|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kernel** | **C** | **Gamma** | **Degree** | **Score (Class acc)** |
| Poly | 500 |  | 1 | .8563 |
| Poly | 500 |  | 3 | .8429 |
| Poly | 500 |  | 5 | .7708 |
| Sigmoid | 100 | .5 | 3 | .7596 |
| Linear | 500 | 1 | 3 | .8557 |
| Rbf | 100 | .5 |  | .8595 |
| Rbf | 10000 | 1/105 |  | .8678 |
| Rbf | 20000 | 1/105 |  | .8680 |
| Rbf | 500 | 1 |  | .8836 |
| Rbf | 500 | .5 |  | .8812 |
| Rbf | 500 | .05 |  | .8696 |
| Rbf | 500 | .01 |  | .8633 |
| Rbf | 500 | .005 |  | .8605 |
| Rbf | 500 | .0005 |  | .8545 |
| Rbf | 500 | 6 |  | .8855 |
| Rbf | 500 | 100 |  | .8938 |
| Rbf | 500 | 50 |  | .8940 |
| Rbf | 500 | 75 |  | .8928 |
| Rbf | 1 | 25 |  | .8753 |
| Rbf | 200 | 35 |  | .8942 |
| Rbf | 50 | 25 |  | .8894 |
| Rbf | 150 | 50 |  | .8941 |

Above are some of my tests that informed my final choice of parameter set. Within several tests I had decided on rbf, as it seemed to be both fast and have high accuracy values, as well as having the best documentation and information provided by the good people at scikit-learn.

My strategy for picking a good parameter set for rbf was to set the C value very high so as to fit the model extremely close to the training set so that it wouldn’t detract from the cross-validations accuracy when modulating gamma. Once I zeroed in on a good gamma, I planned on decreasing the C value to a high but reasonable value so as to “loosen” up the fit and prevent over-fitting from ruining my predictions. Eventually I settled on C=50 and Gamma=25. With a cv accuracy of .8894 even with a (as far as I can tell) reasonably low C value, it seems like a good compromise that performs well while avoiding overfitting too much.

This seemed to me to be a very high gamma, given that the default value is 1/#\_features, (1/105 for me), but I really don’t have much intuition on what constitutes a reasonable or unreasonable gamma, so I trusted scikit-learn when I read that one should check approximately from 10­-3 to 103 for most datasets (for C and gamma), so I took their word and my high accuracy results at face value.