**Background**

**Random Forrest**

Random forest is an ensemble learning algorithms and is one of the best known classifiers. It is essentially an average of tree estimators.

Process used to build random forest:

* Create Bootstrap samples of data
* Fit a decision tree to different bootstrap samples
* Grow tree: select a random sample of m < p predictors to consider in each step
* This leads to different (or “uncorrelated”) trees from each sample
* average the prediction of each tree

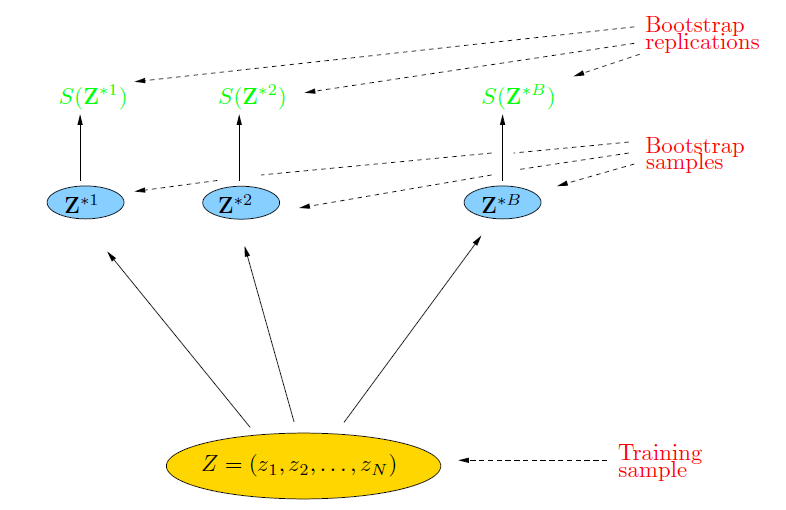
**Bootstraping**

The bootstrap is a method for estimating the variance of an estimator and for finding approximate confidence intervals for parameters. Although the method is nonparametric, it can also be used for inference about parameters in parametric and nonparametric models.

Efron’s insight was that we can simulate replication. we have fitted a model to the data, which is a guess at the mechanism which generated the data. Running that mechanism generates simulated data which, by hypothesis, has the same distribution as the real data. Feeding the simulated data through our estimator gives us one draw from the sampling distribution; repeating this many times yields the sampling distribution. Since we are using the model to give us its own uncertainty, Efron called this “bootstrapping”

the bootstrap seeks to estimate the conditional error ErrT , but typically estimates well only the expected prediction error Err.The bootstrap is broadly applicable and can be used to estimate the SE of a wide variety of statistics including linear regression coefficients, model predictions, principal component loadings

We denote the training set by Z = (z1, z2, . . . , zN) where zi = (xi, yi). The basic idea is to randomly draw datasets with replacement from the training data, each sample the same size as the original training set. This is done B times (B = 100 say), producing B bootstrap datasets. Then we refit the model to each of the bootstrap datasets, and examine the behavior of the fits over the B replications.



**Ensemble Learning**

Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. In contrast to ordinary machine learning approaches which try to learn one hypothesis from training data, ensemble methods try to construct a set of hypotheses and combine them to use.

An ensemble contains a number of learners which are usually called base learners( weak learners eg. naive bayes, log regression). Weak learners have low variance but high bias. The generalization ability of an ensemble is usually much stronger than that of base learners. ensemble learning is appealing because that it is able to boost weak learners which are slightly better than random guess to strong learners which can make very accurate predictions. Base learners are usually generated from training data by a base learning algorithm which can be decision tree, neural network or other kinds of machine learning algorithms. Most ensemble methods use a single base learning algorithm to produce homogeneous base learners, but there are also some methods which use multiple learning algorithms to produce heterogeneous learners.

Constructing Ensembles

an ensemble is constructed in two steps:

First, a number of base learners are produced, which can be generated in a parallel style or in a sequential style where the generation of a base learner has influence on the generation of subsequent learners.

Then, the base learners are combined to use, where among the most popular combination schemes are majority voting for classification and weighted averaging for regression.To get a good ensemble, the base learners should be as accurate as possible and as diverse as possible. Accuracy can be tested using cross-validation. diversity of the base learners can be introduced from different channels, such as subsampling the training examples, manipulating the attributes, manipulating the outputs, injecting randomness into learning algorithms, or even using multiple mechanisms simultaneously

* Classifiers that are most “sure” will vote with more conviction
* Classifiers will be most “sure” about a particular part of the space

**Principal Component Analysis**

PCA is useful for feature reduction, especially when using a large number of features may cause overfitting of the model to the training data. In such scenarios it is better to use PCA which gives a lift of principal components in order of the magnitude of variance they account for.

Principal component 1 (PC1) accounts for the largest variance in the model and should account for majority of the overall variance for PCA to be useful.

**Methodology**

The data is read into the python notebook from the excel file.

The data is in the following order: ["timestamp", "magneto\_x", "magneto\_y", "magneto\_z", "gyro\_x", "gyro\_y", "gyro\_z", "accel\_x", "accel\_y", "accel\_z", "label"]

After initial processing and cleaning, the random forest algorithm is used.

The results of using the Random Forrest Algorithm in 3 different conditions are compared:

* Magneto meter data only
* Magneto meter data with cross validation applied
* Sensor data from 3 different sensors including Magnetometer, Gyroscope and Accelerometer

To explore the results further, Principal Component Analysis is done with the goal to identify the correlation of the variance in data with the label. K-means clustering is presented with respect to the principal components to show that the data can be used to cleanly classify the data into different application groups.

**Results**

Model score for no of trees 1 is : 0.9921514887255513

Model score for no of trees 2 is : 0.9927743864457456

Model score for no of trees 3 is : 0.9972592500311449

Model score for no of trees 4 is : 0.9966363523109505

Model score for no of trees 5 is : 0.9980067272953781

Model score for no of trees 6 is : 0.998131306839417

Model score for no of trees 7 is : 0.998131306839417

Model score for no of trees 8 is : 0.9976329886632614

Model score for no of trees 9 is : 0.9987542045596113

Model score for no of trees 10 is : 0.9983804659274947

Model score for no of trees 11 is : 0.9983804659274947

Model score for no of trees 12 is : 0.9988787841036502

Model score for no of trees 13 is : 0.999003363647689

Model score for no of trees 14 is : 0.9991279431917279

Model score for no of trees 15 is : 0.9986296250155724

Model score for no of trees 16 is : 0.9992525227357668

Model score for no of trees 17 is : 0.9987542045596113

Model score for no of trees 18 is : 0.9991279431917279

Model score for no of trees 19 is : 0.9991279431917279

Model score for no of trees 1 is : 0.9970099667774086

Model score for no of trees 2 is : 0.995265780730897

Model score for no of trees 3 is : 0.9977574750830565

Model score for no of trees 4 is : 0.9967607973421927

Model score for no of trees 5 is : 0.998421926910299

Model score for no of trees 6 is : 0.9975913621262459

Model score for no of trees 7 is : 0.9990863787375416

Model score for no of trees 8 is : 0.9982558139534884

Model score for no of trees 9 is : 0.9988372093023256

Model score for no of trees 10 is : 0.9988372093023256

Model score for no of trees 11 is : 0.998671096345515

Model score for no of trees 12 is : 0.9990863787375416

Model score for no of trees 13 is : 0.9990033222591362

Model score for no of trees 14 is : 0.9988372093023256

Model score for no of trees 15 is : 0.9988372093023256

Model score for no of trees 16 is : 0.9989202657807309

Model score for no of trees 17 is : 0.9987541528239202

Model score for no of trees 18 is : 0.9989202657807309

Model score for no of trees 19 is : 0.9990863787375416

**Principal Component Analysis**

Model score for no of trees 1 is : 0.8348331131883373

Model score for no of trees 2 is : 0.8372729255834628

Model score for no of trees 3 is : 0.8505535212715015

Model score for no of trees 4 is : 0.8409837816493313

Model score for no of trees 5 is : 0.8372209764803678

Model score for no of trees 6 is : 0.8449726430024052

Model score for no of trees 7 is : 0.8513257127459786

Model score for no of trees 8 is : 0.8492069133894182

Model score for no of trees 9 is : 0.8475632924196637

Model score for no of trees 10 is : 0.8422054126729763

Model score for no of trees 11 is : 0.847687005785206

Model score for no of trees 12 is : 0.8533430816409423

Model score for no of trees 13 is : 0.8495798386990596

Model score for no of trees 14 is : 0.8553367641928034

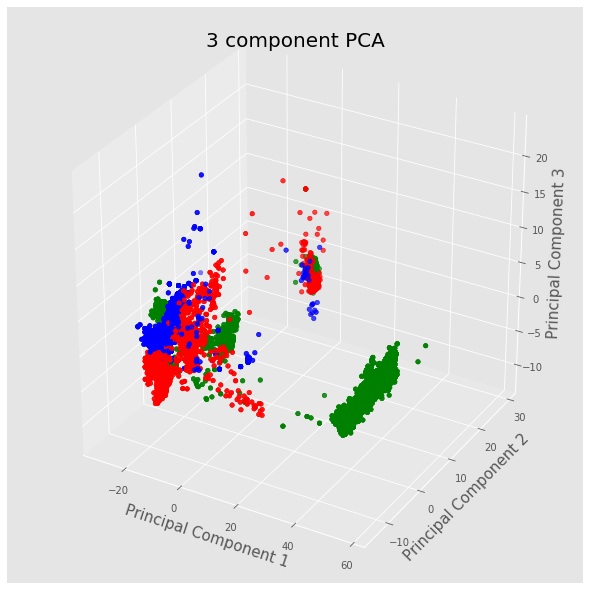
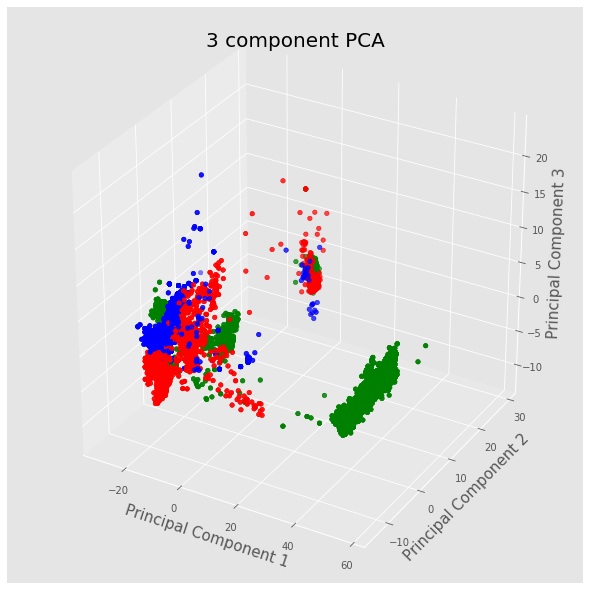
Model score for no of trees 15 is : 0.8471135363656239

Model score for no of trees 16 is : 0.8497307373760152

Model score for no of trees 17 is : 0.8521728969685742

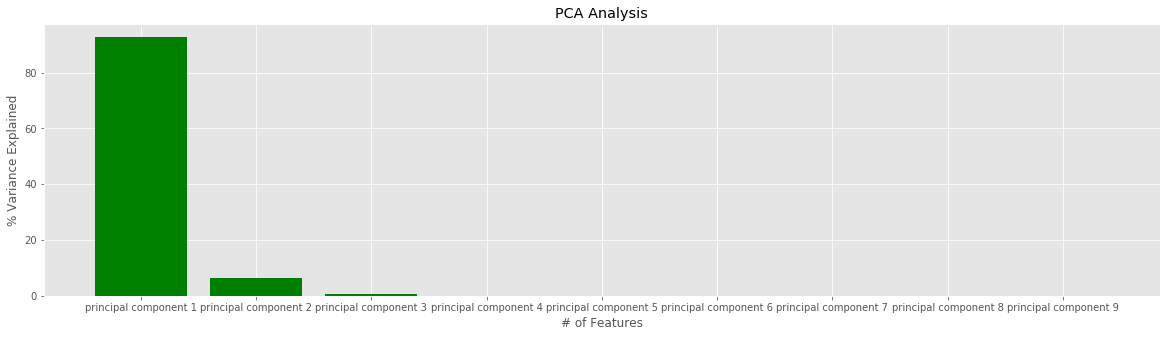
Model score for no of trees 18 is : 0.8449454206107034

Model score for no of trees 19 is : 0.8469144417951714

****

**Left: PCA analysis of magnetometer data only**

**Right: All 3 sensors used**

****

The graph shows that PC1 accounts for 92.7% of the variance in the model and thus can be very useful in classifying the data. The classification process can be seen below in the k-means models.

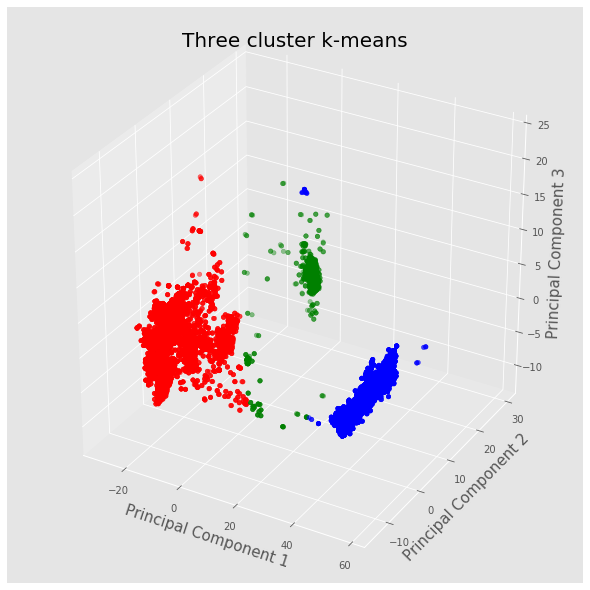
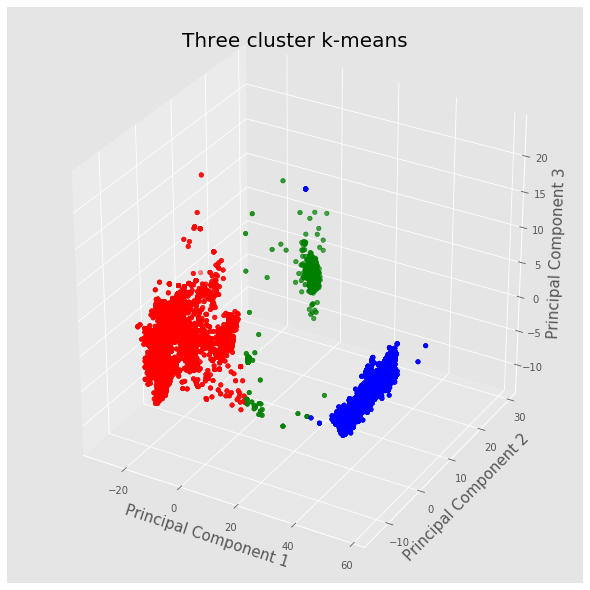
****

Figure 1: k-means on PCs calculated on magnetometer data

Figure 2: K-means on PCs calculated on all 3 sensors