Nghia Lam

1001699317

1.) degree = 1, Lambda = 0

w0=40.2937

w1=-85.3182

w2=40.5272

w3=2.8325

w4=2934.2841

w5=-14575.7107

w6=2403.3571

w7=5.3809

w8=-1217.1594

w9=238.0055

w10=-8.3754

w11=-641.5481

w12=6.1993

w13=-395.2040

ID= 102, output= 25.3395, target value = 25.0000, squared error = 0.1153

----

degree = 1, lambda = 1

w0=23.4505

w1=-4.9610

w2=20.0482

w3=-4.3727

w4=0.1951

w5=-0.0402

w6=2.1384

w7=-8.8236

w8=-0.3145

w9=1.3135

w10=-11.0302

w11=-2.7073

w12=12.2978

w13=-16.1952

ID= 102, output=   19.8046, target value =   25.0000, squared error = 26.9919

----

degree = 2, lambda = 0

w0=166.3681

w1=-298.2493

w2=1754.4640

w3=-43.9698

w4=412.4657

w5=-58.9031

w6=2286.8364

w7=3101.1087

w8=4.3616

w9=-17152.5014

w10=389.2442

w11=-15204.6413

w12=970080.3976

w13=-14.4500

w14=100.9698

w15=-1787.9444

w16=65442.1147

w17=387.9337

w18=-4914.2598

w19=-23.3693

w20=15.3658

w21=-3571.2823

w22=62091.1718

w23=17.6688

w24=-22.6487

w25=-1020.4161

w26=12937.5971

ID= 102, output=   25.0664, target value =   25.0000, squared error = 0.0044

----

degree = 2, lambda = 1

w0=22.4499

w1=-4.7353

w2=-0.3711

w3=19.7559

w4=2.2267

w5=-4.2212

w6=-0.1121

w7=0.1956

w8=0.0003

w9=-0.0382

w10=-0.0001

w11=2.1226

w12=0.0387

w13=-8.4829

w14=-1.7052

w15=-0.3628

w16=-0.0073

w17=1.6790

w18=0.0491

w19=-6.8179

w20=-3.3553

w21=-2.6919

w22=-0.1457

w23=9.2713

w24=5.1126

w25=-16.0614

w26=-0.5957

ID= 102, output= 19.6992, target value = 25.0000, squared error = 28.0982

2.) $ED(w) = (1/2) * [(9.6 - w^T * 5.3)^2 + (4.2 - w^T * 7.1)^2 + (2.2 - w^T * 6.4)^2] + (\lambda/2) * w^Tw$

As λ becomes extremely large the regularization term for λ becomes very big, therefore, to minimize the model, w would need to be as small as possible. Thus, in this case as λ approaches infinity w approaches 0.

3.) $ED(w) = (1/2)\ \Sigma(t_n - w^T \text{phi}(x_n))^2$

$f(x) = 3.1x + 4.2$

$f(x) = 2.4x - 1.5$

You can replace f(x) for w^T phi(x_n) as f(x) is a liner regression model for this problem.

F1(X) = .5* (T1 - (3.1(X1) + 4.2))^2 + (T2 - (3.1(X2) + 4.2))^2 + (T3 - (3.1(X3) + 4.2))^2

= .5 *(9.6 - (3.1(5.3) + 4.2))^2 + (4.2 - (3.1(7.1) + 4.2))^2 + (2.2 - (3.1(6.4) + 4.2))^2

= .5 * 1083.086

= 541.5433

F2(X) = .5 * (T1 - (2.4(X1) - 1.5))^2 + (T2 - (2.4(X2) - 1.5))^2 + (T3 - (2.4(X3) - 1.5))^2

= .5 * (9.6 - (2.4(5.3) - 1.5))^2 + (4.2 - (2.4(7.1) - 1.5))^2 + (2.2 - (2.4(6.4) - 1.5))^2

= .5 * 267.175

= 133.5878

F2(x) is better than F1(x) because the sum of square error of F2 is lower than F1

4.) Even though the algorithm is very good I would not use Bob's algorithm as Bob's algorithm might be very complex and the computationally expensive. In addition automating λ can be bad as it would overfit based on the assumption that it is based on that dataset so that if we were to implement new data it might change drastically. Also there are so many algorithms that are created every year that if we do decide to change everything at once it could be bad. If it were me I would do more testing.