

Lab Assignment 4 - Student Version

Title:

Implement the non-parametric Locally Weighted Regression algorithm to fit data points.

Objective:

To fit data points by assigning different weights to each point based on its proximity to the query point.

Dataset Generation:

Using a synthetic dataset with a sinusoidal pattern.

Python code:

```
import numpy as np
np.random.seed(0)
X = np.sort(5 * np.random.rand(80, 1), axis=0)
y = np.sin(X).ravel() + np.random.normal(0, 0.1, X.shape[0])
```

Tasks and Implementation:

1. Split dataset into features (X) and target variable (y).

2. Implement Locally Weighted Regression (LWR):

```
def kernel(x, xi, tau):
    return np.exp(-np.sum((x - xi)**2) / (2 * tau**2))

def predict(x_query, X, y, tau):
    m = X.shape[0]
    weights = np.array([kernel(x_query, X[i], tau) for i in range(m)])
    W = np.diag(weights)
    theta = np.linalg.pinv(X.T @ W @ X) @ X.T @ W @ y
    return x_query @ theta
```

3. Use multiple query points and bandwidths (tau) to generate fitted curves.

Lab Assignment 4 - Student Version

4. Plot results using matplotlib (not shown in this text version).

Interpretation:

LWR provides better local fitting due to weighted influence of nearby points. Smaller tau gives sharper fits but may overfit. Larger tau smooths out noise but may underfit.