use the **predict** method on the model and pass **x_test** as a parameter to get the output as **y_pred.**

After training the model, the model should be tested by using the test data which is been separated while splitting the data for checking the functionality of the model.

Logistic Regression Model

A function named logisticReg is created and train and test data are passed as the parameters. Inside the function, LogisticRegression() algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

```
def logisticReg(x_train, x_test, y_train, y_test):
    lr = LogisticRegression()
    lr.fit(x_train,y_train)
    yPred = lr.predict(x_test)
    print('***LogisticRegression***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

Random Forest Model

A function named randomForest is created and train and test data are passed as the parameters. Inside the function, RandomForestClassifier algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

```
def randomForest(x_train, x_test, y_train, y_test):
    rf = RandomForestClassifier()
    rf.fit(x_train,y_train)
    yPred = rf.predict(x_test)
    print('***RandomForestClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

Compare The Model

For comparing the above two models compareModel function is defined.

```
def compareModel(x_train, x_test, y_train, y_test):
    logisticReg(x_train, x_test, y_train, y_test)
    print('-'*100)
    randomForest(x_train, x_test, y_train, y_test)
```

```
compareModel(x_train, x_test, y_train, y_test)
```

```
***LogisticRegression***
Confusion matrix
[[3033   82]
 [ 370  214]]
Classification report
              precision    recall  f1-score   support

           0       0.89      0.97      0.93      3115
           1       0.72      0.37      0.49       584

    accuracy                           0.88      3699
   macro avg       0.81      0.67      0.71      3699
weighted avg       0.86      0.88      0.86      3699

----------------------------------------------------------------------------------------------------
***RandomForestClassifier***
Confusion matrix
[[3011  104]
 [ 257  327]]
Classification report
              precision    recall  f1-score   support

           0       0.92      0.97      0.94      3115
           1       0.76      0.56      0.64       584

    accuracy                           0.90      3699
   macro avg       0.84      0.76      0.79      3699
weighted avg       0.90      0.90      0.90      3699
```

After calling the function, the results of models are displayed as output. From the two model random forest is performing well. From the below image, we can see the accuracy of the model, confusion matrix and classification report.

Note: Additionally, we can tune the model with hyper parameter tuning techniques and to oversampling the data smote can be used.

Evaluating Performance Of The Model And Saving The Model

From sklearn, cross_val_score is used to evaluate the score of the model. On the parameters, we have given rf (model name), x, y, cv (as 5 folds). Our model is performing well. So, we are saving the model by pickle.dump().

Note: To understand cross validation, refer this link.

```
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
yPred = rf.predict(x_test)
```

```
cv = cross_val_score(rf,x,y,cv=5)
np.mean(cv)
```

```
0.8944849959448499
```

Save the Model

After building the model we have to save the model.

Pickle in **Python** is primarily **used** in serializing and deserializing a **Python** object structure. In other words, it's the process of converting a **Python** object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. wb indicates write method and rd indicates read method.

This is done by the below code

```
pickle.dump(rf,open('model.pkl','wb'))
```

Milestone 4: Application Building

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

- Building HTML Pages
- Building server side script

Activity 1: Build HTML Code

- In this HTML page, we will create the front end part of the web page. In this page we will accept input from the user and Predict the values.

For more information regarding HTML

https://www.w3schools.com/html/

In our project we have 3 HTML files ,they are

1.home.html

2.predict.html

3.about.html

4.submit.html

home.html

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Home</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <style>
        body
        {
        background-image: url("https://www.thebeijinger.com/sites/default/files/thebeijinger/blog-images/345221/online-shop-website-development-in-kerala1.jpg");
        background-size: cover;
        }
        h3.big
        {
        line-height: 1.8;
        }
    </style>
</head>
<body>

<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <strong><a class="navbar-brand" href="/home">Online Shoppers Intention Using ML</a></strong>
    </div>
    <ul class="nav navbar-nav navbar-right">
        <a href="/" class="btn btn-info btn-lg">Home</a>
        <a href="/about" class="btn btn-info btn-lg">About</a>
        <a href="/predict" class="btn btn-primary btn-lg">Predict</a>
    </ul>
  </div>
</nav>
<center>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</body>
</html>
```

About.html

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>About</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <style>
        body
        {
        background-image: url("https://d1m75rqqgidzqn.cloudfront.net/wp-data/2021/03/22142903/iStock-1206800961.jpg");
        background-size: cover;
        }
        h3.big
        {
        line-height: 1.8;
        }
    </style>
</head>
<body>

<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <strong><a class="navbar-brand" href="/home">Online Shoppers Intention Using ML</a></strong>
    </div>
    <ul class="nav navbar-nav navbar-right">
        <a href="/" class="btn btn-info btn-lg">Home</a>
        <a href="/about" class="btn btn-info btn-lg">About</a>
        <a href="/predict" class="btn btn-primary btn-lg">Predict</a>
    </ul>
  </div>
</nav>
<div class="container">
    <center>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <br>
    <h2 style="color:white">Online shopping is the activity or action of buying products or services over the Internet. It means going online, landing on a
    seller's website, selecting something, and arranging for its delivery. The buyer either pays for the good or service online with a credit or debit card or
    upon delivery. The term does not only include buying things online but also searching for them online. In other words, I may have been engaged in online
    shopping but did not buy anything.
</h2></center>
    </div>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</body>
</html>
```

Predict.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Predict</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <style>
        body
        {
        background-image: url("http://www.kotak.com/content/dam/Kotak/digital-banking/insta-services/kaymall/1440X400-Online-Shopping.jpg");
        background-size: cover;
        }
        h3.big
        {
        line-height: 1.8;
        }
    </style>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <strong><a class="navbar-brand" href="/home">Online Shoppers Intention Using ML</a></strong>
    </div>
    <ul class="nav navbar-nav navbar-right">
        <a href="/" class="btn btn-info btn-lg">Home</a>
        <a href="/about" class="btn btn-info btn-lg">About</a>
        <a href="/predict" class="btn btn-primary btn-lg">Predict</a>
    </ul>
  </div>
</nav>
    <div class="container">
        <h4>
        <form action="/pred", method="post">
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f1">Administrative</label>
                    <input class="form-control" id="f1" name="Administrative" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f2">Administrative_Duration</label>
                    <input class="form-control" id="f2" name="Administrative_Duration" required="required" type="text">
                </div>
            </div>
```

Line 1, Column 1 — 149 Lines INS UTF-8 HTML Spaces: 4

```html
                    <input class="form-control" id="f1" name="Administrative" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f2">Administrative_Duration</label>
                    <input class="form-control" id="f2" name="Administrative_Duration" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f3">Informational</label>
                    <input class="form-control" id="f3" name="Informational" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f4">Informational_Duration</label>
                    <input class="form-control" id="f4" name="Informational_Duration" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f5">ProductRelated</label>
                    <input class="form-control" id="f5" name="ProductRelated" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f6">ProductRelated_Duration</label>
                    <input class="form-control" id="f6" name="ProductRelated_Duration" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f7">BounceRates</label>
                    <input class="form-control" id="f7" name="BounceRates" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f8">ExitRates</label>
                    <input class="form-control" id="f8" name="ExitRates" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
```

Line 85, Column 1 — 150 Lines INS UTF-8 HTML Spaces: 4

```
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f8">ExitRates</label>
                    <input class="form-control" id="f8" name="ExitRates" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f9">PageValues</label>
                    <input class="form-control" id="f9" name="PageValues" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f10">SpecialDay</label>
                    <input class="form-control" id="f10" name="SpecialDay" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f11">Month</label>
                    <input class="form-control" id="f11" name="Month" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f12">OperatingSystems</label>
                    <input class="form-control" id="f12" name="OperatingSystems" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f13">Browser</label>
                    <input class="form-control" id="f13" name="Browser" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f14">Region</label>
                    <input class="form-control" id="f14" name="Region" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f15">TrafficType</label>
```

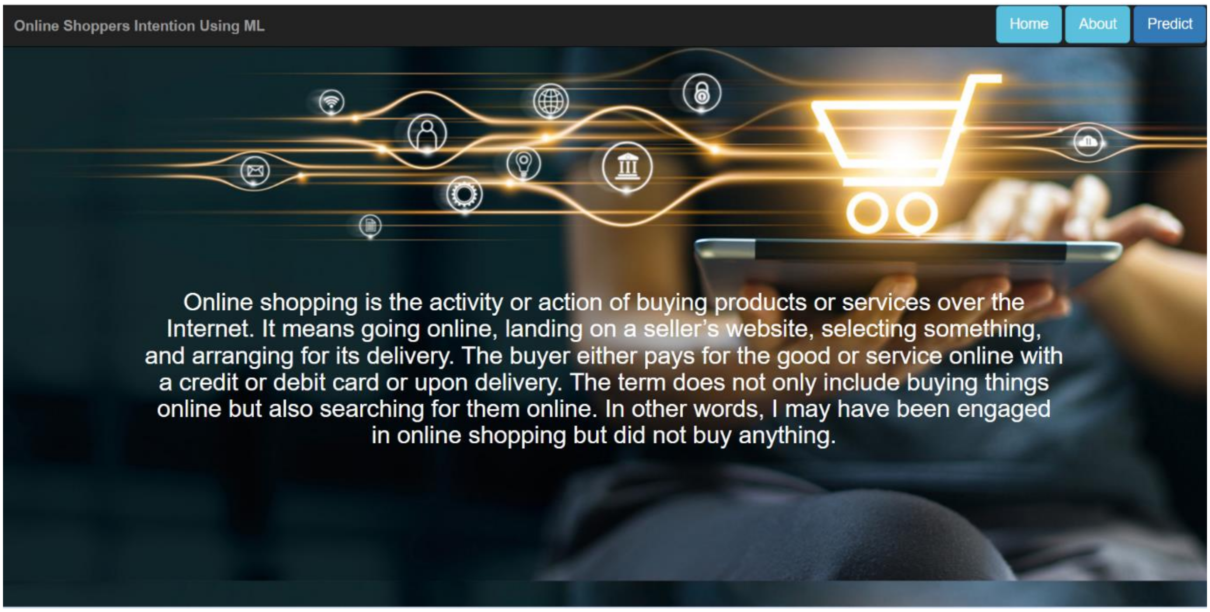
```
                    <input class="form-control" id="f12" name="OperatingSystems" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f13">Browser</label>
                    <input class="form-control" id="f13" name="Browser" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f14">Region</label>
                    <input class="form-control" id="f14" name="Region" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f15">TrafficType</label>
                    <input class="form-control" id="f15" name="TrafficType" required="required" type="text">
                </div>
                <div class="col-xs-1">
                </div>
                <div class="col-xs-3">
                    <label for="f16">VisitorType</label>
                    <input class="form-control" id="f16" name="VisitorType" required="required" type="text">
                </div>
            </div>
            <div class="form-group row">
                <div class="col-xs-3">
                    <label for="f17">Weekend</label>
                    <input class="form-control" id="f17" name="Weekend" required="required" type="text">
                </div>
                <div class="col-xs-2">
                </div>
            </div>
            <button type="submit" class="btn btn-success btn-lg">Submit</button>
        </form>
        <br>
        </h4>
    </div>


    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</body>
</html>
```

Submit.html

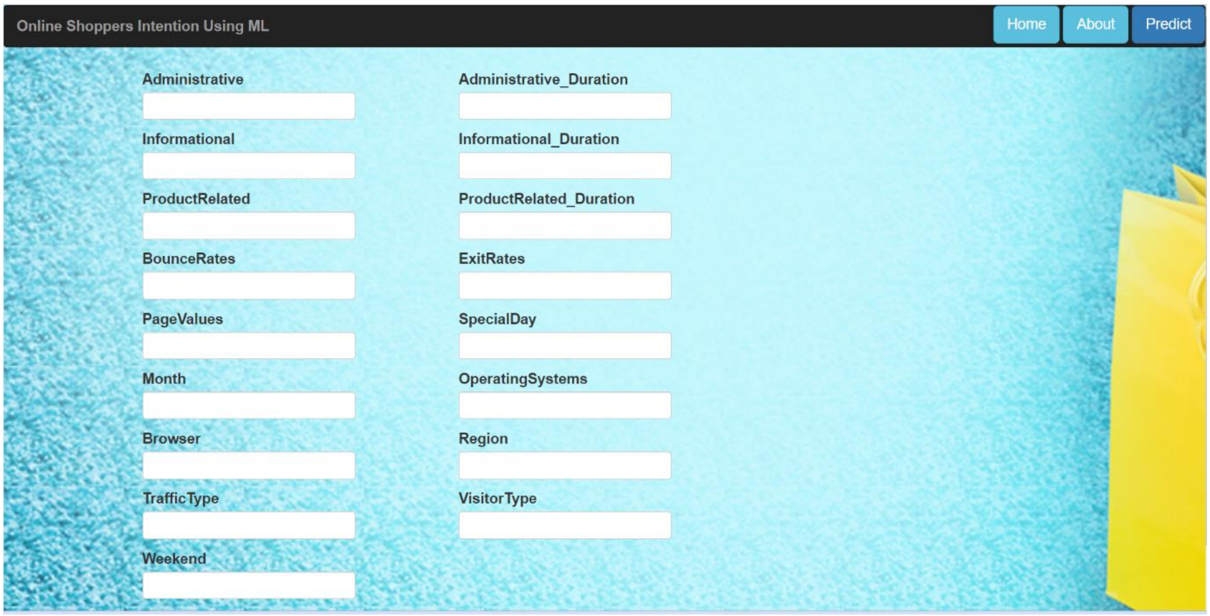
```
<!DOCTYPE html>
<html lang="en">
Submit.html

<head>
    <meta charset="UTF-8">
    <title>Output</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <style>
        body
        {
        background-image: url("https://thinkmarketingmagazine.com/wp-content/uploads/2020/06/10-Advantages-of-Online-Shopping.jpg");
        background-size: cover;
        }
        h3.big
        {
        line-height: 1.8;
        }
    </style>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <strong><a class="navbar-brand" href="/home">Online Shoppers Intention Using ML</a></strong>
    </div>
    <ul class="nav navbar-nav navbar-right">
        <a href="/" class="btn btn-info btn-lg">Home</a>
        <a href="/about" class="btn btn-info btn-lg">About</a>
        <a href="/predict" class="btn btn-primary btn-lg">Predict</a>
    </ul>
  </div>
</nav>
        <div class="container">
            <h3>
            Based on the given input, {{prediction_text}}.
            </h3>

        </div>

</body>
</html>
```

Home.html page looks like

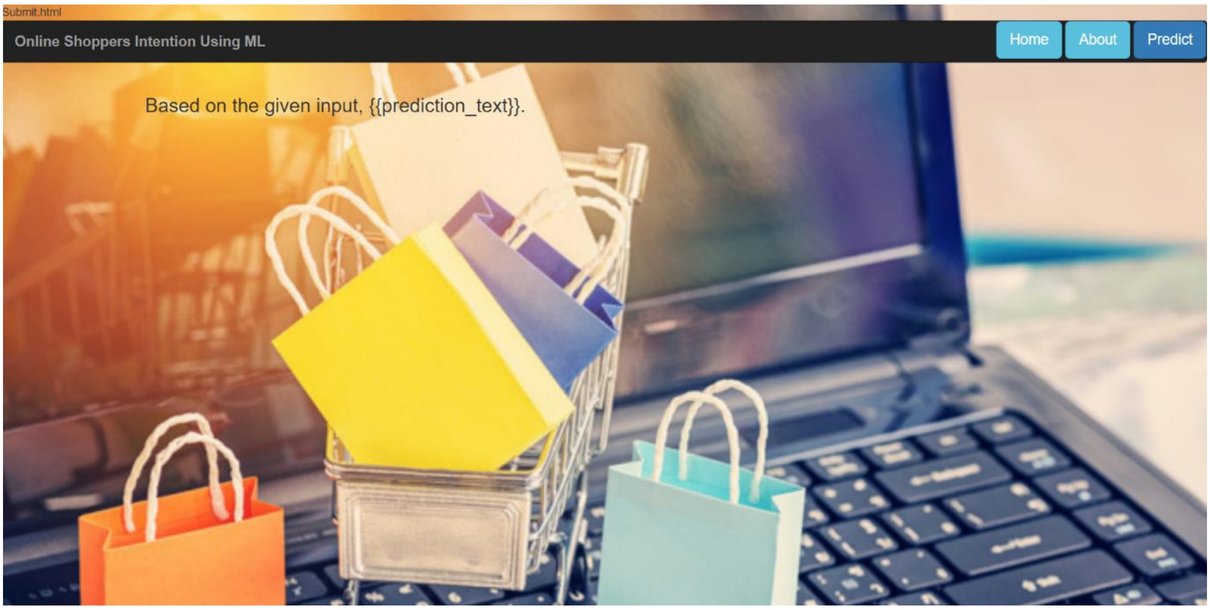
About.html looks like



Predict.html looks like



Submit.html looks like



Activity 2: Main Python Script

Let us build app.py flask file which is a web framework written in python for server-side scripting. Let's see step by step procedure for building the backend application.

Import the libraries

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

Load the saved model. Importing flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument.

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

Render HTML page:

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

Here we will be using declared constructor to route to the HTML page which we have created earlier.

In the above example, '/' URL is bound with home.html function. Hence, when the home page of the web server is opened in browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI:

```python
@app.route("/pred", methods=['POST'])
def predict():
    Administrative = request.form['Administrative']
    Administrative_Duration = request.form['Administrative_Duration']
    Informational = request.form['Informational']
    Informational_Duration = request.form['Informational_Duration']
    ProductRelated = request.form['ProductRelated']
    ProductRelated_Duration = request.form['ProductRelated_Duration']
    BounceRates = request.form['BounceRates']
    ExitRates = request.form['ExitRates']
    PageValues = request.form['PageValues']
    SpecialDay = request.form['SpecialDay']
    Month = request.form['Month']
    OperatingSystems = request.form['OperatingSystems']
    Browser = request.form['Browser']
    Region = request.form['Region']
    TrafficType = request.form['TrafficType']
    VisitorType = request.form['VisitorType']
    Weekend = request.form['Weekend']
    total = [[int(Administrative), float(Administrative_Duration), int(Informational), float(Informational_Duration)
              int(ProductRelated), float(ProductRelated_Duration), float(BounceRates), float(ExitRates),
              float(PageValues), float(SpecialDay), int(Month), int(OperatingSystems), int(Browser), int(Region),
              int(TrafficType), int(VisitorType), int(Weekend)]]
    print(total)
    prediction = model.predict(total)
    print(prediction)
    if prediction == 0:
        text = 'The visitor is not interested in buying products.'
    else:
        text = 'The visitor is interested in buying products'
    return render_template('submit.html', prediction_text=text)
```

Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will rendered to the text that we have mentioned in the submit.html page earlier.

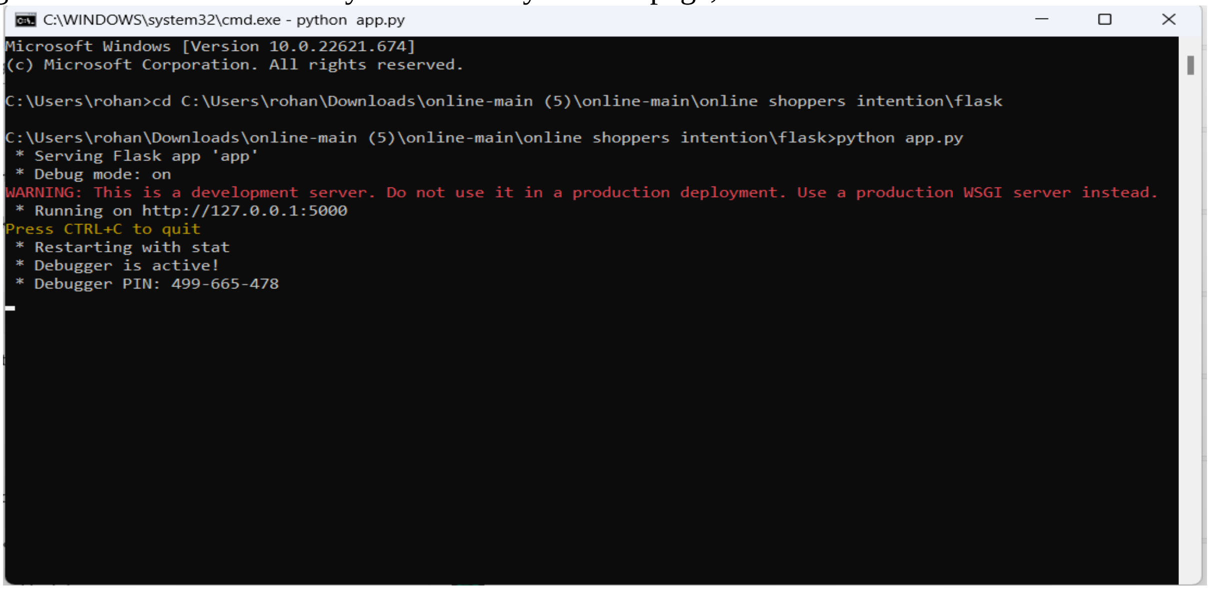
Main Function:

```python
if __name__ == "__main__":
    app.run(debug=False)
```

Activity 3: Run the App

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type “python app.py” command

Navigate to the localhost where you can view your web page, Then it will run on local host:8000



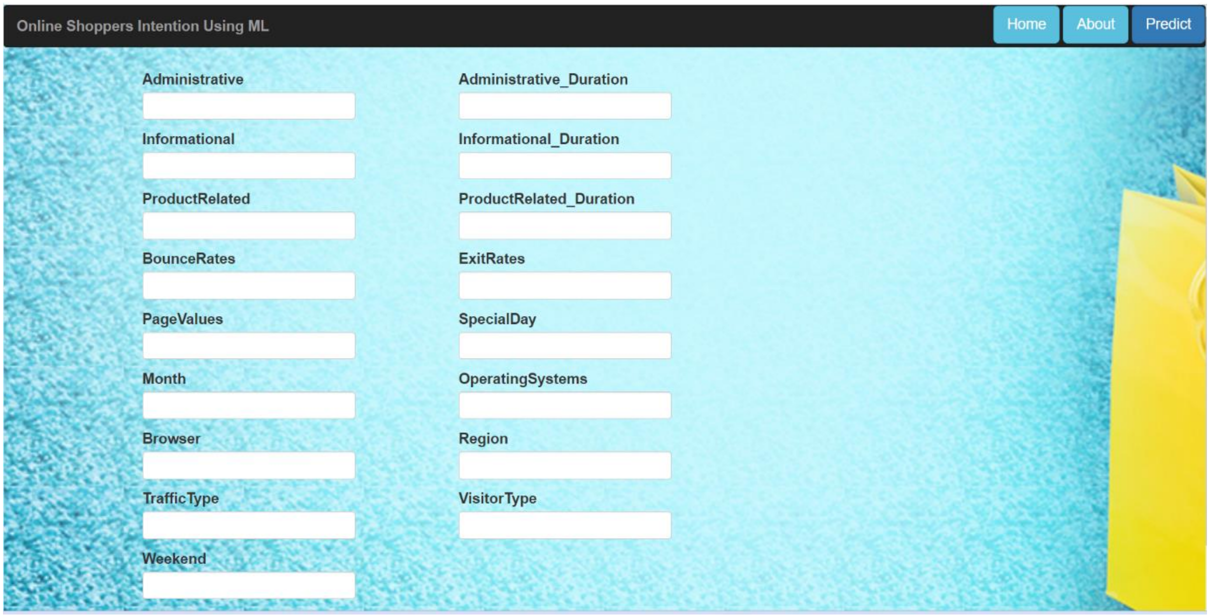
Activity 4:

- Copy the http link and paste it in google link tab,it will display the form page
- Enter the values as per the form and click on predict buttion
- It will redirect to the page based on prediction output

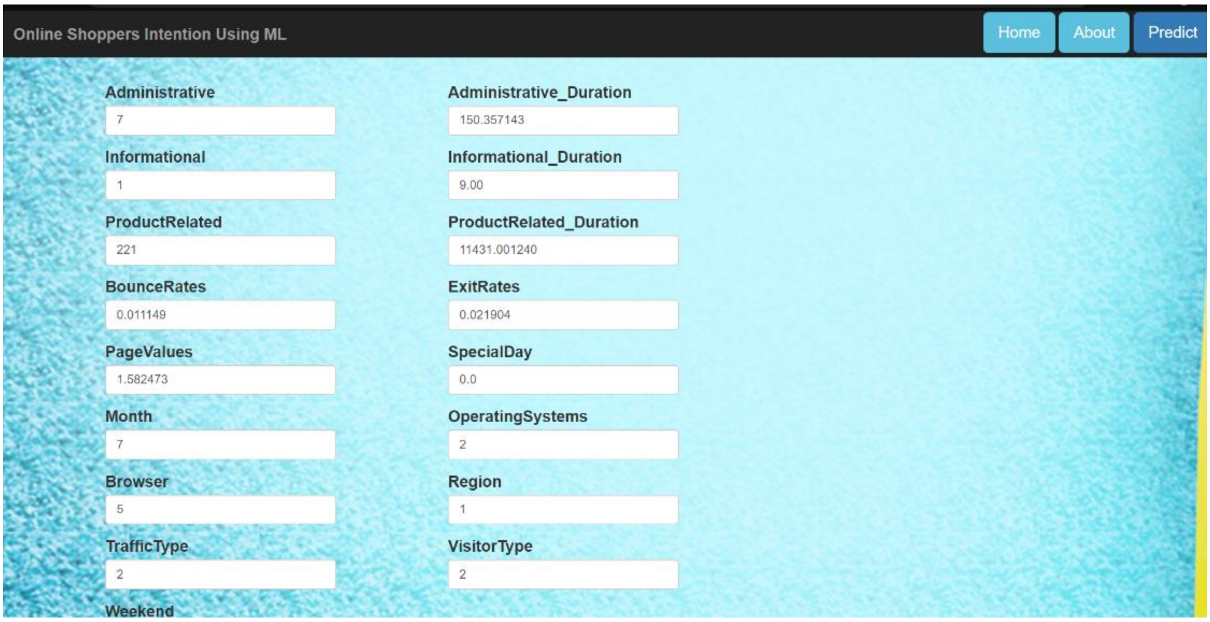
Output:

On opening the link

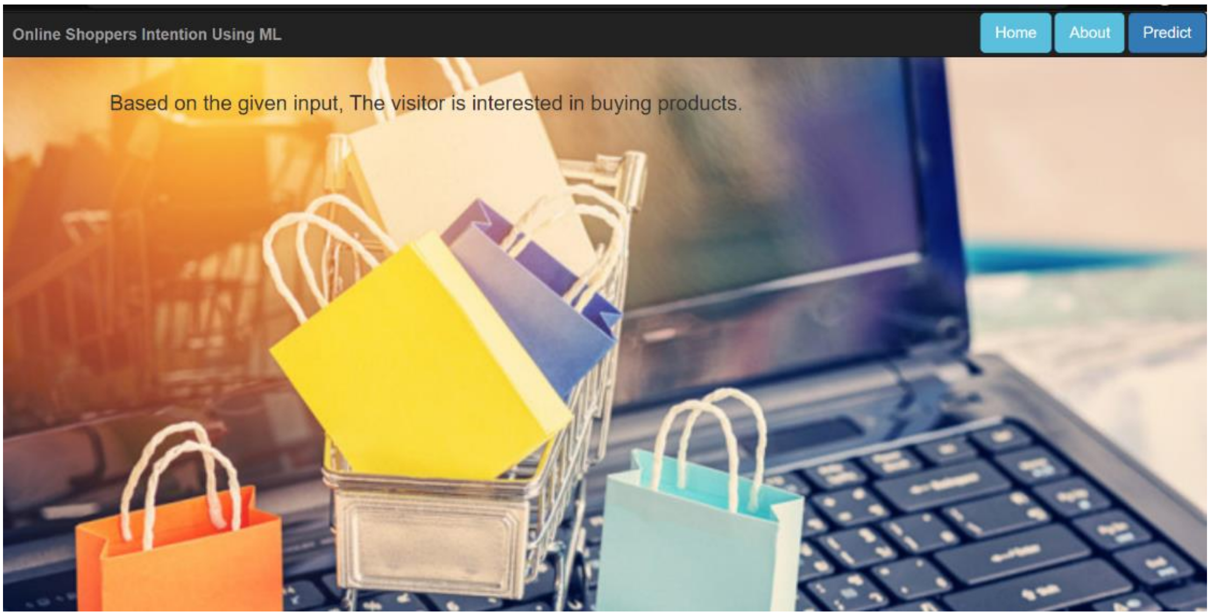
On clicking predict:



Enter the details:



After clicking on submit:



CONCLUSION:

From this research, we can conclude that trust and risk on online purchase played a very important role in making customers have intention to online purchase. As we know, shop online is a very risky activity because there fraud people use internet to make their own benefits and stole money from the customers. The respondents of this research which are students from UiTM Johor are agree that the trust and risk can affect someone behavior to have intention to online purchase.

Many people now are adapting and accepting new technologies like the Internet. Internet can give customers do whatever they want such as make purchases online. Students are tends to shop online because they can save time and the price is reasonable rather than physical shops or vendors.

We also can conclude that female is prefer shop online rather than male because the online retailers offer price discount and of course they can buy more through the web. Other than that, they can buy anything online because there are many websites that sell the product that they want with lower price. But the most important things are risk and trust towards online purchase. They might have some risk if they shop online. So, if we want to that we need to trust the online retailers.

FUTURE SCOPE:

The research used a sample of regular users of the website so future research could investigate whether there are any differences between regular and first-time users of a website. Future studies could also test the appeal of some of the findings, such as different shaped models and displaying the product videos in a different format.

REFERENCES

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). New York: Spring.

- Jolliffe, I. (1982). A note on the use of principal components in regressio. Journal of the Royal Statistical Society, 31(3), 300 – 303.

- Kaggle – the home of data science and machine learning. (2017). Retrieved from https://www.kaggle.com/

- Korpusik, M., Sakaki, S., Chen, F., & Chen, Y. (2016). Recurrent neural networks for customer purchase prediction on twitter. In Proceedings of the CBRecSys.

- Kotu, V., & Deshpande, B. (2014). Predictive analytics and data mining: concepts and practice with rapidminer. Morgan Kaufmann, 562–572.

- Lang, T., & Rettenmeier, M. (2017). Understanding consumer behavior with recur-rent neural networks. In Proceedings of the 3rd International Workshop on Machine Learning Methods for Recommender Systems.

- Lee, M., Ha, T., Han, J., Rha, J., & Kwon, T. (2015). Online footsteps to purchase: Exploring consumer behaviors on online shopping sites. In Proceedings of the ACM Web Science Conference.

- Liu, Y., An, A., & Huang, X. (2006, April). Boosting prediction accuracy on imbal-anced datasets with svm ensembles. PAKDD, 6, 107 – 118.

- Lo, C., Frankowsik, D., & Leskovec, J. (2014). Understanding behaviors that lead to purchasing: A case study of pinterest. In Proceedings of the KDD 16, San Francisco, CA, USA.

- Magrabi, A. (2016, 11). Top 5 machine learning applications for e-commerce. Retrieved from https://techblog.commercetools.com/top-5-machine-learning-applications-for-e- commerce-268eb1c89607 Manning, C., Raghavan, P., & Schuetze, H. (2008). Introduction to information retrieval. Cambridge University Press.

- Moe, W. (2003). Buying, searching, or browsing: Differentiating between online shoppers using in-store navigational clickstream. Journal of Consumer Psychology, 13(1– 2), 29–39.

- Mulder, W. D., Bethard, S., & Moens, M.-F. (2015). A survey on the application of recurrent neural networks to statistical language modeling. Computer Speech Language, 30(1), 61 – 98.

- Nakayama, Y. (2009). The impact of e-commerce: It always benefits consumers, but mayreduce social welfare. Japan and the World Economy, 21(3), 239–247.

- Neural network hyperparameters. (2015, 12). Retrieved fromhttp://colinraffel.com/wiki/neural network hyperparameters