GitHub Code Repository: https://github.com/pharmon0/Harmon_ECGR5105

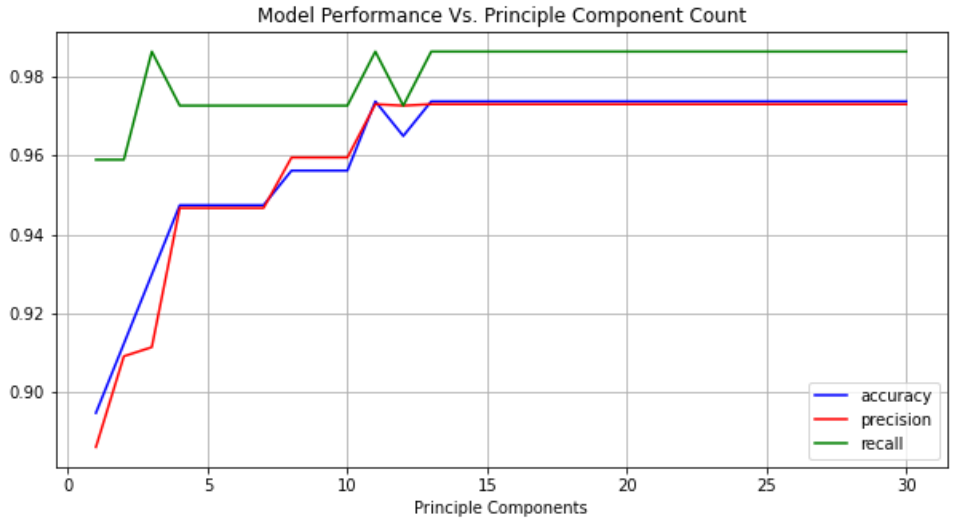
Problem 1 (50pts):

Use the cancer dataset to build an SVM classifier to classify the type of cancer (Malignant vs. benign). Use the PCA feature extraction for your training. Perform N number of independent training (N=1, …, K).

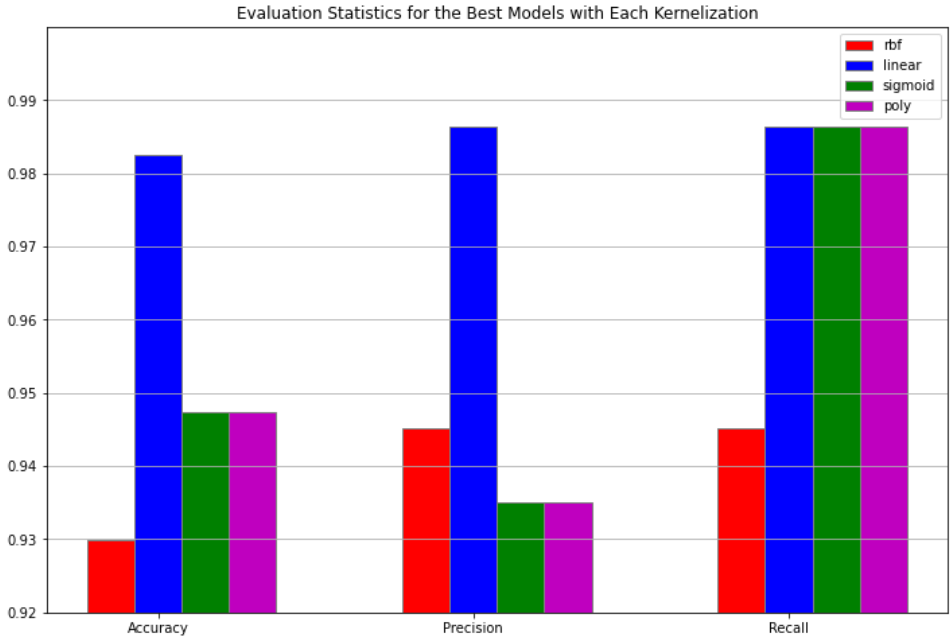
1. Identify the optimum number of K, principal components that achieve the highest classification accuracy.
2. Plot your classification accuracy, precision, and recall over a different number of Ks.
3. Explore different kernel tricks to capture non-linearities within your data. Plot the results and compare the accuracies for different kernels.
4. Compare your results against the logistic regression that you have done in homework 3.

Make sure to explain and elaborate your results.

All possible PCA component counts were tested iteratively. Resultantly, according to the following plot, the optimal number of principal components for an SVC with default settings is about 11.



From here, a wide variety of kernels, parameters, and PCA components were tested in order to find the optimal values. The resultant statistics of the best training for each kernelization method were plotted in the following bar-graph.



From this, it is clear that the linear kernelization model performed the best for this dataset. The resultant training statistics and model parameters of this final best model can be found below.

```
Best model among all parameter options for linear-type kernelization:
{'pca__n_components': 8, 'svc__C': 3, 'svc__gamma': 1e-06}

Report for Best Model Found for linear-type kernelization:
--------------------------------------------------------
              precision    recall  f1-score   support

         0.0       0.98      0.98      0.98        41
         1.0       0.99      0.99      0.99        73

    accuracy                           0.98       114
   macro avg       0.98      0.98      0.98       114
weighted avg       0.98      0.98      0.98       114

Confusion Matrix:
[[40  1]
 [ 1 72]]
```

The best statistics obtained in Assignment 3 were as follows:

	precision	recall	f1-score	support
0.0	0.97	0.85	0.91	41
1.0	0.92	0.99	0.95	73
accuracy			0.94	114
macro avg	0.95	0.92	0.93	114
weighted avg	0.94	0.94	0.94	114

It is clear from comparing this data that the optimized SVC model is performing better in all three major statistical categories than the Linear Regression model from Assignment 3.

GitHub Code Repository: https://github.com/pharmon0/Harmon_ECGR5105

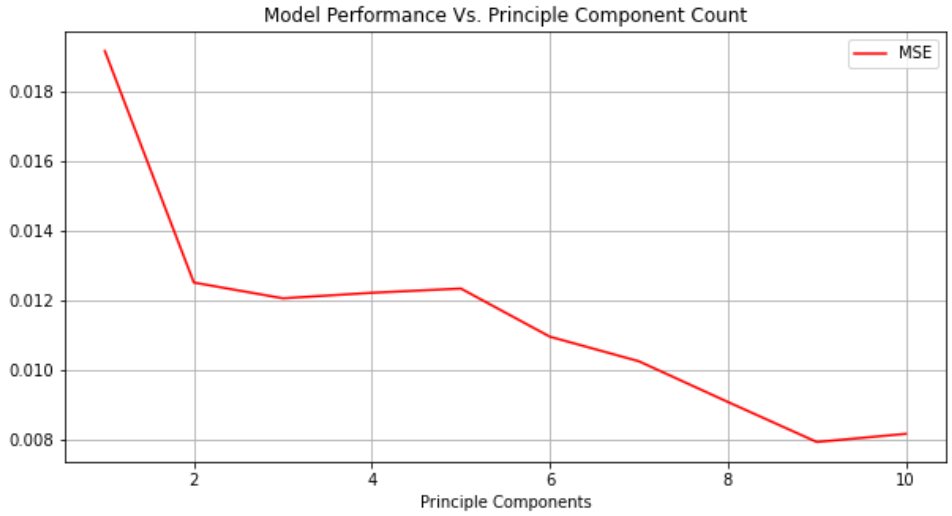
Problem 2 (50pts):

Develop a SVR regression model that predicts housing price based on the following input variables:

Area, bedrooms, bathrooms, stories, mainroad, guestroom, basement, hotwaterheating, airconditioning, parking, prefarea

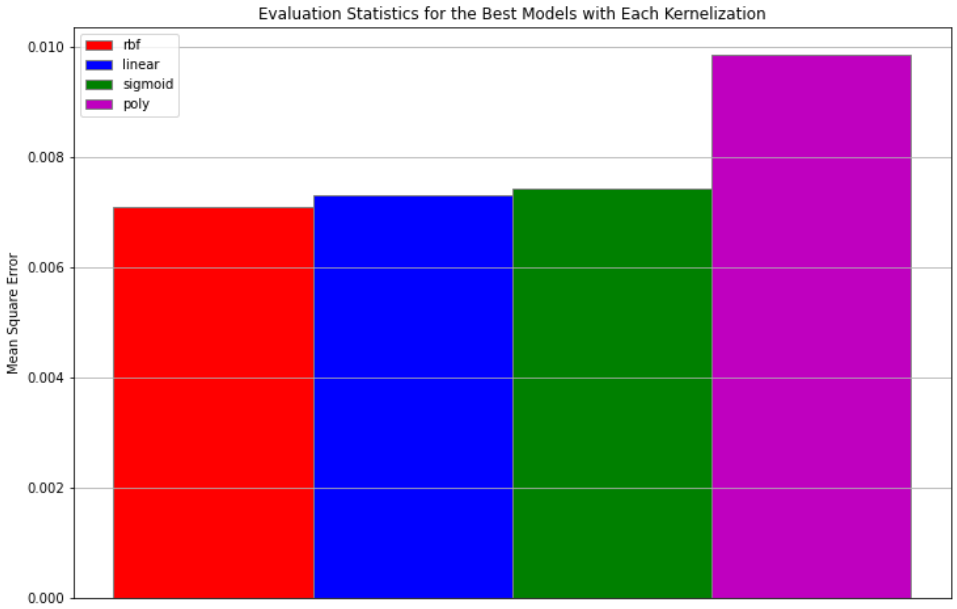
1. ~~Plot your regression model for SVR similar to the sample code provided on Canvas.~~
2. Compare your results against linear regression with regularization loss that you already did in homework1.
3. Use the PCA feature extraction for your training. Perform N number of independent training (N=1, …, K). Identify the optimum number of K, principal components that achieve the highest regression accuracy.
4. Explore different kernel tricks to capture non-linearities within your data. Plot the results and compare the accuracies for different kernels.

The SVR model was trained with a variety of different PCA counts and default model parameters. The mean square error results of this were plotted below. From this plot, it is apparent that the optimal PCA count was 9.



Upon performing a grid search for each kernelization method over a wide range of parameters, the best models for each kernel type were found and the mean square error of these models were placed in the following bar-graph for comparison.

GitHub Code Repository: https://github.com/pharmon0/Harmon_ECGR5105



From this chart, it can be seen that the root-basis-function (rbf) kernelization produced the best error characteristics. The results and parameters of this model are as follows.

```
Best model among all parameter options for rbf-type kernelization:
{'pca__n_components': 10, 'svr__C': 1, 'svr__gamma': 'auto'}
MSE Value for rbf-type kernelization is 0.0071054664994934175
```

Worth noting is that the final mean square loss of the Gradient Descent training done in Assignment 1 (part 3) was only 0.004, meaning that the gradient descent method produced roughly 22.4% less error.

GitHub Code Repository: https://github.com/pharmon0/Harmon_ECGR5105