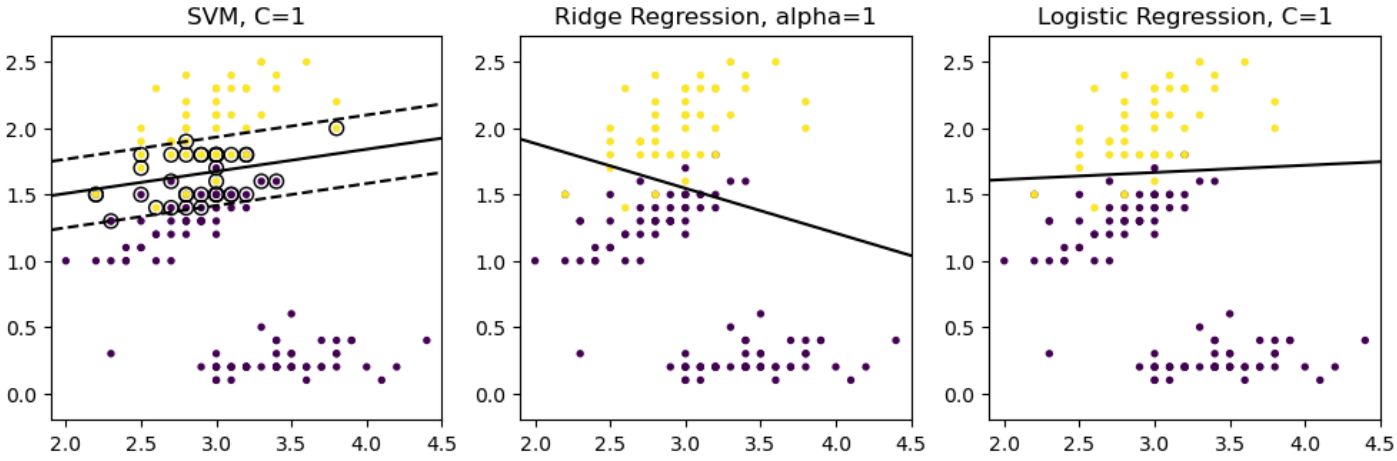
Homework 5

Matthew Carey

Problem 1:

a) The following plot shows how 3 different regression models fit to the toy data we are using:



b) The 2 models that have the most similar decision boundaries are SVM and Logistic Regression. They are similar in that both boundaries have a positive slope with relatively low magnitude, whereas the decision boundary in the Ridge Regression model has a negative slope of relatively large magnitude.

Problem 2:

a)

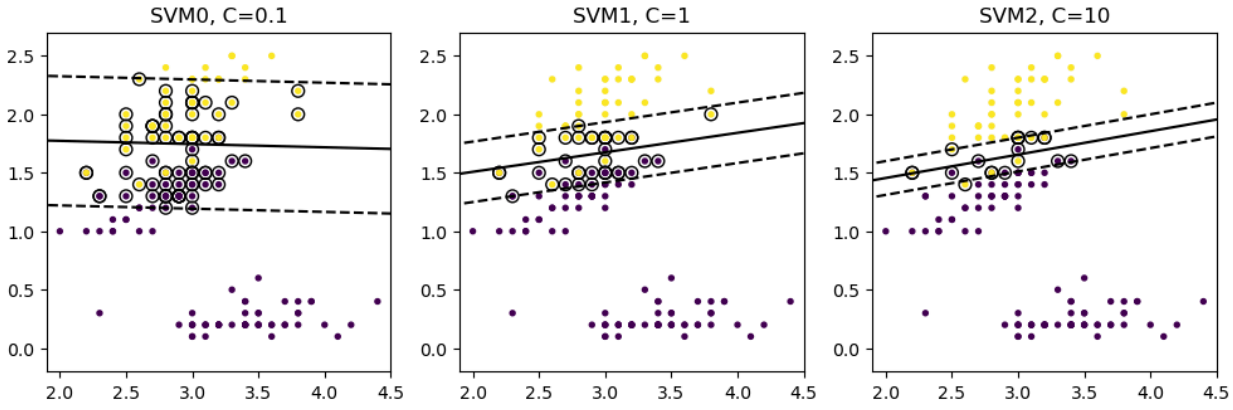


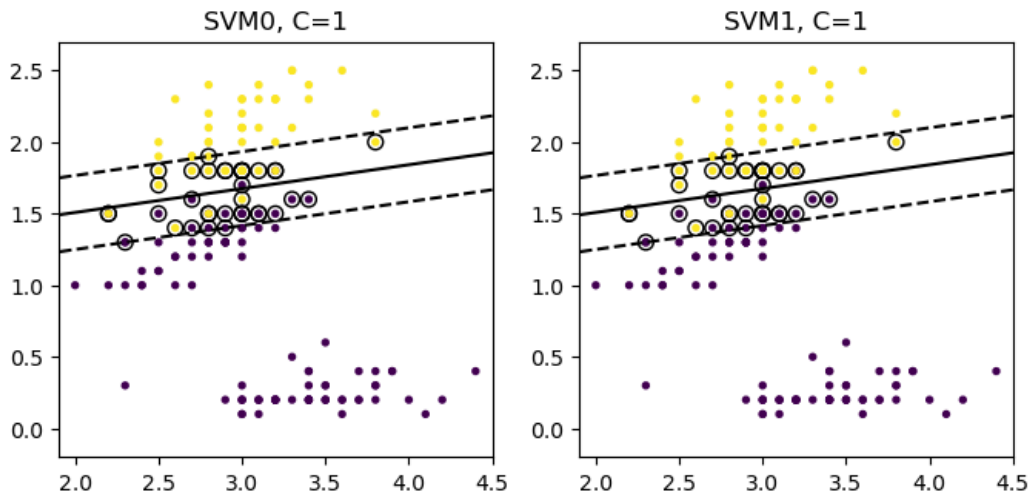
Figure 2a: Plot of 3 SVM models with different C values

b) The model with the largest margin is SVM0, with a C value of 0.1. We can calculate the margin width by using the formula $M(w) = \frac{2}{||w||_2}$, where $||w||_2 = \sqrt{x_1^{\ 2} + ... + x_F^{\ 2}}$. This results aligns with what we would expect given the relationship between C and margin width. A large value of C penalizes misclassifications, resulting in a hard margin approach where the decision boundary may be fitted closer to the data points, which will result in a smaller margin. Conversely, a smaller value of C allows for more misclassifications, allowing the decision boundary to be more loosely fitted to the data resulting in a larger margin. Additionally, we can see this mathematically in the loss function: $min_{w,b} \frac{||w||}{2} + C \sum_{n=0}^{N} \xi_n$ When C is small, the penalty term has minimal effect and the model is able to prioritize maximizing the margin in the $||w||$ term.

c) The model with the least number of support vectors is SVM2, with a C value of 10. If we look at the training objective of SVM, $min_{w,b} \frac{||w||}{2} + C \sum_{n=0}^{N} \xi_n$, we see that a large value of C limits the ability to maximize the margin. Thus, a large C corresponds to a smaller margin, which will require less support vectors to define than a large margin. The smaller the margin, the less data points that are within the margin that will become support vectors that define it.

Problem 3:

a)



b) The two SVMs have the same decision boundary, despite SVM1 being trained on a reduced dataset.

c) If you can achieve an identical model despite training it on a reduced dataset, this means that prediction speed can be increased and model storage can be used more efficiently. The time complexity of predictions in SVMs is a function of the number of support vectors, which is related to the number of data points. In a reduced dataset, there will be less support vectors and thus will have faster predictions. Additionally, this will result in less storage being required for the model, as it is defined using less data.

Problem 4:

a)

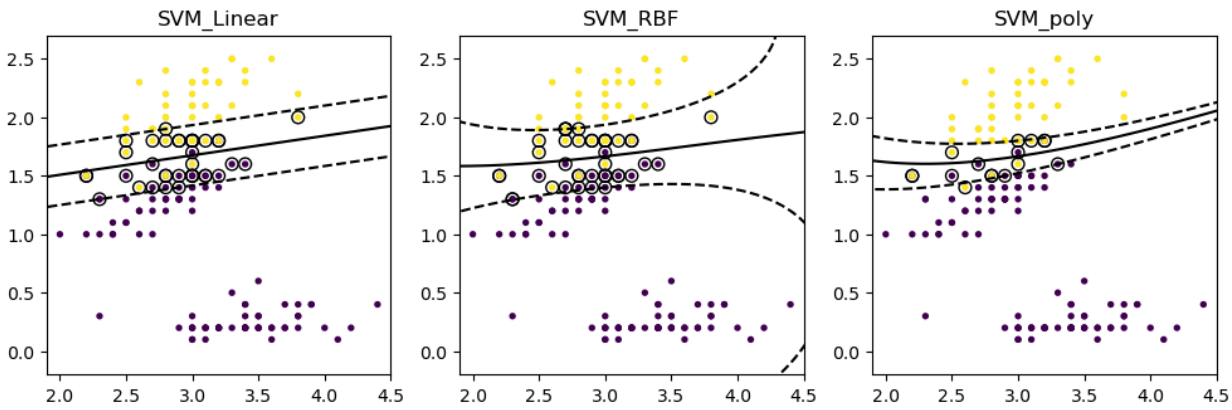
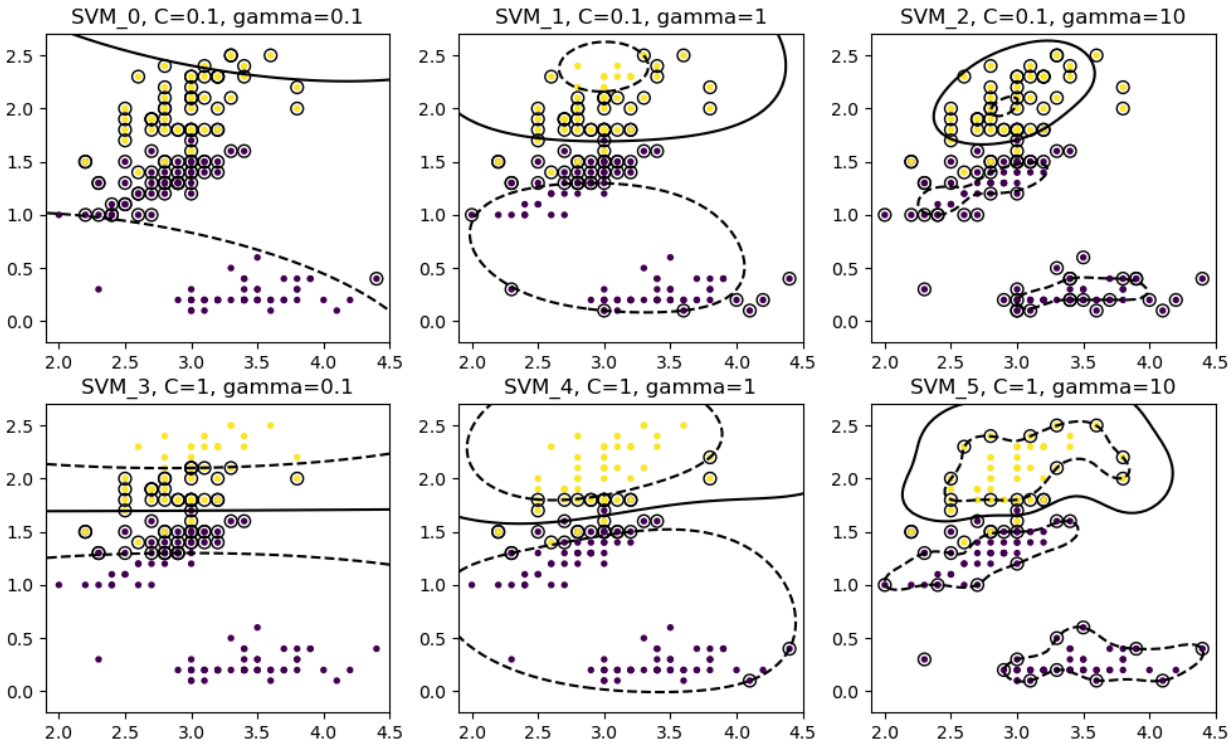


Figure 4a: Decision boundaries of SVM models with linear, RBF, and polynomial kernels

b) The Linear kernel model has a linear decision boundary. The RBF kernel is close, but upon analysis we can see that it is not.

Problem 5

a) F

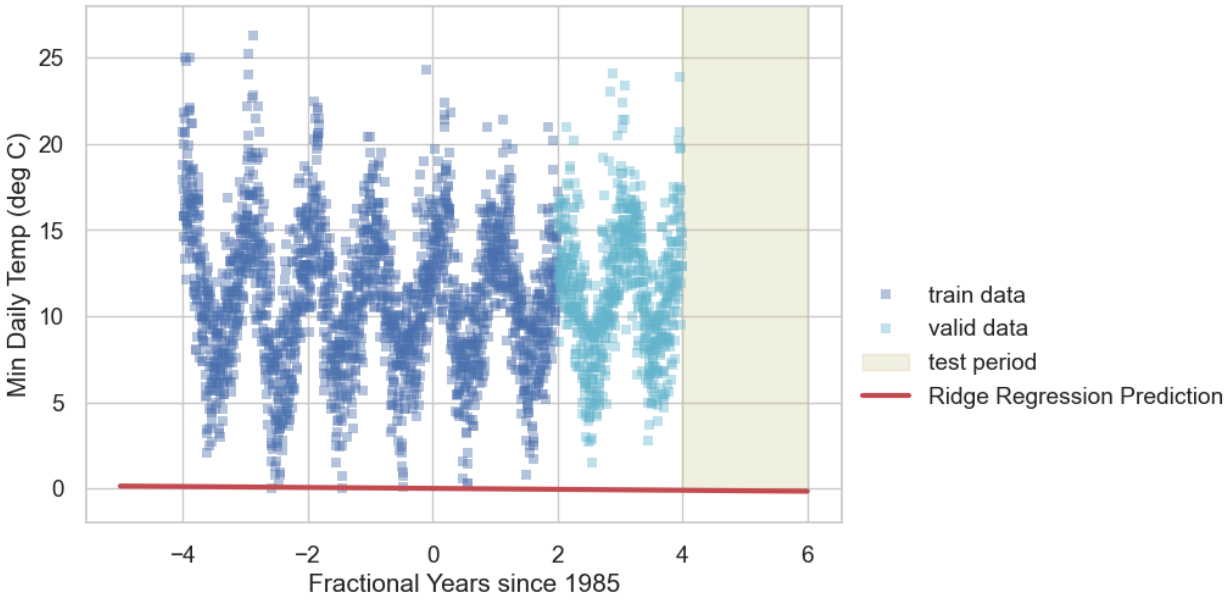


b) To reduce training error and increase overfitting, we would want to decrease the hyperparameter C. mathematically, this can be explained by the loss function $min_{w,b} \frac{||w||}{2} + C \sum_{n=0}^{N} \xi_n$. with a decreased value of C, the model is able to minimize the $\frac{||w||}{2}$ term by overfitting to the data as much as it can. As C increases, the $C \sum_{n=0}^{N} \xi_n$ penalty term will have an increased effect on the loss function if the model is overfitted, preventing it from being selected by the *min* operator.

c) To increase overfitting and reduce the training error, the gamma hyperparameter should be increased. In a situation where no two data points are the same and a very high gamma value, it is possible to achieve 0 training error. This is because a high gamma value corresponds to a data poin's influence on the decision boundary being very small, and so as gamma approaches infinity, the effect of a single data point is only in its own 'location', and so the boundary will fit perfectly around the data. The only case where this could still result in training error is when two data points have the same X value(s) but different y, so it will be impossible for the model to fit around them, but we know this is not the case in this dataset. Thus, it is possible to achieve 0 training error.

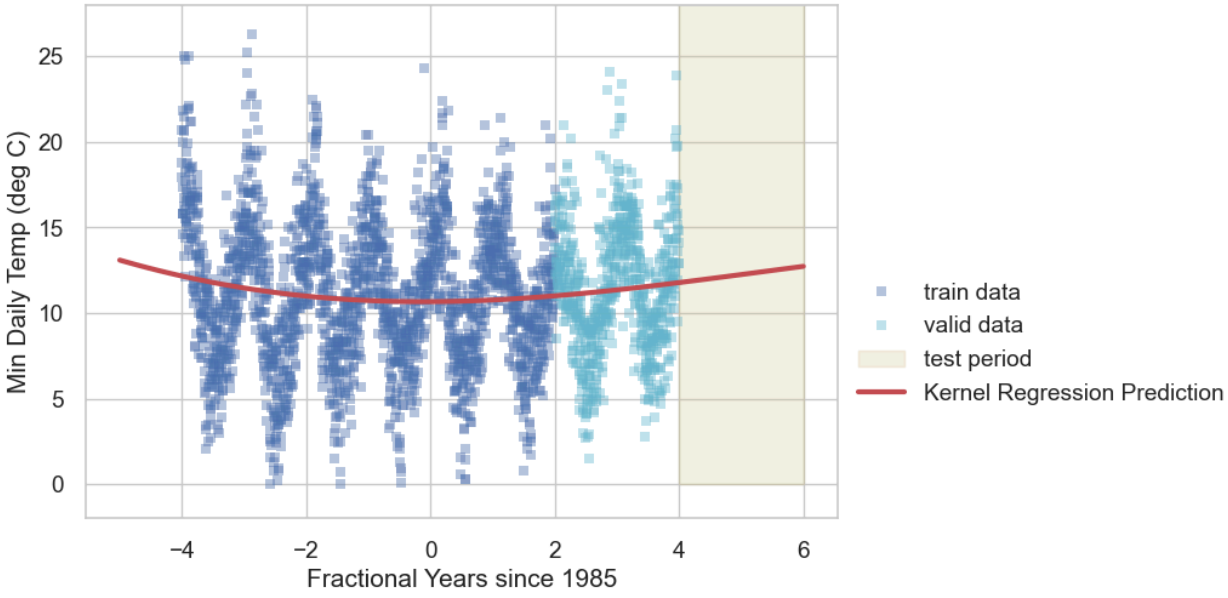
Problem 6:

```
The best hyperparameter setting found is:
{'alpha': 100}
```



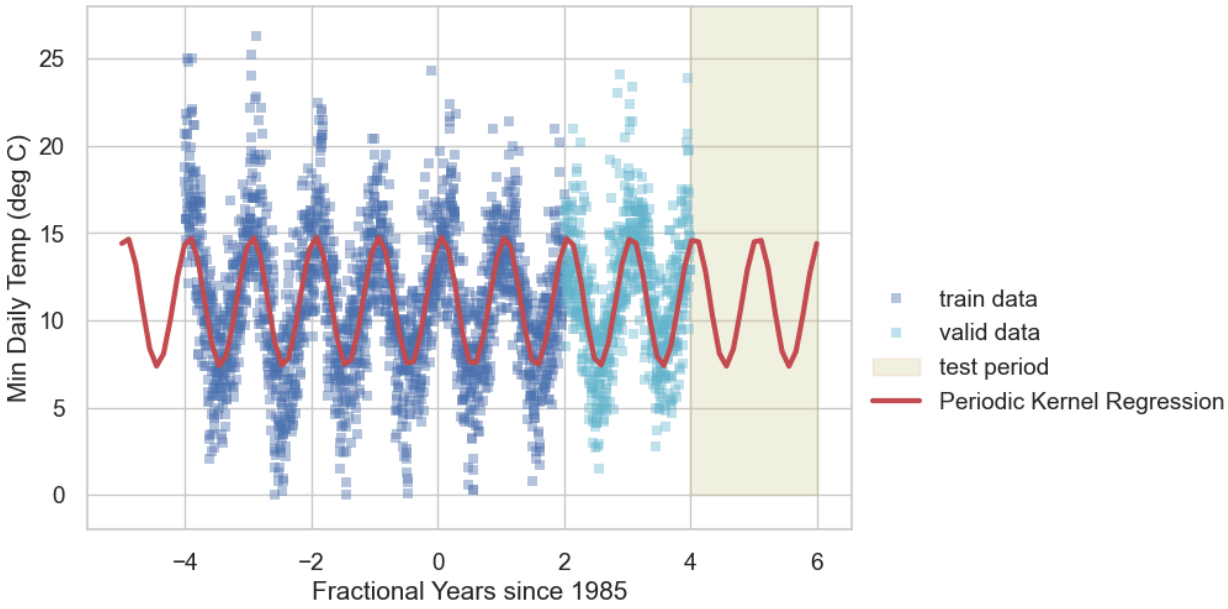
Problem 7:

```
The best hyperparameter setting found is:
{'alpha': 0.01, 'kernel': RBF(length_scale=16)}
```



Problem 8:

```
The best hyperparameter setting found is:
{'alpha': 10, 'kernel': ExpSineSquared(length_scale=4, periodicity=1)}
```



Problem 9:

	method	train RMSE	validation RMSE	test RMSE
0	linear	11.74	12.08	12.31
1	RBF	4.14	3.72	4.22
2	ExpSineSquared	2.86	2.75	2.64