```
X1, X2 = getX1X2(X, model.degree) x1_range =
np.linspace(X1.min(), X1.max(), n_) x2_range =
np.linspace(X2.min(), X2.max(), n_) x1_range, x2_range =
np.meshgrid(x1_range, x2_range) contours =
plt.contour(X1_plot, X2_plot, Y, 20, colors=‘black’)
plt.clabel(contours, inline=True, fontsize=8) plt.imshow(Y,
extent=[0, 3, 0, 3], origin=‘lower’, cmap=‘viridis’)

plt.colorbar();



ZZ = predict(X, theta)

fig = plt.figure()

ax = fig.gca(projection=‘3d’)

ax.scatter3D(x1, x2, x3, c=x3, cmap=‘Greens’); plt.show()

pdb.set_trace()



X_plane = []

for i in range(n_):

for j in range(n_):

X_plane.append([1, x1_range[i], x2_range[j]]) X_plane =
np.array(X_plane)

Yhat = X_plane.dot(theta)
```

```
Yhat_surface = []

for i in range(n_):

Yhat_surface.append(Yhat[ n_*i : n_*i+n_]) Yhat_surface = np.array(Yhat_surface) Yhat = Yhat_surface

pdb.set_trace()

XX2, XX1 = np.meshgrid(x1_range, x2_range) # intentional ordering: X2, then X1

fig = plt.figure()

fig, ax = plt.subplots(subplot_kw={“projection”: “3d”}) ax = fig.add_subplot(111, projection=‘3d’) ax.scatter(X1,X2, Y) # supplied data ax.plot_surface(XX1, XX2, Yhat, color=‘y’, alpha=0.1)
```