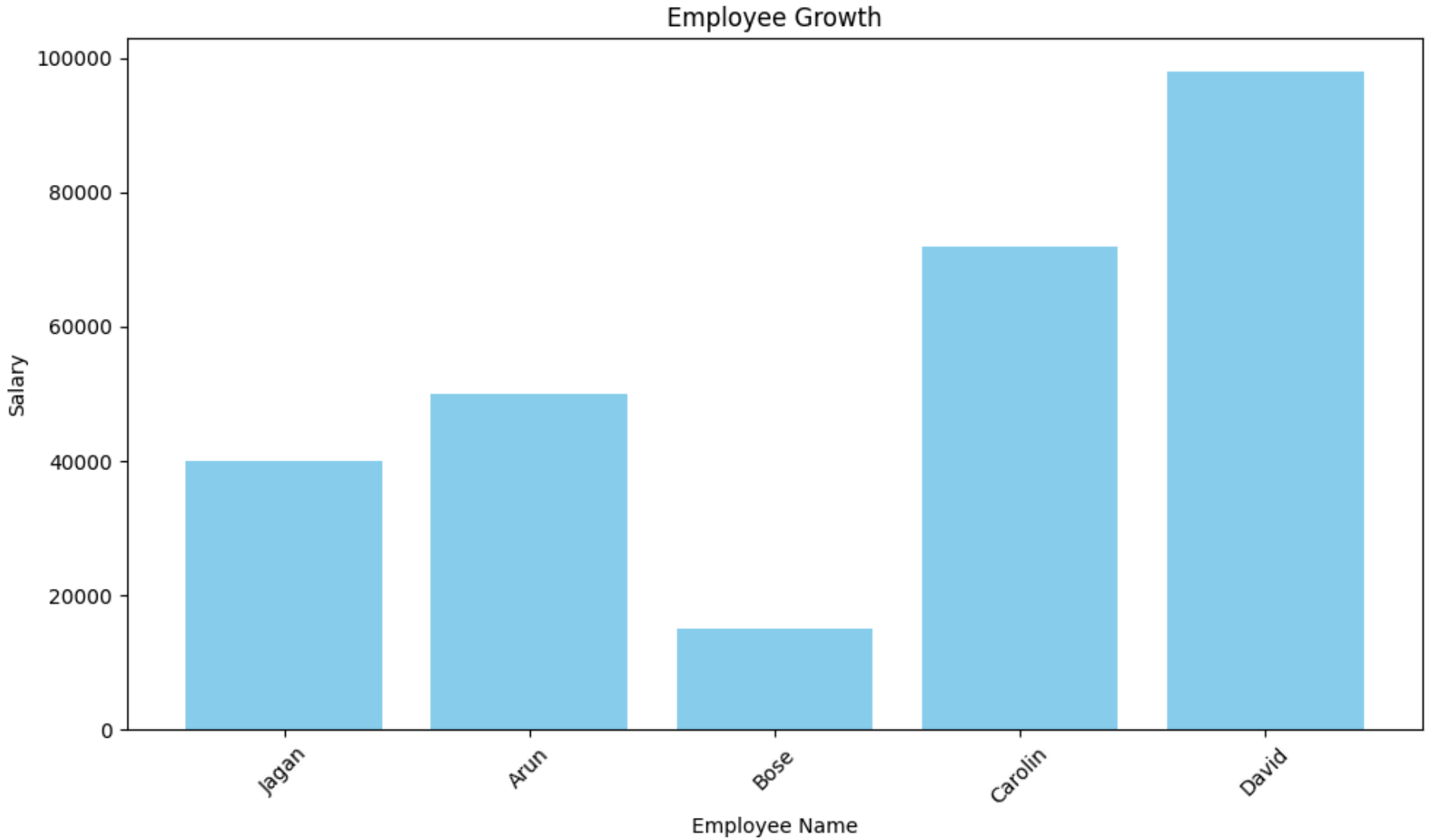
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Creating the initial dataset
data = {
    'Name': ['Jagan', 'Arun', 'Bose', 'Carolin', 'David'],
    'Employee ID': [101, 102, 103, 104, 105],
    'Age': [32, 28, 35, np.nan, 40],  # Introducing a NaN value for demonstration
    'Salary': [40000, 50000, 15000, 72000, 98000],
    'Department': ['HR', 'Finance', 'IT', 'HR', 'IT']
}

# Creating DataFrame
df = pd.DataFrame(data)

# Data Cleaning Process
# 1. Handling Missing Values
df['Age'].fillna(df['Age'].median(), inplace=True)  # Filling NaN with median age

# 2. Removing Outliers (Optional)
# We can remove outliers based on Salary using some statistical methods like Z-score or IQR.
# For simplicity, let's assume there are no outliers in this example.

# Visualizing the growth of each employee using a bar chart
plt.figure(figsize=(10, 6))
plt.bar(df['Name'], df['Salary'], color='skyblue')
plt.xlabel('Employee Name')
plt.ylabel('Salary')
plt.title('Employee Growth')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



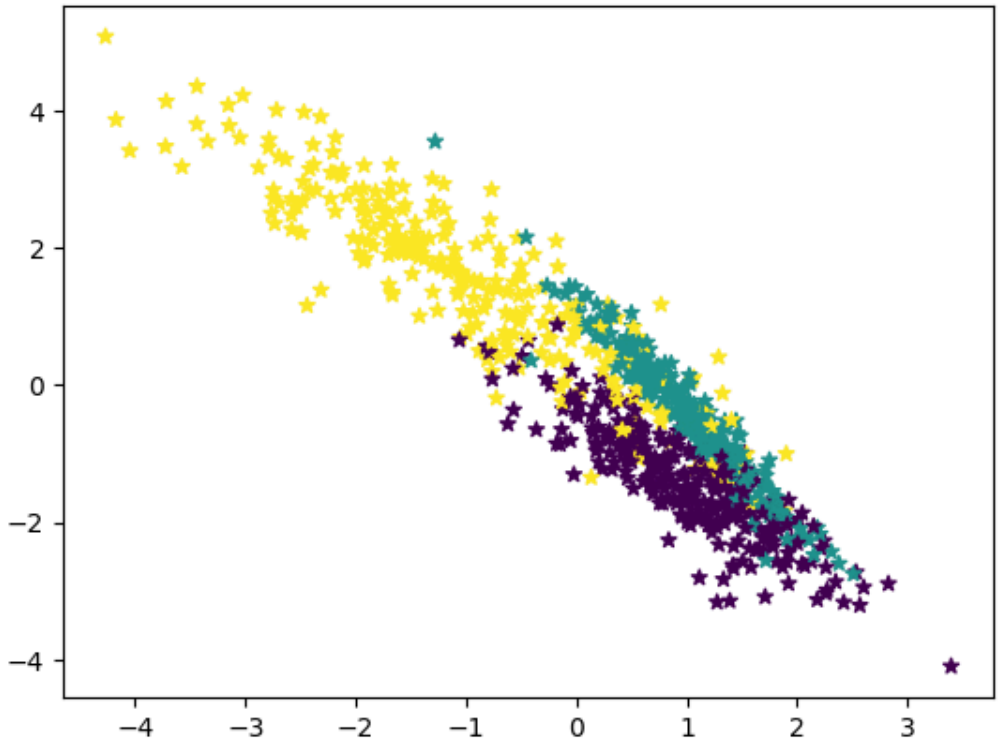

```
import pandas as pd
from sklearn.datasets import make_classification
X, y = make_classification(
n_features=6,
n_classes=3,
n_samples=800,
n_informative=2,
```

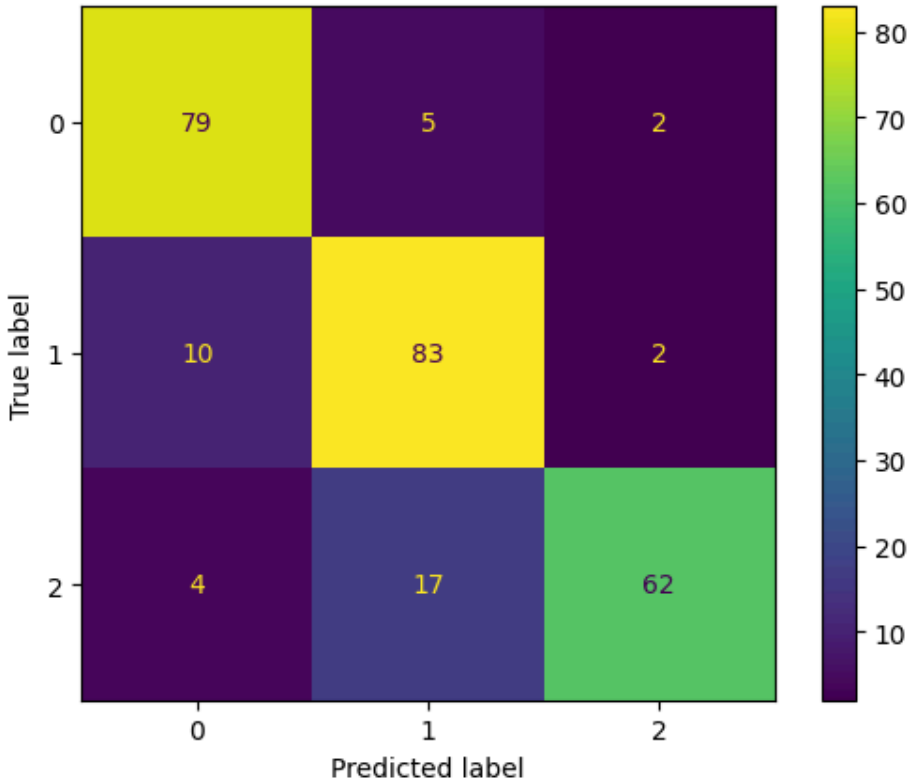
```
random_state=1,
n_clusters_per_class=1,
)
import matplotlib.pyplot as plt
plt.scatter(X[:, 0], X[:, 1], c=y, marker="*");
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
 X, y, test_size=0.33, random_state=125)
from sklearn.naive_bayes import GaussianNB
# Build a Gaussian Classifier
model = GaussianNB()
# Model training
model.fit(X_train, y_train)
# Predict Output
predicted = model.predict([X_test[6]])
print("Actual Value:", y_test[6])
print("Predicted Value:", predicted[0])
from sklearn.metrics import (
accuracy_score,
confusion_matrix,
ConfusionMatrixDisplay,
f1_score,
)
y_pred = model.predict(X_test)
accuray = accuracy_score(y_pred, y_test)
f1 = f1_score(y_pred, y_test, average="weighted")
print("Accuracy:", accuray)
print("F1 Score:", f1)
labels = [0,1,2]
cm = confusion_matrix(y_test, y_pred, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot();
import pandas as pd
df = pd.read_csv('/content/archive (1).zip')
df.head()
```

```
Actual Value: 0
Predicted Value: 0
Accuracy: 0.8484848484848485
F1 Score: 0.8491119695890328
```

	Marital status	Application mode	Application order	Course	Daytime/evening attendance	Previous qualification	Nacionality	Mother's qualification	Father's qualification	Mother's occupation	...
0	1	8	5	2	1	1	1	13	10	6	...
1	1	6	1	11	1	1	1	1	3	4	...
2	1	1	5	5	1	1	1	22	27	10	...
3	1	8	2	15	1	1	1	23	27	6	...
4	2	12	1	3	0	1	1	22	28	10	...

5 rows × 35 columns




79
5
2
10
83
2
4
17
62
0
1
2
True label
0
1
2
Predicted label
80
70
60
50
40
30
20
10