TEXTILE INDUSTRY PRODUCTION TEST AND TRAIN MODEL USING MACHINE LEARNING

TEAM MEMBERS:

SOWMYA R

SHALINI R K

SAVITHA J V

CODE:

```
# Required imports
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error,
accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import StandardScaler

# Load data - remember to replace the path with your own file path
data = pd.read_csv('C:\\Users\\sowmy\\Downloads\\FDS_PROJECT.csv')

# Data preprocessing
# Creating a binary target: 1 for gain (Profit > 0), 0 for loss (Loss > 0)
data['Target'] = (data['Profit'] > 0).astype(int)  # Binary target
```

```
# Selecting features (excluding 'Product Name', 'Year', 'Profit', 'Loss')
features = data.drop(columns=['Year', 'Product Name', 'Profit', 'Loss', 'Target'])
target = data['Target']

# Splitting data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

# Scaling the features for better model performance
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Training the model
model = LogisticRegression()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Calculating metrics
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
```

```
report = classification_report(y_test, y_pred)

# Print metrics
print("Mean Square Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("Accuracy:", accuracy)
print("Classification Report:\n", report)

# Visualization of Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False,
            xticklabels=['Loss', 'Gain'], yticklabels=['Loss', 'Gain'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()

# Bar plot for classification report metrics
report_dict = classification_report(y_test, y_pred, output_dict=True)
metrics = ['precision', 'recall', 'f1-score']
metrics_values = [report_dict['1'][metric] for metric in metrics]

plt.figure(figsize=(10, 6))
```

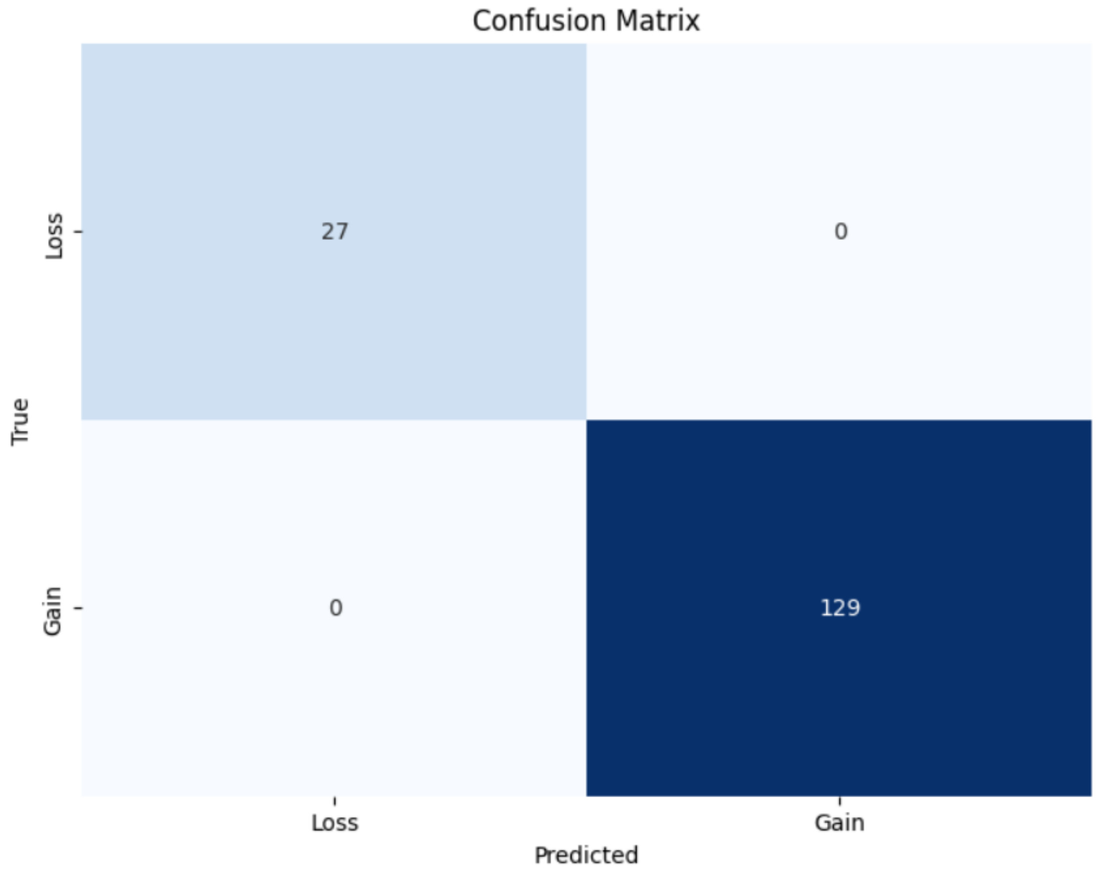
```
sns.barplot(x=metrics, y=metrics_values)
plt.ylim(0, 1)
plt.title('Classification Metrics for Gain')
plt.ylabel('Score')
plt.xlabel('Metrics')
plt.show()
```

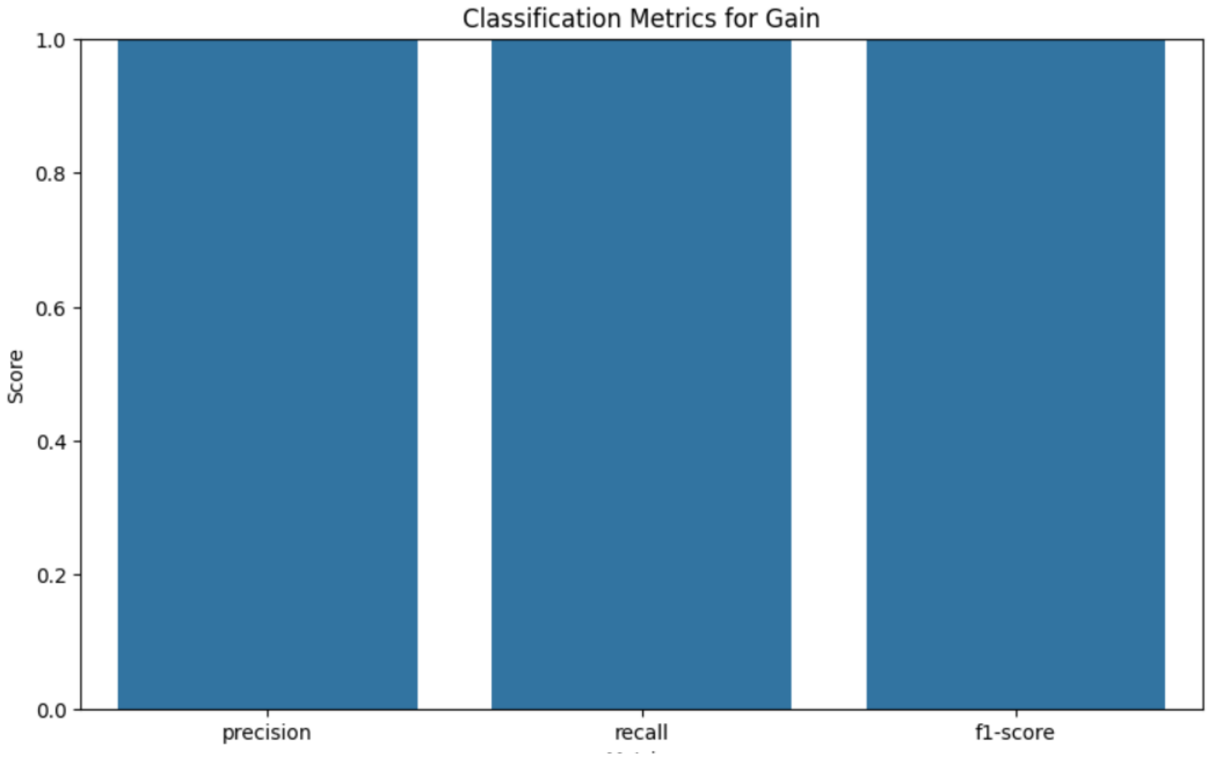
OUTPUT:

```
Mean Square Error (MSE): 0.0
Mean Absolute Error (MAE): 0.0
Accuracy: 1.0
Classification Report:
            precision    recall  f1-score   support

           0       1.00      1.00      1.00        27
           1       1.00      1.00      1.00       129

    accuracy                           1.00       156
   macro avg       1.00      1.00      1.00       156
weighted avg       1.00      1.00      1.00       156
```


Confusion Matrix
True
Loss
Gain
27
0
0
129
Loss
Gain
Predicted


Classification Metrics for Gain
Score
1.0
0.8
0.6
0.4
0.2
0.0
precision
recall
f1-score

DATASET:

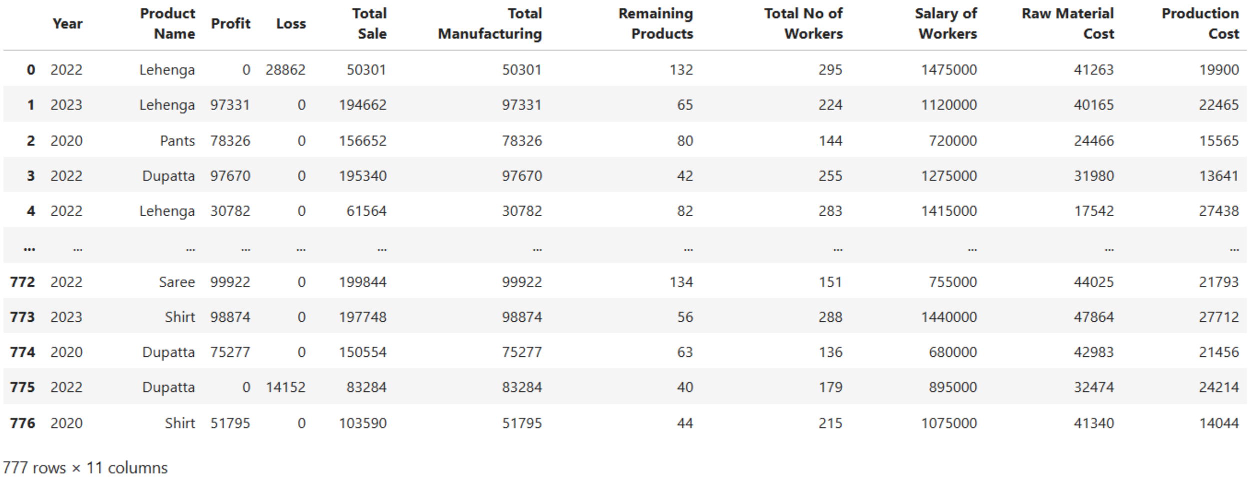
```
import pandas as pd

import numpy as np

df=pd.read_csv("FDS_PROJECT.csv")

df
```

OUTPUT:



	Year	Product Name	Profit	Loss	Total Sale	Total Manufacturing	Remaining Products	Total No of Workers	Salary of Workers	Raw Material Cost	Production Cost
0	2022	Lehenga	0	28862	50301	50301	132	295	1475000	41263	19900
1	2023	Lehenga	97331	0	194662	97331	65	224	1120000	40165	22465
2	2020	Pants	78326	0	156652	78326	80	144	720000	24466	15565
3	2022	Dupatta	97670	0	195340	97670	42	255	1275000	31980	13641
4	2022	Lehenga	30782	0	61564	30782	82	283	1415000	17542	27438
...	...	...	...	...	...	...	...	...	...	...	...
772	2022	Saree	99922	0	199844	99922	134	151	755000	44025	21793
773	2023	Shirt	98874	0	197748	98874	56	288	1440000	47864	27712
774	2020	Dupatta	75277	0	150554	75277	63	136	680000	42983	21456
775	2022	Dupatta	0	14152	83284	83284	40	179	895000	32474	24214
776	2020	Shirt	51795	0	103590	51795	44	215	1075000	41340	14044

777 rows × 11 columns