## A

**Mini Skill Based Project On**

### Data Mining & Data Warehousing(280601)

*In fulfilment of the requirement for the award of the degree*



### SUBMITTED TO

Dr. Subha Mishra

### SUBMITTED BY

Anshika Tripathi (0901AM211012) Muskan Sharma (0901AM211032) Sahil Choudhary (0901AM211047)

VI semester

Artificial Intelligence and Machine Learning

### Department Of Information Technology

**Madhav Institute of Technology and Science, Gwalior**

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

**Session: Jan-June 2024**

## DECLARATION

I hereby declare that the mini skill-based project for the course Data Mining & warehousing(280601) is being submitted in the partial fulfilment of the requirement for the award of **Bachelor of Technology** in Artificial Intelligence and Machine Learning**.**

All the information in this document has been obtained and presented in accordance with academic rule and ethical conduct.

**Date:**

**Place: Gwalior**

Anshika Tripathi (0901AM211012) Muskan Sharma (0901AM211032) Sahil Choudhary (0901AM211047)

## ACKNOWLEDGEMENT

I would like to express my greatest appreciation to all the individuals who have helped and supported me throughout this lab file. I am thankful to whole Information Technology department for their ongoing support during the experiments, from initial advice and provision of contact in the first stages through ongoing advice and encouragement, which led to the finals report of this project.

A special acknowledgement goes to my colleagues who help me in completing the project and by exchanging interesting ideas to deal with problems and sharing the experience.

I wish to thank Shubha Mam as well for her undivided support and interests which inspired me and encouraged me to go my own way without whom I would be unable to complete my project.

**Table of content**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Title** | **Date** | **Signature** |
| 1. | Micro Project |  |  |
| 2. | Macro Project |  |  |
| 3. | Mini Project |  |  |

# Micro-Project

**Aim : Develop an E-commerce website of your choice.**

**Description:** An E-commerce website serves as a digital platform where businesses showcase and sell their products or services to consumers online. It typically features a user-friendly interface that allows customers to browse products, add items to their cart, and securely complete transactions. E-commerce websites often include features such as product search functionality, detailed product descriptions, reviews and ratings, and various payment options to enhance the shopping experience. Additionally, they may incorporate personalized recommendations, promotions, and discounts based on user behavior and preferences. With the convenience of 24/7 accessibility and global reach, E-commerce websites enable businesses to reach a broader audience and streamline the purchasing process for customers, ultimately facilitating seamless online shopping experiences for both buyers and sellers.

**Program Code:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title> Cart | E-COMMERCE WEBSITE BY SAHIL CHOUDHARY </title>

<link rel="stylesheet" href="css/cart.css">

<!-- favicon -->

<link rel="icon" href="https://yt3.ggpht.com/a/AGF-l78km1YyNXmF0r3- 0CycCA0HLA\_i6zYn\_8NZEg=s900-c-k-c0xffffffff-no-rj-mo" type="image/gif" sizes="16x16">

<!-- header links -->

<script src="https://kit.fontawesome.com/4a3b1f73a2.js"></script>

<link href="https://fonts.googleapis.com/css?family=Lato&display=swap" rel="stylesheet">

</head>

<body>

<!-- HEADER -->

<div id="1"></div>

<script>

load("header.html"); function load(url)

{

req = new XMLHttpRequest(); req.open("GET", url, false); req.send(null);

document.getElementById(1).innerHTML = req.responseText;

}

</script>

<!-- CART CONTAINER -->

<div id="cartMainContainer">

<h1> Checkout </h1>

<h3 id="totalItem"> Total Items: 0 </h3>

<div id="cartContainer">

<!-- JS rendered code -->

</div>

</div>

</body>

<!-- FOOTER -->

<div id="4"></div>

<script>

load("footer.html"); function load(url)

{

req = new XMLHttpRequest(); req.open("GET", url, false); req.send(null);

document.getElementById(4).innerHTML = req.responseText;

}

</script>

<script src="/cart.js"></script>

</html> body

{

margin: 0;

font-family: 'Lato', sans-serif;

}

#cartMainContainer

{

padding: 90px 50px;

}

#cartContainer

{

}

img

{

display: grid;

grid-template-columns: 1fr 1fr; grid-gap: 20px;

width: 80px; float: left;

border-radius: 10px; margin-right: 50px;

|  |  |  |  |
| --- | --- | --- | --- |
| }  #boxContainer |  | | |
| { |
| display: grid; |
| grid-gap: 20px;  padding: 20px; |
| width: 100%; |
| /\* float: left; \*/ |
| }  #box |
| { |
| background-color: white; |
| box-shadow: 1px 2px 6px 2px  border-radius: 10px; | rgb(219, | 219, | 219); |
| width: 100%; |  |  |  |
| /\* margin: auto \*/ |  |  |  |
| padding: 20px 10px;  } |  |  |  |
| #box:hover |  |  |  |
| { |  |  |  |
| box-shadow: 0px 1px 3px 2px  } | rgb(185, | 185, | 185); |
| #totalContainer |  |  |  |
| { |  |  |  |
| width: 50%;  padding: 20px; |  |  |  |
| margin: auto; |  |  |  |
| background-color: white; |  |  |  |
| border-radius: 10px;  box-shadow: 1px 2px 6px 2px | rgb(219, | 219, | 219); |
| } |  |  |  |
| #totalContainer:hover |  |  |  |
| {  box-shadow: 0px 1px 3px 2px | rgb(185, | 185, | 185); |
| } |  |  |  |
| button |  |  |  |
| {  padding: 8px 12px; |  |  |  |

background-color: rgb(3, 122, 122); border: none;

border-radius: 5px; cursor: pointer;

}

button a

{

text-decoration: none;

color: white; font-size: 15px;

}

button:hover

{

background-color: rgb(3, 94, 94);

}

button a:focus

{

outline: none;

}

/\* MEDIA QUERY \*/

@media(max-width: 800px)

{

img

{

}

width: 60px;

margin-right: 20px;

h3, h4

{

font-size: 14px;

}

h2

{

font-size: 18px;

}

#cartMainContainer

{

padding: 90px 20px;

}

}

@media(max-width: 555px)

{

#cartContainer

{

display: grid;

grid-template-columns: 1fr;

}

#boxContainer

{

padding: 20px; width: 80%;

}

}

console.clear();

if(document.cookie.indexOf(',counter=')>=0)

{

let counter = document.cookie.split(',')[1].split('=')[1] document.getElementById("badge").innerHTML = counter

}

let cartContainer = document.getElementById('cartContainer')

let boxContainerDiv = document.createElement('div') boxContainerDiv.id = 'boxContainer'

// DYNAMIC CODE TO SHOW THE SELECTED ITEMS IN YOUR CART

function dynamicCartSection(ob,itemCounter)

{

let boxDiv = document.createElement('div') boxDiv.id = 'box' boxContainerDiv.appendChild(boxDiv)

let boxImg = document.createElement('img') boxImg.src = ob.preview boxDiv.appendChild(boxImg)

let boxh3 = document.createElement('h3')

let h3Text = document.createTextNode(ob.name + ' × ' + itemCounter)

// let h3Text = document.createTextNode(ob.name) boxh3.appendChild(h3Text) boxDiv.appendChild(boxh3)

let boxh4 = document.createElement('h4')

let h4Text = document.createTextNode('Amount: Rs' + ob.price) boxh4.appendChild(h4Text)

boxDiv.appendChild(boxh4)

// console.log(boxContainerDiv);

buttonLink.appendChild(buttonText) cartContainer.appendChild(boxContainerDiv) cartContainer.appendChild(totalContainerDiv)

// let cartMain = document.createElement('div')

// cartmain.id = 'cartMainContainer'

// cartMain.appendChild(totalContainerDiv)

return cartContainer

}

let totalContainerDiv = document.createElement('div') totalContainerDiv.id = 'totalContainer'

let totalDiv = document.createElement('div') totalDiv.id = 'total' totalContainerDiv.appendChild(totalDiv)

let totalh2 = document.createElement('h2')

let h2Text = document.createTextNode('Total Amount') totalh2.appendChild(h2Text) totalDiv.appendChild(totalh2)

// TO UPDATE THE TOTAL AMOUNT

function amountUpdate(amount)

{

let totalh4 = document.createElement('h4')

// let totalh4Text = document.createTextNode(amount)

let totalh4Text = document.createTextNode('Amount: Rs ' + amount) totalh4Text.id = 'toth4'

totalh4.appendChild(totalh4Text) totalDiv.appendChild(totalh4) totalDiv.appendChild(buttonDiv) console.log(totalh4);

}

let buttonDiv = document.createElement('div') buttonDiv.id = 'button' totalDiv.appendChild(buttonDiv)

let buttonTag = document.createElement('button') buttonDiv.appendChild(buttonTag)

let buttonLink = document.createElement('a') buttonLink.href = '/orderPlaced.html?' buttonTag.appendChild(buttonLink)

buttonText = document.createTextNode('Place Order') buttonTag.onclick = function()

{

console.log("clicked")

}

//dynamicCartSection()

// console.log(dynamicCartSection());

// BACKEND CALL

let httpRequest = new XMLHttpRequest() let totalAmount = 0

httpRequest.onreadystatechange = function()

{

if(this.readyState === 4)

{

if(this.status == 200)

{

// console.log('call successful'); contentTitle = JSON.parse(this.responseText)

+ counter)

let counter = Number(document.cookie.split(',')[1].split('=')[1]) document.getElementById("totalItem").innerHTML = ('Total Items: '

let item = document.cookie.split(',')[0].split('=')[1].split(" ") console.log(counter)

console.log(item)

let i;

let totalAmount = 0 for(i=0; i<counter; i++)

{

let itemCounter = 1

for(let j = i+1; j<counter; j++)

{

if(Number(item[j]) == Number(item[i]))

{

itemCounter +=1;

}

}

itemCounter

}

totalAmount += Number(contentTitle[item[i]-1].price) \*

dynamicCartSection(contentTitle[item[i]-1],itemCounter) i += (itemCounter-1)

}

}

else

{

}

}

amountUpdate(totalAmount)

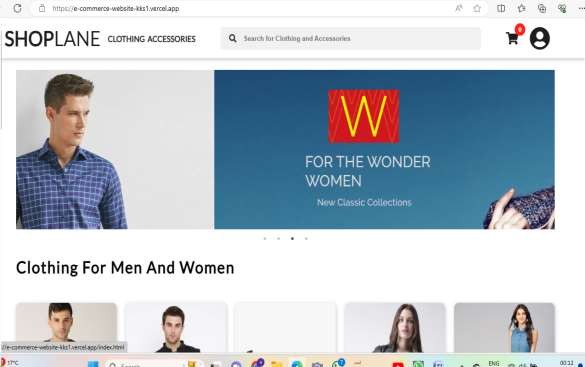
console.log('call failed!');

httpRequest.open('GET', 'https://5d76bf96515d1a0014085cf9.mockapi.io/product', true)

httpRequest.send()

**Output:** Website: https://e-commerce-website-kks1.vercel.app/

nbbnnvn



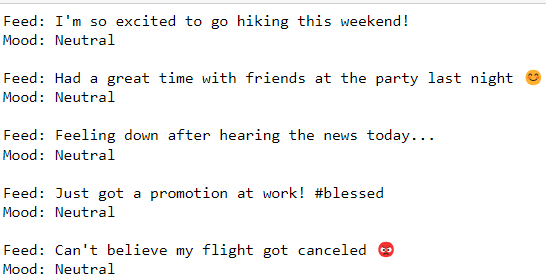
# Macro-Project

**Aim:** Design a system that extracts users' feeds from facebook and determines their moods based on it.

**Description:** The system for extracting users' feeds from Facebook and determining their moods operates through a series of interconnected processes. Initially, the system accesses users' feeds via the Facebook Graph API, requiring user authentication and permissions. Subsequently, relevant data such as posts, comments, and reactions are extracted, encompassing textual, visual, and interactive elements. Natural Language Processing (NLP) techniques are then employed to analyze the textual content, encompassing tokenization, sentiment analysis, and entity recognition. This analysis discerns the prevailing sentiment within each post or comment, categorizing them as positive, negative, or neutral. Advanced techniques for emotion recognition further enrich the analysis, capturing nuanced emotional states beyond mere sentiment. Over time, aggregated sentiment and mood data unveil trends and patterns in users' emotional responses, shedding light on underlying factors influencing mood fluctuations, such as topical discussions or life events. Through visualization tools and comprehensive reporting, the system presents these insights in an accessible format, facilitating a deeper understanding of users' emotional behaviors. Additionally, stringent privacy measures are integrated to ensure compliance with Facebook's API policies and safeguard user data throughout the entire process.

**Program Code:**



**Output:**

# Mini-Project

**Aim:** Design a Credit Card Fraud Detection System.

**Description:**

A Credit Card Fraud Detection System employs machine learning techniques to distinguish between legitimate and fraudulent transactions, ensuring the security of financial transactions for cardholders and financial institutions alike. At its core, the system begins by collecting transaction data, encompassing essential attributes such as transaction amount, timestamp, merchant ID, and cardholder details. This data undergoes preprocessing, where techniques like cleaning and normalization ensure its suitability for analysis. Feature engineering follows, where relevant features are extracted or engineered from the transaction data to capture patterns indicative of fraudulent activity. These features may include transaction frequency, amount, geographical location, and time of day.

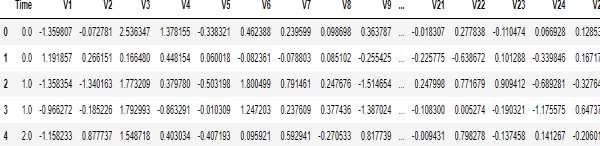
Machine learning models are then trained on labeled historical data, learning patterns that differentiate fraudulent transactions from legitimate ones. Various algorithms are utilized, including logistic regression, decision trees, random forests, support vector machines (SVM), and neural networks, each offering distinct advantages in detecting fraud. Ensemble methods like boosting and bagging are often employed to enhance model performance by combining multiple base models.

**Program Code:**

Libraries used :



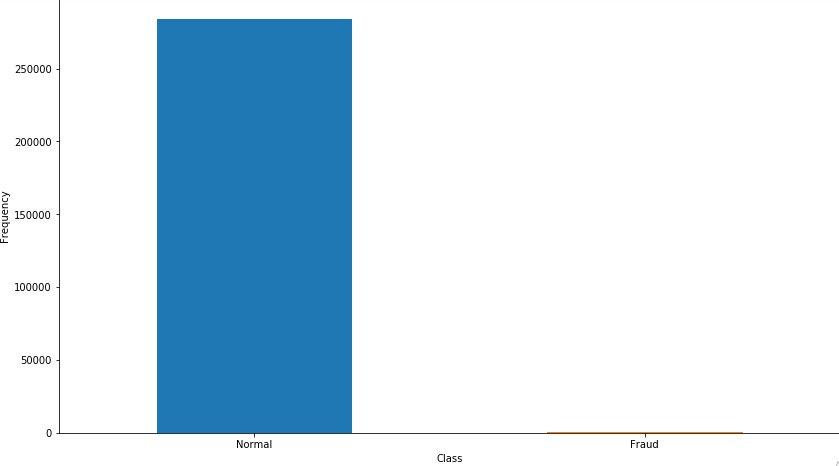
**Datasets:**



**Transaction Class Distribution:**



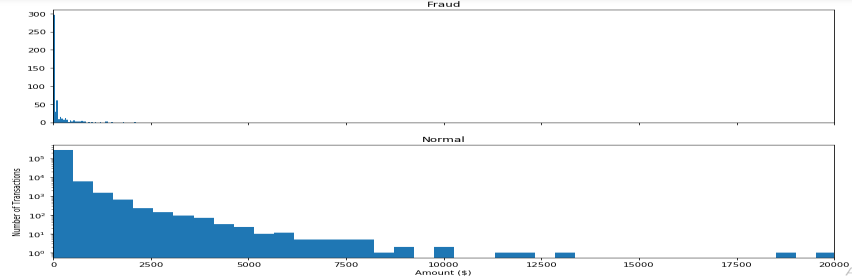
**Graph of Transaction :**



**Amount per Transactions by Class:**



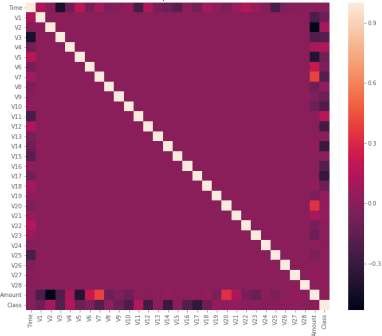
**Graph of Amount Transactions:**



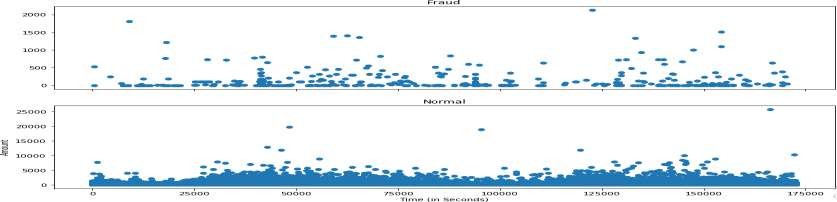
**Correlation:**



**Graph of Correlation :**

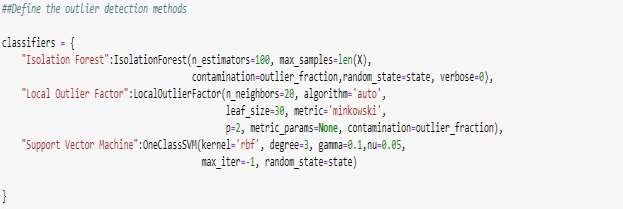


**Time of Transaction vs Amount by Class:**

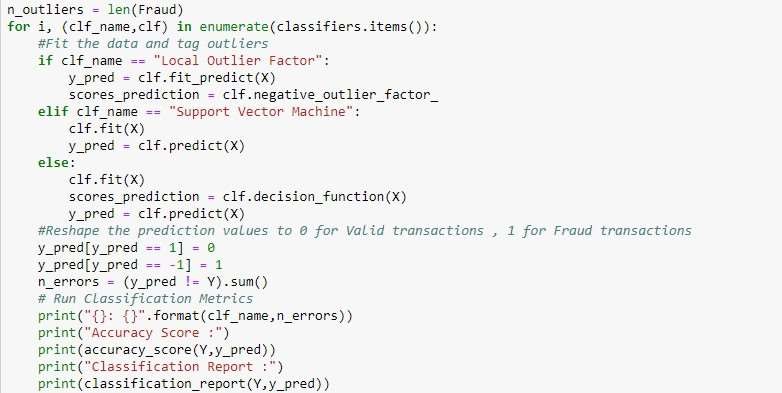


**LOCAL OUTLIER FACTOR:**

**Outlier Detection Method :**

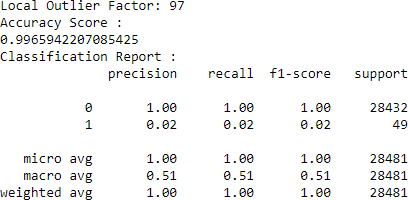


**Outlier Detection Method cont.:**

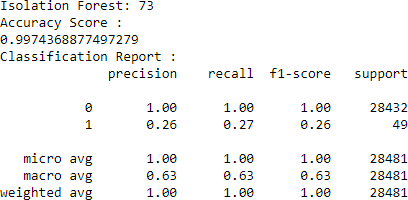


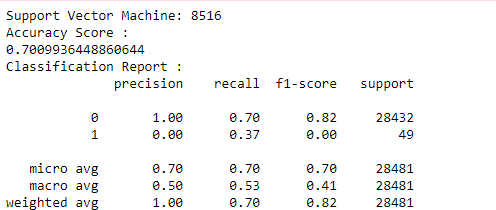
**Screen Shots:**

**Local Outlier Factor:**



**Isolation Forest:**



**Support Vector Machine :**

* While the algorithm does reach over 99.6% accuracy, its precision remains only at 28% when a tenth of the data set is taken into consideration.
* However, when the entire dataset is fed into the algorithm, the precision rises to 33%.
* This high percentage of accuracy is to be expected due to the huge imbalance between the number of valid and number of genuine transactions.