

```

1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.preprocessing import LabelEncoder, StandardScaler
4 from sklearn.svm import SVC
5 from sklearn.metrics import accuracy_score, classification_report

```

```

1 df = pd.read_csv('/content/manufacturing_6G_dataset.csv')
2 df.head()

```



	Operation_Mode	Temperature_C	Vibration_Hz	Power_Consumption_kW	Network_Latency_ms	Packet_Loss_%
0	Idle	74.137590	3.500595	8.612162	10.650542	0.207764
1	Active	84.264558	3.355928	2.268559	29.111810	2.228464
2	Active	44.280102	2.079766	6.144105	18.357292	1.639416
3	Active	40.568502	0.298238	4.067825	29.153629	1.161021
4	Idle	75.063817	0.345810	6.225737	34.029191	4.796520

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

```

1 # Separate features (X) and target (y)
2 X = df.drop('Efficiency_Status', axis=1)
3 y = df['Efficiency_Status']

```

```

1 # Encode the 'Operation_Mode' column
2 le = LabelEncoder()
3 X['Operation_Mode'] = le.fit_transform(X['Operation_Mode'])

```

```

1 # Split data into training and testing sets
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```

1 # Scale numerical features
2 scaler = StandardScaler()
3 X_train = scaler.fit_transform(X_train)
4 X_test = scaler.transform(X_test)

```

```

1 # Train an SVM classifier
2 svm_classifier = SVC(kernel='linear') # You can experiment with different kernels
3 svm_classifier.fit(X_train, y_train)

```



SVC

SVC(kernel='linear')

```

1 # Make predictions
2 y_pred = svm_classifier.predict(X_test)

```

```

1 # Evaluate the model
2 accuracy = accuracy_score(y_test, y_pred)
3 print(f"Accuracy: {accuracy}")
4 print(classification_report(y_test, y_pred))

```



Accuracy: 0.91685

	precision	recall	f1-score	support
High	0.90	0.84	0.87	625
Low	0.94	0.96	0.95	15464
Medium	0.81	0.75	0.78	3911
accuracy			0.92	20000
macro avg	0.88	0.85	0.87	20000
weighted avg	0.91	0.92	0.92	20000