CS412

Introduction to Machine Learning

Homework Assignment - 2

Submitted By

Saikrishna Kalahasti Karthik

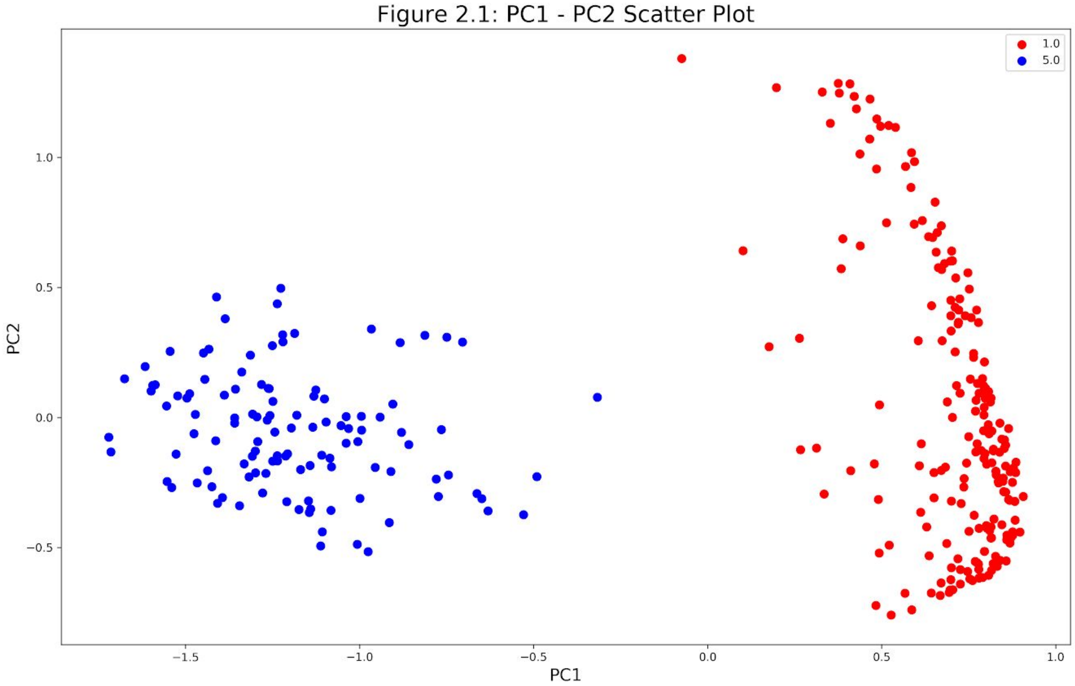
UIN: 655128301

Due Date: 20th February 2019

1 Feature Extraction

Use your ML package to extract two features using Kernel Primary Component Analysis (kPCA) and graph the training data in 2D space with the axes as the first and second components.

Figure 2.1: PC1 - PC2 Scatter Plot



a) Compare kernel PCA features with the features you selected from HW1. Do these features seem to better separate the data?

On visually comparing both the scatter plots, it seems like the PCA separates the clusters better than the features selected in HW1. To quantitatively understand which features separate data points of 1 and 5 better, I performed k-means clustering on the data with k = 2 and measured the distance between cluster centers. I obtained the following result:

Feature Set	Distance between the cluster center
HW1 Features	1.1157
kPCA Features	1.8678

Clearly kPCA features separate data points of 1 and 5 better (Although statistical significance is not computed).

b) Give the explained variance ratio for each of the two feature extractions given above.

I calculated the explained variance ratio in the following way:

$$Explained\ Variance\ Ratio = \frac{Var(DerivedFeature1) + Var(DerivedFeature2)}{Var(OriginaFeature1) + Var(OriginaFeature2) + \ldots + Var(OriginaFeature256)}$$

I obtained the following results for explained variance ratio for HW1 features and kPCA features.

Feature Set	**Explained Variance Ratio**
HW1 Features	0.0061
kPCA Features	0.0228

Is there one of the methods which explains more variance than the other?

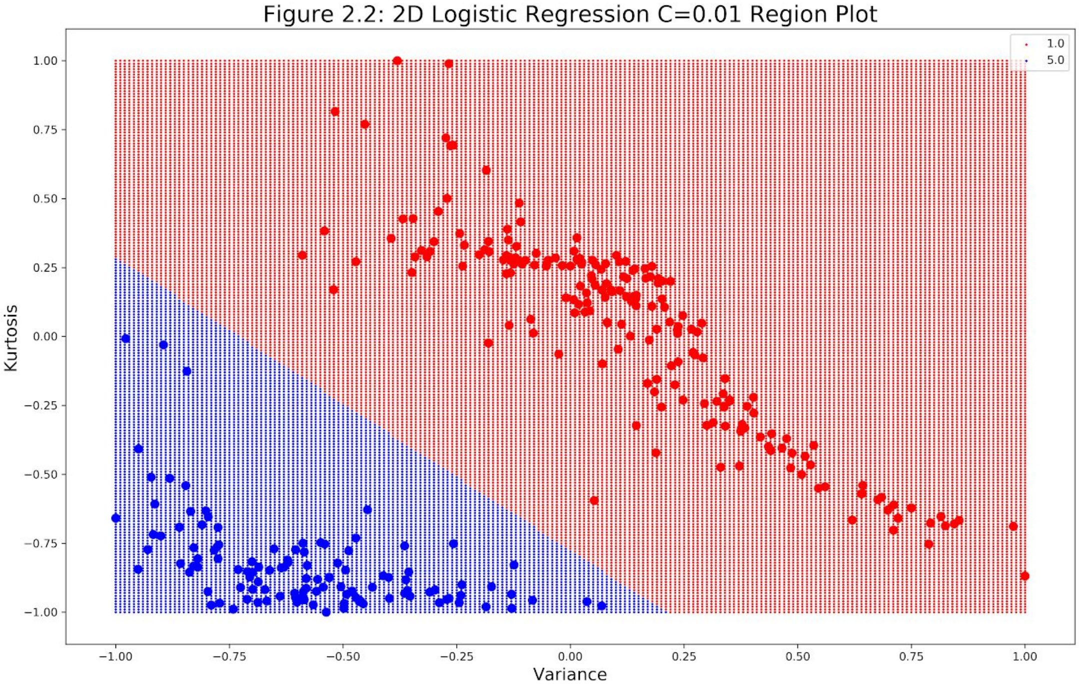
As you can see in the table above, the kPCA has higher variance than the HW1 features.

Is this what you expect? Explain your answer.

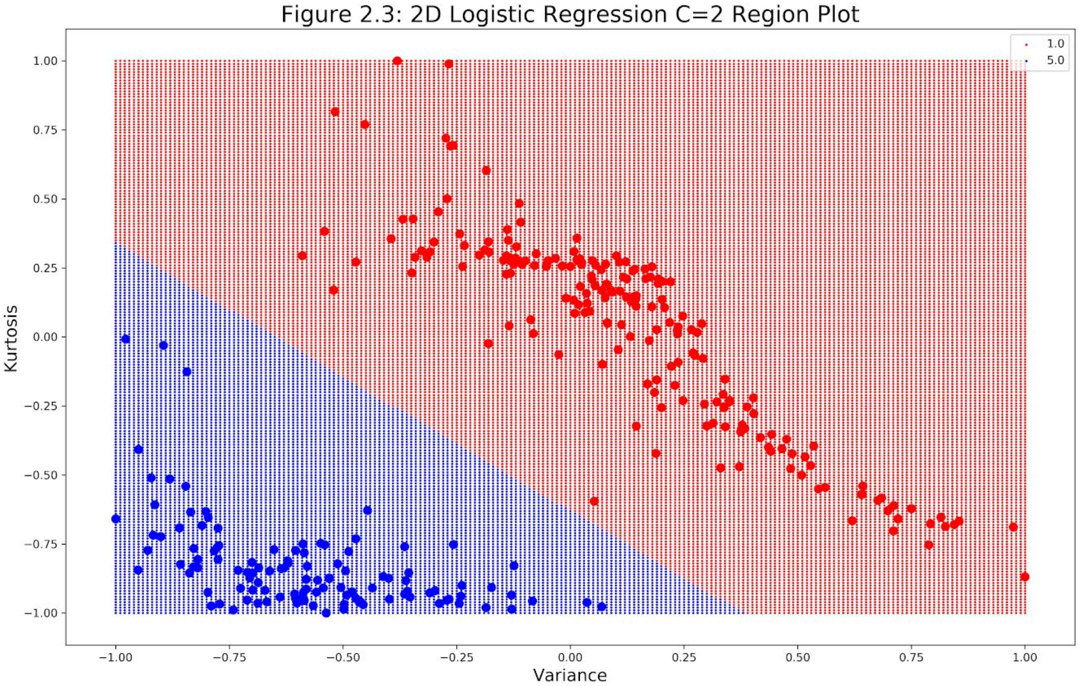
Yes, I did expect this answer. Since kPCA orders the principal components (pc) based on the explained variance ratio and selects the top n pc's into the model, it is likely that they account for high portions of explained variance ratio. Also, from 1a) we see that kPCA separates the clusters better, which means that the top principal components will account for high portions of variations in the data. This may or may not be the case for manually selected features.

2 Logistic Regression

Plot the decision region for your 2D space with a logistic regression solver where c is 0.01.

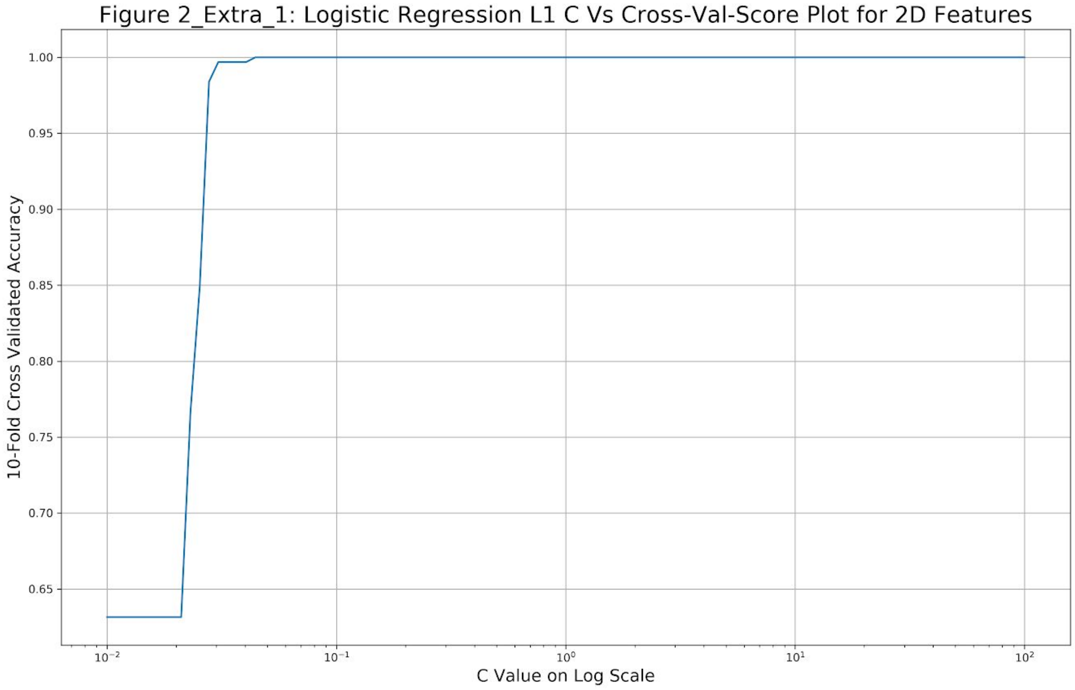


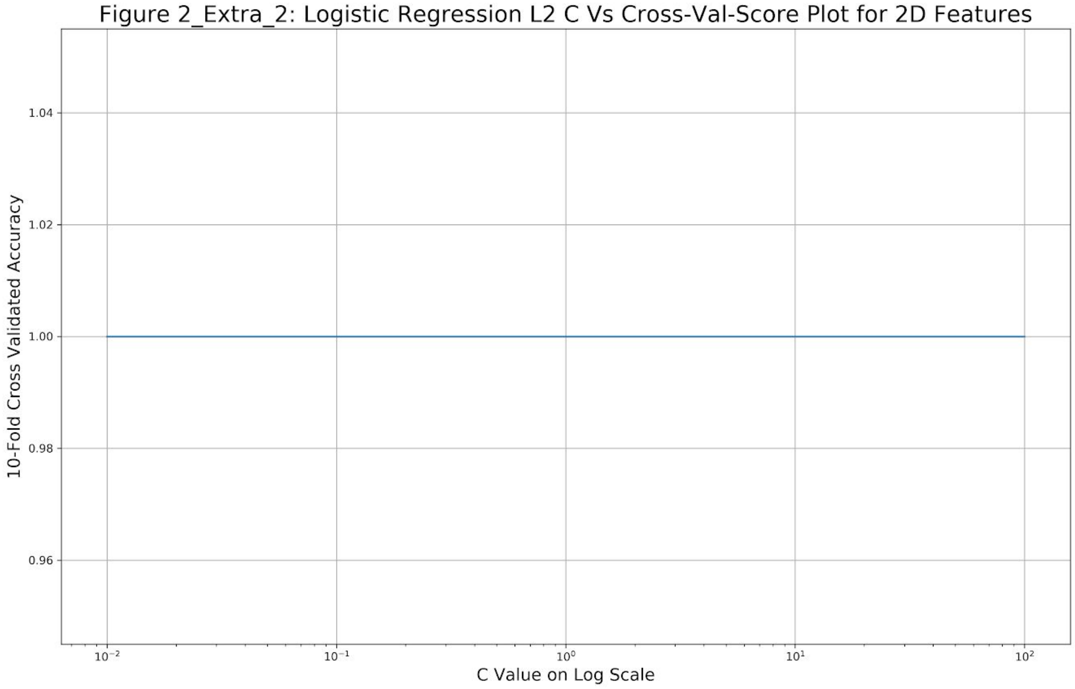
Plot the decision region for your 2D space as above but with C = 2.0.



<u>Graduate student question:</u> Repeat the experiment for Figure 2.2 and 2.3 using L1 regularization. Does this regularization method make it more or less likely for the model to overfit the data. If you don't think there is any overfitting, defend your answer.

When we run L1 and L2 regularization for various C values between 0.01 and 100, we get the following figures.





We also have the following coefficient values at the optimum C.

Regularization	**C Value**	**Coefficient 1**	**Coefficient 2**
L1	0.0443	-1.4271	-1.3888
L2	0.01	-0.42236641	-0.39706025

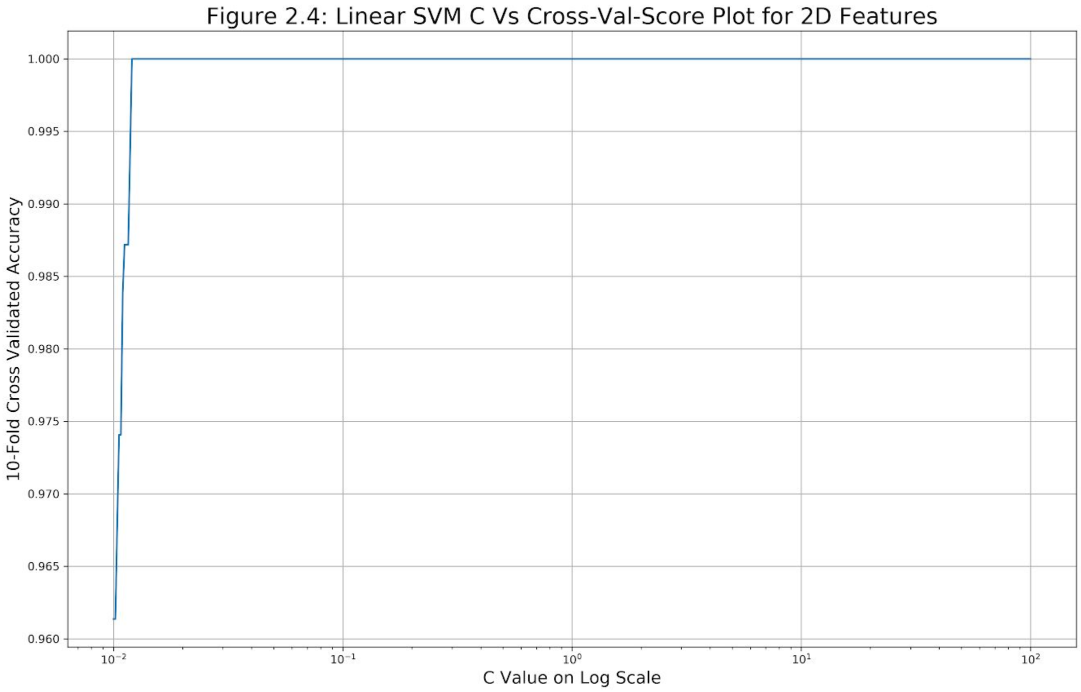
We observe that we have a lower value for Coefficients in case of L1. This indicates a lower regularization (lower λ), implying a higher C value compared to L2, which we can observe in the table given above.

The higher C is indicative of the fact that the model tries to avoid any misclassification in the training set, i.e it overfits. Thus we can safely say that **L1 regularization is more likely to overfit the data than L2 regularization**.

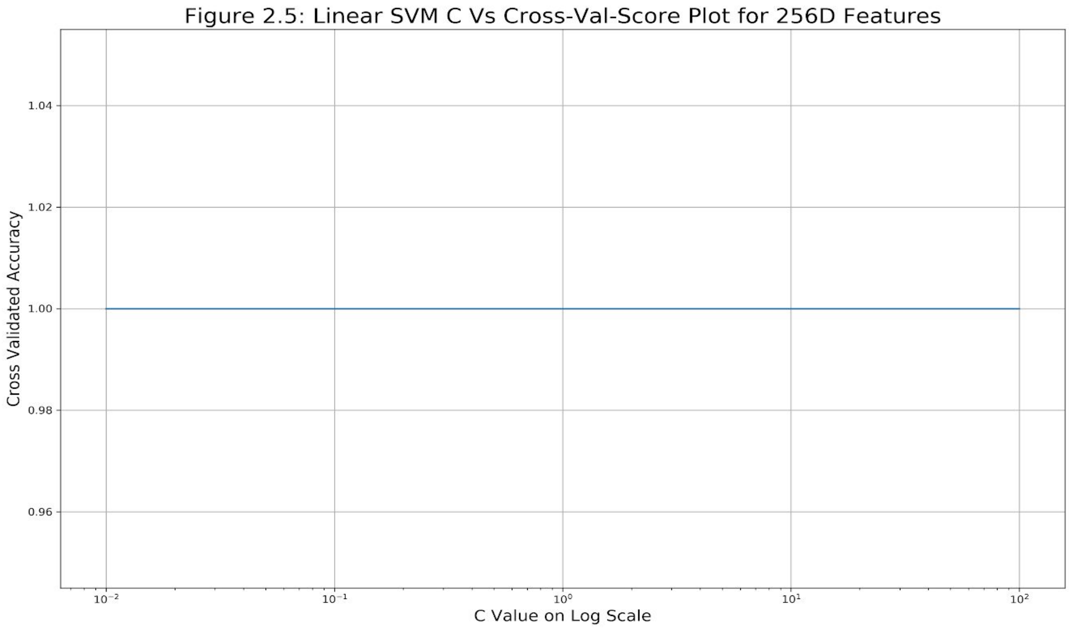
However please note that we cannot observe any overfitting with L1 regularization for this data.

3 Support Vector Machines

On your 2D data from HW1, find the cross-validation errors for all values of c from 0.01 to 100. Your x axis should have logarithmic scale and you should have at least 20 data points. Label this figure 2.4

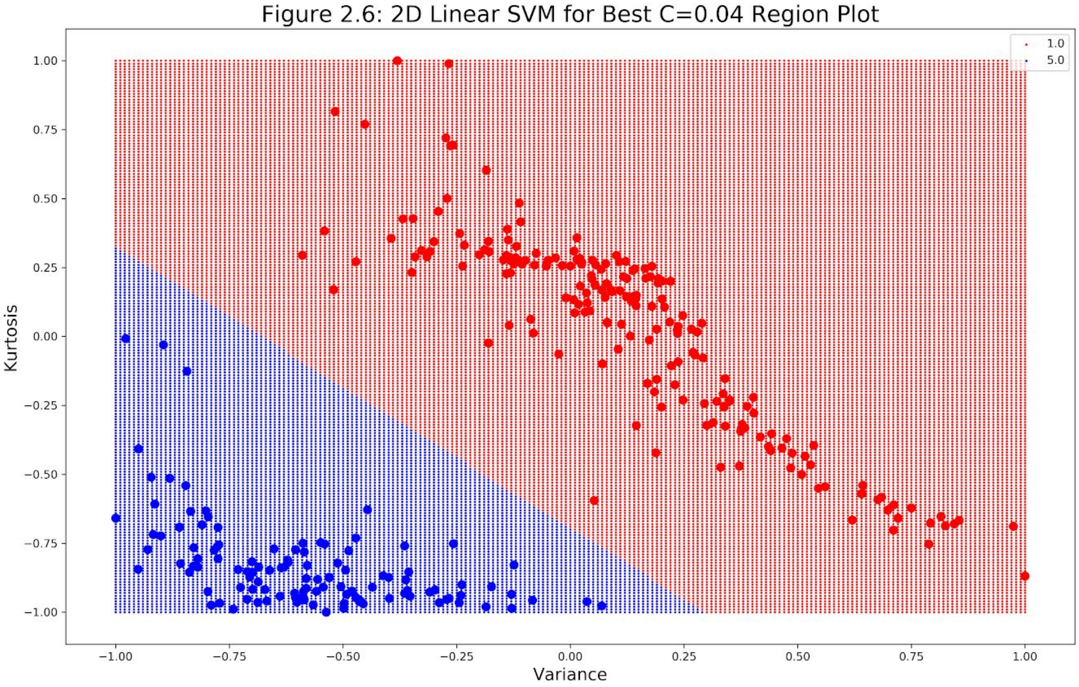


Repeat the above experiment for the 256D degree data. Label this figure 2.5



Find the value of c from Figure 2.4 which has the lowest cross validation error and plot the decision region in your 2D space. Label this 2.6

Figure 2.6: 2D Linear SVM for Best C=0.04 Region Plot



Now, repeat the experiment for the polynomial kernel model for your 2D data. Let the degree of the kernel be each of 2,5,10,20. Give the value of c for each of these kernel degrees which gives the lowest cross validation error.

Degree	C Value	Cross Validation Accuracy
2	4.2292	1.0
5	83.0218	1.0
10	83.0218	0.9416
20	91.1163	0.6666

Explain the tradeoff between the ’degree’ and ’c’ as far as overfitting is concerned.

As we can observe in the table above, the value of C increases with the polynomial degree. The reason behind is that we need a higher C inorder to even out the higher degree polynomials.

The higher C is indicative of the fact that the model tries to avoid any misclassification in the training set, i.e it overfits. The higher C is an effect of higher degree of the polynomial which tries to fit a curve as close to the training data as possible (High Bias). This leads to poor performance in the unseen test dataset. As you can observe in the table above, the cross-validation-score gets increasingly poor with the increase in degree.

Using your 2D data from HW1, find the values for ’degree’ and ’c’ that you believe finds the best fit for the model.

I found the corresponding c values with minimum error for every degree of the polynomial from 1 to 20. The table below has the results. The best given below is for the model with degree = 1 and C = 0.0254.

Degree	**C Value**	**Cross Validation Accuracy**
1	**0.0254**	**1**
2	4.2292	1
3	8.9022	1
4	20.5651	1
5	83.0218	1
6	75.6463	0.9871
7	91.1163	0.9807
8	100	0.9709
9	91.1163	0.9676
10	83.0218	0.9416
11	100	0.8902
12	100	0.8617
13	91.1163	0.8296
14	100	0.8041
15	91.1163	0.7786
16	100	0.7532
17	100	0.7278
18	100	0.6984
19	100	0.6826
20	91.1163	0.6666

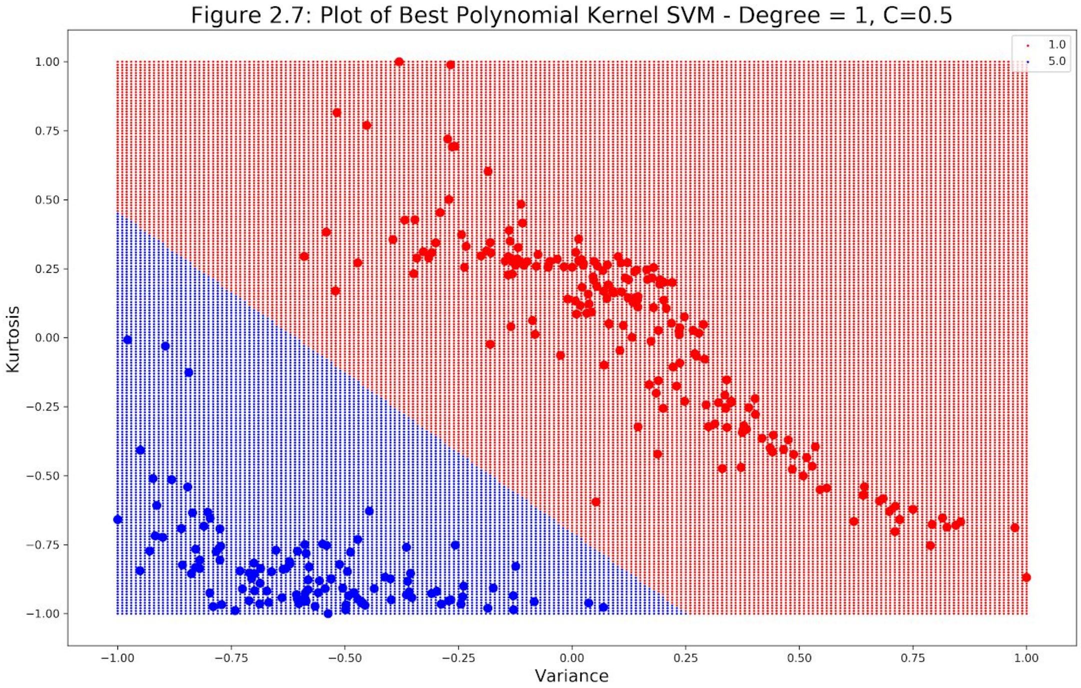
Plot the decision region for this model (From figure 2.6) and explain why you think this model is best. Support your conclusion with data.

I chose a model of degree = 1 and C = 0.5.

By visually looking at the data, the data clearly looks linearly separable. Therefore, a polynomial of degree 1 should be sufficient to build an ideal model. Therefore a model of degree 1

Now coming to the C value, we should not choose a high C, as that will cause the decision boundary to be narrow, which will cause overfitting. And a low C will make the decision boundary to be too wide, which will cause underfitting. Therefore we need to choose a C that creates a boundary of a decent size. That is the reason I choose C = 0.5.

Using Degree=1 and C=0.5, we get the following region plot.



As you can clearly see, the decision boundary is neither too close to the red data points, nor too close to the blue data points. This model gives a cross validation accuracy score of 100%.

Graduate student question: Provide two graphs of decision regions for SVC models in the 2D space. One should have evidence of overfitting, and the other should have evidence of underfitting. Explain the parameters that lead to each of the graphs and their cross-validation errors

Overfitting: As the C value increases, the model tries to achieve a high training accuracy, or creates a higher bias. Also we see here that we are trying to fit a second degree polynomial for a linearly separable data.

You can clearly see that this model classifies the region on the top right corner as blue, even though one would argue against it. This is a clear result of overfitting, and it is due to the higher degree and higher C values. The cross validation score for this model is 1, however we know it won't perform well on unseen data set.

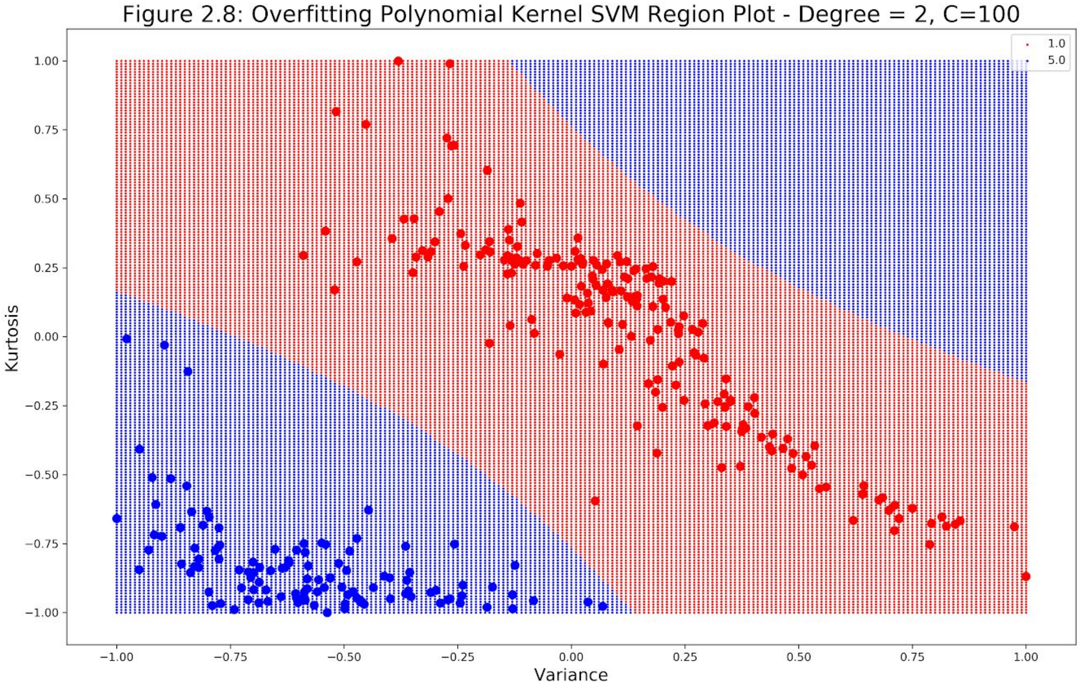
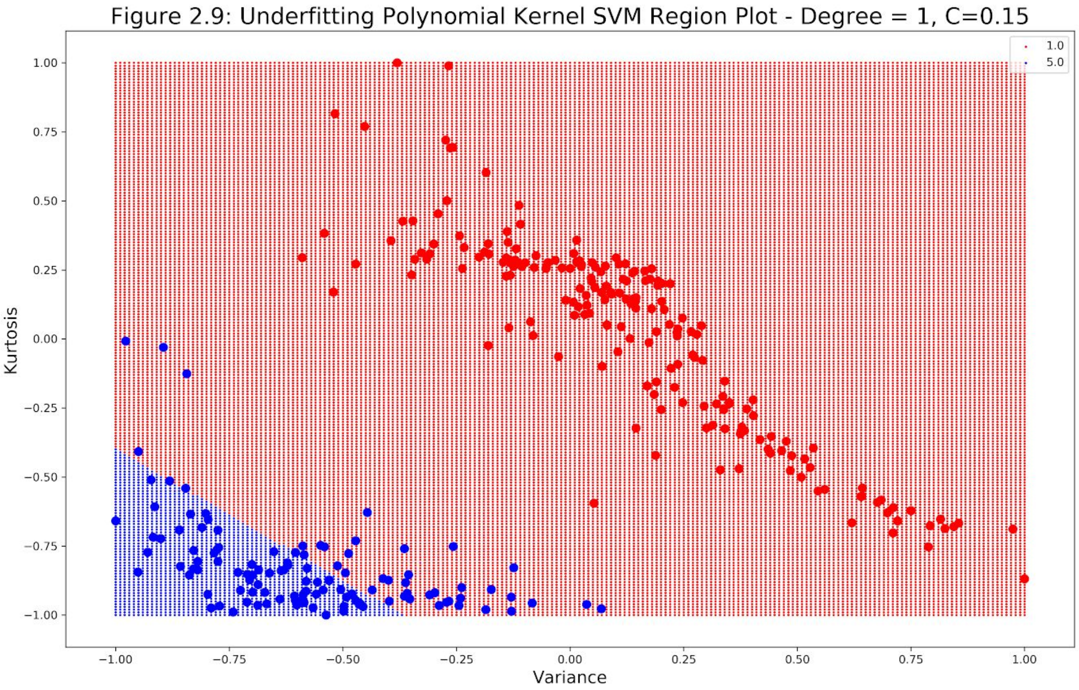


Figure 2.8: Overfitting Polynomial Kernel SVM Region Plot - Degree = 2, C=100

Underfitting: The given data points are clearly linearly separable. With a significantly low C, the model does poorly on the training data. When the C value is too low, we are trying to generalise the model too much, and it does not classify well enough. In other words, the decision boundary is too wide, and it leads to underfitting.

We choose a model with degree 1, and a C value of 0.15. This model has a low cross-validation accuracy of 0.71. The reason for this is clearly underfitting.



Extra Credit

For the kernel support vector classifiers (both radial and polynomial), examine the parameter gamma. Conduct experiments for different values of gamma, c and degree.

The results when we run experiments for Polynomial Kernel for 0.01<=Gamma<=100, Degree in [1,2,5,10], and 0.01<=C<=100 can be found in

out_data/poly_c_vs_gamma.xlsx in the code zip.

The results when we run experiments for Radial Kernel for 0.001<=Gamma<=1000 and 0.01<=C<=100 can be found in

out_data/rbf_c_vs_gamma.xlsx in the code zip.

Due to a high computation time for higher degree polynomial kernels and higher Gammas, I restricted the degree of the polynomial kernel to 10.

Try to determine the relationship between gamma and over/under fitting for the model. Support your explanation with figures.

Overfitting and Gamma:

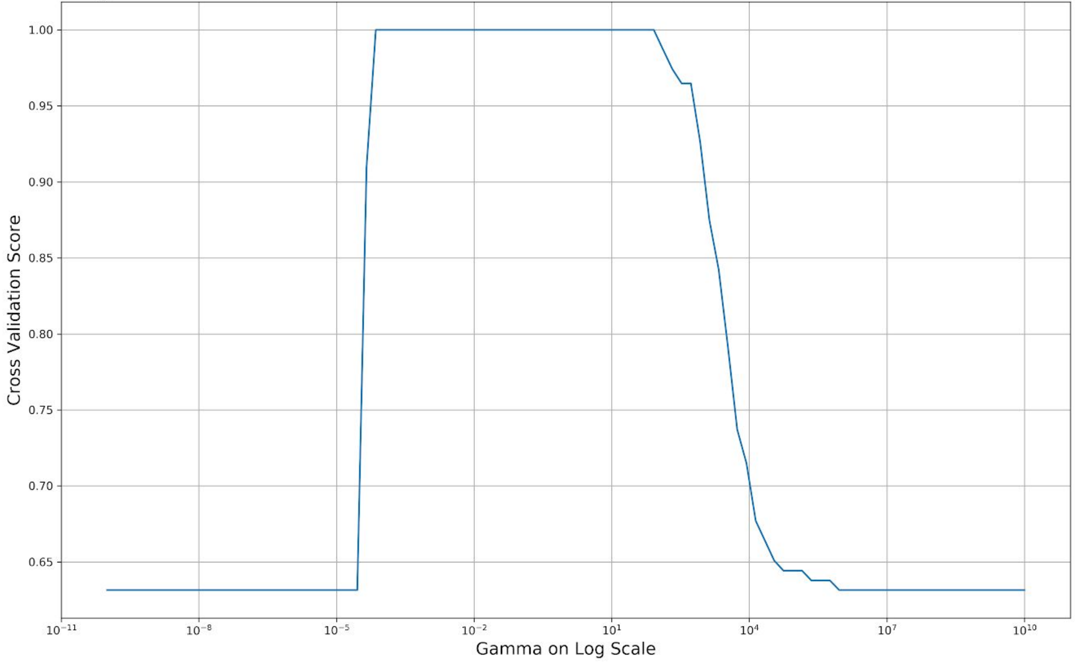
When the gamma value is large, only the closer training examples will affect the label of a data point during classification. If the gamma value is too large, the radius of the area of influence of a support vector is shrunk to include only itself. Any value of C cannot cause regularization because no matter how wide the soft margin is, the support vectors are influenced only by the close points. **This is the case of overfitting.**

Underfitting and Gamma:

When gamma value is too small, each support vector is influenced by almost every training point in the region, and thus the model does not learn the nature/shape of the training data well enough. **This can cause underfitting**.

The figure below illustrates the explanation given above. We have a small cross validation score for very low Gamma value and very high Gamma value, indicating underfitting and overfitting respectively.

Figure 2.10: Radial Kernel SVM Gamma Vs Cross Validation Score Plot for 2D Features


Cross Validation Score
10^{-11}
10^{-8}
10^{-5}
10^{-2}
10^{1}
10^{4}
10^{7}
10^{10}
Gamma on Log Scale