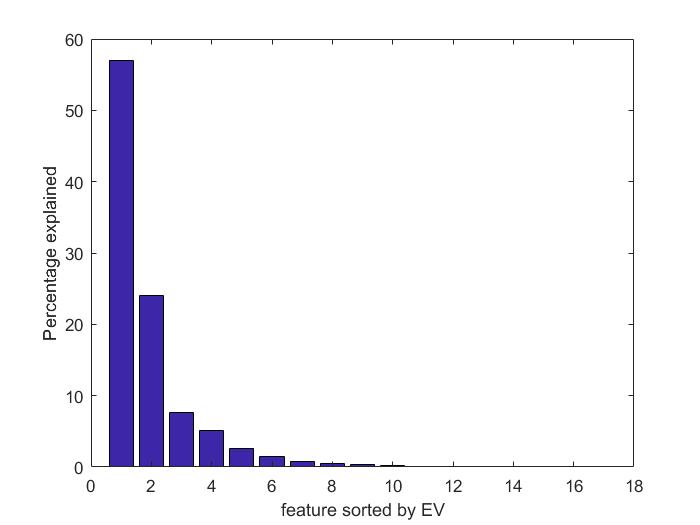
PCA first 3 PCA

Milestone 3 – Dimensionality Reduction (PCA)

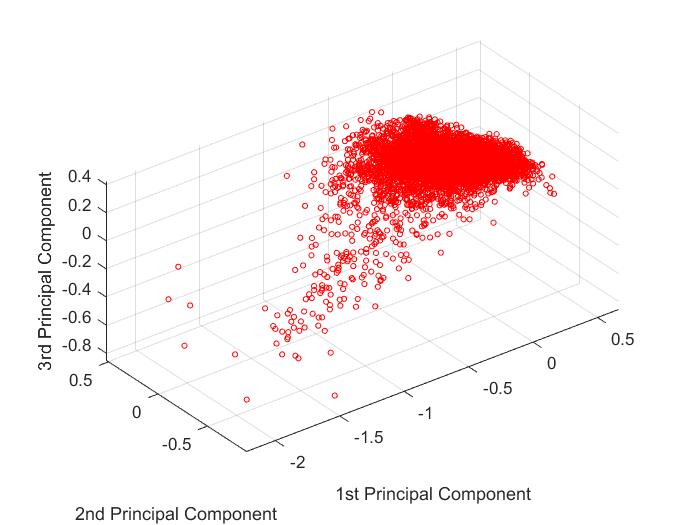
Each feature of the data is first scaled to with formula

Perform the PCA analysis on the data. Below is the graph of the percentage of variance explained by each feature in descending order.

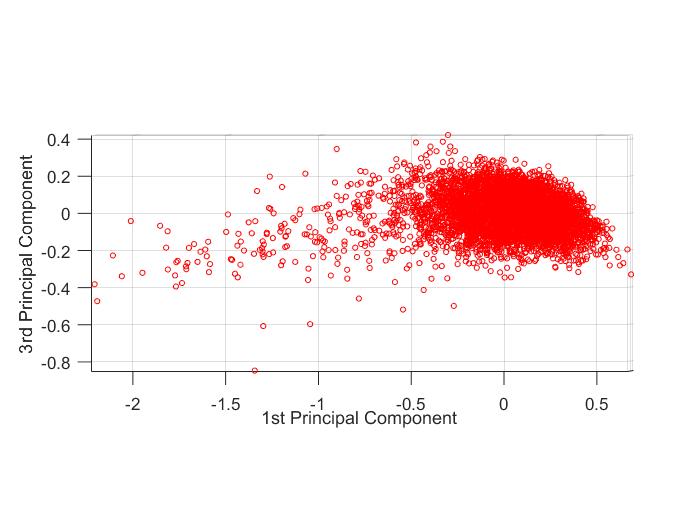
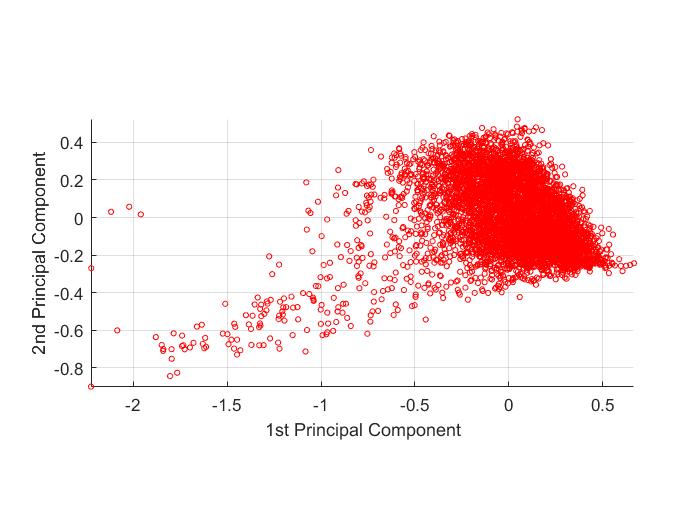
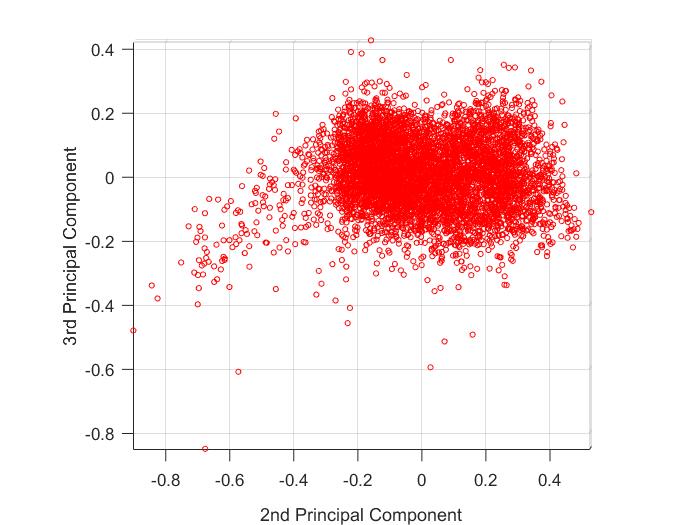


The first three principal components explain 57.04%, 24.04% and 7.6% of the data variance respectively and in total cover the 88.67% of the variance.

The data is visualized to observe the distribution in relate to the principal eigenvalues. The first graph below presents the scatter data points plotted in the first three principal components coordinates and the distribution is non-linear.



Viewing from sides to form a two dimensional graph, the data is still more close to a cluster instead of linear. (The original documentation of the data set on the website suggests that this particular data is mostly studied for linear regression, therefore it is expected that the data would display linear correlation to some extent; however this is not the case with principal component coordinates)



Training efficiency comparison with linear regression

The first model training is with the data of 16 features. It takes around 20 seconds to finish the gradient descent algorithm. The training with data of 3 principal components is much faster. As soon as the “train” button is clicked, the training finishes, which is ~1 second.

There is a slightly increase in the training error. The full data set produced 60.3566 least squared error for the motor score label while the reduced training data produced 65.5012 least squared error. For the total score label, the training with full dataset produced 104.3331 least squared error while the reduced data produced 113.8450 least squared error.

Motor score label training error: 60.3566 -> 65.5012

Total score label training error: 104.3331 -> 113.8450

This is reasonable because when we reduce the dimensionality to the major three components, only around 88% of the information is reserved. With 12% of the data information lost, there is the accuracy loss in the model training.

Training efficiency comparison with Gaussian processes

Unlike to linear regression, Gaussian processes do not benefit very much from the reduced dataset. The training time for either full dataset or reduced dataset both fluctuates around 1 minute.

In most cases, the training error suffers when the dimensionality is reduced. Below is the training output for Gaussian processes with different kernel functions. The numbers marked in red are error measure with the left being produced with full dataset and the right being produced with reduced dataset. In all cases the error becomes larger when the dimensionality is reduced, except for training with squared exponential kernel and total score, the training loss (least square error (not the negative log likelihood)) actually decreased from 58 to 53 with the reduced dataset, which is possible since the feature discarded might just be noises.

Round 1 without cross validation

Training with squared exponential kernel

Negative log likelihood is -6739.7306 -> -6952.3651

The training loss is 36.8477 -> 53.2991

...

Training with exponential kernel

Negative log likelihood is -6760.3959 -> -6981.5661

The training loss is 22.4103 -> 44.4686

...

Training with matern 3/2 kernel

Negative log likelihood is -6744.0904 -> -6967.5319

The training loss is 31.3263 -> 51.4816

Training with Total Score

Round 1 without cross validation

Training with squared exponential kernel

Negative log likelihood is -6739.7306 -> -6990.2324

The training loss is 58.0305 -> 53.3257

...

Training with exponential kernel

Negative log likelihood is -6760.3959 -> -6967.0961

The training loss is 36.0366 -> 47.3598

...

Training with matern 3/2 kernel

Negative log likelihood is -6744.0904 -> -6962.2244

The training loss is 49.0757 -> 50.5256