Format:

****Name of Classifier**

Notes:

- notes

No NAN

Accuracy Score: [paste here, decimal form]

[Insert Confusion Matrix]

Filled Data

Accuracy Score: [paste here, decimal form]

[Insert Confusion Matrix]

*****PARAMETERS*****

train_test_split() → test_size = 0.3, random_state = 35

Random forest → n_estimators = 10, random_state = 0

KNN → n_neighbors = 7

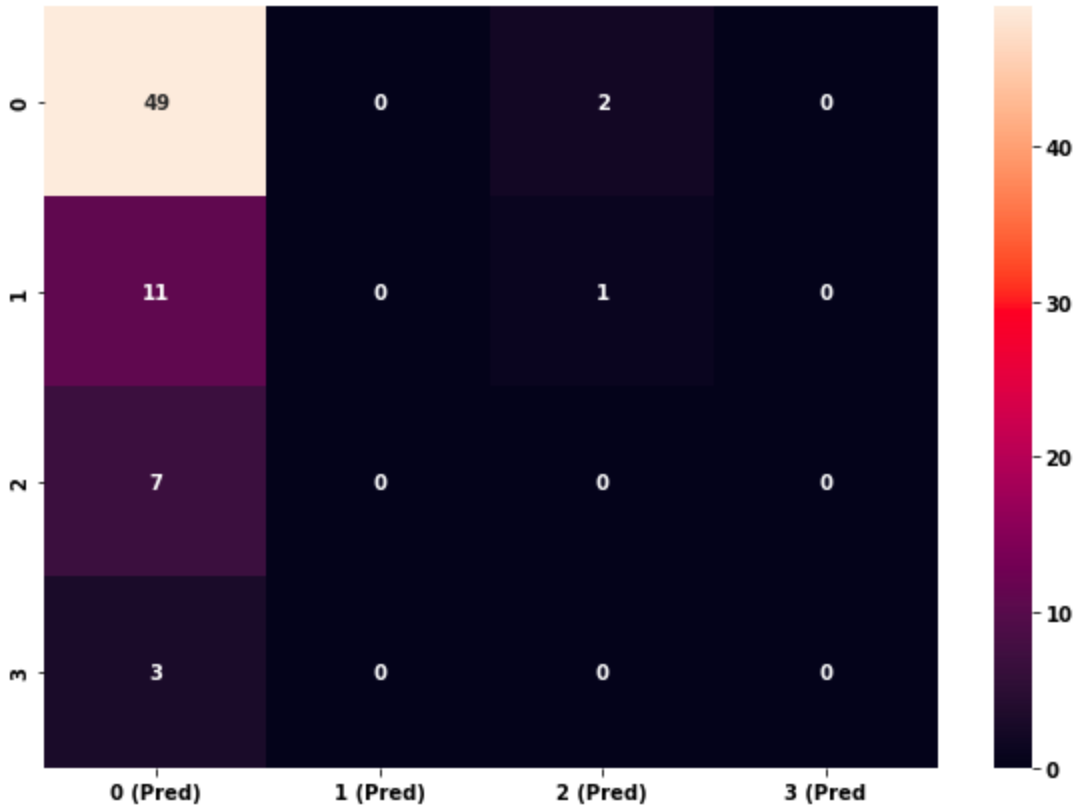
Decision tree → max_depth = 4, random_state = 35

Gaussian Process Classifier

Notes:

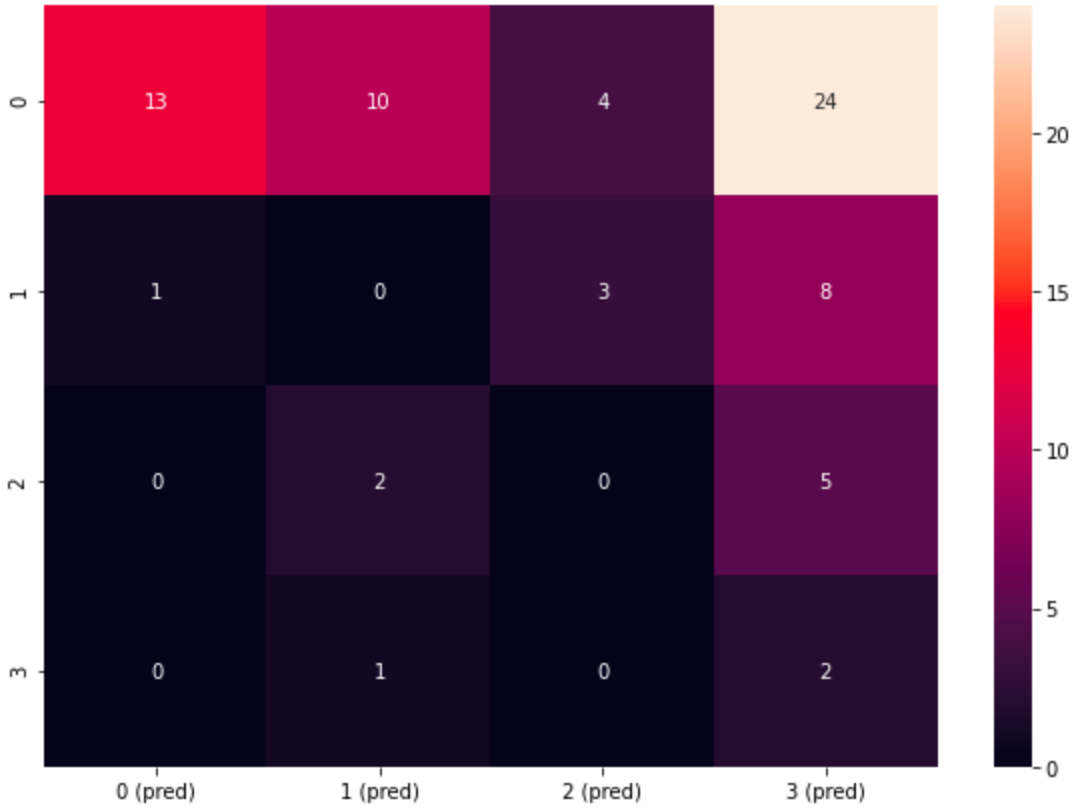
- special parameter: kernel = 1.0 * RBF(1.0) → "The kernel used for prediction. In case of binary classification, the structure of the kernel is the same as the one passed as parameter but with optimized hyperparameters. In case of multi-class classification, a CompoundKernel is returned which consists of the different kernels used in the one-versus-rest classifiers." (https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessClassifier.html#sklearn.gaussian_process.GaussianProcessClassifier)

Accuracy Score: 0.6900826446280992



Support Vector Machines

Accuracy Score: 0.698630

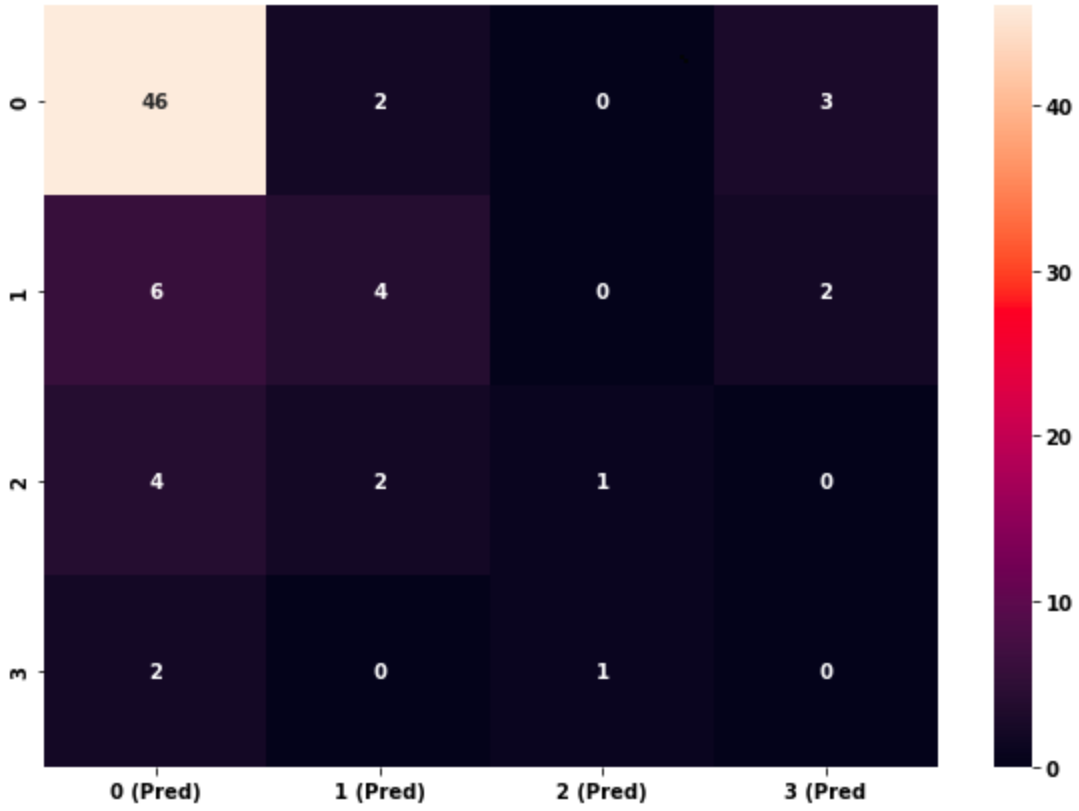


Decision Tree Classifier

Notes:

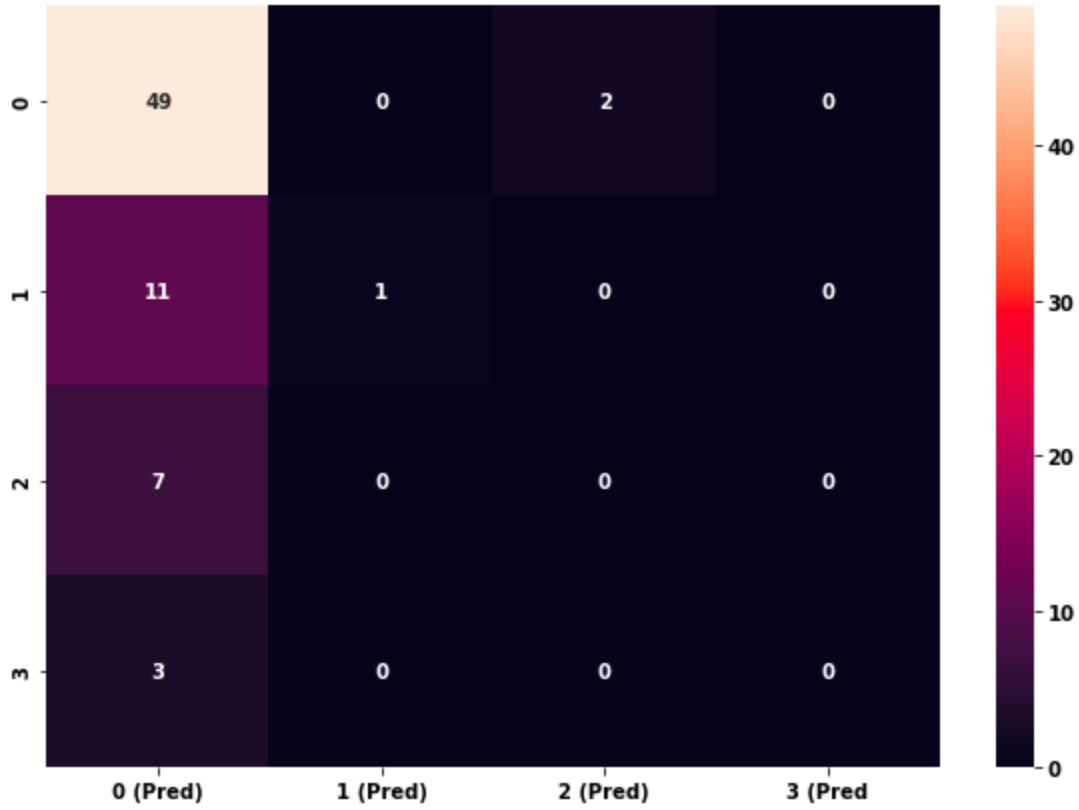
- max depth = 4

Accuracy Score: 0.6986301369863014



Quadratic Discriminant Analysis

Accuracy Score: 0.684931506849315

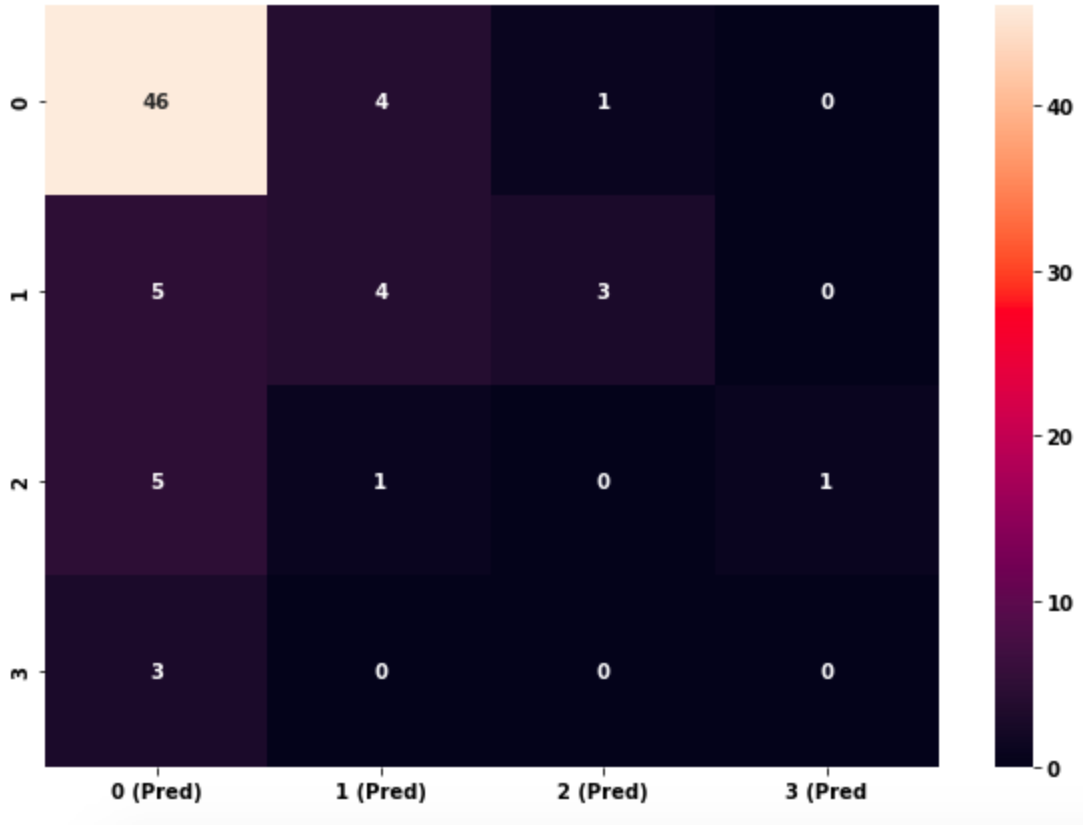


Multi-layer Perceptron Classifier

Notes:

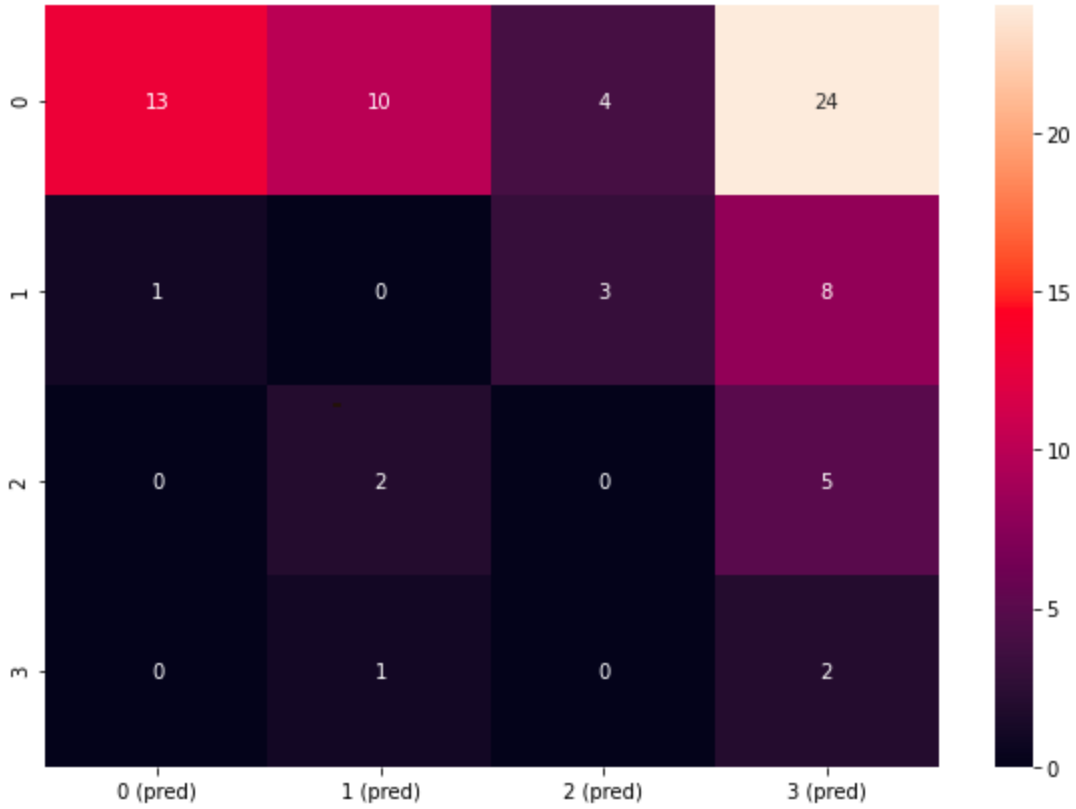
- max_iter = 100 → max_iter denotes the number of *epochs* → an epoch is one cycle through the full training dataset (https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/)
- 100 epochs got the highest accuracy score after some messing around w the parameter; 50, 200+ were in the lower 80% accuracy range

Accuracy Score: 0.684932



Naive Bayes

Accuracy Score: 0.2054794520547945

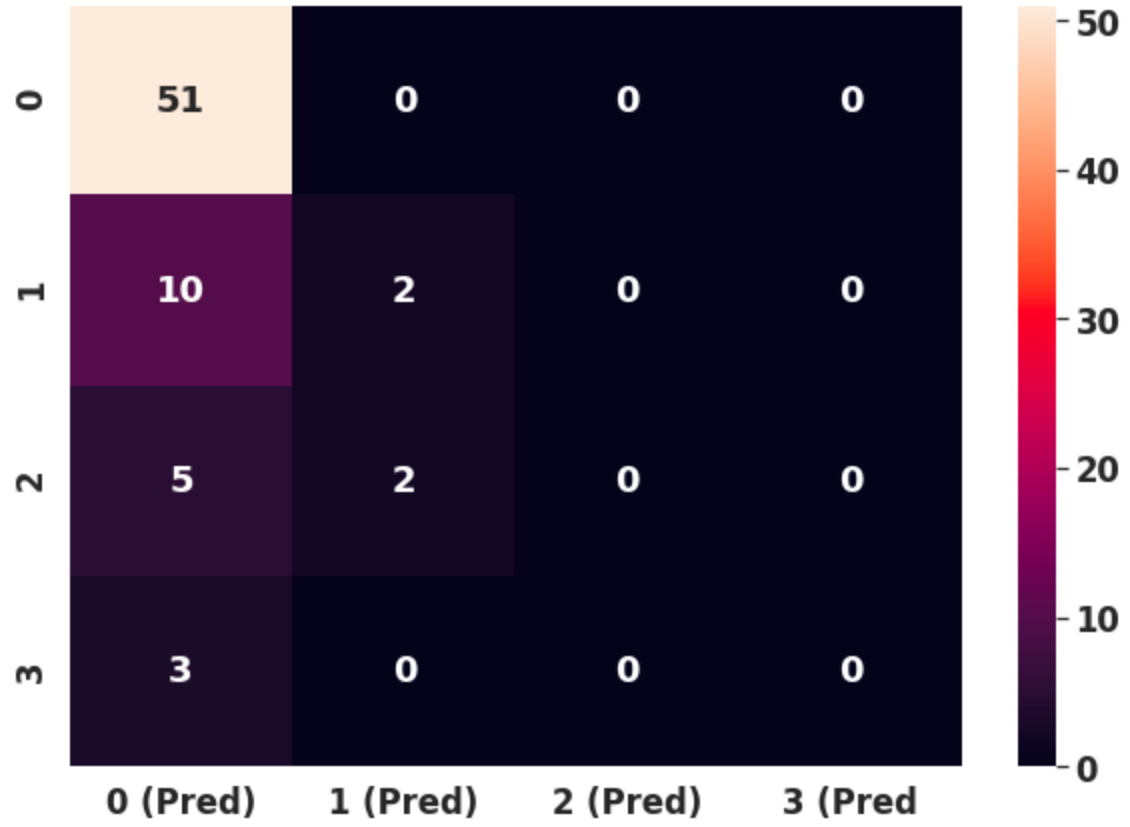


K-Nearest Neighbors

Notes:

- k = 5

Accuracy Score: 0.726027

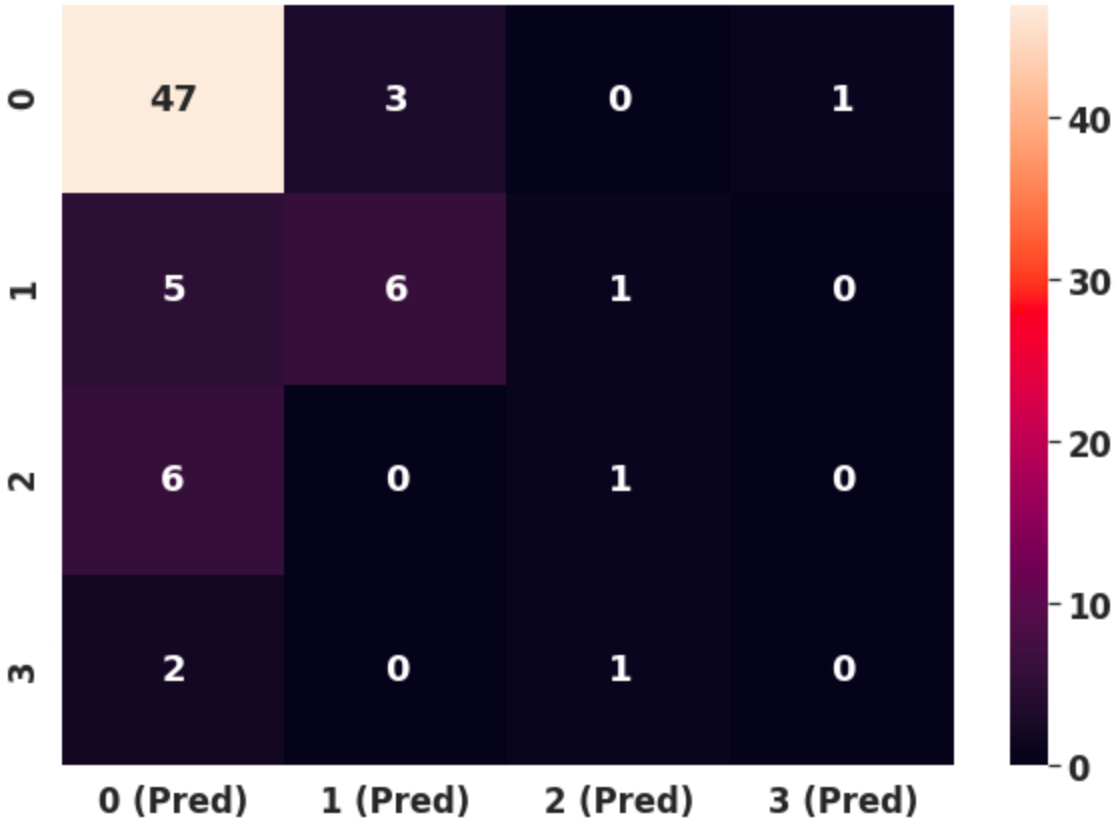


Random Forest Classifier

Notes:

- n_estimators = 10 → "the number of trees you want to build before taking the maximum voting or averages of predictions; higher number of trees give you better performance but makes your code slower" (https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/)

Accuracy Score: 0.739726



ROC/AUC Compilation!