1. Discuss potential ethical concerns that can arise with the use of AI and ML in decision-making processes in sensitive areas such as finance, healthcare, or criminal justice

AI presents three major areas of ethical concern for society:

1. Bias and Discrimination - AI systems can perpetuate and amplify existing biases and discrimination in society, leading to unfair and discriminatory outcomes For example, if an AI system is trained on biased data on criminal records, it can learn and replicate those biases to identify new records that are potential not crime and make a bad decision.
2. Responsibility and Accountability - It can be challenging to determine who is responsible and accountable for the decisions made by AI systems For example, if an AI system makes a wrong diagnosis in healthcare, who can be considered as responsible for the consequences faced by the patient.
3. Privacy and Security - AI systems can collect and process vast amounts of personal data, raising concerns about privacy and security. For example, if an AI system is used to screen job applicants, it can collect and process sensitive personal information which is out of concent from the users of the system.

https://www.captechu.edu/blog/ethical-considerations-of-artificial-intelligence (https://www.captechu.edu/blog/ethical-considerations-of-artificial-intelligence)
https://news.harvard.edu/gazette/story/2020/10/ethical-concerns-mount-as-ai-takes-bigger-decision-making-role/ (https://news.harvard.edu/gazette/story/2020/10/ethical-concerns-mount-as-ai-takes-bigger-decision-making-role/)

2. Investigate a real-world incident where the use of AI in decision-making led to unintended consequences. Detail the incident, identify where things went wrong, and discuss what could have been done to avoid the incident.

Recruiting bias - Amazon built an AI-based tool to “out recruit” other tech firms in the tech brains arms race. The company trained their models to look for top talent in the resumes. However, the AI models were trained using tainted data collected over a 10-year period in which the vast majority of candidates were men. The AI model gave higher priority to male resumes, and low scoring for the resumes that participated in women’s activities, even if the names were anonymized, such as “Women’s chess club captain.” After many attempts to make the program gender-neutral, Amazon gave up and disbanded the tool and the team.

The analysis on the data for gender disparity could be done before training the model and also would segment the data in such a way that both could give equal weightage.
https://hbr.org/2022/09/ai-isnt-ready-to-make-unsupervised-decisions (https://hbr.org/2022/09/ai-isnt-ready-to-make-unsupervised-decisions)

1. Data Exploration and Visualization: a. Load the dataset using pandas.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [3]:
```
# Load the dataset using pandas
data = pd.read_csv('e_commerce_clv_dataset.csv')
data
```

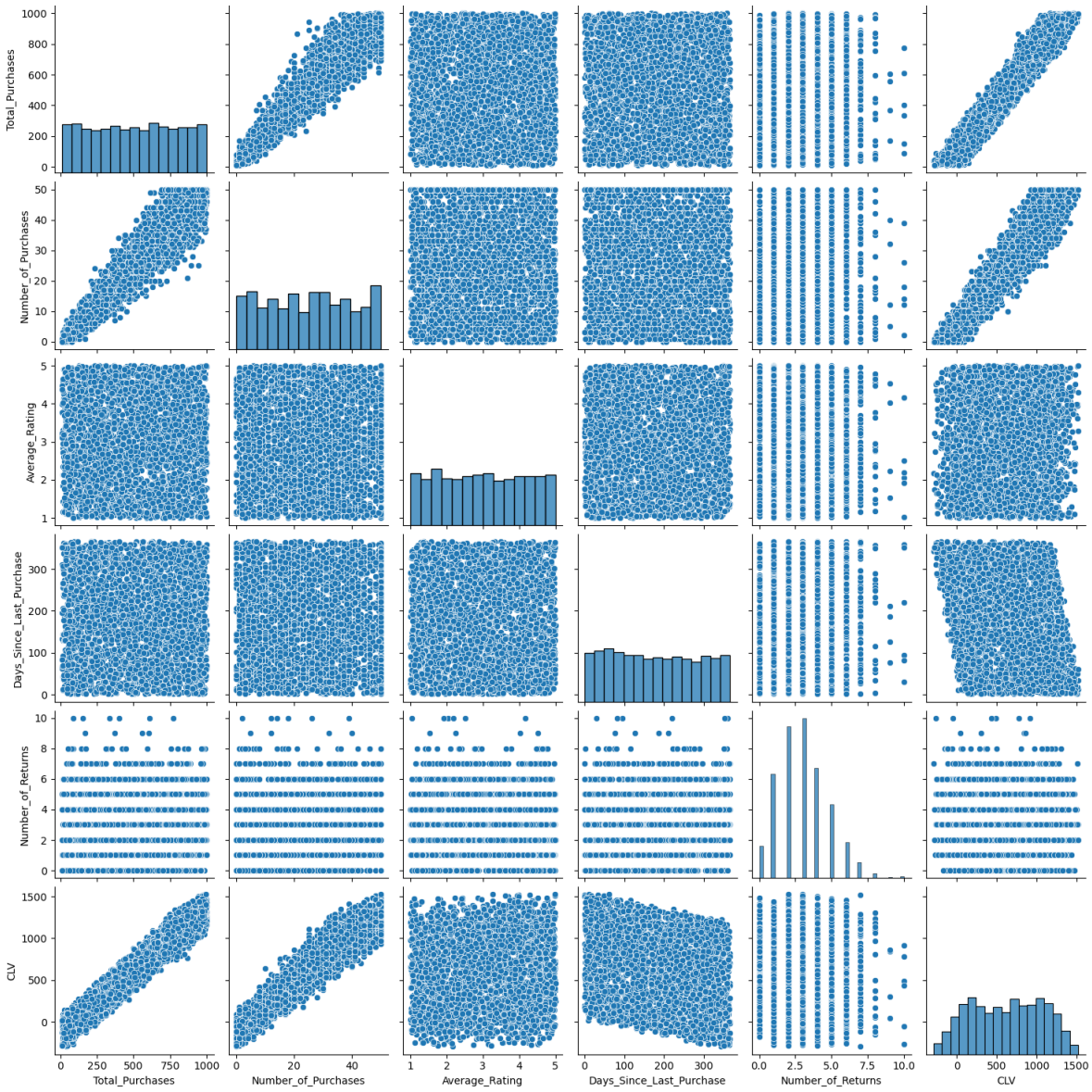
Out[3]:

	Total_Purchases	Number_of_Purchases	Average_Rating	Days_Since_Last_Purchase	Numb
0	380.794718	21	4.226859	337	
1	951.207163	42	1.066158	57	
2	734.674002	38	1.747232	67	
3	602.671899	39	3.619621	46	
4	164.458454	7	2.503037	94	
...	...	...	...	...	
2995	866.642801	37	4.239015	44	
2996	165.700476	7	2.630393	73	
2997	316.689981	14	1.105818	261	
2998	297.145077	17	1.232253	231	
2999	872.699894	49	4.424976	33	

3000 rows × 6 columns

In [7]:
```
# Visualize the relationships between features and the target variable ie. CLV
sns.pairplot(data)
```

Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x2033a8186d0>
```



Observations:

CLV with Total Purchases: In this case it is observer that there is a very high linearity between the two attributes considered here as all the datapoints plotted here are in a linear format and close to each other.

CLV with Number of Purchases: It has been observed a siminal high linerity between these two attributes and the datapoints are closed to each other stating these attributes are related to each other, if one gose up the other gose up.

CVL with Average Rating: In this case there is no linearity between the attributes considered and all the datapoints are scattered across the ploted area.

2. Data Pre-processing: a. Split the data into training and test sets (80% train, 20% test).

In [14]:
```
ind_variable = data[['Total_Purchases', 'Number_of_Purchases', 'Average_Rating
dep_variable = data['CLV'].values
```

In [15]:
```
ind_variable
```

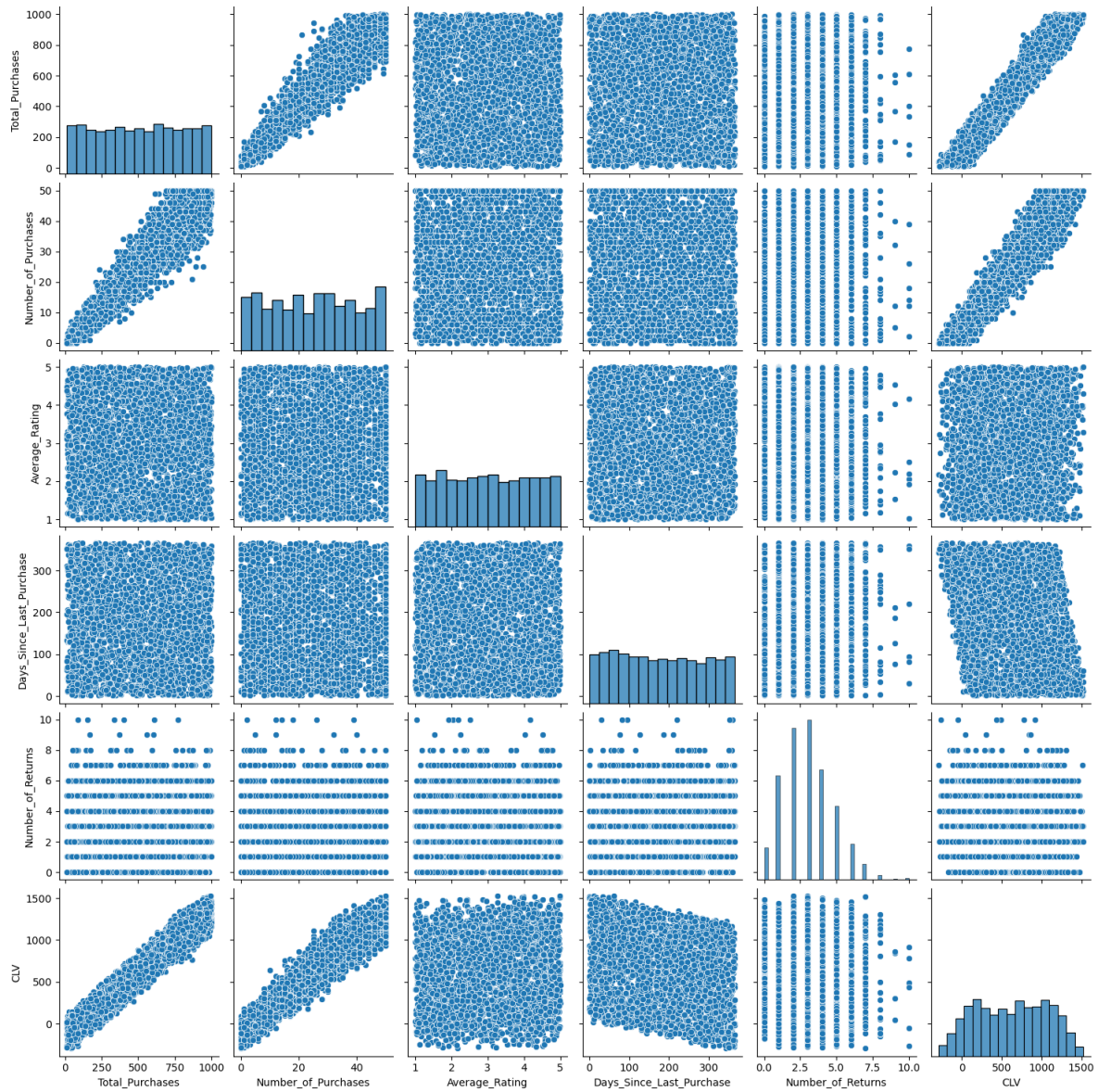
Out[15]:

	Total_Purchases	Number_of_Purchases	Average_Rating	Days_Since_Last_Purchase	Numb
0	380.794718	21	4.226859	337	
1	951.207163	42	1.066158	57	
2	734.674002	38	1.747232	67	
3	602.671899	39	3.619621	46	
4	164.458454	7	2.503037	94	
...	...	...	...	...	
2995	866.642801	37	4.239015	44	
2996	165.700476	7	2.630393	73	
2997	316.689981	14	1.105818	261	
2998	297.145077	17	1.232253	231	
2999	872.699894	49	4.424976	33	

3000 rows × 5 columns

3. Model Development: a. Using the libraries and functions we used in the lab session of LR and MLR, create an MLR model. b. Train the model on the training set. c. Predict CLV values on the test set.

In [16]:
```
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split

# Split the data into training and test sets (80% train, 20% test).
X_train, X_test, y_train, y_test = train_test_split(ind_variable, dep_variable

model = LinearRegression()
model.fit(X_train, y_train)

y_predictions = model.predict(X_test)

sns.pairplot(data)
plt.show()

#Print y intercept
print(f"Intercept (Bias) : {model.intercept_}\n")
```



```
Intercept (Bias) : 3.410605131648481e-13
```

In [20]:
```
# a. Evaluate the model using appropriate metrics such as Mean Absolute Error

#Calculate Mean Absolute Error
mae = metrics.mean_absolute_error(y_test, y_predictions)
print(f"Mean Absolute Error (MAE) : {mae}")
#Calculate MSE

mse = metrics.mean_squared_error(y_test, y_predictions)
print(f"Mean Square Error (MSE) : {mse}")

root_mse = np.sqrt(metrics.mean_squared_error(y_test, y_predictions))
print(f"Root Mean Square Error (RMSE) : {root_mse}")

r2squared = metrics.r2_score(y_test, y_predictions)
print(f"RSquared (R2) : {r2squared}")
```

```
Mean Absolute Error (MAE) : 7.840927906954675e-13
Mean Square Error (MSE) : 9.293614763573067e-25
Root Mean Square Error (RMSE) : 9.640339601680568e-13
RSquared (R2) : 1.0
```

In [21]:
```
# b. Interpret the coefficients of the model.

print('Coefficients:')
coef_arr = model.coef_

print(f"Average Response Time (s) : {coef_arr[0]}")
print(f"Number of Features : {coef_arr[1]}")
print(f"Number of Bugs Reported : {coef_arr[2]}")
print(f"Training Hours Provided : {coef_arr[3]}")
```

```
Coefficients:
Average Response Time (s) : 0.9999999999999962
Number of Features : 10.00000000000011
Number of Bugs Reported : 19.999999999999996
Training Hours Provided : -1.0000000000000062
```

c. Interpret the significance and impact of each feature on the CLV based on the coefficients.

Average Response Time (s):

Coefficient: 0.9999999999999962 For every one-unit increase in the average response time, the Customer Lifetime Value (CLV) is expected to increase by approximately 1 unit.

Number of Features:

Coefficient: 10.00000000000011 For every one-unit increase in the number of features, the CLV is expected to increase by approximately 10 units.

Number of Bugs Reported:

Coefficient: 19.999999999999996 For every one-unit increase in the number of bugs reported, the CLV is expected to increase by approximately 20 units.

Training Hours Provided:

Coefficient: -1.0000000000000062 For every one-unit increase in the number of training hours provided, the CLV is expected to decrease by approximately 1 unit.

#WEKA

=== Run information ===

Scheme: weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4
Relation: e_commerce_clv_dataset Instances: 3000 Attributes: 6 Total_Purchases Number_of_Purchases Average_Rating Days_Since_Last_Purchase Number_of_Returns CLV
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

Linear Regression Model

CLV =

```
      1      * Total_Purchases +
     10      * Number_of_Purchases +
     20      * Average_Rating +
     -1      * Days_Since_Last_Purchase +
     -5      * Number_of_Returns +
      0
```

Time taken to build model: 0.07 seconds

=== Cross-validation === === Summary ===

Correlation coefficient 1
Mean absolute error 0
Root mean squared error 0
Relative absolute error 0 % Root relative squared error 0 % Total Number of Instances 3000