

```

# Step 1: Simulate sensor-like drainage data
data = {
    'water_level': [20, 35, 60, 28, 45, 50, 90, 45, 70, 100],
    'flow_rate': [4, 5, 1, 3, 6, 4, 2, 4, 2, 1],
    'turbidity': [10, 25, 30, 12, 18, 35, 50, 40, 30, 50],
    'citizen_reports': [0, 0, 1, 0, 1, 0, 1, 1, 0, 1],
    'blockage_risk': [0, 0, 1, 0, 1, 0, 1, 1, 0, 1] # 0 = Low Risk, 1 = High Risk
}

df = pd.DataFrame(data)

# Step 2: Train/test split
X = df[['water_level', 'flow_rate', 'turbidity', 'citizen_reports']]
y = df['blockage_risk']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

# Step 3: Model training
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Step 4: Prediction and evaluation
y_pred = model.predict(X_test)
print("Classification Report:")
print(classification_report(y_test, y_pred))

# Optional: Predict for a new drain condition
new_data = pd.DataFrame([
    'water_level': 75,
    'flow_rate': 1,
    'turbidity': 38,
    'citizen_reports': 2
])

risk = model.predict(new_data)
print(f"Predicted Blockage Risk (1 = High, 0 = Low): {risk[0]}")

```