(Diagonal) Linear Discriminant Analysis:

A simplified version of linear discriminant analysis (LDA) where the covariance matrices are assumed to be tied, or diagonal.

References:

- https://topepo.github.io/sparsediscrim/reference/lda_diag.html
- https://scikit-learn.org/stable/modules/lda_qda.html

For doing it in Python:

- https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html

Support Vector Machine:

Makes a hyperplane/a set of hyperplanes to make the class differentiation margin the largest, using kernel functions to classify all types of data (linear and nonlinear).

References:

- https://scikit-learn.org/stable/modules/svm.html

For doing it in Python:

- https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Nearest Centroid

An algorithm that represents each class by the centroid of its neighbors.

References:

- https://scikit-learn.org/stable/modules/neighbors.html
- https://www.sciencedirect.com/topics/computer-science/nearest-centroid#:~:text=Nearest%20Centroid%20is%20a%20classification,metric%2C%20such%20as%20Euclidean%20distance.

For doing it in Python:

- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestCentroid.html

K-Nearest Neighbors

Classifies a data point by finding the ‘k’ nearest training examples (k is a user given value) and assigning the most common class among the neighbors

References:

- https://scikit-learn.org/stable/modules/neighbors.html
- https://www.ibm.com/think/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20(KNN)%20algorithm%20is%20a%20non,used%20in%20machine%20learning%20today.

For doing it in Python:

- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Random Forest

Builds multiple decision trees (the forest!) using random subsets of features and samples. Then, it aggregates their prediction through voting to make sure overfitting isn’t occurring nad generalization is improved.

References:

- https://scikit-learn.org/stable/modules/ensemble.html

For doing it in Python:

- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html