Report on Creating and Training models with and without PCA using [RandomForest, KNearest Neighbor, SVC] Classifiers on Stellar Classification Dataset

The dataset consists of 100,000 observations of space taken by the SDSS (Sloan Digital Sky Survey). Every observation is described by 17 feature columns and 1 class column which identifies it to be either a star, galaxy or quasar.

The attributes are as follows:

obj\_ID = Object Identifier, the unique value that identifies the

object in the image catalog used by the CAS

alpha = Right Ascension angle (at J2000 epoch)

delta = Declination angle (at J2000 epoch)

u = Ultraviolet filter in the photometric system

g = Green filter in the photometric system

r = Red filter in the photometric system

i = Near Infrared filter in the photometric system

z = Infrared filter in the photometric system

run\_ID = Run Number used to identify the specific scan

rereun\_ID = Rerun Number to specify how the image was processed

cam\_col = Camera column to identify the scanline within the run

field\_ID = Field number to identify each field

spec\_obj\_ID = Unique ID used for optical spectroscopic objects

(this means that 2 different observations with the same

spec\_obj\_ID must share the output class)

class = object class (galaxy, star or quasar object)

redshift = redshift value based on the increase in wavelength

plate = plate ID, identifies each plate in SDSS

MJD = Modified Julian Date, used to indicate when a given piece

of SDSS data was taken

fiber\_ID = fiber ID that identifies the fiber that pointed the light

at the focal plane in each observation

First we define the attributes/column names and read the csv file. Then we split the data into X and Y, where X is the attributes and Y is the classification answers.

Then we split the whole data into training data and test data. In this case, 80,000 rows are being taken as training data and rest 20,000 rows are used as testing data.

Then the data is preprocessed by using StandardScaler() function from sklearn module which standardizes features by removing the mean and scaling to unit variance.

We then use Random Forest Classifier to train and test the model, and the performance is given below:

Time Taken: ~39sec

Confusion Matrix:

[[11679 138 34]

[ 263 3571 1]

[ 8 0 4306]]

Accuracy: 97.78%

The same was done for K-Nearest Neighbor and the performance is:

Time Taken: ~13sec

Confusion Matrix:

[[11424 122 305]

[ 670 3127 38]

[ 899 9 3406]]

Accuracy: 89.785%

And in case of SVC:

Time Taken: ~2min

Confusion Matrix:

[[11469 135 247]

[ 393 3432 10]

[ 92 0 4222]]

Accuracy: 95.615%

After getting the results before dimensionality reduction, we then apply PCA and reduce the dimensionality. Performance after PCA is as follows;

Random Forest Classifier:

Time Taken: ~8sec

Confusion Matrix:

[[11543 11 297]

[ 1636 1918 281]

[ 2839 6 1469]]

Accuracy: 74.65%

K Nearest Neighbor:

Time Taken: ~11sec

Confusion Matrix:

[[10536 945 370]

[ 2036 1537 262]

[ 1725 558 2031]]

Accuracy 70.52%

SVC:

Time Taken: ~11min 30sec

Confusion Matrix:

[[11603 248 0]

[ 3394 441 0]

[ 4179 135 0]]

Accuracy: 60.22%

Hence we can conclude from the performance result that dimensionality reduction using PCA helped in using less resource and time, even though there was a slight compromise in the accuracy.