(1) the averages of the RMSE values obtained during the 6-fold CV for each case;

Ans: The averages of the RMSE values obtained during the 6-fold CV foreach degree from 0 to 12 (inclusive) is as follows:

[1.0155605589710912,
1.083556161737652,
0.7754293123513891,
0.7830016853813812,
0.49348184425645386,
0.5701350133901167,
0.14235262539341095,
0.18750847543431423,
0.14631007436529808,
0.23647291096697745,
0.16488080003157965,
0.6257835760348095,
0.6869783309072769]

(2) the optimal degree d*and regularization parameter λ* obtained via the 6-fold CV;

Ans: The optimal degree d* = 6 (with λ = 0)

The optimal lambda λ* = exp(-3) where exp(z) = e^z (with degree=12)

(3) the coefficient-weights of the d* -degree polynomial λ* -regularized 12-degree learned on all the training data;

Ans: The coefficient-weights of the d* - degree polynomial:

array([[0. , 0.02770501, 0.26189729, 0.01058892, -0.22895198, 0.00065375, 0.04178481]])

```
In [19]: model_optD.coef_

Out[19]: array([[ 0.        ,  0.02770501,  0.26189729,  0.01058892, -0.22895198,
          0.00065375,  0.04178481]])
```

The coefficient-weights of the λ * - degree polynomial:

array([[0.00000000e+00, 3.10354670e-02, 1.89229715e-01,
 6.44582979e-03, -7.92323648e-02, -3.00615259e-04,
-6.47732564e-02, 1.09511371e-03, 1.82647791e-02,
-4.68465148e-05, 7.10722790e-03, -5.21040251e-06,
-2.03791326e-03]])

```
In [26]: model_optL.coef_

Out[26]: array([[ 0.00000000e+00,  3.10354670e-02,  1.89229715e-01,
          6.44582979e-03, -7.92323648e-02, -3.00615259e-04,
         -6.47732564e-02,  1.09511371e-03,  1.82647791e-02,
         -4.68465148e-05,  7.10722790e-03, -5.21040251e-06,
         -2.03791326e-03]])
```

(4) the training and test RMSE of that final, learned polynomials;

Ans: The training and testing RMSE for d*=6:

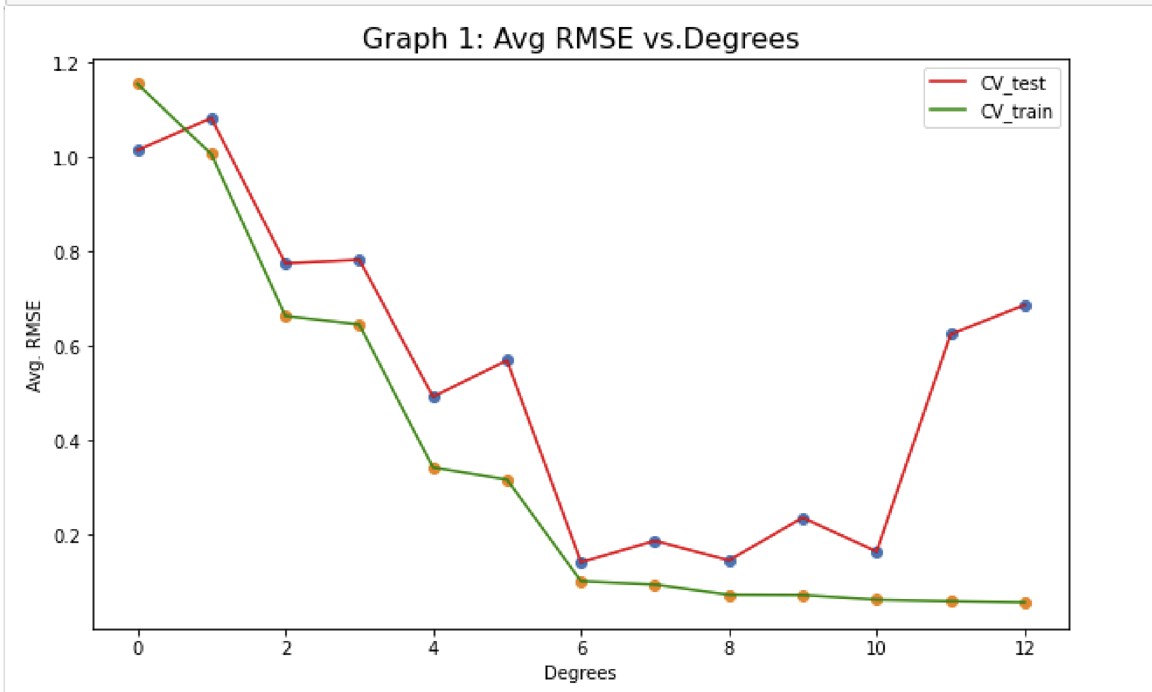
```
rmse_train = np.sqrt(mean_squared_error(y, y_pred_train_inv))

In [18]: rmse_train

Out[18]: 0.10540106673272033

In [21]: rmse_test

Out[21]: 0.11432570919476512
```

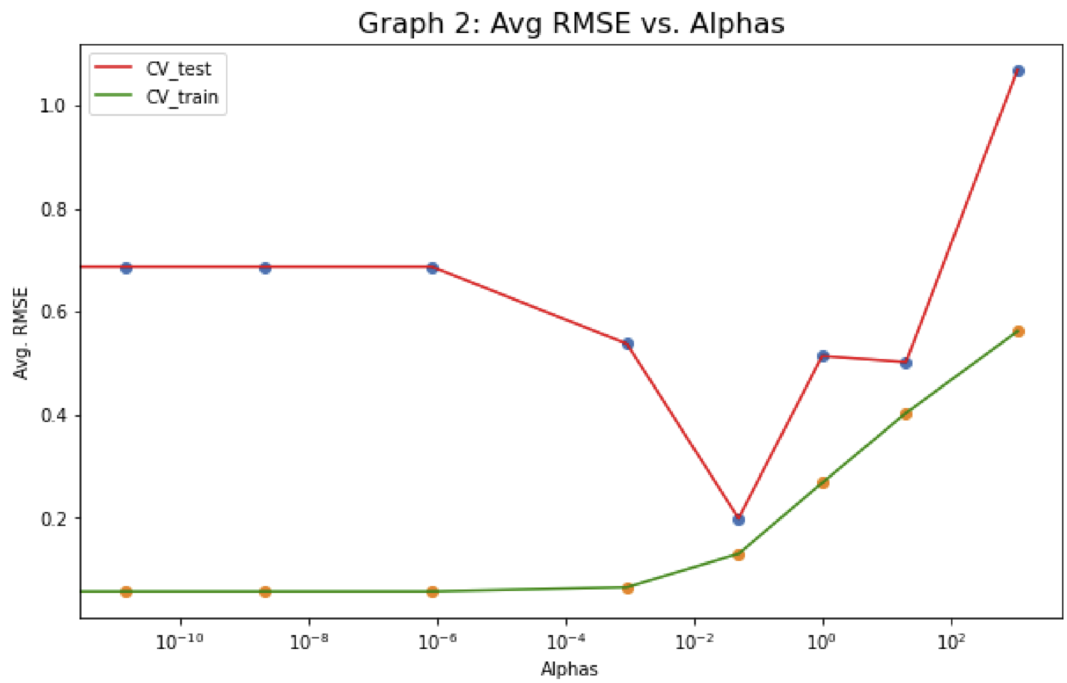
The training and testing RMSE for λ * -regularized 12-degree:

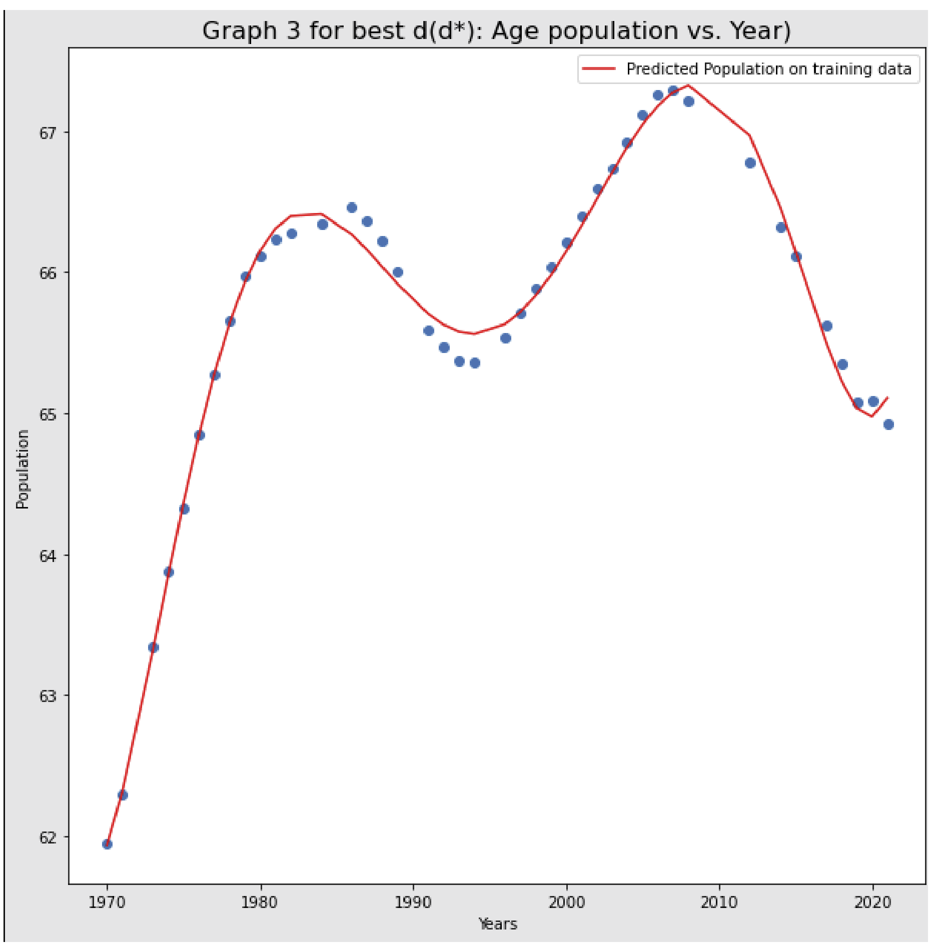
```
In [25]: rmse_train_l
Out[25]: 0.12755963811590315

In [28]: rmse_test_l
Out[28]: 0.12863909044507685
```

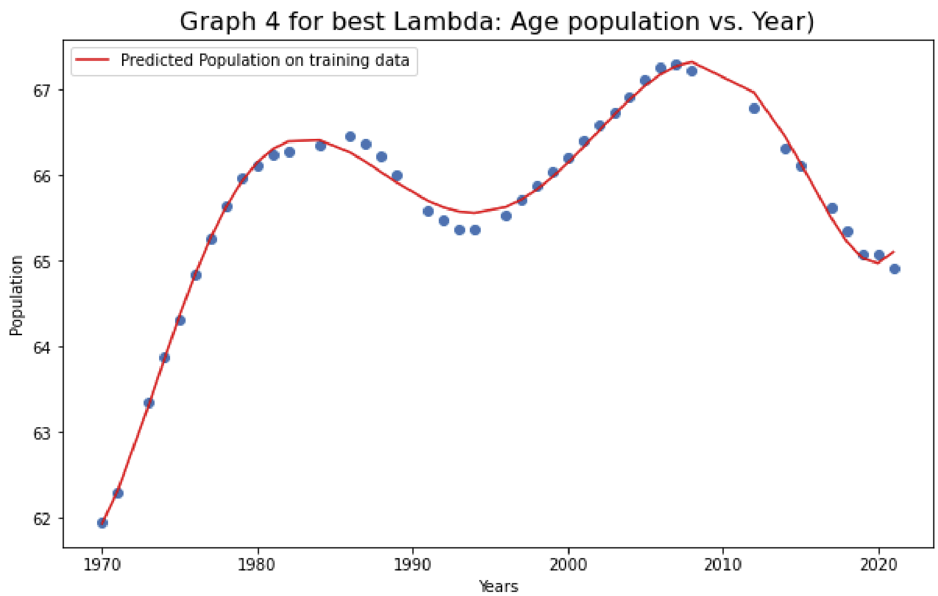
(5) the 2 plots containing all the training data along with the resulting polynomial curves for d* and λ* for the range of years 1968-2023 as input;

<u>Ans:</u>




Graph 2: Avg RMSE vs. Alphas
CV_test
CV_train
Avg. RMSE
Alphas


Graph 3 for best d(d*): Age population vs. Year)
Predicted Population on training data
Population
Years



(6) a brief discussion of your findings and observations.

Ans: Based on the graphs presented above, it can be observed that the CV_train and CV_test error are closest to each other at degree=6. The CV_train error graph shows underfitting before degree=6 and overfitting after degree=6. This implies that the model is not exposed to complex functions before degree=6 for underfitting, and the model predicts the model values precisely for overfitting. The minimum distance between CV_train and CV_test errors at degree=6 suggests that the model's predictions are accurate and the model fits well. Additionally, the Avg_RSME vs Alphas graph demonstrates that the optimal regularized parameter value is exp(-3) because the distance between the errors of CV-Train and CV_test is the smallest at that point compared to other values in the set. Graphs 3 and 4 illustrate the performance of the model when optimized parameters are used, which were selected using the cross-validation model selection method.