February 13, 2018

**Lab Assignment 2**

Sanchit Singhal

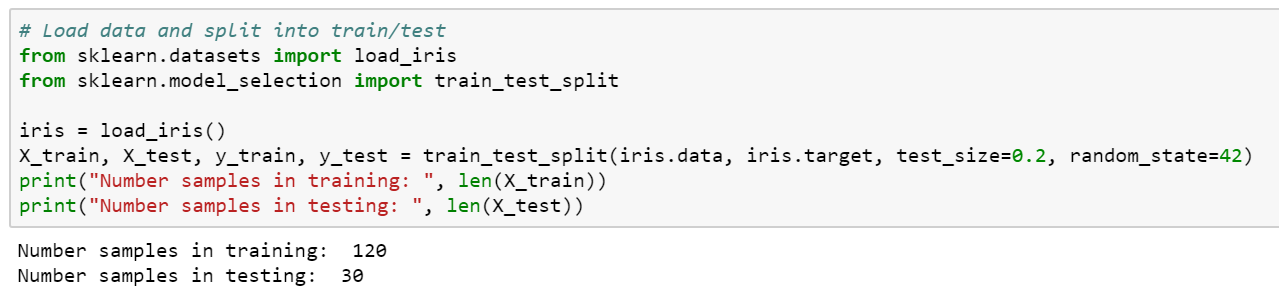
INF 385T – Introduction to Machine Learning with Danna Gurari

Spring 2018

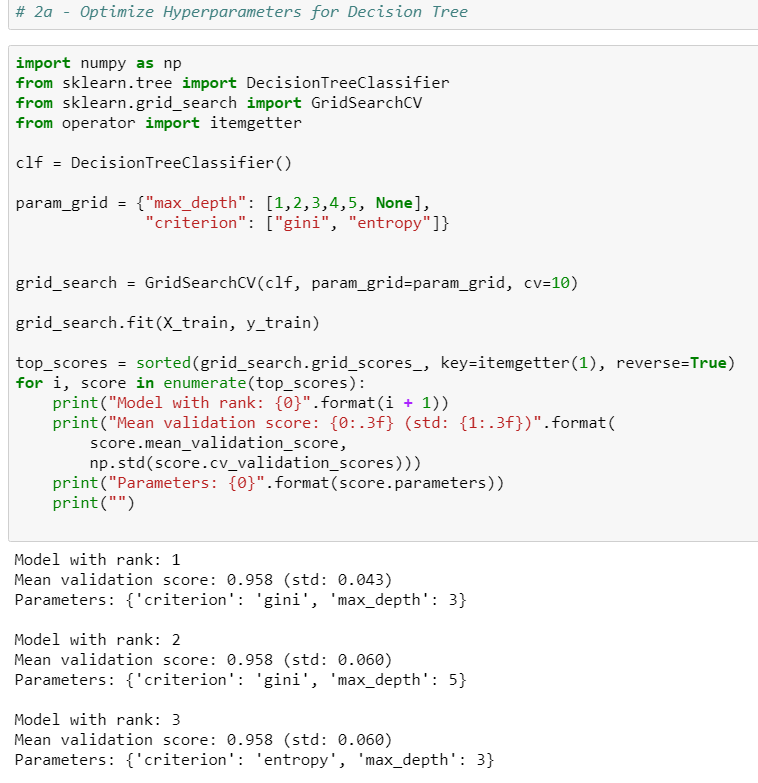
School of Information

**Hyperlink to code:** <https://introtoml-sanchit1276.notebooks.azure.com/nb/notebooks/IntroToML/LabAssignment2.ipynb>

1. **Construct Datasets for Training and Evaluation**



1. **Optimize Hyperparameters for Each Classification Model :**
2. **Decision Tree**



I tested 12 combinations of hyper parameters in total and the optimal ones were:

Criterion: Gini, Max\_depth: 3

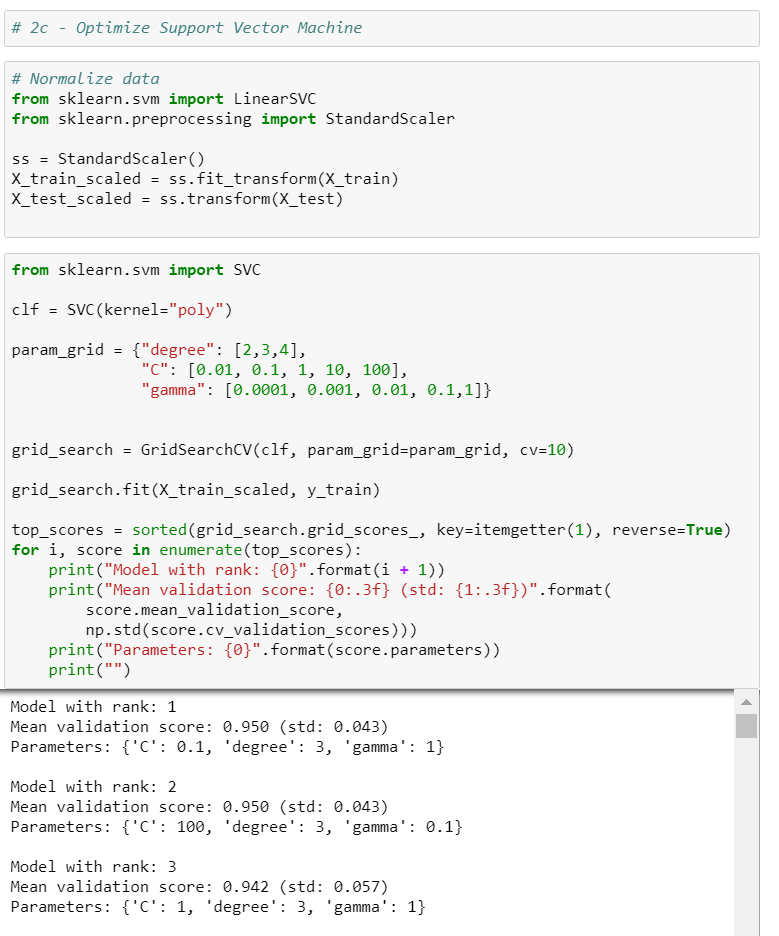
1. **K-Nearest Neighbor**



I tested 10 combinations of hyper parameters in total and the optimal ones were:

Metric: Euclidean, n\_neighbors = 1

1. **Support Vector Machine**

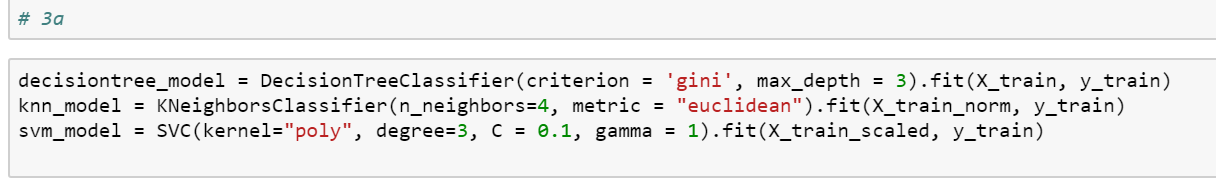


I tested 75 combinations of hyper parameters in total and the optimal were either of the two:

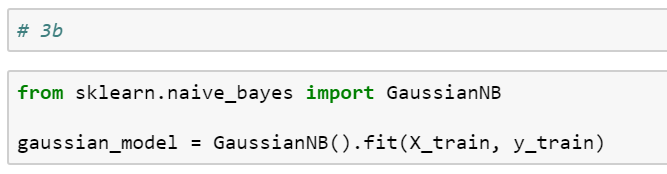
C: 0.1, degree: 3, gamma: 1

C: 100, degree: 3, gamma: 0.1

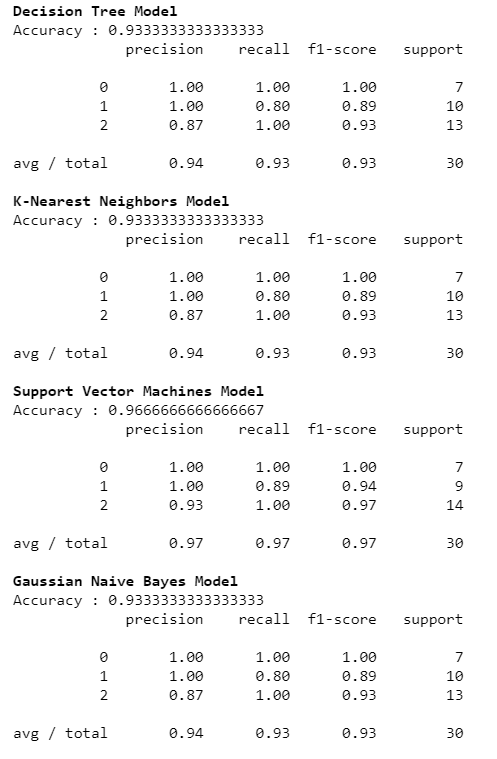
1. **Comparative Analysis of Optimized Classification Models**
2. **Retrain Decision Tree, K-NN, SVM**

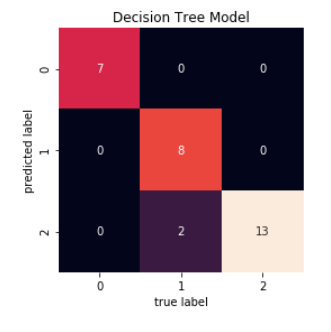


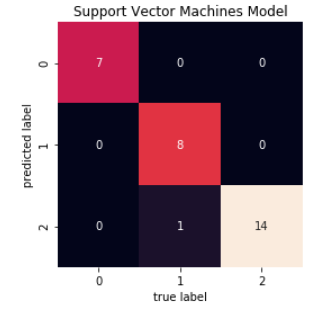
1. **Retrain Gaussian Naïve Bayes**

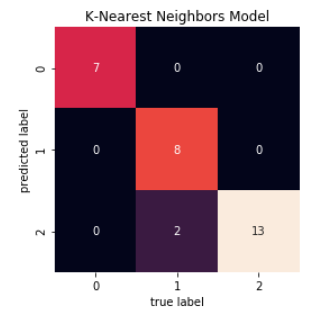


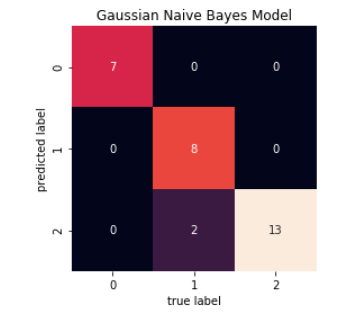
1. **Predictive Performance**



1. **Confusion Matrix**







1. **Analysis and Comparison of Performance**

The Support Vector Machine model performed the best on the test set while the Decision Tree was the worst. KNN was very close to SVM but slightly worse in recall. Both of these were better than the Decision Tree and Naïve Bayes models – although not by much. Changing the data set split state (during reruns of the code) alters the model performance slightly – presumably by how the data is initially split up.

The performance metrics tell us that all the models were very close to perfect with almost all the points falling on the diagonal center line. There do appear to be some records that were predicted to be in the level 2 but the true label was 1. SVM model was able to reduce this error and had one less data point in this bracket.

The algorithms performed the way they did because of how they are calculated. SVM’s are usually less vulnerable to outliers whereas KNN are more susceptible to irrelevant or noisy data. It is possible that the dataset contains a few bad data elements which causes the SVM model to perform slightly better.