AI Project

<http://scikit-learn.org/stable/supervised_learning.html#supervised-learning>

Part 2:

1. Since a dataset is being provided to learn first, this problem will be formatted as a supervised learning problem. The goal is to determine if a user’s application should be granted or denied based on different features. Therefor the problem is also a classification problem as there are two distinct states each observation should result in.
2. The first solution used is the k-nearest neighbors classifier using the kd-tree algorithm. Neighbors to check value (n\_neighbors) resulted in the highest accuracy for the given dataset when set to exactly 5.

The exact parameters used for KNeighbors was:

KNeighborsClassifier(algorithm='kd\_tree', leaf\_size=30, metric='minkowski', metric\_params=None, n\_neighbors=5, p=2, weights='uniform')

Data was tested using cross-validation with 80% of the sample data being used to train, and 20% used to test.

The resulting test completed with 86.57% accuracy. On a 6-core, 12GB RAM system, operational time to fit was ~0.0019 seconds, and ~0.003 seconds to test.

The second method used is a linear support vector machine

LinearSVC(C=1.0, class\_weight=None, dual=True, fit\_intercept=True,

intercept\_scaling=1, loss='squared\_hinge', max\_iter=1000,

multi\_class='ovr', penalty='l2', random\_state=None, tol=0.0001,

verbose=0)

The same data and training samples used for the KNeighbors classifier was used in fitting and testing the linear SVM.

Each run would largely vary the accuracy between 52 and 83.58%

On a 6-core, 12GB RAM system, operational time to fit was ~0. 04096seconds, and ~0. 001 seconds to test. This took longer to train, but faster to test.

[Kneighbors <http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier>]

[kd-tree <http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KDTree.html#sklearn.neighbors.KDTree>]