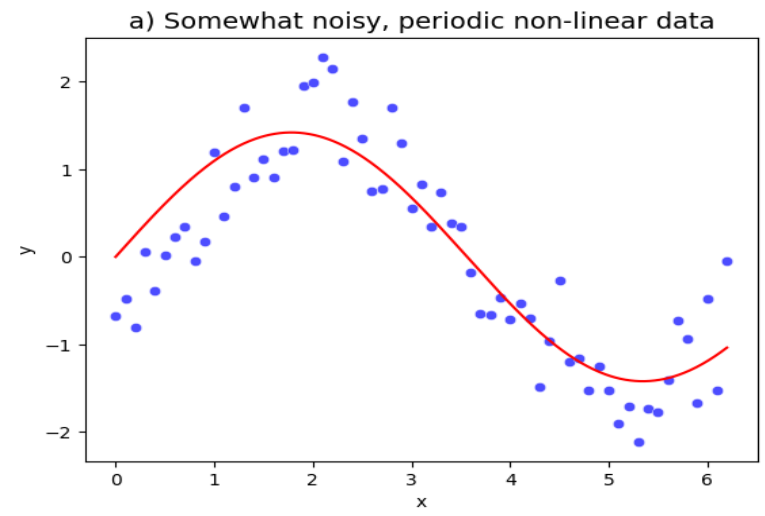

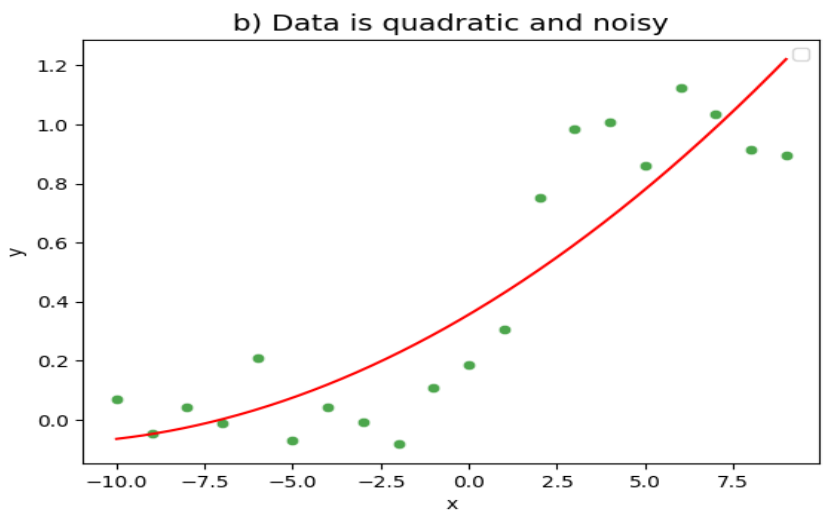

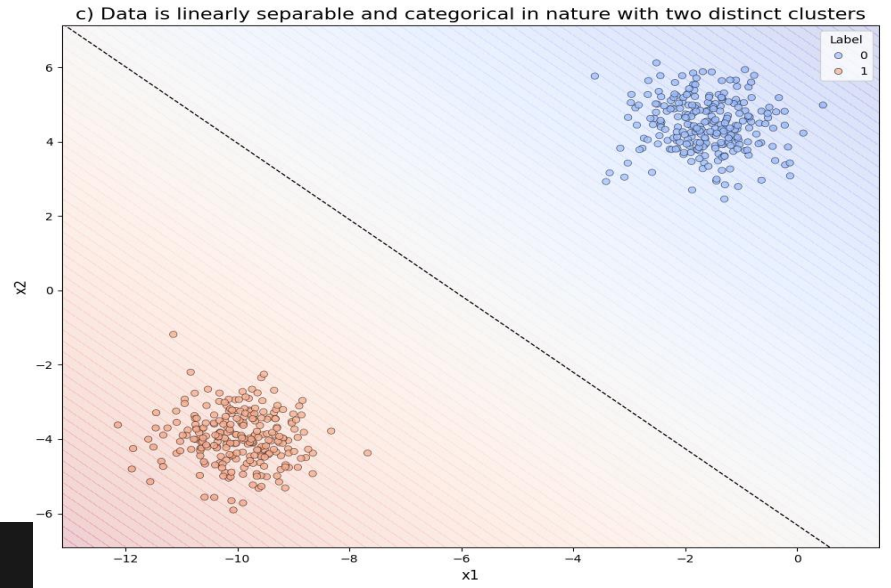


```python
def sine_function(x, A, B):
    return A * np.sin(B * x)

x = regression_1['x1']
y = regression_1['x2']
params, _ = curve_fit(sine_function, x, y, p0=[1, 1])
A, B = params

x_fit = np.linspace(x.min(), x.max(), 500)
y_fit = sine_function(x_fit, A, B)

sns.scatterplot(data=regression_1, x=x, y=y, color="blue", alpha=0.7, )
sns.lineplot(x=x_fit, y=y_fit, color="red", )
plt.title("a) Somewhat noisy, periodic non-linear data", fontsize=14)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

```python
from sklearn.pipeline import make_pipeline

poly = PolynomialFeatures(degree=2)
model = make_pipeline(poly, Ridge(alpha=1.0))

x = regression_2[['x1']].values
y = regression_2['x2'].values
model.fit(x, y)

x_fit = np.linspace(x.min(), x.max(), 500).reshape(-1, 1)
y_fit = model.predict(x_fit)

sns.scatterplot(x=regression_2["x1"], y=regression_2["x2"], color="green", alpha=0.7)
sns.lineplot(x=x_fit.ravel(), y=y_fit, color="red")
plt.title("b) Data is quadratic and noisy", fontsize=14)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

```python
svc = SVC(kernel="linear")
x = classification[['x1', 'x2']]
y = classification['label']
svc.fit(x, y)
xx, yy = np.meshgrid(
    np.linspace(x['x1'].min() - 1, x['x1'].max() + 1, 100),
    np.linspace(x['x2'].min() - 1, x['x2'].max() + 1, 100)
)
grid = np.c_[xx.ravel(), yy.ravel()]
decision_function = svc.decision_function(grid).reshape(xx.shape)
plt.figure(figsize=(10, 8))
sns.scatterplot(data=classification, x="x1", y="x2", hue="label", palette="coolwarm", alpha=0.8, edgecolor="k")
levels = np.linspace(decision_function.min(), decision_function.max())
plt.contour(xx, yy, decision_function, levels=[0], colors='black', linestyles='--', linewidths=1)
plt.contourf(xx, yy, decision_function, alpha=0.2, cmap="coolwarm", levels= levels)
plt.title("c) Data is linearly separable and categorical in nature with two distinct clusters", fontsize=16)
plt.xlabel("x1", fontsize=12)
plt.ylabel("x2", fontsize=12)
plt.legend(title="Label")
plt.tight_layout()
plt.show()
```

The scatter plot showed a **periodic trend**, so I used a **sine function** with curve fitting to model the data. The optimal parameters are **Amplitude**: 1.42, **Angular Frequency**: 0.88. The sine curve effectively captures the peaks and troughs of the data, minimizing residual error and aligning well with its periodic behavior.

The data exhibited an **upward curvature**, suggesting a **quadratic relationship**. I applied **2nd degree polynomial regression** with **Ridge regularization** to handle noise. The key coefficients are **Intercept**: 0.357, **Linear**: 0.07, **Quadratic**: 0.0028. This model balances noise smoothing while capturing the non-linear trend accurately.

The data was **linearly separable**. So I used an **SVM with a linear kernel** to determine the decision boundary. The model maximizes the margin (7.99) between the two classes, with **3 support vectors** contributing weights [−0.031,0.017,0.013]. The resulting boundary cleanly separates the two clusters without misclassifications.

Rohit Potdukhe, M. N.: 22985091, IdM: vx49uxvm