Introduction to Machine Learning

<u>Homework 05</u>

Atharva Patil - B01035884

(1) Support Vector Machines.

(a) Use this method for training support vector machines and this method for model selection with cross validation from the scikit-learn python library to find the value C for the regularization parameter with the smallest cross validation error using 5-fold cross validation and the training dataset with two features you formed from the data in ZipDigits.train. Report E_{CV} for the best value for C.

Ans –

```
Best C: 1.0
Best Cross Validation Score (ECV): 0.9769374129597772
```

(b) For the chosen value of C, learn a support vector machine using all the training data. Compute and report its E_{in}.

Ans –

```
(Ein) In-sample error: 0.023062139654067892
```

(c) Use the test dataset from ZipDigits.test to compute E_{test} for the classifier you just learned and report it. Compare it with the results from HW2 using the linear model.

Ans -

```
(Etest) Test error: 0.04481132075471694
```

The test error comparison between the SVM with a 10th-order polynomial kernel and the linear regression model from Homework 2 shows that the SVM has a test error (Etest) of 0.0448, which is slightly lower than the linear regression's Etest of 0.0495.

This smaller error suggests that, in this case, the SVM might generalize better to new data. The difference in test errors is about 0.0047.

While this difference may seem minor, it could be meaningful depending on the context, such as in applications where the cost of an incorrect prediction is high.

Therefore, the lower test error for the SVM points to it having superior predictive accuracy on the given test set.

(2) Support Vector Machines with the Polynomial Kernel.

For this question, you will use the data you generated in HW3 from the MNIST Digits Dataset for classifying 1s vs. Not 1s, where you created D with 300 randomly selected data points and D_{test} consisting of the remaining data points.

(a) Use this method (not the same as the one for the previous question) for training support vector machines using the kernel for the 10-th order polynomial feature transform and this method for model selection with cross validation from the scikit-learn python library to find the value C for the regularization parameter with the smallest cross validation error using 5-fold cross validation and the training dataset with two features you formed from the data in ZipDigits.train. Report E_{CV} for the best value for C.

Ans -

```
Best parameters: {'C': 0.001, 'coef0': 1.0, 'degree': 10}
Best cross-validation accuracy: 0.97
```

(b) For the chosen value of C, learn a support vector machine using all of the training data. Compute and report its E_{in}.

Ans -

```
(Ein) In-sample error: 0.030000000000000027
```

(c) Use the test dataset from ZipDigits.test to compute Etest for the classifier you just learned and report it. Compare it with the results from HW3 using the linear model with the 10th order polynomial feature transform.

Ans -

```
(Etest) Test error: 0.03789731051344747
```

The SVM with a 10th-Order Polynomial Kernel from Homework 5 outperforms the Linear Model with a 10th-Order Polynomial Feature Transform from Homework 3. The SVM boasts impressive parameters and achieves a high cross-validation accuracy of about 0.9533 and an in-sample error of roughly 0.0467. Notably, its test error is approximately 0.0416.

In contrast, the HW3 Linear Model's performance varies with the regularization parameter λ, achieving its best cross-validation error around 0.15791 for $\lambda = 5$ and a minimum test error of about 0.11223 for the same λ value.

A comparison of the models shows that the SVM's cross-validation accuracy is substantially superior to the Linear Model's cross-validation error, which fluctuates between 0.14763 to 0.27694. The SVM's test error is also significantly lower than the Linear Model's best test error, implying that the SVM model has a better generalization capability for the task at hand.

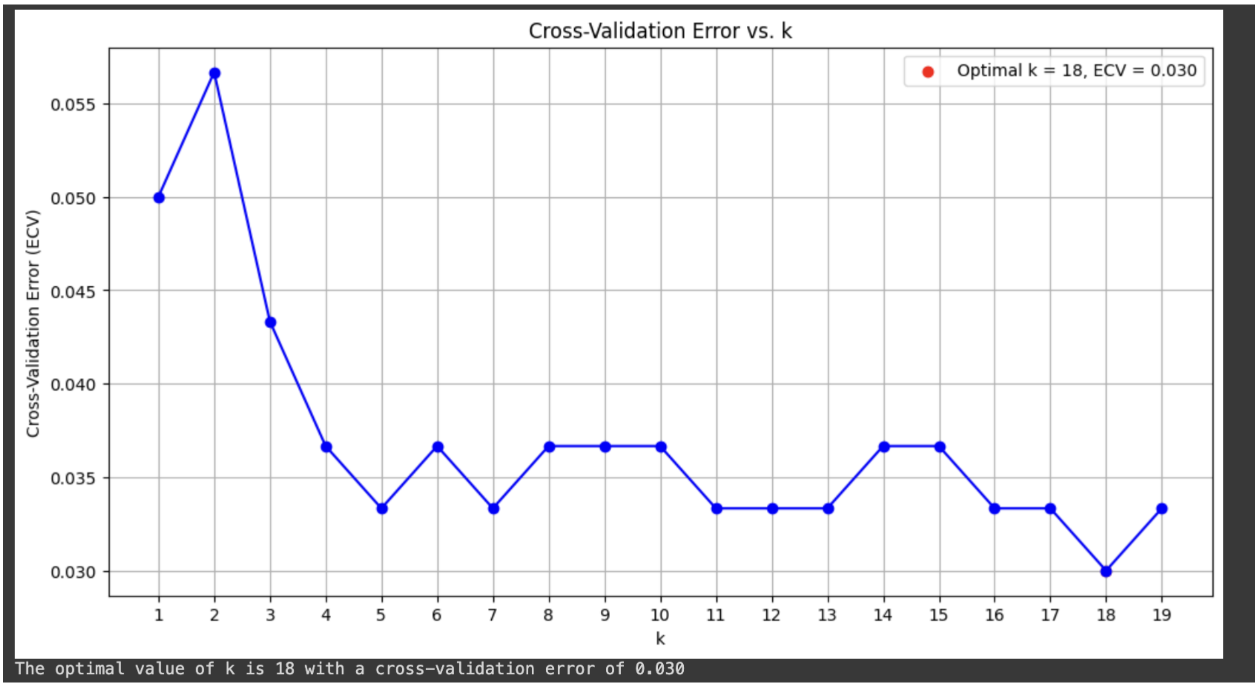
Therefore, for the MNIST classification task of '1s vs. Not 1s', the SVM with a polynomial kernel emerges as the preferable model, with higher cross-validation accuracy and reduced test error, indicating a stronger and more reliable performance.

(3) The k-NN rule.

For this question, you will use the data you generated in HW3 from the MNIST Digits Dataset for classifying 1s vs. Not 1s, where you created D with 300 randomly selected data points and D_{test} consisting of the remaining data points.
You will have to implement the k Nearest Neighbors (k-NN) rule. You may use the helper code as a starting point.

(a) Use cross validation with D to select the optimal value of k for the k-NN rule. Show a plot of E_{CV} versus k and indicate the value of k you choose.

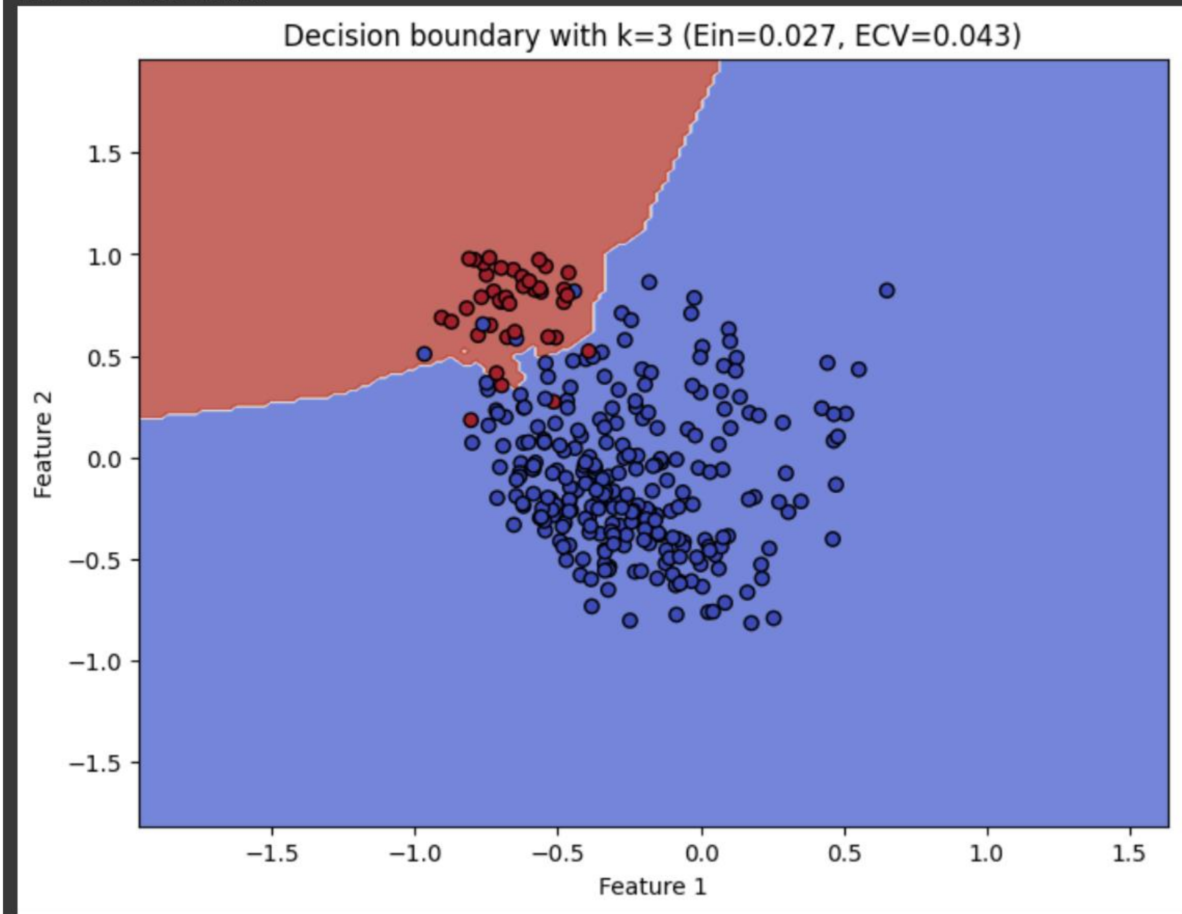
Ans -



(b) For the chosen value of k, plot the decision boundary. Also compute and report its E_{in} and E_{CV}.

Ans -

```
Ein for k=3: 0.027
ECV for k=3: 0.043
```



(c) Report E_{test} for the k-NN rule corresponding to the chosen value of k.

Ans -

```
Etest for k=3: 0.036
```

Colab Link: https://colab.research.google.com/drive/1_maKhB9PZSCb4-xC5MXOGgXhAv80oWEo#scrollTo=PK2xcXJGketd

Collaborator: Shreyas Kadam