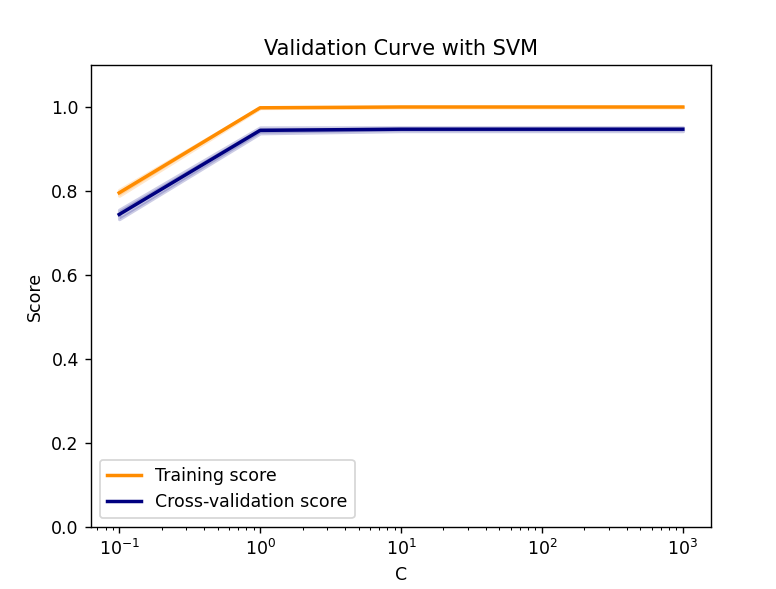
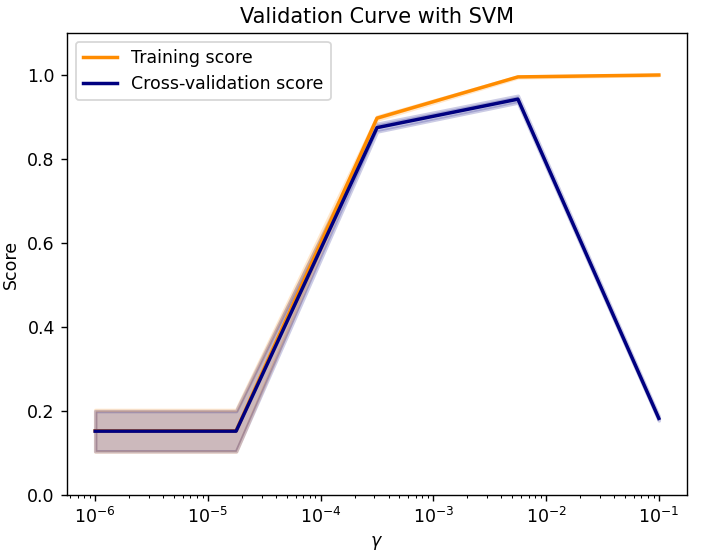
**Introduction**

**Method:**

**- SVM:**

In SVM when data are not linearly separable we are using the kernel trick which is transformation of data to higher dimensional space.



We can conclude that the best value for gamma is equal to 10^-2, because the cross-validation score is the biggest for this value. What can be observed from the first graph is that for values bigger than 10^-2 the score drops significantly, whereas in training scores it increases a little bit, this is the result of overfitting. For C values bigger than 1. the score stops increasing. This is because the bigger the C is the smaller the margin is, so for these values margin cannot be smaller compared to what was previously achieved.

**Results:**

| **classifier** | kmeans | linear regression | svm |
| --- | --- | --- | --- |
| **accuracy** |  |  | 0.962363 |

**Discussion/Conclusion:**

**- SVM:**

This algorithm is of much importance to tune hyperparameters as the classification is highly based on chosen values. Kernels like polynomial and rbf are useful in terms of non-linear hyperplanes, which are in higher dimensions. Penalty term could help us achieve an accurate tradeoff between margin and accuracy. Also different values of gamma make a big difference whereas low gamma's use only nearby points to calculate the separation line, the bigger one considers more of samples.

Advantages vs Disadvantages

SVM works well on data which can be clearly separated, also in high dimension, but its training is time costly. Because of that we should not apply it to large datasets. Also choosing a properly working kernel can be problematic and makes this algorithm harder to acquire for novices.

**Runtime analysis (in seconds):**

**- SVM:**

| **classifier** | kmeans | logistic regression | svm |
| --- | --- | --- | --- |
| **training** |  |  | 3.4276695999999998 |
| **prediction** |  |  | 0.5364348999999997 |