Machine Learning Engineer Nanodegree¶

Capstone Project

Eoin Kelly

June 9th, 2019

I. Definition

Project Overview

This project is based upon a study by Smart et al. in 2011 [1] in St. Vincents Hospital in Dublin. It is a study into three different types of musculoskeletal back pains. These are "nociceptive," "peripheral neuropathic," and "central sensitization" pain in patients with lower back pain disorders. We are looking to find a way to diagnose which group these patients belong to based on what symptoms they are feeling using both clustering and classification methods.

Problem Statement

The problem with back pain is that it can be difficult to diagnose exactly what the root cause of the problem is. For neuropathic pain alone, clinical practice guidelines typically suggest that the prevalence of neuropathic pain in LBP is approximately 5%; however, some reports suggest that as many as 16–55% of patients with chronic LBP have possible neuropathic pain components *(R. Baron et al. 2016)*. [2]

If we are able to analyse a number of symptoms that our patients show, we can try to cluster them into groups and certain groupings we hope to show are part of different types of lower back pain. This would help doctors to diagnose the best treatments for their patients. We have the response variable for each of these patients so we will be able to measure how well our models have performed using some format of training, validation and testing models so as not to bias our research.

Once we have these clusters of symptoms, we will be using the classification method of learning to attempt to diagnose which type of LBP the patient is showing. The symptoms/inputs that we will be analysing are different symptoms resulting from back pain. We will be putting the binary yes/no answer to symptoms through our classifier algorithm and this will give us an output which is the type of LBP exhibited by our patient. We will then compare our classifications against the real symptoms which are also recorded in the dataset. This is therefore a supervised learning multi-class classification problem.

Metrics

Accuracy is the most general way to measure how well our model is defined at predicting our response variables from binary inputs. Accuracy is found using the following formula:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Dataset\ Size}$$

However as our dataset is unbalanced, we will also be looking at some other metrics for measuring the performance of our models. Specificity and sensitivity allow us to prioritise false positives or false negatives depending on the scenario we want. In the medical profession we prefer a method of recall, meaning that we minimise the false negative rate, and that is what we will mainly look to do here. The below two formulae are how we measure the sensitivity and the specifiity of our model.

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Specificity = \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

II. Analysis

Data Exploration

The dataset we are using contains data acquired by doctors and physiotherapists examining patients with lower back pain. There are 464 subjects in this study. It contains the group of back pain the patient belongs to, which is our response variable and can be any of the below categorical responses:

1. Nociceptive
2. Peripheral Neuropathy
3. Central Neuropathy

Each patient is also asked whether or not they have each of 36 different symptoms, described below, and all of these answers are recorded. We will be using this data to try and predict the group of back pain that the patient hasbased on what his/her symptoms are. The symptoms analysed leave us a binary yes/no set of data points regarding the following 36 LBP symptoms:

1. Pain of recent onset.
2. Pain associated with trauma, pathology, and movement.
3. History of nerve injury, trauma.
4. Pain disproportionate to injury, pathology.
5. Intermittent and sharp or constant dull ache.
6. More constant, unremitting.
7. Burning, shooting, sharp, electric-shock like.
8. Localised to area of injury/dysfunction.
9. Referred in dermatomal, cutaneous distribution.
10. Widespread, non-anatomical distribution.
11. Mechanical nature to aggs + eases.
12. Mechanical pattern associated with movement, loading, and compression of neural tissue.
13. Disproportionate, non-mechanical pattern to aggs + eases.
14. Spontaneous, paroxysmal pain.
15. Pain with dyesthesias.
16. Pain of high severity and irritability.
17. Pain with neurological symptoms.
18. Night pain, disturbed sleep.
19. Response to simple analgesia, NSAIDS.

20. Rapidly resolving, resolving with expected tissue healing, pathology recovery times.
21. Pain persisting beyond expected tissue healing, pathology recovery times.
22. History of failed interventions.
23. Strong association with maladaptive psychological factors.
24. Pain with high levels of functional disability.
25. Antalgic postures, movement patterns.
26. Consistent, proportionate pain reproduction on mechanical testing.
27. Pain, symptom provocation with mechanical tests that move, load, compress neural tissue.
28. Disproportionate, non-mechanical pattern of pain provocation on mechanical testing.
29. Positive neurological findings.
30. Localised pain on palpation.
31. Diffuse, non-anatomic areas of pain on palpation.
32. Positive findings of allodynia.
33. Positive findings of hyperalgesia.
34. Positive findings of hyperpathia.
35. Pain, symptom provocation on palpation of neural tissues.
36. Positive identification of psychosocial factors

A quick analysis of the data shows us that there are 44 null values present in the data. This would imply the patient was not checked for these symptoms or was not able to answer confidently whther they had this type of symptom or not. For this reason I am going to fill in these values with 0, indicating 'no', they do not have this symptom.

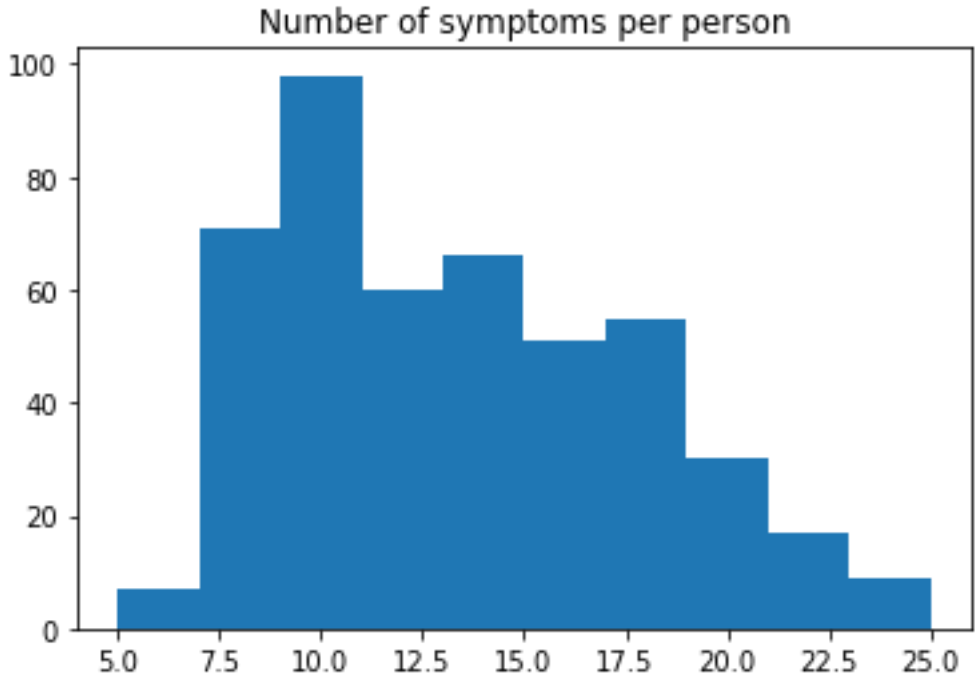
Exploratory Visualization

A further look into the format this data takes shows that the dataset is not balanced:

1. Nociceptive patients = 256 = 0.5517
2. Peripheral Neuropathic patients = 102 = 0.2198
3. Central Neuropathic patients = 106 = 0.2284

This means we will likely be leveraging specificity and sensitivity measures in analysing how well our models have performed in the results section.

The number of symptoms displayed by a patient is displayed in a histogram below



It is vital to recognise that most patients exhibit a large number of these symptoms at once with the majority of patients having between 6 and 22 symptoms. We will be looking to find patterns within this set.

Algorithms and Techniques

1. Clustering

First up, we will look to cluster our symptoms into groups in which patients show many of these at once. This will give us an insight as to whether or not we will be able to group together the symptoms with a view to classifying them into the overall pain group they patient is in. We will be using the k-means clustering algorithm to do this.

K-Means clustering is an iterative, partitioning clustering algorithm that aims to divide n objects in the data into K disjoint groups which minimizes the within group sum of squares. It is a simple and computationally efficient algorithm. It's formula is:

$$SS = \sum_{k=0}^{K} \sum_{j=0}^{n} \sum_{i \in C_k} (X_{ij} - \bar{X}_j^{C_k(i)})^2$$

where $C_k(i)$ is the cluster that observation i belongs

To begin the algorithm, the data are partitioned into K different clusters.

1. Next, we calculate the centroid of each cluster.

2. For each data point, the distance to each cluster centroid is computed.
3. Each observation is then reassigned to the cluster whose centroid it is closest to.
4. We recalculate the centroid for each new cluster and repeat steps 1 - 4 until convergence.

The algorithm is said to have converged once the cluster memberships do not change for any point. Note that we can get caught in local minima here so it is wise to run this test more than once to find our best or most common clusterings.

2. Classification

i. Decision Tree

Decision trees can provide highly understandable outputs. We can usually visualise them to see exactly what route something takes in getting classified. It is this simplicity in visualisation that makes these an extremely powerful tool in machine learning. It will be an easy way for doctors to understand the output and apply this to their patients in making their diagnosis.

We will be using the in-built DecisionTreeClassifier from sklearn to implement this. This basically finds the best tree by fitting the data to it. It finds the best feature to split the data at each partition and follows this algorithm to decide on how a new observation should be classified.

ii. Random Forest

A random forest is simply a collection of decision trees whose results are aggregated into one final result. They often make a better generalised solution than a decision tree for the following reasons.

1. **Reduces Overfitting** by averaging its results over several trees.
2. **Reduces Variance**: Reduces chances of a bad performance in classification of sample of data due to the relationship between the test and training sets.

Their ability to limit overfitting without substantially increasing error due to bias is why they are such powerful models. The original basic construction of a random forest is as below. We have used a similar method to this.

1. Take a bootstrap sample of the data.
2. Select a random sample of the data variables (default number is pp)
3. Build a classification tree for this smaller dataset.
4. Test how well the method works on the Out of Bag (OOB) samples.
5. Repeat B times (default B = 500).

iii. Adaboost

I also want to check out how the Adaboost classifier performs on this dataset to see the differences between it and our previous two classifiers. The adaboost classifier is a boosting algorithm whereas our random forest is a bootstrapped algorithm. Adaboost uses many weak learners to combine to make one strong classifier. Each of these weak learners are better at predicting than a random chance model.

It initially trains to the data using whichever properties work best. It will then apply a higher weighting to each of the misclassified points. From there it will now try to make the best classifier combining the scores of the higher weighted points and the initial points.

It does this iteratively to find many 'weak' models or learners. If we then combine these models, we create our one strong learner.

Benchmark

The benchmark model for this project has been decided to be the logistic regression multi-class classifier. It will provide us a baseline to see how well our models actually perform on the data. However as it transpires this logistic regression does exceptionally well on this dataset with an accuracy of 93%.

As this is the case I have also looked at other supervised models on the data, including the K-nearest neighbors algorithm. This algorithm gave me a 87% accuracy rating, so I have decided that this will be our benchmark model. The multinomial model will be discussed later in this report as a result.

KNN Model

The K-Nearest Neighbors algorithm bases itself upon the proposition that in a dataset, an observation can be classified based upon the observations it lies beside or has similar characteristics to. If we choose K of these nearest neighbors, a majority vote between all of these neighbors decides which group our observation will be assigned to.

III. Methodology

Data Pre-processing

As we can see from the data exploration section there were a number of null values in the dataset and as I had explained there, I have filled those values in to be 0. We need to do this as our classifier's will not work if these values are nulls. Outside of this there is no other required pre-processing for this dataset.

I had considered encoding each of the categorical variables but as I have gone through the dataset, I did not see it as being required for any of my calculations. With regard to our input data, this is binary and so does not need normalised or anything applied to it.

Implementation

Clustering: To implement the k-means algorithm required first finding the optimal number of clusters present. To do this we used the elbow method. This involved running the K-means algorithm over a number of possible values for k. We then plot each value of k, against the within sum of squares as detailed in the *Analysis, Clustering* section. We look for the elbow in the graph and this will give us the optimal number of clusters.

We would then analyse more in-depth on the optimal k-values to see which fits to our data the best. We would look at silhouette scores and also check the Adjusted Rand Index for each of these values of k.

Classification: For the classification section, the first and possibly the most important part of this section is to make sure that we split our dataset into a training set and a testing dataset. This removes any bias that would be in our model's testing. That would arise from using the same data to create our model and then to test the same data which our model has already trained on. Giving inaccurate biased results.

From here we will fit each of our classifiers using their corresponding algorithms within the loaded python libraries. We will fit the data on the training set of data, which we have set to be 75% of our set. Once we have fitted each model, we will assess how well this classifier has done by predicting the outcomes of the test dataset, and comparing them to the true results. We will choose our best model based on the results from the testing dataset checks.

One difficulty I had was in getting GridSearchCV to work on the classifiers to test over a number of hyperparameters at once for best results so I found another functon which kind of did the same thing for the RF and Adaboost algorithms.

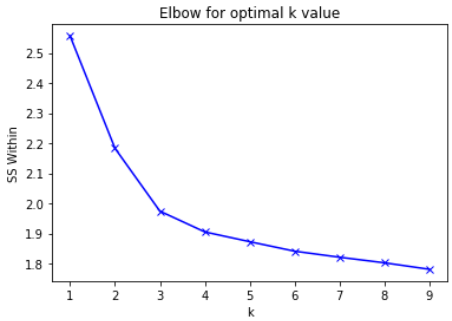
Refinement

Within the Random Forest section I checked for the best number of trees over multiple numbers and whilst there were a number of options that alll gave quite similar results, I decided to go with n=8 as the optimal number in this case. If we used more than this amount there was a larger possibility of overfitting or the model getting over complex. The max_depth of these trees were limited at n = 4 also. This is to prevent the tree from becoming too large and being allowed to separate every single observation in our dataset out and basically overfitting to the training set.

IV. Results

Model Evaluation and Validation

A. Clusters



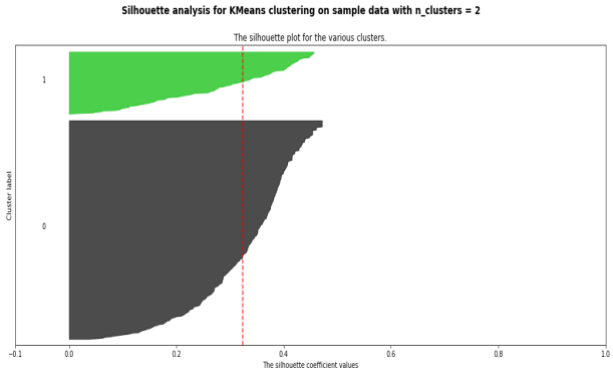
Adjusted Rand Index for different values of k

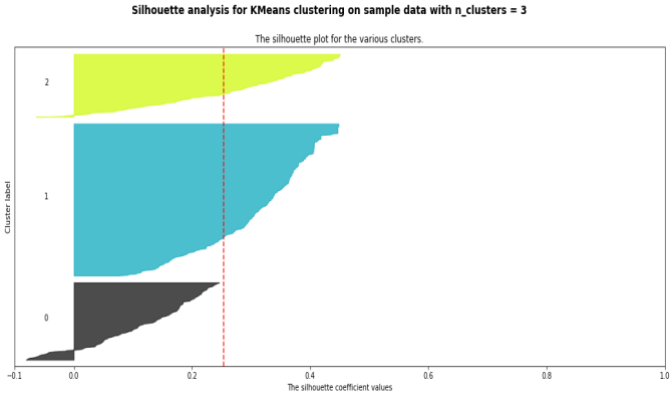
```
ARI for 1 groups = 0.0
ARI for 2 groups = 0.4597765735531884
ARI for 3 groups = 0.7775006133940894
ARI for 4 groups = 0.5413851097696029
ARI for 5 groups = 0.48992146061271186
ARI for 6 groups = 0.43020674529209924
ARI for 7 groups = 0.4072538600256582
ARI for 8 groups = 0.39858401111458813
ARI for 9 groups = 0.31385375153049566
```

Above we can see the elbow diagram used in finding our optimal value of k clusters. From this, we can see the elbow is around the 2,3 or 4 values of k. So I analysed all these three values. The ARI was strongest for k=3, however this is quite expected since we are comparing it to three groups.

Looking further into this, we see the silhouettes below and it makes me understand that the best value for number of clusters in this dataset is actually two. This means to me that one of the clusters is easily distinguished by solely clustering the symptoms whilst the oter two are a little loser together. The silhouette scores below show the average value for how tight each cluster is grouped together. The values range from -1 to 1 with 1 being god and 0 showing no correlation. Negative numbers are bad and show that some points may be better off in another group. With k = 4, the average was way off and I have not included that in this report.

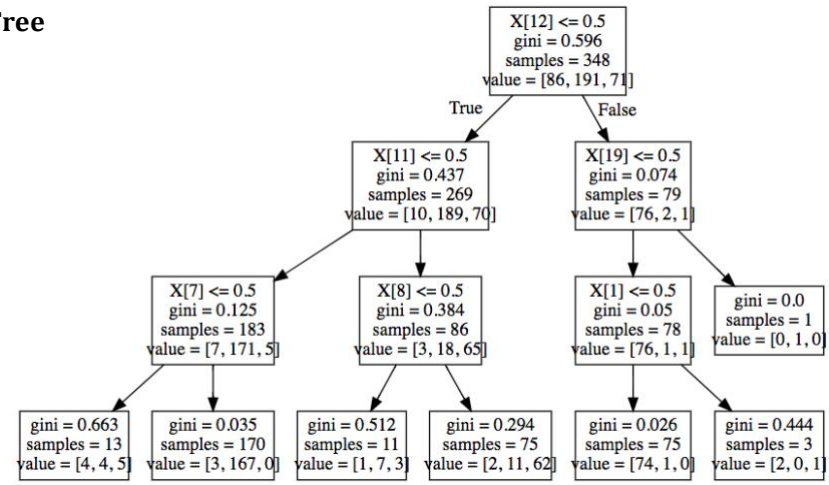
Average value for k=2 was 0.323 vs for k=3 the value was 0.253 meaning two is the optimal number of clusters in this dataset.

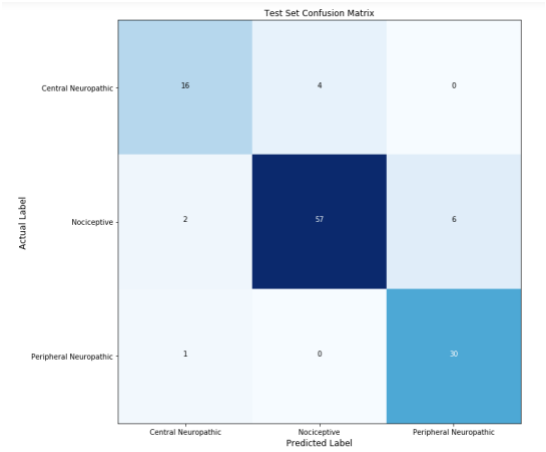




B. Classification

i. Decision Tree





	precision	recall	f1-score	support
Central Neuropathic	0.84	0.80	0.82	20
Nociceptive	0.93	0.88	0.90	65
Peripheral Neuropathic	0.83	0.97	0.90	31
avg / total	0.89	0.89	0.89	116

The three images pictured above are taken from my notebook. The top one is a diagram showing how the Decision Tree looks like and how it comes to make the decision as to which group a new observation should be placed.

The second diagram is a confusion matrix and shows how well our model fits to the real data. This image is gotten from running the test data through our algo to gain predictions. You can see that it performs quite well. To see just how well, we will look to the right of this diagram.

Recall is the statistic we are looking to maximise here. Looking at this we can see that our model is very good at finding patients with 'Peripheral Neuropathic' back disorder, whereas finding 'Central Neuropathic' is a lot more difficult.

ii. Random Forest

All we will analyse here is the recall score that our random forest gives us. You can see here that it is much higher and better at predicting our back pain. All three of the values within are well found and in the 90% success rate. This improves upon the above Decision Tree mainly due to the fact that any overfitting will have been removed from the decision tree as this random forest is aggregated over a number of small samples of Decision Tree.

	precision	recall	f1-score	support
Central Neuropathic	0.79	0.95	0.86	20
Nociceptive	0.98	0.91	0.94	65
Peripheral Neuropathic	0.91	0.94	0.92	31
avg / total	0.93	0.92	0.92	116

iii. Adaboost

As can be seen from the result of this is slightly worse than that of the random forest above. This seems to be once again due to the 'Central Neuropathic' disease being difficult to diagnose. This shows Random Forest's in action as the reason this is likely to be difficult to diagnose is due to the training dataset overfitting on some aspects of the data. Random Forest is our best decision tree type classifier.

	precision	recall	f1-score	support
Central Neuropathic	0.81	0.85	0.83	20
Nociceptive	0.95	0.92	0.94	65
Peripheral Neuropathic	0.91	0.94	0.92	31
avg / total	0.92	0.91	0.91	116

iv. Multinomial Logistic Regression

I had not anticipated that the multi-class logistic regression classifier would be a great model but it has turned out to be tied with the random forest model. However in looking at where its accuracy comes from we can see that it seems to have a better understanding of the 'Nociceptive' group but worse on the other two.

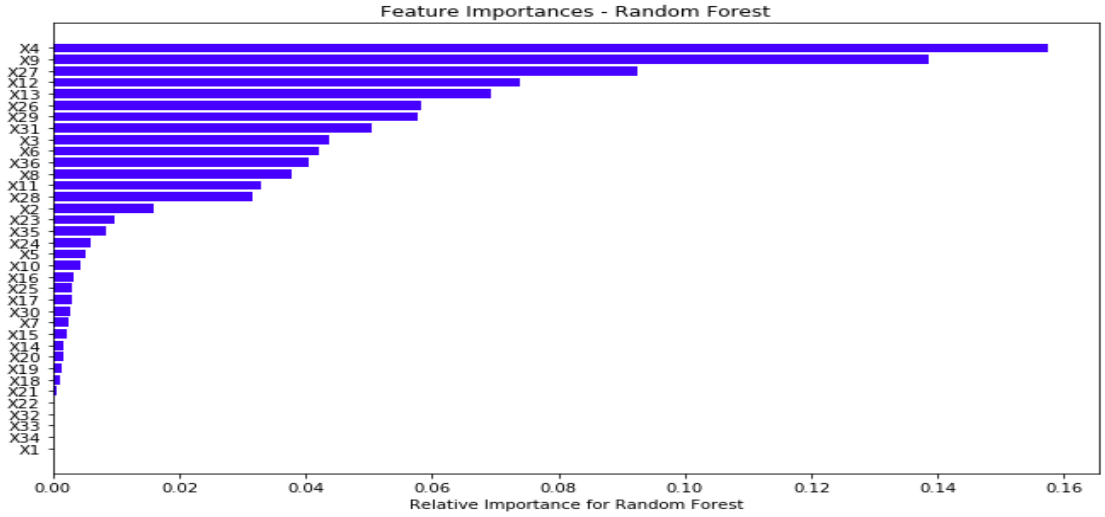
	precision	recall	f1-score	support
Central Neuropathic	0.82	0.90	0.86	20
Nociceptive	0.94	0.94	0.94	65
Peripheral Neuropathic	0.97	0.90	0.93	31
avg / total	0.92	0.92	0.92	116

Justification

The best model for this problem in my opinion is the Random Forest Classifier as it picks out the majority of patients' pains better than the other algorithms. It would be interesting to dive further into this study to see if there is a way to combine the Random Forest algorithm with the multinomial algorithm to see if we were able to improve again as the multinomial would help in diagnosing 'Nociceptive' pains whereas Random Forest classifier could diagnose the other pains.

V. Conclusion

Free-Form Visualization



The above diagram is a representation on how the random forest works. It shows the relative importance of each feature in how the classifier picks the next step. If we analyse this and look back to

our Decision Tree in Section IV. We notice that X11, X12 and X8 are showing as important in both. So we can see some similarities in that these three symptoms should show more weight by doctors in being able to properly diagnose what form of Lower Back Pain a patient is displaying.

Reflection

Initially I had intended a logistic regression model to be the benchmark model that I would compare against as I had mentioned in my proposal. This benchmark soon turned out to be a close second best to our random forest classifier. For this reason I pursued another supervised learning algorithm in the K-nearest neighbors algo. This provided me with a better benchmark as to how our results had actually improved against another similar classification model.

There were two main parts to this project, the unsupervised clustering algorithms and the supervised classification algorithms. With the clustering what we discovered through the silhouettes is that the optimal clusterings of the dataset, there were two. This to me shows that within the three groups that one is possibly easier to differentiate out than the other two which may have a number of similarities.

It is also cool to see the difference in accuracy of a random forest against a simpler model in the decision tree classifier. Just by taking an average over a number of decision trees, the random forest eliminates any bias or variance and actually makes the model so much more accurate. I say 3% is a lot more accurate because we are already at the 89% accuracy rate so any increments above this are difficult to find as the model is quite good anyway.

In terms of how well this actually works in real life sense of diagnosing patients, I think this will be a welcome help to all doctors involved in the area, but it should not be taken as the be all and end all. This model still has a 7% miss rate, meaning by following this policy exactly patients will still get misdiagnosed. This should be a good preliminary examination to carry out, but there still needs to be further checks carried out on the patient.

Improvement

One way that this project could have been improved is early on we could possibly use PCA to find the principal components in the data. This would allow us to remove most of the symptoms that are present in all or None of the pain groups. It would be slightly easier to analyse the data from that.

Another area of improvement is to see if we could someway manage to get a predictor created whereby in some cases we look at the multinomial model and in other cases the random forest. This would be a help because, even though the two models are similarly accurate, they are accurate in different areas. The multinomial shows good at predicting ‘Nociceptive’ pain and the random forest, if it was to struggle with any is this pain. The other two it is quite impressive at predicting with over 94% in each.

I also wonder whether we could run this data through a neural network to see if we could improve the results any way by doing so.

VI. References

[1] Keith M. Smart, PhD,* Catherine Blake, PhD,w Anthony Staines, PhD,z and Catherine Doody, PhDw - https://www.elleboogkliniek.nl/wp-content/uploads/Smart-et-al-The-Discriminative-Validity-of-Nociceptive-Peripheral-Neuropathic-and-Central-Sensitization-kopie.pdf

[2] R. Baron, Binder, N. Attal, R. Casale, A.H. Dickenson and R-D. Treede - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5069616/