week5a

September 6, 2024

[1]:

```
import numpy as np
def pl_regressor(X, y, degree):
    X_poly = np.ones((X.shape[0], degree + 1))
    for i in range(1, degree + 1):
        X_poly[:, i] = X**i
    X_transpose = X_poly.T
    theta = np.linalg.inv(X_transpose.dot(X_poly)).dot(X_transpose).dot(y)
    return theta

def get_input():
    X_input = input("Enter the X values as a comma-separated list: ")
    X_values = [float(x.strip()) for x in X_input.split(",")]
    y_input = input("Enter the corresponding y values as a comma-separated list:
 ↪ ")
    y_values = [float(y.strip()) for y in y_input.split(",")]
    if len(X_values) != len(y_values):
        raise ValueError("The number of X and y values must be the same.")
    X = np.array(X_values)
    y = np.array(y_values)
    return X, y

def main():
    X, y = get_input()
    degree = len(X) - 1
    theta = pl_regressor(X, y, degree)
    print(f"\nThe computed polynomial regression coefficients are: {theta}")

if __name__ == "__main__":
    main()
```

```
The computed polynomial regression coefficients are: [ 2.43010056e+03
-1.77365106e+01  5.24073763e-02 -7.59337136e-05
  4.58414150e-08  1.50509491e-11 -4.30161852e-14  2.86639060e-17
 -8.72626555e-21  1.04264000e-24]
```

- The pl_regressor function is implemented to perform polynomial regression.
- It begins by transforming the input array X into a polynomial feature matrix, where each

column corresponds to increasing powers - of X up to the specified degree.

- The polynomial feature matrix is built by raising X to powers ranging from 1 to the specified degree, with an additional column of ones for the intercept term.