Q#2

Customer	Age	Gender	Loyalty_M	Product_T	Rating	Order_Sta	Payment_	Total_Pric	Unit_Price	Quantity	Shipping_Type
1000	53	Male	No	Smartpho	2	Cancelled	Credit Car	5538.33	791.19	7	Standard
1000	53	Male	No	Tablet	3	Complete	Paypal	741.09	247.03	3	Overnight
1002	41	Male	No	Laptop	3	Complete	Credit Car	1855.84	463.96	4	Express
1002	41	Male	Yes	Smartpho	2	Complete	Cash	3164.76	791.19	4	Overnight
1003	75	Male	Yes	Smartpho	5	Complete	Cash	41.5	20.75	2	Express
1004	41	Female	No	Smartpho	5	Complete	Credit Car	83	20.75	4	Standard
1005	25	Female	No	Smartwat	3	Complete	Paypal	7603.47	844.83	9	Overnight
1005	25	Female	No	Laptop	3	Complete	Debit Carc	4175.64	463.96	9	Overnight
1006	24	Male	No	Smartpho	2	Cancelled	Debit Carc	5538.33	791.19	7	Standard
1006	24	Male	Yes	Laptop	3	Complete	Cash	4175.64	463.96	9	Express
1006	24	Male	Yes	Tablet	3	Complete	Paypal	2470.3	247.03	10	Overnight
1007	35	Male	No	Smartpho	2	Cancelled	Credit Car	7120.71	791.19	9	Overnight
1008	66	Female	No	Smartwat	3	Complete	Cash	3379.32	844.83	4	Express
1011	72	Male	No	Smartpho	2	Complete	Credit Car	7911.9	791.19	10	Standard
1013	42	Female	No	Smartpho	2	Cancelled	Paypal	5538.33	791.19	7	Overnight
1014	73	Female	No	Smartwat	3	Cancelled	Credit Car	5068.98	844.83	6	Standard
1014	73	Female	Yes	Smartpho	2	Complete	Cash	4747.14	791.19	6	Standard
1015	22	Male	No	Smartpho	5	Complete	Paypal	83	20.75	4	Standard
1016	61	Male	Yes	Smartpho	5	Complete	Paypal	103.75	20.75	5	Overnight
1018	34	Male	No	Smartpho	5	Complete	Paypal	124.5	20.75	6	Express
1019	45	Male	No	Smartpho	5	Cancelled	Cash	83	20.75	4	Express
1019	45	Male	No	Tablet	3	Complete	Cash	988.12	247.03	4	Express
1019	45	Male	Yes	Smartwat	3	Cancelled	Paypal	844.83	844.83	1	Express
1020	57	Female	No	Smartwat	3	Cancelled	Debit Carc	3379.32	844.83	4	Express

Q#4

Customer	Age	Product_T	Rating	Payment_	Total_Pric	Unit_Price	Quantity	Shipping_	Gender_F	Gender_M	Gender_n	Loyalty_M	Loyalty_M	Order_Status
0	0.56452	0	2	0	0.48502	2	0.52754	0	0	1	0	1	0	0
0	0.56452	1	3	1	0.06332	0	-0.86581	1	0	1	0	1	0	1
0.00011	0.37097	2	3	0	0.16131	1	-0.51747	2	0	1	0	1	0	1
0.00011	0.37097	0	2	2	0.27637	2	-0.51747	1	0	1	0	0	1	1
0.00016	0.91935	0	5	2	0.00182	0	-1.21415	2	0	1	0	0	1	1
0.00021	0.37097	0	5	0	0.00547	0	-0.51747	0	1	0	0	1	0	1
0.00026	0.1129	4	3	1	0.66655	2	1.22421	1	1	0	0	1	0	1
0.00026	0.1129	2	3	3	0.36523	1	1.22421	1	1	0	0	1	0	1
0.00032	0.09677	0	2	3	0.48502	2	0.52754	0	0	1	0	1	0	0
0.00032	0.09677	2	3	2	0.36523	1	1.22421	2	0	1	0	0	1	1
0.00032	0.09677	1	3	1	0.21533	0	1.57255	1	0	1	0	0	1	1
0.00037	0.27419	0	2	0	0.62411	2	1.22421	1	0	1	0	1	0	0
0.00042	0.77419	4	3	2	0.29523	2	-0.51747	2	1	0	0	1	0	1
0.00058	0.87097	0	2	0	0.69366	2	1.57255	0	0	1	0	1	0	1
0.00068	0.3871	0	2	1	0.48502	2	0.52754	1	1	0	0	1	0	0
0.00074	0.8871	4	3	0	0.44376	2	0.1792	0	1	0	0	1	0	0
0.00074	0.8871	0	2	2	0.41547	2	0.1792	0	1	0	0	0	1	1
0.00079	0.06452	0	5	1	0.00547	0	-0.51747	0	0	1	0	1	0	1
0.00084	0.69355	0	5	1	0.0073	0	-0.16914	1	0	1	0	0	1	1
0.00095	0.25806	0	5	1	0.00912	0	0.1792	2	0	1	0	1	0	1
0.001	0.43548	0	5	2	0.00547	0	-0.51747	2	0	1	0	1	0	0
0.001	0.43548	1	3	2	0.08504	0	-0.51747	2	0	1	0	1	0	1
0.001	0.43548	4	3	1	0.07244	2	-1.56248	2	0	1	0	0	1	0
0.00105	0.62903	4	3	3	0.29523	2	-0.51747	2	1	0	0	1	0	0

Q#5

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, StandardScaler, OneHotEncoder, OrdinalEncoder
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load the data
data = pd.read_csv('Electronics_data.csv')
print("No of data rows:", data.shape[0])

# Normalization for numerical columns
scalar = MinMaxScaler()
data[['Customer_ID', 'Age', 'Total_Price']] = scalar.fit_transform(data[['Customer_ID', 'Age', 'Total_Price']])

# Standardization for 'Quantity'
standard_scalar = StandardScaler()
data[['Quantity']] = standard_scalar.fit_transform(data[['Quantity']])

# One-hot encoding for categorical variables
one_hot_encoder = OneHotEncoder(sparse_output=False).set_output(transform="pandas")
one_hot_encoded = one_hot_encoder.fit_transform(data[['Gender', 'Loyalty_Member']])
data = pd.concat([data.drop(columns=['Gender', 'Loyalty_Member']), one_hot_encoded], axis=1)


order_status = data['Order_Status'].copy()
data = data.drop(columns=['Order_Status'])

# Ordinal encoding for Product_Type, Payment_Method, Shipping_Type, Order_Status
product_type_categories = [['Smartphone', 'Tablet', 'Laptop', 'Headphones', 'Smartwatch']]
ordinal_encoder_product = OrdinalEncoder(categories=product_type_categories)
data['Product_Type'] = ordinal_encoder_product.fit_transform(data[['Product_Type']])

payment_method_categories = [['Credit Card','Paypal', 'Cash', 'Debit Card', 'Bank Transfer']]
ordinal_encoder_payment = OrdinalEncoder(categories=payment_method_categories, handle_unknown='use_encoded_value', unknown_value=-1)
data['Payment_Method'] = ordinal_encoder_payment.fit_transform(data[['Payment_Method']])
```

```
shipping_type_category = [['Standard', 'Overnight', 'Express','Expedited', 'Same Day']]
ordinal_encoder_shipping = OrdinalEncoder(categories=shipping_type_category)
data['Shipping_Type'] = ordinal_encoder_shipping.fit_transform(data[['Shipping_Type']])

order_status_categories = [['Cancelled', 'Completed']]
ordinal_encoder_status = OrdinalEncoder(categories=order_status_categories)
encoded_order_status = ordinal_encoder_status.fit_transform(order_status.values.reshape(-1
data['Order_Status'] = encoded_order_status

# Binning for Unit_Price
data['Unit_Price'] = pd.qcut(data['Unit_Price'], q=3, labels=False)

data.to_csv('processed_data.csv', index=False)

x = data
y = order_status

# Split the data
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

print("No. of training rows:", x_train.shape[0])
print("No. of testing rows:", x_test.shape[0])

# Initialize and train the MLP classifier
mlp_classifier = MLPClassifier(hidden_layer_sizes=(100,), max_iter=1000, random_state=42)
mlp_classifier.fit(x_train, y_train)
# Make predictions
y_pred = mlp_classifier.predict(x_test)
# Print model details and evaluation metrics
print("Hidden_layer_size:", mlp_classifier.hidden_layer_sizes)
print("No. of layers:", mlp_classifier.n_layers_)
print("No. of iterations:", mlp_classifier.n_iter_)
print("Classes:", mlp_classifier.classes_)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
PS C:\Users\HP\OneDrive\Desktop\Personal Data> & "C:/Program Files/Python312/python.exe" "c:/Users/HP/OneDrive/Desktop/Personal Data/iris.py"
No of data rows: 20000
No. of training rows: 16000
No. of testing rows: 4000
Hidden_layer_size: (100,)
No. of layers: 3
No. of iterations: 27
Classes: ['Cancelled' 'Completed']
Accuracy: 1.0
              precision    recall  f1-score   support

   Cancelled       1.00      1.00      1.00      1310
   Completed       1.00      1.00      1.00      2690

    accuracy                           1.00      4000
   macro avg       1.00      1.00      1.00      4000
weighted avg       1.00      1.00      1.00      4000
   Cancelled       1.00      1.00      1.00      1310
   Completed       1.00      1.00      1.00      2690
```