linear

July 2, 2024

1 Least Squares Linear Regression

1.0.1 Imports

[1]:
```python
import numpy as np
import matplotlib.pyplot as plt
```

1.0.2 Implementation

[68]:
```python
def linear_regression(data):
    # Split dataset into x matrix and y vector
    # Add ones column to x matrix to account for intercept
    x = np.column_stack((np.ones(len(data)), data[:, :-1]))
    y = data[:,-1:]

    # Compute coefficients using matrix arithmetic
    return np.linalg.inv(x.T @ x) @ x.T @ y
```

1.0.3 Generate Test Dataset

[69]:
```python
n = 100
slope = 5
intercept = 10
```

[70]:
```python
x_vals = np.random.uniform(0, 5, n)
y_vals = np.random.uniform(0, 5, n)

z_vals = slope * x_vals + intercept
```

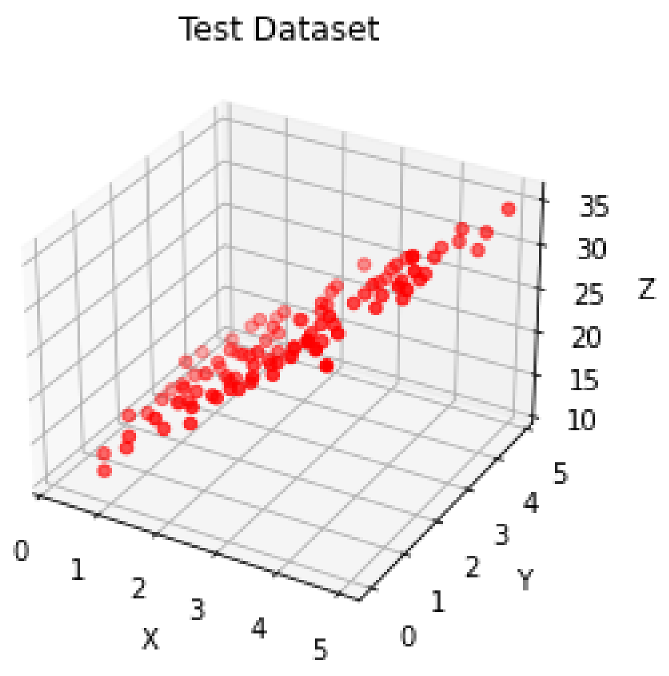
[71]:
```python
noise = np.random.normal(0, 1, n)
z_vals += noise
```

[72]:
```python
fig = plt.figure()
ax = fig.add_subplot(111, projection = "3d")

ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.set_title("Test Dataset")

ax.scatter(x_vals, y_vals, z_vals, c = "red", marker = "o")
```

[72]:
```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x1ff77d32fd0>
```



1.0.4 Visualization

[73]:
```
data = np.array(list(map(list, zip(x_vals, y_vals, z_vals))))
coefs = linear_regression(data)
```

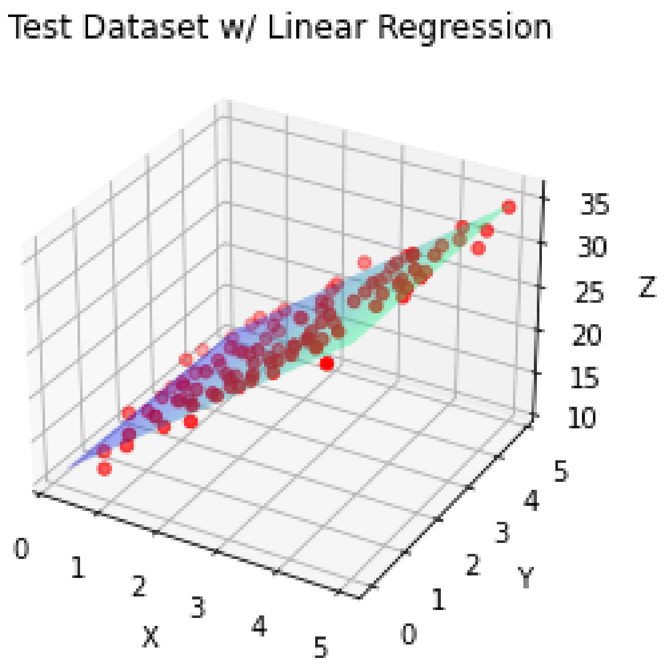
[74]:
```
x_grid, y_grid = np.meshgrid(np.linspace(min(data[:,0]), max(data[:,0]), 100),␣
 ↪np.linspace(min(data[:,1]), max(data[:,1]), 100))
z_grid = coefs[0] + coefs[1] * x_grid + coefs[2] * y_grid
```

[81]:
```
fig = plt.figure()
ax = fig.add_subplot(111, projection = "3d")

ax.set_xlabel("X")
ax.set_ylabel("Y")
ax.set_zlabel("Z")
ax.set_title("Test Dataset w/ Linear Regression")

ax.scatter(x_vals, y_vals, z_vals, c = "red", marker = "o")
ax.plot_surface(x_grid, y_grid, z_grid, alpha = 0.4, cmap = "winter")
```

```
[81]: <mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x1ff79eeef40>
```


Test Dataset w/ Linear Regression
X
Y
Z