Wet section

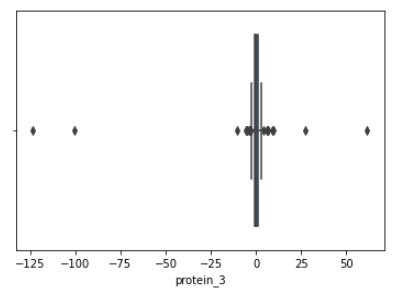
Since this work was done in pairs, many independent methods were used.

# Data Preprocessing

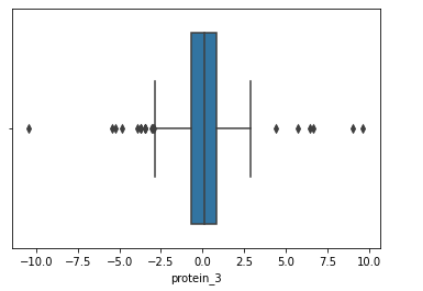
We have noticed that data has a lot of outliers and a lot of missing data. So the following steps were taken to deal with it, without losing much information (without simply dropping rows with missing data or outliers):

1. For each feature, outliers were detected using the z-score equal 2 (empirical number)
2. The mean was calculated on the remaining feature values
3. The outliers and the missing values were replaced with the means

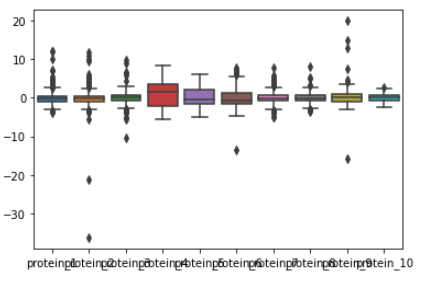
We can see the results visually. Box plot of the distribution of one of the features before the preprocessing was applied:



And after:



And the distribution of all the features (after):



# Addressing the ML tasks

We have examined the existing Clustering ML approached, chose 3 models we want to use based on their diversified approached and the fact that all of them were taught in the lectures. We evaluate the performance of each clustering method using Silhouette metrics. Later, we use different methods to choose the most important features.

This is the short examination:

## Pros/Cons of various Clustering Methods

K Means Clustering

Pros

* Easy to implement.
* Guaranteed convergence.
* Scales to large datasets

Cons

* a negligent edge of each cluster, because the priorities are set on the center of the cluster, not on its borders. (Because optimizing k-means equivalent to optimizing MLE for mixed Gaussian model with covariance matrix )
* an inability to create a structure of a dataset with objects that can be classified to multiple clusters in equal measure.
* a need to guess the optimal *k*number, or a need to make preliminary calculations to specify this gauge.
* NP-hard.
* Handling of outliers.
* Requires initialization of centers.
* Difficulty clustering data of varying densities and sizes
* Curse of dimensionality- with increasing amount of features the distances between points obtain a lower ratio of std. variation to mean.

Mixed Gaussians Model

Pros

* Unlike the centroid-based models, the EM algorithm allows the points to classify for two or more clusters – it simply presents you the possibility of each event, using which you can conduct further analysis.
* The borders of each cluster compose ellipsoids of different measures unlike k-means, where the cluster is visually represented as a circle.

Cons

* The algorithm simply would not work for datasets where objects do not follow the Gaussian distribution. It is more applicable to theoretical problems rather than the actual measurements or observations.
* Insufficient data leads to overfitting and difficulty estimating the covariance matrices which leads to divergence

Spectral Clustering

Pros

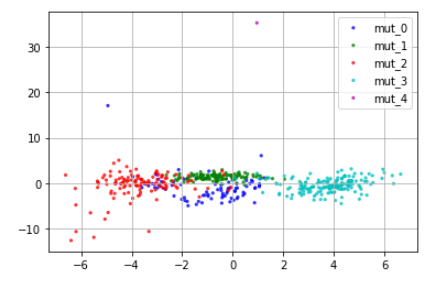
* Can recognize clusters that do not have a clear blob-shape, such as concentric rings.
* Polynomial time optimizable for fixed kfor cost function G\_cut (defined in lecture)
* Avoids curse of dimesionality- projects data onto lower dimensional space and then perform clustering using k-means (or other method)

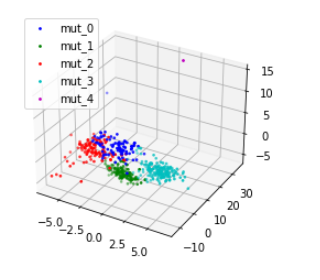
Cons

* tends to produce small clusters
* NP hard for cost function G\_cost\_cut for k=2 (defined in lecture)

## Clustering using KMeans

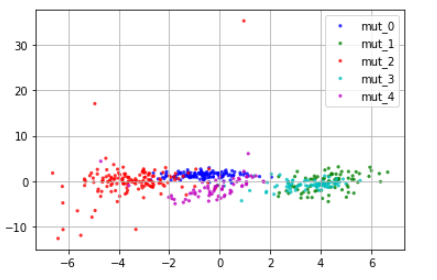
Using the first method, we cluster the data into 5 clusters. To observe the results visually (we will use this method a lot) we use the PCA, with 2 and 3 components. PCA makes each sample to contain only 2 or 3 PC, which retain the biggest amount of information from all the features, and allows us to plot them on a 2D, 3D graphs. Using the clusters labels on those samples we obtain:

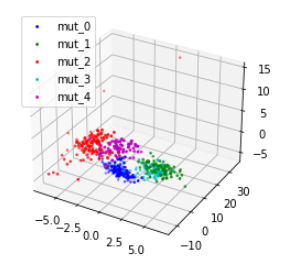




## Clustering using Spectral Clustering

Similarly, using Spectral Clustering (We use nearest neighbor as criteria for Affinity Matrix), we obtain:

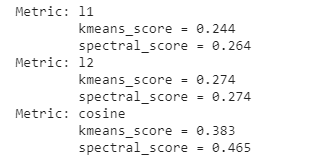




The colors are different, but we can see that in general the clusters are very similar, and visually we see indeed different group in both KMeans and Spectral Clustering algorithms.

## Comparing between KMeans and Spectral Clustering

We use the silhouette score to compare between the two. We use different distance metrics, such as L1, L2, or cosine (measures ‘angle’ between feature vectors projections). Results:



The cosine score seems to be much higher for the spectral. Other metrics seem to be similar

## Choosing the most dominant features

Several methods were used for this task.

### Intra-class Feature Similarity

NOTE – we didn’t use this method out of bad performance.

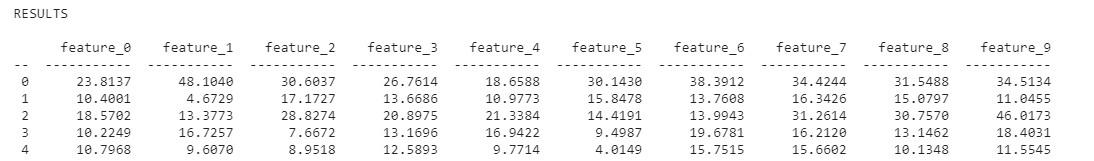
The base assumption was that the for a specific class and a specific feature, the distribution of the feature (if it’s an important feature for this class) will be close to the mean value of this feature. E.G for mutation no. 2, the protein 4 levels is important – so this feature levels for this class will be similar.

We have used the similarity function:

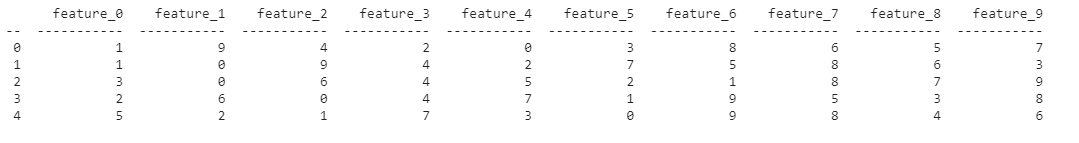
Where:

Results:

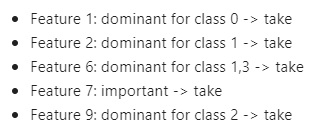
By using the value of 0.5, those are the results:



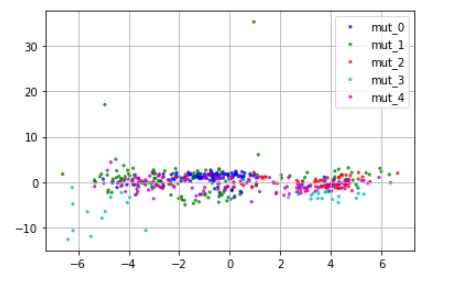
For each class, we reorder the features according to their similarity:



Analyzing:



To validate the results we leave only those features, and try to cluster the samples again. We get the following visualization:



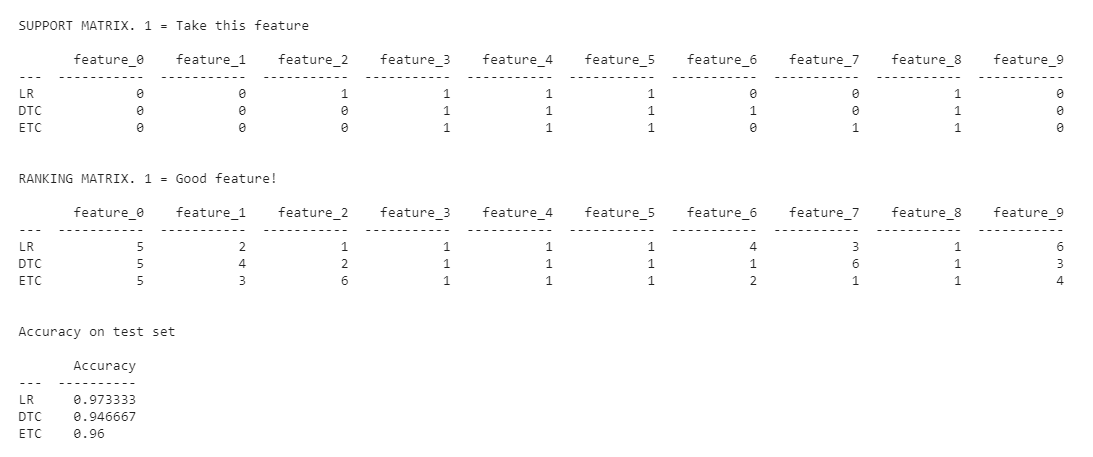
Which looks bad. From here, we have left this method and didn’t use its results further. It didn’t work, Perhaps out of erroneous assumptions on the nature

### Multiclass Classification

Another way to choose the best features is to train classifier on the labels that were created during the clusterin. We will train some classifier and detect the most dominant features it used for the classification. We will use the labels from the Spectral Clustering with 10 features.

First we use the RFE - Recursive Feature Elimination. It works by recursively removing attributes and building a model on those attributes that remain. It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute.

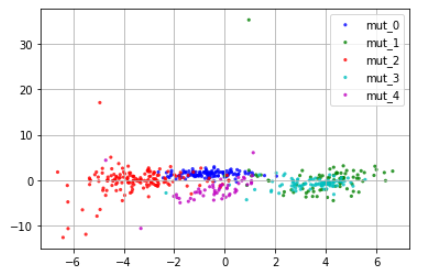
To diversify, we use 3 classifier models – Logistic Regression, DecisionTreeClassifier and ExtraTreesClassifier. Those are the results:

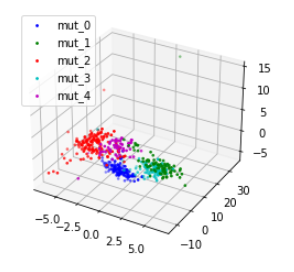


Features 3,4,5,7 agreed on all classifiers to be best features. Feature 6 is good feature for LR, which has highest accuracy score.

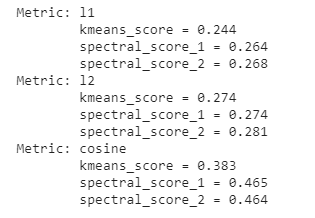
So from this analysis, best features are 3, 4, 5, 6, 7.

We make the visualization test again – I make clustering **using only those 5 chosen features**. The results are:





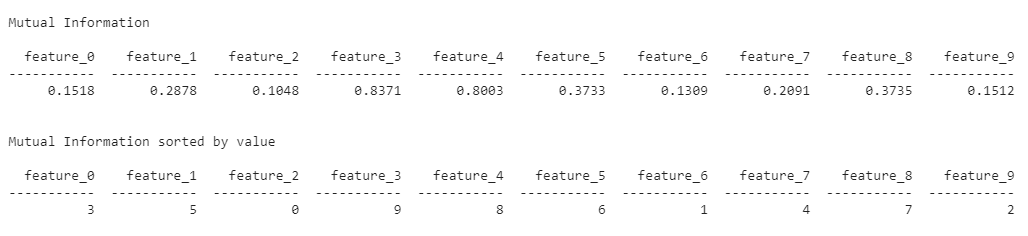
And it looks really good. So those features really are most important and contain most information for classification. But visualization is not enough, we also use silhouette score in this case. Spectral\_score\_2 applies to clustering using Specrtal Clustering, but only using 5 selected features:



We can see that the metrics didn’t get worse, and even improved a bit. This can be explained by removing features which did adversary effect on the clustering in the beginning.

### Mutual Information Score

To validate the previous method, we use the Mutual Score between the feature values and the classification. Those are the results:

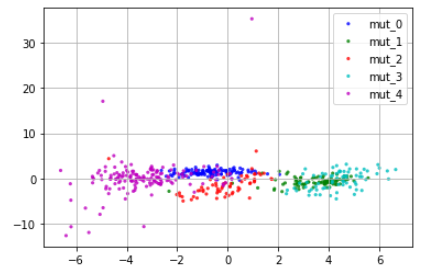


best\_features\_list:

[1, 5, 8, 4, 3]

Which are mostly (3/5) similar to the Features found by the RFE (3, 4, 5, 7, 8).

By using those features from MI, we can also see quiet a good classification visually:



### BEST FEATURES: CONCLUSION

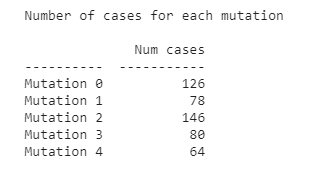
The most important features were chosen as the output of the RFE method, since it used various models, and used various metrics.

Best features:

(3, 4, 5, 7, 8).

## Choosing the 3 mutations to treat

The best way will be to treat the 3 mutations which are the most prevalent in the dataset



We will choose to treat mutation 0, 1 and 3