WEEK1_CSRESEARCH

June 1, 2025

1 LINEAR REGRESSION FUNCTION

This function takes 4 parameters: - X: Input feature matrix (NumPy array) - Y: Target vector (NumPy array) - lr: Learning rate (float) - lambda_: L1 regularization coefficient (float)

It returns the weights, i.e. the model parameters of the linear regression model. We have only used numpy and pandas to code this function.

[ ]:
```
def linearRegression(X: np.array, Y: np.array, lr: float, lambda_: float):
  import numpy as np
  import pandas as pd

  m, n = X.shape
  weights = np.zeros(n)
  bias = 0.0
  epochs = 1000

  for epoch in range(epochs):
    y_pred = np.dot(X, weights) + bias
    error = y_pred - y_train
    weights_grad = (1/m) * np.dot(X.T, error) + (lambda_/m) * np.sign(weights)
    bias_grad = (1/m) * np.sum(error)
    weights -= lr * weights_grad
    bias -= lr * bias_grad

  return weights
```

2 NEWTON-RAPHSON METHOD

This function will compute the roots of a function, and consequently the mth root of a number by the Newton-Raphson method.

[ ]:
```
"""def newtonRaphson(x: float, m: float):
    root = x / m
    def find_root_recursive(current_root, tolerance=1e-7, max_iterations=100):
        if abs(current_root**m - x) < tolerance or max_iterations == 0:
            return current_root
        else:
```

```
            next_root = current_root - (current_root**m - x) / (m *␣
↪current_root**(m-1))
            return find_root_recursive(next_root, tolerance, max_iterations - 1)
    return find_root_recursive(root)"""
```

[ ]: 1.4142135623746899