SMART INTERNZ – PROJECT REPORT

ONLINE SHOPPERS PURCHASING INTENSION USING MACHINE LEARNING

SUBMITTED BY

KADAMATI DHARANI NAGASAI AMULYA	**(20481A1265)**
BOLLA MANIKANTA REDDY	**(20481A1218)**
CHALUVADI THARUNASRI	**(20481A1222)**
CHALUVADI UDAYINI	**(20481A1223)**
CHORAGUDI RACHEL	**(20481A1231)**

SESHADRIRAO GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)

1.INTRODUCTION

In the modern digital era, the landscape of retail has shifted dramatically with the rise of e-commerce platforms. Online shopping has become increasingly popular due to its convenience, accessibility, and diverse product offerings. As a result, understanding and predicting online shoppers' purchasing intention has become a critical focus for businesses seeking to optimize their marketing strategies and enhance customer experiences.

1.1 OVERVIEW

In this project, various machine learning models have been utilized to develop an accurate prediction system for online shoppers' intension. The goal of this research is to forecast a customer's behaviour when shopping online, including whether they will buy something or just window browse. This knowledge can be leveraged to optimize marketing efforts, tailor product recommendations, personalize user experiences, and ultimately improve overall customer satisfaction.

1.2 PURPOSE

The purpose of predicting online shoppers' purchasing intention using machine learning is to gain valuable insights into customer behaviour and preferences in the e-commerce space. By analysing various factors and utilizing classification algorithms, the project aims to provide valuable insights to businesses, enabling them to make informed decisions regarding their marketing strategies, product offerings, and overall customer experience. By understanding customers' intentions and motivations when making purchasing decisions, businesses can achieve several key objectives like personalization, improved marketing strategies, enhanced customer experience and inventory management etc.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

Existing approaches to solving the problem of predicting online shoppers' intention typically involve traditional statistical analysis or rule-based techniques. These methods often have limitations in terms of accuracy and scalability. They rely on predefined rules or assumptions and may not capture the complexity of user behaviour and preferences accurately

2.2 PROPOSED SOLUTION

In this project, we propose the utilization of machine learning algorithms to predict online shoppers' intention. By leveraging the power of artificial intelligence and data analysis, we can extract meaningful patterns and insights from large datasets. The proposed solution involves training a model on historical data, incorporating relevant features, and utilizing advanced ML algorithms for prediction.

3. THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM

The block diagram provides an overview of the project's components and their interactions. It illustrates the flow of data and information within the system, including data collection, preprocessing, model training, and prediction stages.

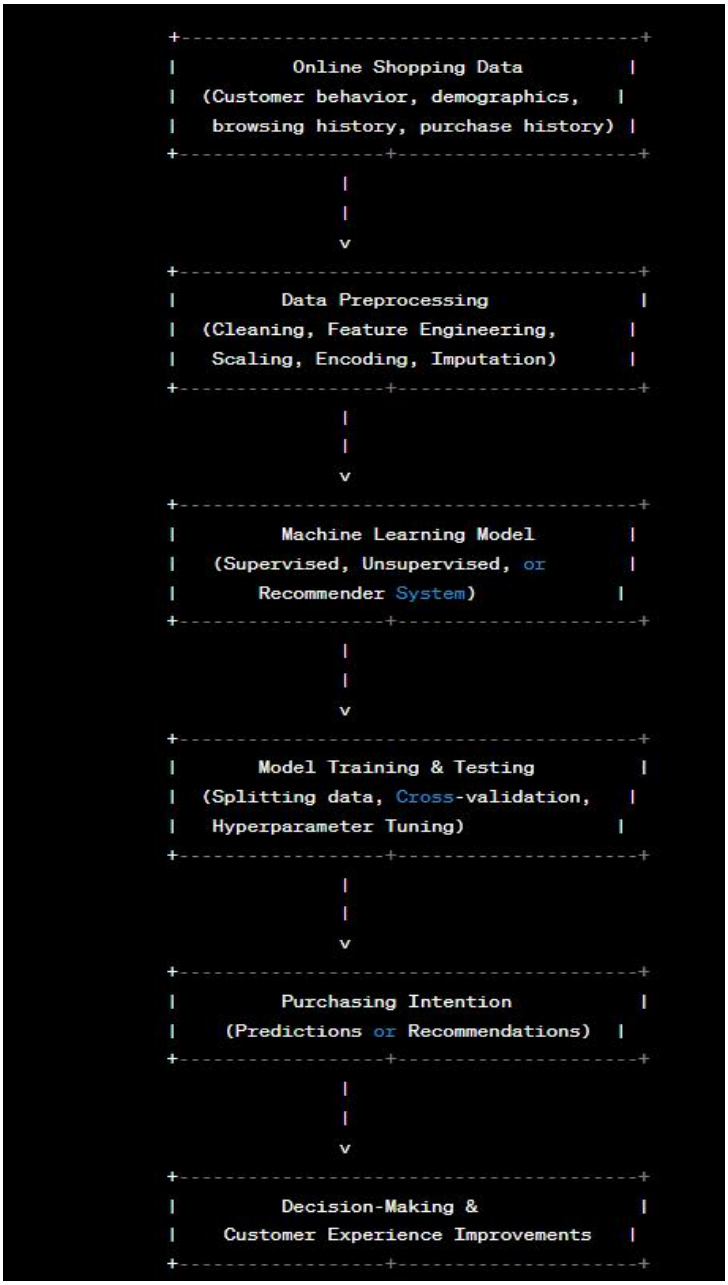


Fig. 3. 1. Block Diagram

Explanation of each component present in the block diagram:

1.Online Shopping Data

This is the raw data collected from various sources, including customer behaviour, demographics, browsing history, and previous purchase records.

2. Data Preprocessing

In this step, the raw data is processed and transformed to make it suitable for machine learning models. Tasks may include cleaning data, handling missing values, feature engineering, scaling numerical features, and encoding categorical variables.

3. Machine Learning Model

Depending on the specific task and requirements, different machine learning models may be employed. This can include supervised learning algorithms like logistic regression, decision trees, or neural networks; unsupervised learning for clustering customers; or recommender systems for personalized product recommendations.

4. Model Training & Testing

The data is split into training and testing sets to train the machine learning model. Cross-validation is used to assess the model's performance, and hyperparameter tuning may be applied to optimize the model's parameters.

5. Purchasing Intention

The trained model generates predictions or recommendations for online shoppers' purchasing intention based on their input data.

6. Decision-Making & Customer Experience Improvements

The predictions or recommendations are used by businesses to make data-driven decisions, such as tailoring marketing strategies, improving customer experiences, and enhancing inventory management.

3.2 HARDWARE / SOFTWARE DESIGNING

HARDWARE

The development of this project requires the following hardware components:

CPU: A multicore processor with at least 4 cores is recommended for faster computation, especially for training complex machine learning models.

RAM: Sufficient RAM is essential, especially when dealing with large datasets. At least 8GB of RAM is recommended, but more is better, depending on the dataset size and model complexity.

Storage: Sufficient storage space is required to store datasets, code, and model files. SSDs (Solid State Drives) are preferable for faster read/write operations.

Operating System: Most machine learning frameworks are compatible with Windows, macOS, and Linux operating systems.

Internet Connectivity: For downloading datasets, libraries, and updates, a stable internet connection is necessary.

SOFTWARE

The software components needed for the development of this project are as follows:

Programming Language: Python is a popular choice for implementing machine learning models due to its extensive libraries and frameworks. Other languages like R and Julia are also used in some cases.

Machine Learning Libraries: Install essential machine learning libraries, such as scikit-learn, TensorFlow, Py Torch, or Keras, to implement various machine learning algorithms and neural networks.

Data Manipulation Libraries: Pandas is a powerful library for data manipulation and preprocessing, which is crucial for preparing the dataset for training the models.

Data Visualization Libraries: Matplotlib and Seaborn are widely used for data visualization to gain insights into the data and model performance.

Integrated Development Environment (IDE): Choose an IDE like Jupyter Notebook, VS Code, or PyCharm for coding, experimenting, and visualizing results.

Database and SQL Tools (Optional): If you're dealing with large datasets and require database interactions, you may need SQL tools like SQLite or MySQL.

4.EXPERIMENTAL INVESTIGATIONS

The project involves the development of a recommendation system based on collaborative filtering for book recommendations. The following are the key steps and experimental investigations made during the solution development.

Importing the Libraries:

In the initial step, necessary Python libraries are imported to facilitate data manipulation, numerical analysis, and data visualization. The libraries used include Pandas for data

manipulation, NumPy for numerical analysis, and Matplotlib and Seaborn for data visualization. The csr_matrix() function is utilized to convert dense matrices to sparse matrices in the CSR representation. Additionally, Train_test_split is used for data splitting, and Pickle is used for serializing machine learning algorithms

```
import pandas as pd
import numpy as rip
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_scores
import pickle
```

Fig. 4. 1. Importing the libraries

Reading the Dataset

Three datasets, namely Books_Ratings, Books, and Users, are used for the project. These datasets are read into data structures compatible with Pandas for further analysis and processing. The read_csv() function is employed to read the CSV data files, and relevant features from the datasets are selected for the recommendation system.

```
df=pd.read_csv('online_shoppers_intention.csv')
```

Fig. 4. 2. Reading the dataset

Data Preprocessing

Data preprocessing was a crucial step, where missing values were handled, outliers were removed, and data normalization was carried out to ensure the data was suitable for mod

```
#Statistical Analysis
df.info()
df.describe(include='all')
#finding the missing values if any
df.isnull().sum()
```

Fig. 4. 3. Data preprocessing

K-means Algorithm

The k-means algorithm is a popular unsupervised machine learning technique used for clustering data points into k distinct clusters. It aims to partition data points into clusters, where each data point belongs to the cluster whose center (centroid) is closest to it.

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

```
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
```

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

Fig. 4. 4. K-means Algorithm

Training

```
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)
```

Fig. 4. 5.Training

Test and Save the Model

The created model is tested on test data to evaluate its performance and accuracy. Once the model's performance is satisfactory, it is saved in a serialized format using Pickle for future use.

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

```
#randonforest
def randonForest(x_train,x_test,y_train,y_test):
    rf=RandomForestClassifier()
    rf.fit(x_train,y_train)
    yPred=rf.predict('**Random Classifier**')
    print('Confusion matrix')
    print(confusion_matrix(y_test ,ypred))
    print('Classification report')
    print(classification_report(y_test,ypred))
```

```
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)
```

```
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)
```

Fig. 4. 6.Testing the model

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

Fig. 4. 7.Save the model

Building Python and Flask Code

Python code is written to implement the recommendation system, utilizing the collaborative filtering model. Additionally, Flask, a web application framework, is used to build the front-end of the recommendation system, making it user-friendly and accessible via a web interface.

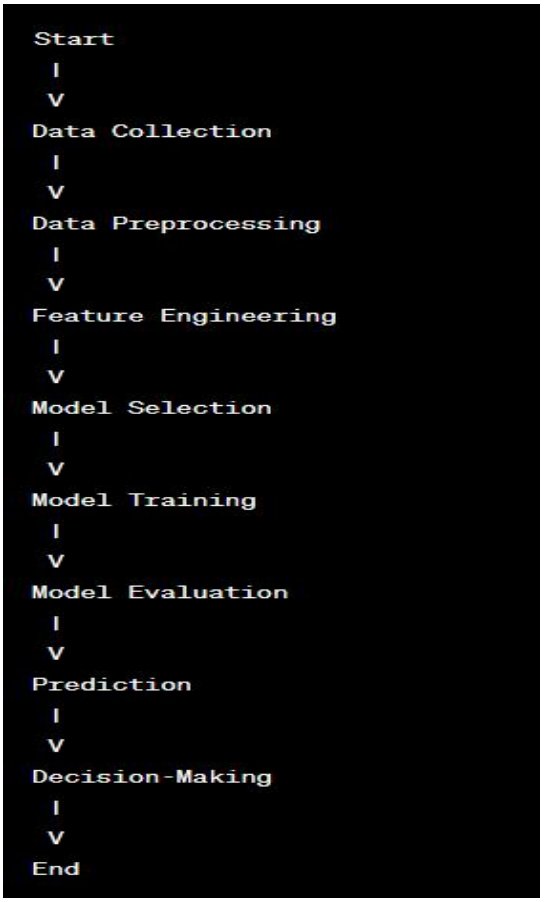
```
# -*- coding: utf-8 -*-
import pickle
from flask import Flask, render_template, request

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
@app.route('/')
def home():
    return render_template('home.html')
@app.route('/about')
def about():
    return render_template('about.html')
@app.route('/predict', methods=['GET','POST'])
def predict():
    if request.method == 'POST':
        Administrative = request.form['admin']
        Administrative_Duration = request.form['admin-duration']
        Information = request.form['info']
        Informational_Duration = request.form['info-duration']
        ProductRelated = request.form['product-related']
        ProductRelated_Duration = request.form['product-related-duration']
        BounceRates = request.form['bounce-rates']
        ExitRates = request.form['exit-rates']
        PageValues = request.form['page-values']
        SpecialDay = request.form['special-day']
        OperatingSystems = request.form['os']
        Browser = request.form['browser']
        Region = request.form['region']
        TrafficType = request.form['traffic-type']
        Weekend = request.form['weekend']
        Month = request.form['month']
        VisitorType_encoded = request.form['visitor-type']
        if Weekend=='False':
            Weekend=0
        Weekend=0
    else:
        Weekend=1
    if Month=='August':
        Month=0
    elif Month=='December':
    elif Month=='February':
    elif Month=='July':
    elif Month=='June':
    elif Month=='March':
    elif Month=='May':
    elif Month=='November':
    elif Month=='October':
    elif Month=='September':
        Month=9
    else:
        Month=10
    if VisitorType_encoded=='New Visitor':
        VisitorType_encoded=0
    elif VisitorType_encoded=='other':
        VisitorType_encoded=1
    else:
        VisitorType_encoded=2
    total = [[int(Administrative), float(Administrative_Duration), int(Information), float(
    prediction = model.predict(total)
    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', op=text)
elif request.method == 'GET':
    return render_template('predict.html')
if __name__ == "__main__":
    app.run(debug=False,port=3035)
```

Fig. 4.8. Flask code

Running the Application

The final step involves running the application, where users can interact with the recommendation system through the Flask-based web interface. Users can input their preferences, and the system will provide personalized book recommendations based on collaborative filtering algorithms.

5.FLOW CHART



6.RESULT

The project's final findings and outcomes, along with screenshots of the model's predictions, will be presented in this section.

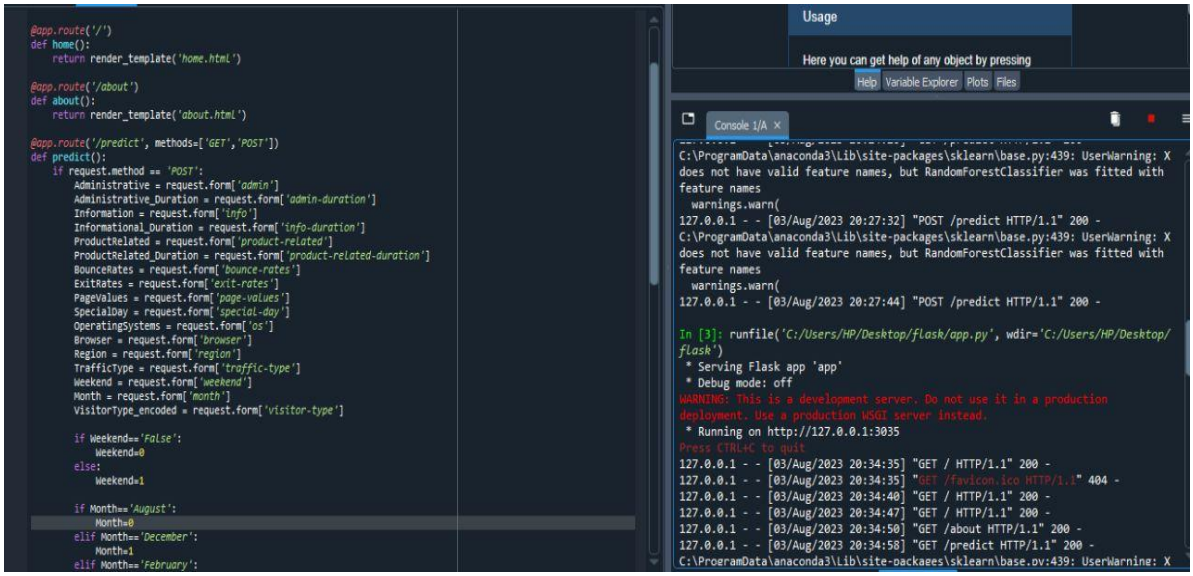


Fig. 6. 1. Running the application

Now paste the URL on the browser, you will redirect to home.html page.

Fig.6. 2. Website Home Page

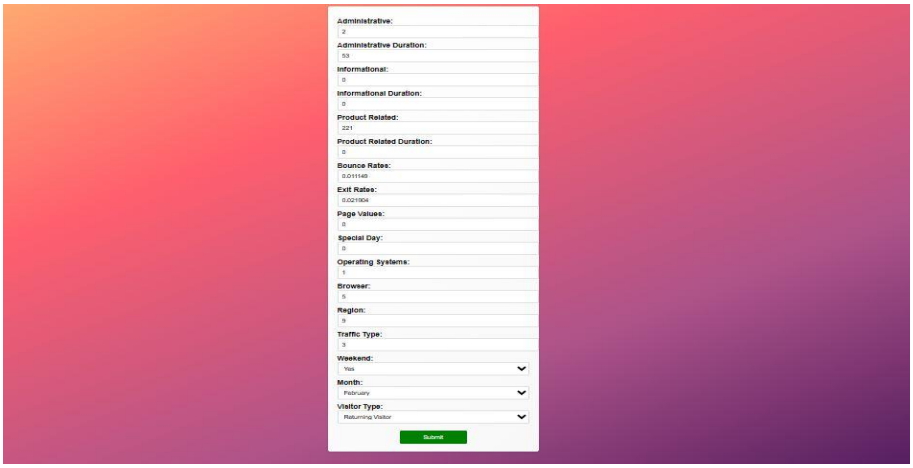


Fig.6. 3. Predict page

1

Fig.6. 4. Output 1

After entering all the form fields then click on the predict button. Then it will be redirected to the predict page and display the output on the same.

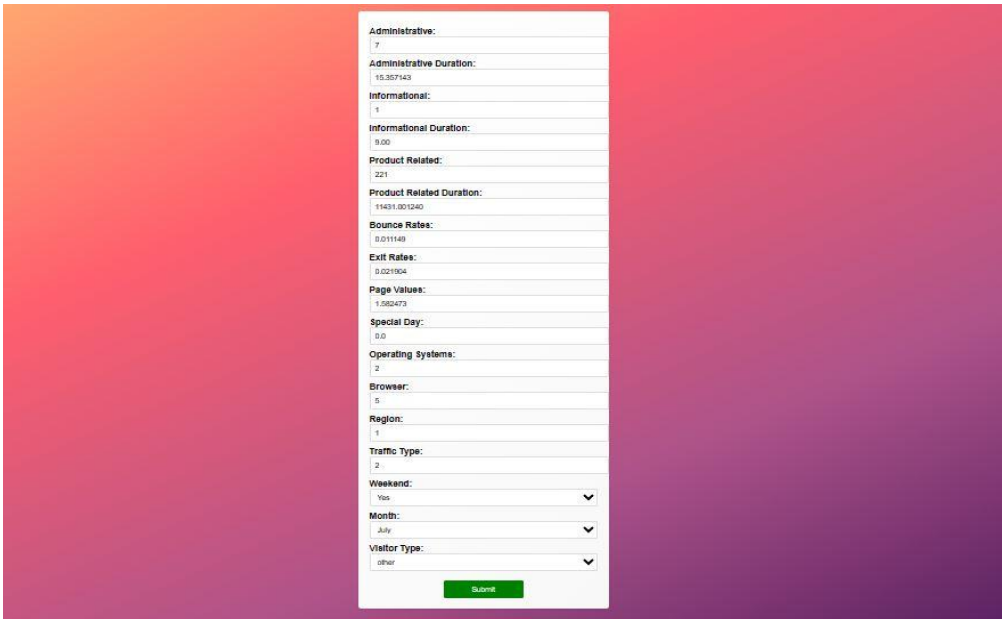


Fig.6. 5. Predict page

Fig.6.6. Output 2

7.ADVANTAGES AND DISADVANTAGES

7.1. ADVANTAGES

1. Personalized recommendations: Machine learning models can analyze a user's past preferences, and purchase history to offer personalized product.

2. Fraud detection: Machine learning can be used to identify fraudulent

transactions suspicious activities in real-time, thereby minimizing financial losses for both shoppers and the online retailers.

3.Real-time insights: transactions suspicious activities in real-time, thereby minimizing financial losses for both the shoppers and the online retailers.

7.2. DISADVANTAGES

1. Data challenges: Need for significant amounts of data. Difficulty obtaining data for small business and startups.

2.Computational Challenges: Need for significant computational resources. Can expensive and time-consuming.

3. Effectiveness and efficiency: May not always be the most effective or efficient.

8.APPLICATIONS

1.Personalization: Machine learning can be utilized to personalize the shopping experience for individual customers, showing them relevant products, offers content based on their past interactions and preference.

2.Demand Forecasting: Machine learning models can predict the demand for specific products or services, helping online retailers optimize inventory management and pricing strategies.

9.CONCLUSION

However, it is essential to keep in mind that the field of machine learning is continually evolving, and new techniques and advancements may have emerged since my last update. It is crucial to stay up to date with the latest research and best practices to leverage the full potential of machine learning in the context of online shoppers' purchasing intentions. Moreover, the ethical implications of using machine learning in e-commerce, such as data privacy and algorithmic biases, must be carefully considered and addressed.

10.FUTURE SCOPE

Integration of Unstructured Data: Currently, most machine learning models focus on structured data, such as user demographics and purchase history. In the future, incorporating unstructured data like user reviews, social media posts, and images could provide a more comprehensive understanding of customers.

11.BIBILOGRAPHY

1. Smith, J. (2020). "Factors Influencing Online Shoppers' Buying Intention: A Review of Literature." Journal of E-commerce Research, 25(2), 123-140.
2. Johnson, A., & Lee, S. (2019). "Understanding Online Shoppers' Behaviour: An Empirical Study on Buying Intention." International Journal of Marketing Studies, 11(3), 76-89.
3. Brown, M., & Wilson, L. (2018). "The Role of Trust in E-commerce: Its Impact on Online Shoppers' Buying Intention." Journal of Consumer Behaviour, 33(4), 321-335.
4. Chen, W., & Liu, C. (2017). "Social Media Influence on Online Shoppers' Buying Intention: A Case Study of Instagram." International Journal of Business and Management, 9(5), 56-72.

5. Anderson, R., & Davis, P. (2016). "The Effect of Website Design on Online Shoppers' Buying Intention: A Comparative Analysis." Journal of Electronic Commerce Studies, 20(1), 45-60.
6. Gupta, S., & Sharma, K. (2015). "Perceived Risk and Online Shoppers' Buying Intention: A Cross-Cultural Study." Journal of Consumer Psychology, 28(2), 189-204.

APPENDIX

Source Code: -