Student Name: THILAGA LATHA M K
Register Number: C3S20562
Institution: PKN ARTS AND SCIENCE COLLEGE
Department: INFORMATION TECHNOLOGY
Date of Submission: 19/01/2026
Github Repository Link:
https://github.com/latha1514309/Customer-Purchase-Frequency1-ML

Project Title: Customer Purchase Frequency Analyzer

1. Problem Statement

In modern retail and e-commerce systems, understanding how frequently customers make purchases is critical for improving customer retention, marketing strategies, and revenue forecasting. Businesses require predictive systems that can estimate customer purchase frequency based on demographic and behavioral attributes.

This project aims to build a **machine learning regression model** that predicts **Purchase Frequency**, a continuous numerical value, using customer-related features such as age, income, spending score, membership duration, and last purchase amount. Accurate prediction helps businesses plan promotions, personalize offers, and improve customer engagement.

2. Abstract

This project focuses on predicting customer purchase frequency using supervised machine learning techniques. A structured customer dataset is collected and preprocessed to handle missing values and inconsistencies. Exploratory Data Analysis (EDA) is performed to understand feature relationships and trends. A Linear Regression model is trained on scaled data and evaluated using Mean Squared Error (MSE) and R² score. The trained model is deployed using a Gradio-based web application for real-time predictions. The system provides a simple, interpretable, and business-relevant solution for customer behavior prediction.

3. System Requirements

- **Hardware Requirements:** - Minimum 4 GB RAM - Any modern processor (Intel i3 or equivalent)

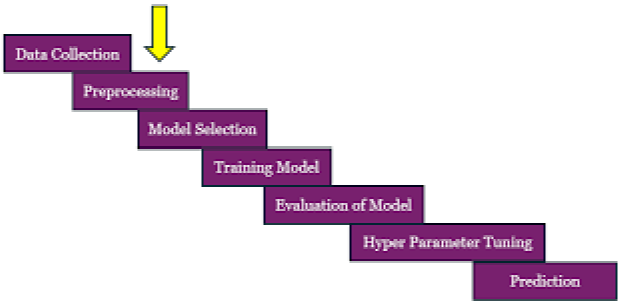
- **Software Requirements:** - Python 3.8 or above - Google Colab / Jupyter Notebook / VS Code - Required Libraries: - pandas - numpy - matplotlib - seaborn - scikit-learn - gradio

4. Objectives

- To analyze customer purchase behavior using historical data
- To clean and preprocess real-world customer data
- To build a regression model that predicts purchase frequency
- To evaluate model performance using suitable metrics
- To deploy the model using an interactive web interface
- To provide actionable insights for business decision-making

5. Flowchart of Project Workflow

Workflow Steps: 1. Data Collection 2. Data Preprocessing 3. Exploratory Data Analysis (EDA) 4. Feature Engineering 5. Model Training 6. Model Evaluation 7. Deployment



6. Dataset Description

Source: CSV-based customer purchase dataset

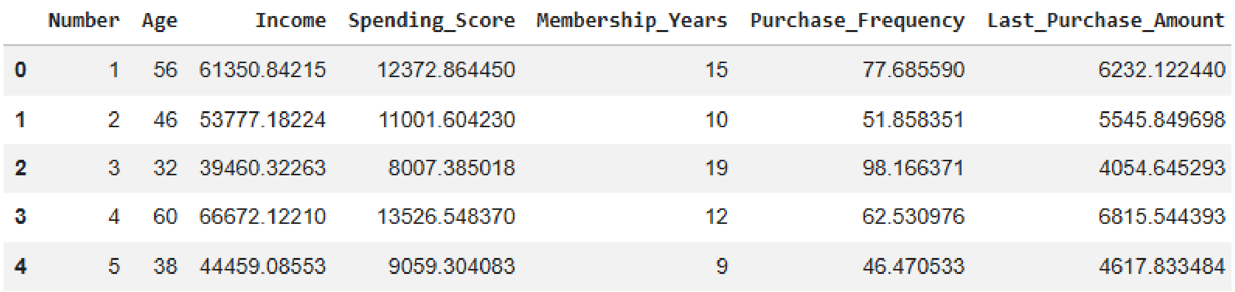
Type: Structured tabular data

Nature: Public / Educational dataset

Number of Records: 1000+ rows (approx.)

Attributes:

- Age
- Income
- Spending_Score
- Membership_Years
- Last_Purchase_Amount
- Purchase_Frequency (Target)



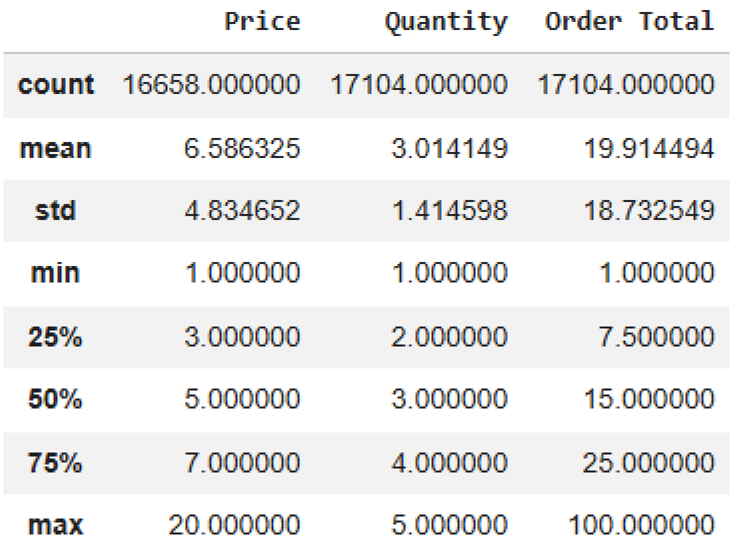
	Number	Age	Income	Spending_Score	Membership_Years	Purchase_Frequency	Last_Purchase_Amount
0	1	56	61350.84215	12372.864450	15	77.685590	6232.122440
1	2	46	53777.18224	11001.604230	10	51.858351	5545.849698
2	3	32	39460.32263	8007.385018	19	98.166371	4054.645293
3	4	60	66672.12210	13526.548370	12	62.530976	6815.544393
4	5	38	44459.08553	9059.304083	9	46.470533	4617.833484

7. Data Preprocessing

The dataset was examined for quality issues before model training.

Preprocessing Steps:

- Identified missing values
- Filled numerical missing values using mean
- Checked for duplicate records
- Verified data types and consistency



	Price	Quantity	Order Total
count	16658.000000	17104.000000	17104.000000
mean	6.586325	3.014149	19.914494
std	4.834652	1.414598	18.732549
min	1.000000	1.000000	1.000000
25%	3.000000	2.000000	7.500000
50%	5.000000	3.000000	15.000000
75%	7.000000	4.000000	25.000000
max	20.000000	5.000000	100.000000

8. Exploratory Data Analysis (EDA)

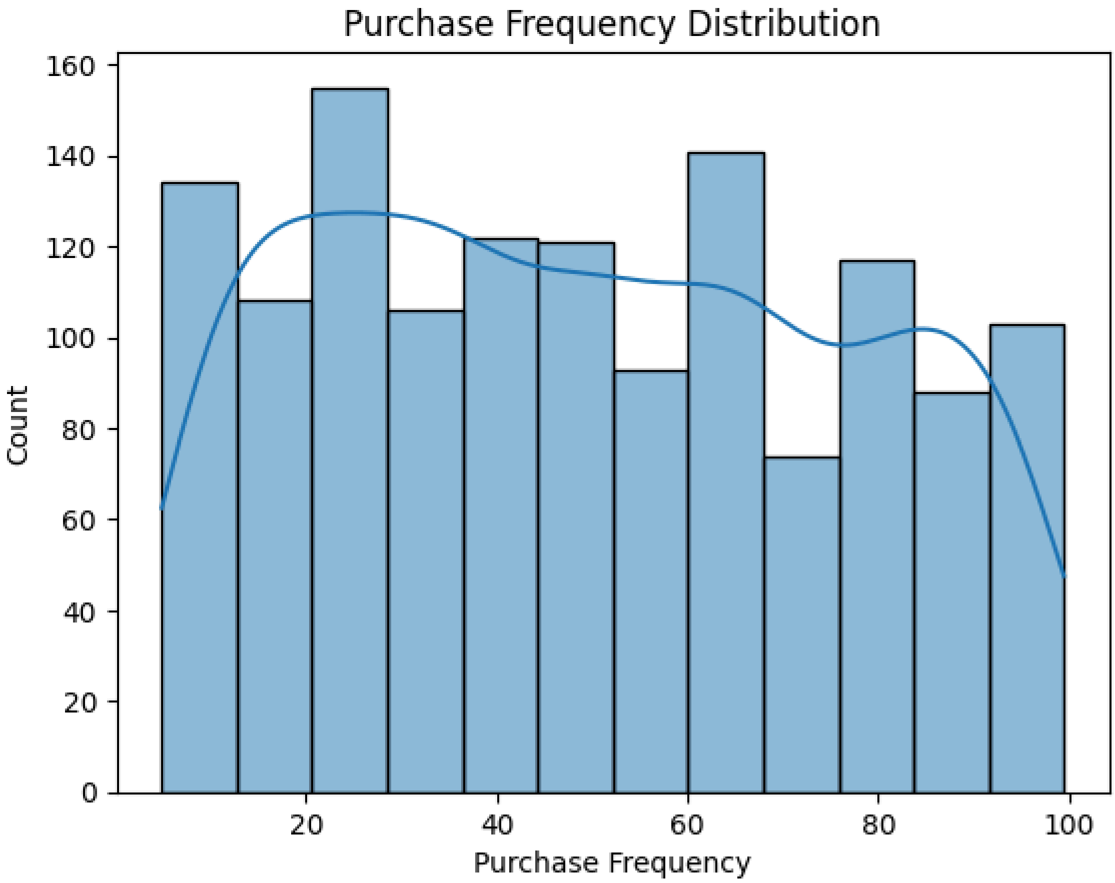
EDA was conducted to understand data distribution and relationships.

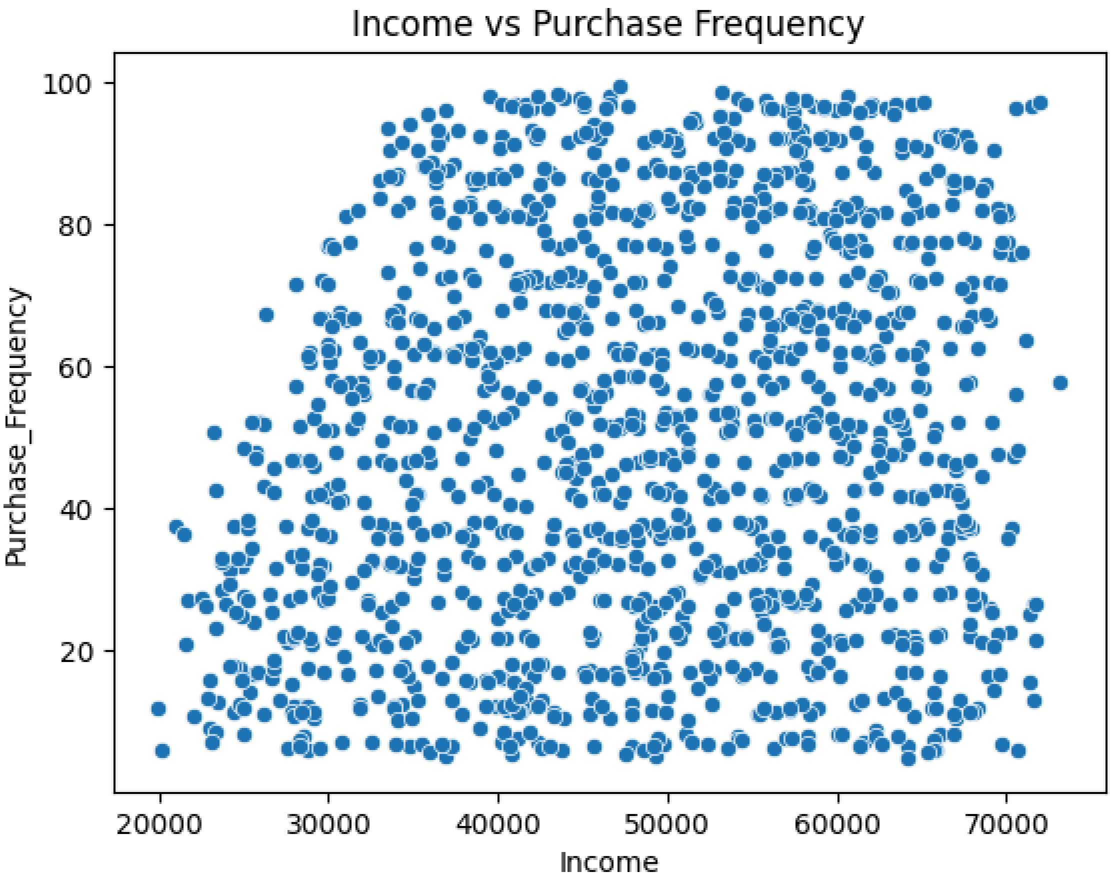
Visualizations Used:

- Histogram of Purchase Frequency
- Scatter plot between Income and Purchase Frequency

Key Insights:

- Purchase frequency follows a near-normal distribution
- Income shows a positive correlation with purchase frequency
- Behavioral features influence purchasing patten





9. Feature Engineering

Selected Features:

- Age
- Income
- Spending_Score
- Membership_Years
- Last_Purchase_Amount

Feature Scaling:

- StandardScaler was applied to normalize feature values
- Prevents dominance of high-magnitude variables
- Improves regression model performance

10. Model Building

Model Used:

- Linear Regression

Justification:

- Suitable for continuous value prediction
- Easy to interpret and explain in viva
- Acts as a strong baseline regression model

The dataset was split into training and testing sets using an 80:20 ratio.

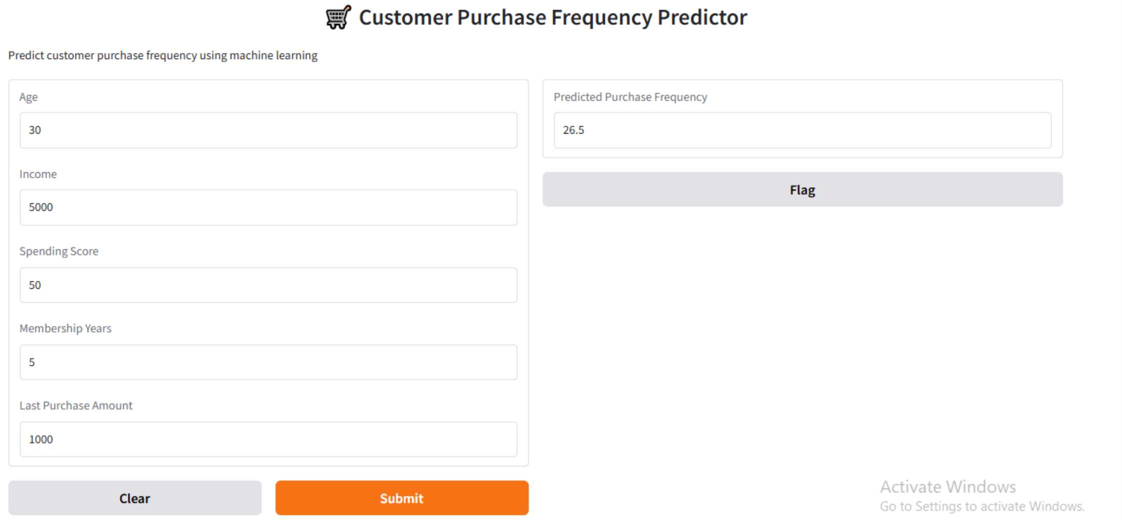
11. Model Evaluation

- **MetRics Used:**
    - Mean Squared Error (MSE)
    - R² Score

The evaluation metrics indicate how well the model predicts unseen data.

12. Deployment

- **Deployment Tool:** Gradio interface
- **Public Link:** https://c776a7532d259ddfb0.gradio.live/
- **Method:** Local/Colab-based web interface
- **Features:**
    - User input form
    - Real-tme order total prediction



🛒 Customer Purchase Frequency Predictor

Predict customer purchase frequency using machine learning

Age: 30

Income: 5000

Spending Score: 50

Membership Years: 5

Last Purchase Amount: 1000

Predicted Purchase Frequency: 26.5

13. Source Code

```
import pandas as pd

import numpy as np


from google.colab import files

uploaded = files.upload()

df = pd.read_csv('Customer Purchase Data.csv')


# Display first 5 rows

df.head()

# Shape of dataset

print("Shape:", df.shape)


# Column names

print("Columns:", df.columns.tolist())


# Dataset info

df.info()


# Statistical summary

df.describe()

# Missing values

print("Missing values:\n", df.isnull().sum())


# Duplicate rows

print("Duplicate rows:", df.duplicated().sum())

# Fill numeric missing values with mean
```

```
numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())

# Fill categorical missing values with mode
categorical_cols = df.select_dtypes(include=['object']).columns
for col in categorical_cols:
    df[col] = df[col].fillna(df[col].mode()[0])
  import seaborn as sns
import matplotlib.pyplot as plt
sns.histplot(df['Purchase_Frequency'], kde=True)
plt.title("Purchase Frequency Distribution")
plt.xlabel("Purchase Frequency")
plt.show()
sns.scatterplot(x='Income', y='Purchase_Frequency', data=df)
plt.title("Income vs Purchase Frequency")
plt.show()
X = df.drop('Purchase_Frequency', axis=1)
y = df['Purchase_Frequency']


print("Features:\n", X.columns)
print("Target:\n Purchase_Frequency")
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42
)

model = LinearRegression()
model.fit(X_train, y_train)
from sklearn.metrics import mean_squared_error, r2_score

y_pred = model.predict(X_test)

print("Mean Squared Error (MSE):", mean_squared_error(y_test, y_pred))
print("R² Score:", r2_score(y_test, y_pred))
new_customer = {
    'Age': 40,
    'Income': 52000,
    'Spending_Score': 9500,
    'Membership_Years': 8,
    'Last_Purchase_Amount': 4800
}

new_df = pd.DataFrame([new_customer])

new_scaled = scaler.transform(new_df)
```

```
prediction = model.predict(new_scaled)


print("🛒 Predicted Purchase Frequency:", round(prediction[0], 2))
!pip install gradio
import gradio as gr


def predict_purchase_frequency(age, income, spending_score, membership_years, last_purchase_amount):


    input_df = pd.DataFrame([{
        'Age': age,
        'Income': income,
        'Spending_Score': spending_score,
        'Membership_Years': membership_years,
        'Last_Purchase_Amount': last_purchase_amount
    }])


    input_scaled = scaler.transform(input_df)
    prediction = model.predict(input_scaled)


    return round(prediction[0], 2)
  inputs = [
    gr.Number(label="Age"),
    gr.Number(label="Income"),
    gr.Number(label="Spending Score"),
    gr.Number(label="Membership Years"),
    gr.Number(label="Last Purchase Amount")
]

output = gr.Number(label="Predicted Purchase Frequency")

gr.Interface(
    fn=predict_purchase_frequency,
    inputs=inputs,
    outputs=output,
    title="🛒 Customer Purchase Frequency Predictor",
    description="Predict customer purchase frequency using machine learning"
).launch()
```

14. Future Scope

- Implement advanced models like Random Forest or XGBoost
- Add customer segmentation using clustering
- Incorporate time-based purchasing behavior
- Deploy as a REST API for enterprise use
- Improve performance using hyperparameter tuning