Data Science MSc 2016/2017

Final Project

ARTIFICIAL NEURAL NETWORK MODELING IN PREDICTING 1ST EPISODE PSYCHOSIS ASSOCIATED TO CANNABIS USE.

Irene Volpe

INDEX

abstract

An extensive amount of information is currently available to clinical specialists, ranging from details of clinical symptoms to various types of biochemical data and outputs of imaging devices.
Each type of data provides information that must be evaluated and assigned to a particular pathology during the diagnostic process.

To streamline the diagnostic process in daily routine and avoid misdiagnosis, artificial intelligence methods (especially computer aided diagnosis and artificial neural networks) can be employed. These adaptive learning algorithms can handle diverse types of medical data and integrate them into categorized outputs.

In this paper, we propose a novel symbiotic machine learning and statistical approach to pattern detection and to developing predictive models for the onset of first-episode psychosis.

1) introduction

The relationship between cannabis use and psychosis has, in recent decades, become a focus of controversy. The National Institute of Mental Health has stated that “research has found increasing evidence of a link between marijuana and schizophrenia symptoms.” In a report issued in 2000, the National Academy of Sciences noted that some researchers had proposed a link between cannabis use and schizophrenia, as well as between cannabis use and a unique type of psychosis. They observed that “marijuana use alone—without the influence of additional risk factors—is unlikely to provoke a psychosis that persists longer than intoxication.” Likewise, a number of reviews have concluded that cannabis use only results in a significant increase in risk of psychosis when coupled with additional risk factors, in particular, an underlying genetic vulnerability.

However, there is still scope for further understanding of the links between patterns of cannabis use and psychosis.
The field of machine learning has advanced at tremendous pace in recent years, with advanced predictive techniques being developed and improved upon.
In this study, I propose a novel symbiotic machine learning and statistical approach to pattern detection and to developing predictive models for the onset of first-episode psychosis.

The dataset I based my study upon was collected in previously conducted medical studies and comprises a wide set of variables including demographic, drug-related, as well as several other variables with specific information on the participants’ history of cannabis use.

Prior to the prediction modelling, a significant effort was involved in the data pre-processing due to inherent challenges present in data collected in a case-control study involving many missing values, unbalanced number of records for each class, a significantly large number of variables, etc.

The prediction modelling phase consisted of investigating several machine learning techniques such as Support Vector Machine, Decision Trees, Neural Nets, Deep Neural Nets and Random Forests, whose predictive models have been optimised in a computationally intensive framework.

In the next chapter i’m going to explain how the dataset was collected, how the models I used, and those I haven’t, have been succesfully been used in the field of bioinformatics so far, with a particular attention to ANNs models; also a brief description of their working principles will be carried on within the chapter.

Chapter 3 will deal with the preprocessing part in a more deep way: each step will be accompanied with plots and summaries to discuss the results.

Chapter 4 will show the filtering processes with the aim of selecting different sets of predictors based on their importance in predicting the outcome; for each method different thresholds will be chosen.

Chapter 5 will focus on the technologies used and those that have not but could potentially improve the prediction performances.
Next is the conclusion together with the achievements, and finally the reference section.

2.1) the dataset

The data used to develop my novel approach to predicting the first-episode psychosis is a part of a case-control study at the inpatient units of the South London and Maudsley NHS Foundation Trust.

The clinical data consisted of 1106 records, including 489 patients, 370 controls and 247 unlabeled records. Those described as patients were patients of the Trust who at one time presented with first-episode psychosis; controls were recruited from the same local area by a dedicated research team.

Each record had 255 possible attributes, both demographic attributes and drug- related attributes.

In order to build my approach to predicting the first- episode psychosis, the data required a set of pre-processing transformations, including feature selection, data sampling, data type conversions as needed by training certain types of models, and missing value imputation. So the dataset I was finally given consists of 777 records and 29 variables: again both demographic and drug related ones.

The prediction modelling consisted of training different machine learning techniques, including k-Nearest Neighbours, Support Vector Machine, Decision Trees, Neural networks, Deep Learing and Random Forests.

The models were evaluated based on accuracy, area under curve, precision, sensitivity, specificity.

All experiments, including model training and optimisation based on repeated cross validations, were conducted in a computationally intensive framework, using R packages such as Caret.

2.2) Machine learning algorithms in Bioinformatics

Given the complexity and gigantic volume of biological data, the traditional computer science techniques and algorithms fail to solve complex biological problems of the real world. However, there are modern computational approaches called machine learning that can address the limitations of the traditional techniques.

Machine learning is an adaptive process that enables computers to learn from experience, learn by example, and learn by analogy. Learning capabilities are essential for automatically improving the performance of a computational system over time on the basis of previous results. A basic learning model typically consists of the following four components:

- learning element, responsible for improving its performance,
- performance element,which decides the choice of actions to be taken,
- critical element, which tells learning element how the algorithm performs, and
- problem generator, responsible for suggesting actions that could lead to new or informative experiences (Adeli, 1995; Finlay and Dix 1996;Kuonen, 2003; Narayanan et al., 2002; Negnevitsky, 2002; Nilsson, 1996; Baldi and Brunak, 2001; and Westhead et al., 2002).

Machine learning typically can be divided into three phases, as follows:
1. analysis of a training set of examples and generation of a set of rules from training set,
2. verification of the rules by human experts or automatic knowledge based components and
3. use of the validated rules in responding to some new testing datasets (Finlay and Dix 1996).

There are a number of reasons why machine learning approaches are widely used in practice, especially in bioinformatics (Narayanan et al., 2002; Nilsson, 1996; Baldi and Brunak, 2001; and Westhead et al., 2002):

- Traditionally, a human being builds such an expert system by collecting knowledge from specific experts. The experts can always explain what factors they use to assess a situation; however, it is often difficult for the experts to say what rules they use, for example, for disease analysis and control. This problem can be resolved by machine learning mechanisms. Machine learning can extract the description of the hidden situation in terms of those factors and then fire rules that match the expert's behavior.

- In molecular biology research, new data and concepts are generated every day, and those new data and concepts update or replace the old ones. Machine learning can be easily adapted to a changing environment. This benefits system designers, as they do not need to redesign systems whenever the environment changes.

- Missing and noisy data is one characteristic of biological data. The conventional computer techniques fail to handle this. Machine learning techniques are able to deal with missing and noisy data.

- With advances in biotechnology, huge volumes of biological data are generated. In addition, it is possible that important hidden relationships and correlations exist in the data. Machine learning methods are de signed to handle very large data sets, and can be used to extract such relationships.

- There are some biological problems in which experts can specify only input/output pairs, but not the relationships between inputs and outputs, such as the prediction of protein structure and structural and functional sequences. This limitation can be addressed by machine learning methods. They are able to adjust their internal structure to produce approximate results for the given problems.

Machine learning mechanisms form the basis of adaptive systems. In bioinformatics research, a number of machine learning approaches are applied to discover new meaningful knowledge from the biological data- bases, to analyze and predict diseases, to group similar genetic elements, and to find relationships or associations in biological data. Examples of machine learning approaches in bioinformatics research are shown in Table 1 below.

Research Area	**Application**	**Reference**
Sequence allignment	BLAST FASTA	https://blast.ncbi.nlm.nih.gov/Blast.cgi http://www.ebi.ac.uk/Tools/sss/fasta/
Multiple sequence allignment	ClustalW MultAlin DiAlign	http://www.ebi.ac.uk/Tools/msa/clustalw2/ http://www.sacs.ucsf.edu/Documentation/multalin-help.html#Introduction https://bibiserv.cebitec.uni-bielefeld.de/dialign/
Gene finding	GenScan GenomeScan GeneMark	http://genes.mit.edu/GENSCANinfo.html http://genes.mit.edu/genomescan/ https://www.ncbi.nlm.nih.gov/genomes/MICROBES/genemark.cgi
Protein domain analysis and identification	Pfam BLOCKS ProDom	http://pfam.xfam.org http://blocks.fhcrc.org http://prodom.prabi.fr/prodom/current/html/home.php
Pattern identification	Gibbs Sampler AlignACE MEME	http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-486 http://www1.spms.ntu.edu.sg/~chenxin/W-AlignACE/ http://meme-suite.org
Protein folding prediction	PredictProtein SwissModle	https://www.predictprotein.org https://swissmodel.expasy.org

2.3) Artificial Neural Networks

The process of learning is a complex phenomenon. Many puzzling questions arise from of it. How can one recognize the faces of others? How can one identify complex patterns from the faces? How does one discriminate images and backgrounds? How does one learn a shortcut to go to his or her university? In order to answer these questions, one needs to know how the brain works.

The human brain has been studied since the late Middle Ages; however, its detailed structure began to be unraveled only in the nineteenth century. Neuronists claim that the brain is a collection of about 10 billion densely interconnected cellular units called neurons. The structure of a neuron and its network is shown in Fig. 2.

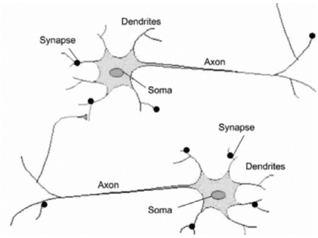


Fig 2 (Biological neural network (Adapted from: http://ffden2.phys.uaf.edu/- 212_ fall2003.web.dir/Keith_Palchikoff/Intro_page.html)

Each neuron consists of a cell body called soma, a number of root-like extensions connected to a thousand adjacent neurons called dendrites, and a single transmission line extending out from the soma called axon. The two specialized extensions of a soma are responsible for carrying information from/to a cell body. Dendrites bring information to a cell body and axons take information away from a cell body. The connection between two neurons, in particular, between an axon terminal and another neuron, is called synapse.

Each neuron uses biochemical reactions to receive processes and transmit information. Neurons communicate with each other through an electrochemical process. This means that chemicals create an electrical signal. When a neuron does not send a signal, it is in a resting state. The inside of the neuron has a negative electric potential. When a neuron sends a signal, it causes a change in the electrical potential of the cell body. The change occurs due to the release of chemical substances from the synaptic cell, called neurotransmitter. When the potential exceeds a certain threshold, an action potential occurs. Consequently, the neuron will fire the electrical signal down the axon. The occurrence of action potential can be increased or decreased by changing the constitution of various neurotransmitters.

An essential characteristic of biological neural networks is plasticity, an ability of the brain to reorganize with learning, based on experience or sensory stimulation. Scientists believe that there are two types of modifications that form the basis of learning in the brain:
1) a change in the internal structure of the synapses
2) an increase in the number of synapses between neurons.

The natural and power of a biological neural network, in particular, the potential of learning process, motivated computer scientists to design and develop a new network platform that worked in a way similar to that of the biological neurons (Adeli, 1995; Freeman and Skapura, 1991; Haykin, 1994; Müller and Reinhardt, 1990; and Negnevitsky, 2002). This leads to the introduction of Artificial Neural Networks (ANNs).

An Artificial Neural Network (ANN) is an information processing model that is able to capture and represent complex input-output relationships. The motivation the development of the ANN technique came from a desire for an intelligent artificial system that could process information in the same way the human brain. Its novel structure is represented as multiple layers of simple processing elements, operating in parallel to solve specific problems.

An architecture of a typical artificial neural network is illustrated in Fig. 3.

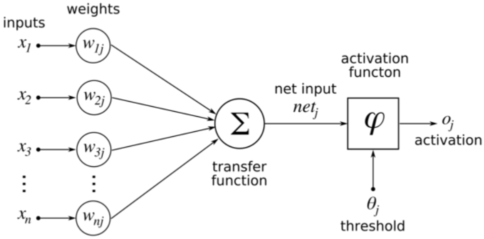


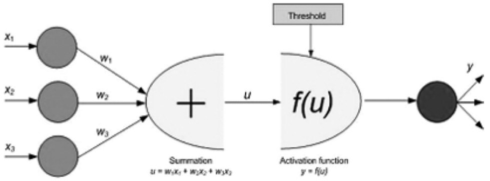
Fig 3: Schematic representation of a generic ANN (https://upload.wikimedia.org/wikipedia/commons/thumb/6/60/ArtificialNeuronModel_english.png/600px-ArtificialNeuronModel_english.png)

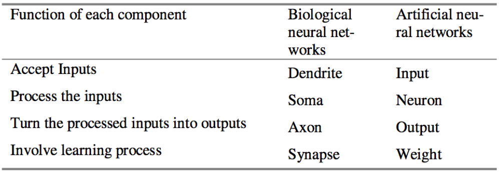
ANNs resemble human brain in two respects:
1) learning process
2) storing experiential knowledge.

An artificial neural network learns and classifies a problem through repeated adjustments of the connecting weights between the elements. In other words, an ANN learns from examples and generalizes the learning beyond the examples supplied. For instance, human beings learn to recognize faces from examples of faces.

Each element (analogous to a neuron) in the network is connected to its neighbors with weights (analogous to synapses) that represent the strengths of the connections. Typically, a single processing element receives a number of inputs (analogous to dendrites) through its connection, combines them, performs a (non-)linear operation on the result, and then produces the final result (analogous to an axon). The input can be information from external environments or outputs of other neurons. The output can be either a final solution to the problem or an input to other neurons.

Figure 4 illustrates a neuron model, and Figure 5 shows that the artificial neural network concepts are similar to those of the biological brain.





Function of each component	Biological neural networks	Artificial neural networks
Accept Inputs	Dendrite	Input
Process the inputs	Soma	Neuron
Turn the processed inputs into outputs	Axon	Output
Involve learning process	Synapse	Weight

The neuron determines its output on the basis of the weighted sum of the inputs, a threshold value (θ), and an activation function. An activation function of a neuron can be any mathematical function. In practice, four functions are commonly used. They are step function, sign function, sigmoid function, and linear function. If one chooses a sign function as an ac- tivation function and the net input is less thanθ , the neuron output is 1; otherwise, it is -1.

To build an artificial network, one must decide which network architec- ture and learning algorithm should be used. Network architecture tells how the neurons are used, and how they are connected in a network. The aim of the learning function is to modify the weights of the inputs to achieve the desired outputs.

Based on the arrangement of the internal nodes in the network layer, the neural network architecture can be classified into different types:
1) perceptron,
2) feedforward networks
3) feedback networks.

The simplest type of neural network is a perceptron (Rosenblatt, 1958). It consists of a single layer wherein weights are trained to produce a correct output when presented with inputs. The perceptron is typically used for class classification, where the classes are linearly separable. Therefore, perceptrons are suitable only for simple problems in pattern classification. The limitations of the single layered perceptron were mathematically analyzed. The outcome of this analysis was the multilayer perceptron (Minsky and Papert, 1969).

The multilayer perceptron expands the basic single layer network by having one or more hidden layers. In the multilayer structure, the input layer accepts information from the external environment and passes the information to all units in the first hidden layer. The outputs from the first hidden layer are redistributed to the next hidden layer, and so on. The output layer accepts output from the last hidden layer and generates the final output of the entire network.

A feedforward network is a network of neurons that have signals traveling from input layer to output layer only. In contrast, feedback networks allow signals traveling in both directions (from input layer to output layer and vice versa). A type of feedback network is a recurrent neural network.

One important function of the human brain is to collect down and recall the memories. This is done with short and long term memories. The human memory is associative, that is, people recognize an input pattern by comparing it with patterns stored in their memories. If the input pattern is noisy, the associative memory returns the closest stored pattern. In other words, if a corrupted image is given to a network, the network will automatically reconstruct a perfect image. A recurrent neural network, a variation of the multi-layer perceptron, is able to emulate the associative characteristics. It is a modification to the multilayer neural networks, trained with the backpropagation algorithm; that is, a recurrent neural network has feedback loops from its outputs to its inputs. As in backpropagation learning, the feedbacks are used to adjust the weights of inputs. Then the output is computed again. The algorithm is repeatedly iterated until output becomes convergent.

Learning in neural networks can be divided into two types:
1) supervised learning
2) unsupervised learning.

In supervised learning, an artificial neural network is trained by an external teacher who presents inputs, weights, and desired outputs to a network. Weights are randomly initialized to the inputs of the network to compute the actual outputs. The actual outputs are compared to the desired outputs. The weights are then adjusted by the network to produce actual outputs that are close to the desired outputs. The input weights are continuously modified until acceptable actual outputs are achieved.
In contrast, unsupervised learning, also known as self-supervised learning, does not require an external teacher. During the training phase, a neural network receives a number of inputs, discovers regularities in the inputs, and learns how to organize itself.

With remarkable abilities such as nonlinearity, adaptive learning, self organization, real-time operation, very large-scale integrated implementation, and fault tolerance via redundant information coding, neural networks are able to solve complex problems that human and other computer techniques cannot do.

For example, neural networks outperform the decision tree approach on the same data. However, neural networks have some limitations. For instance, complex neural network models lack explanations to interpret the decisions of each node in the network as rules; as testing and verification. This problem comes from adaptive learning capability, in which a network learns how to solve problem by itself, and its operations cannot therefore be interpreted.

The neural network is one of several machine learning approaches that have been successfully applied to solving a wide variety of bioinformatics problems. In sequence analysis, ANNs have been applied or integrated with other methods or systems. For example, a knowledge-based neural network system was applied to analyzing DNA sequence (Fu, 1999). An artificial neural network was trained to predict the sequence of the human TP53 tumor suppressor gene based on a p53 GeneChip (Spicker et al., 2002). A multilayered feed-forward ANN was developed as a tool to predict a mycobacterial promoter sequence in a nucleotide sequence (Kalate et al., 2003).

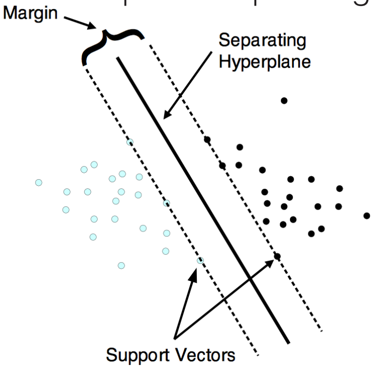
2.4) Support Vector Machines

Another machine learning approaches that have been successfully applied to solving a wide variety of bioinformatics problems is SVM.

A supervised machine learning method, the support vector machine (SVM) algorithm, has demonstrated high performance in solving classification problems in many biomedical fields, especially in bioinformatics. In contrast to logistic regression, which depends on a pre-determined model to predict the occurrence or not of a binary event by fitting data to a logistic curve, SVM discriminates between two classes by generating a hyperplane that optimally separates classes after the input data have been transformed mathematically into a high-dimensional space. Because the SVM approach is data-driven and model-free, it may have important discriminative power for classification, especially in cases where sample sizes are small and a large number of variables are involved (high-dimensionality space). This technique has recently been used to develop automated classification of diseases and to improve methods for detecting disease in the clinical setting.

SVMs were developed by Cortes & Vapnik (1995) for binary classification. Their approach may be roughly sketched as follows:

1) Class separation: basically, we are looking for the optimal separating hyperplane between the two classes by maximizing the margin between the classes' closest points (see Figure below)—the points lying on the boundaries are called support vectors, and the middle of the margin is our optimal separating hyperplane;



2) Overlapping classes: data points on the "wrong" side of the discriminant margin are weighted down to reduce their influence ("soft margin");
3) Nonlinearity: when we cannot find a linear separator, data points are projected into an (usually) higher-dimensional space where the data points effectively become linearly separable (this projection is realised via kernel techniques);

Problem solution: the whole task can be formulated as a quadratic optimization problem which can be solved by known techniques. A program able to perform all these tasks is called a Support Vector Machine.

2.5) Deep Learning

Different to more traditional use of NNs, deep learning accounts for the use of many hidden neurons and layers—typically more than two—as an architectural advantage combined with new training paradigms.

While resorting to many neurons allows an extensive coverage of the raw data at hand, the layer-by-layer pipeline of nonlinear combination of their outputs generates a lower dimensional projection of the input space. Every lower-dimensional projection corresponds to a higher perceptual level. Provided that the network is optimally weighted, it results in an effective high-level abstraction of the raw data or images. This high level of abstraction renders an automatic feature set, which otherwise would have required hand-crafted or bespoke features.

In domains such as health informatics, the generation of this automatic feature set without human intervention has many ad- vantages. For instance, in medical imaging, it can generate fea- tures that are more sophisticated and difficult to elaborate in descriptive means. Implicit features could determine fibroids and polyps, and characterize irregularities in tissue morphology such as tumors.
In translational bioinformatics, such features may also determine nucleotide sequences that could bind a DNA or RNA strand to a protein.

3) Data cleaning:

3.1) row cutting

3.2) imputation methods

3.3) class balance

3.1) row cuttings

The whole dataset counted 777 records for 62 (dummified) variables, and it was splitted into a training (545 62) and testing (232 62) datasets.

I tried 6 different cutoffs for deleting those rows containing a certain amount of missing values, then trained decision tree models to evaluate the best resampled dataset. Every time, the model was controlled by 10 cross validation methods, missing values were imputed with the means, and it was evaluated with ROC metrics. Below are the results:

1.a) keep all the predictors (29/29). dataset final size: 777 29

1.b) delete predictors with more than 41 % (12/29) of missing values. dataset final size: 654 29

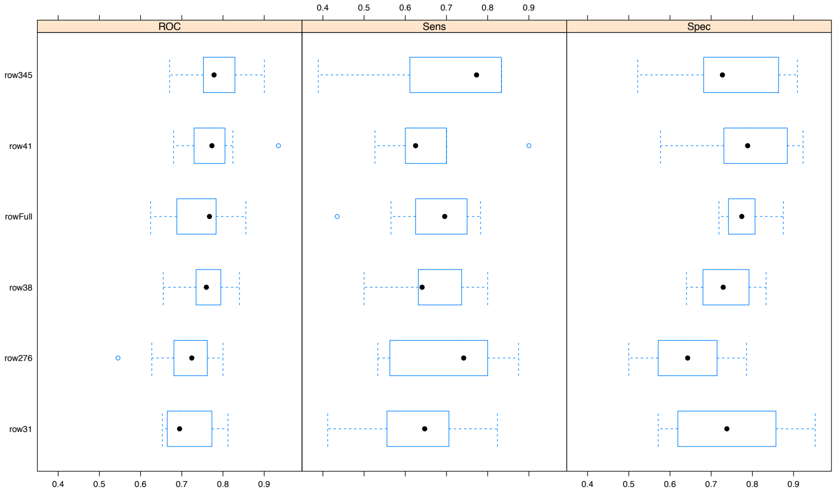
1.c) delete predictors with more than 38% (11/29) of missing values. dataset final size: 625 30

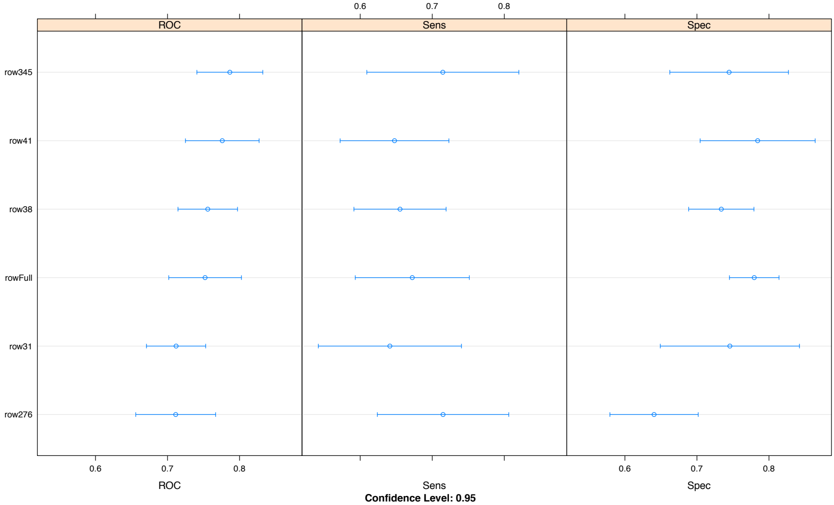
1.d) delete predictors with more than 34,5 % (10/29) of missing values. dataset final size: 571 30

1.e) delete predictors with more than 31% (9/29) of missing values. dataset final size: 543 30

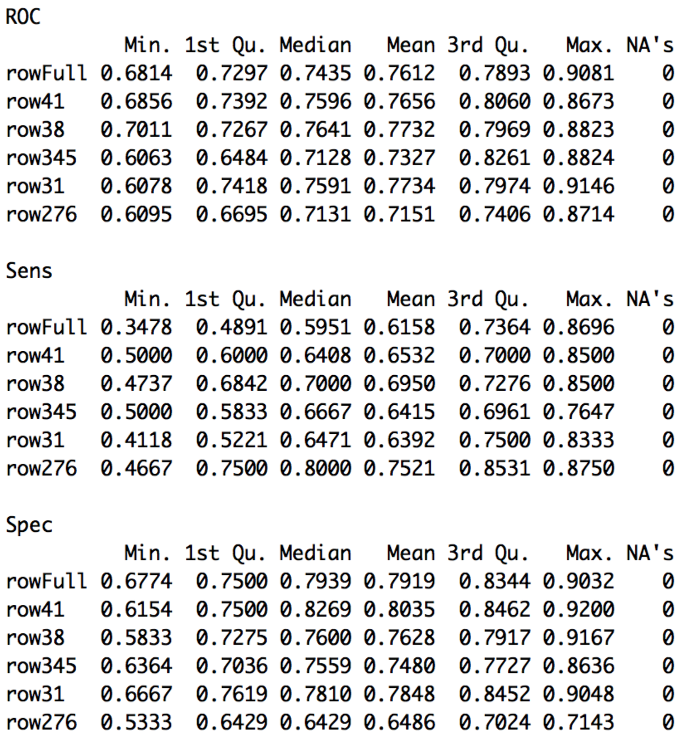
1.f) delete predictors with more than 27,6% (8/29) of missing values. dataset final size: 421 30

Results on the training dataset





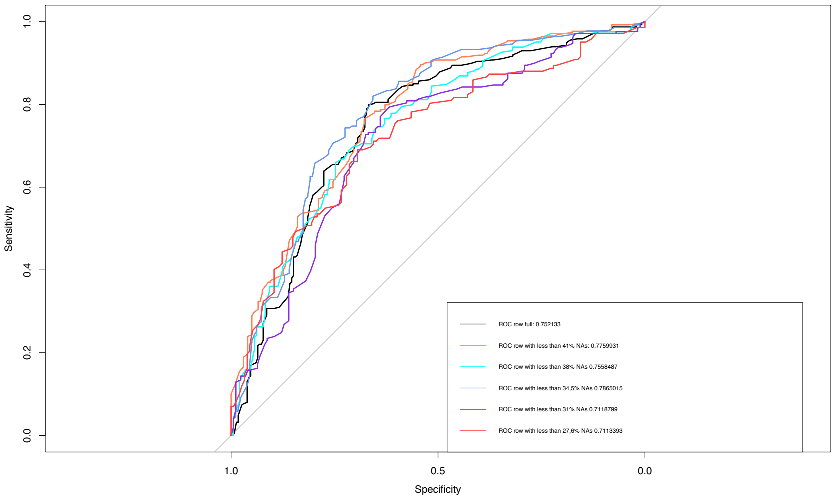
Summary of the training



```
ROC
            Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
rowFull 0.6814  0.7297 0.7435 0.7612  0.7893 0.9081    0
row41   0.6856  0.7392 0.7596 0.7656  0.8060 0.8673    0
row38   0.7011  0.7267 0.7641 0.7732  0.7969 0.8823    0
row345  0.6063  0.6484 0.7128 0.7327  0.8261 0.8824    0
row31   0.6078  0.7418 0.7591 0.7734  0.7974 0.9146    0
row276  0.6095  0.6695 0.7131 0.7151  0.7406 0.8714    0

Sens
            Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
rowFull 0.3478  0.4891 0.5951 0.6158  0.7364 0.8696    0
row41   0.5000  0.6000 0.6408 0.6532  0.7000 0.8500    0
row38   0.4737  0.6842 0.7000 0.6950  0.7276 0.8500    0
row345  0.5000  0.5833 0.6667 0.6415  0.6961 0.7647    0
row31   0.4118  0.5221 0.6471 0.6392  0.7500 0.8333    0
row276  0.4667  0.7500 0.8000 0.7521  0.8531 0.8750    0

Spec
            Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
rowFull 0.6774  0.7500 0.7939 0.7919  0.8344 0.9032    0
row41   0.6154  0.7500 0.8269 0.8035  0.8462 0.9200    0
row38   0.5833  0.7275 0.7600 0.7628  0.7917 0.9167    0
row345  0.6364  0.7036 0.7559 0.7480  0.7727 0.8636    0
row31   0.6667  0.7619 0.7810 0.7848  0.8452 0.9048    0
row276  0.5333  0.6429 0.6429 0.6486  0.7024 0.7143    0
```

Comments: even though cutting rows with more than 34,5% of NAs may seems the better solution given the results, its specificity and sensitivity show a wider range of results among the mean than the other models.

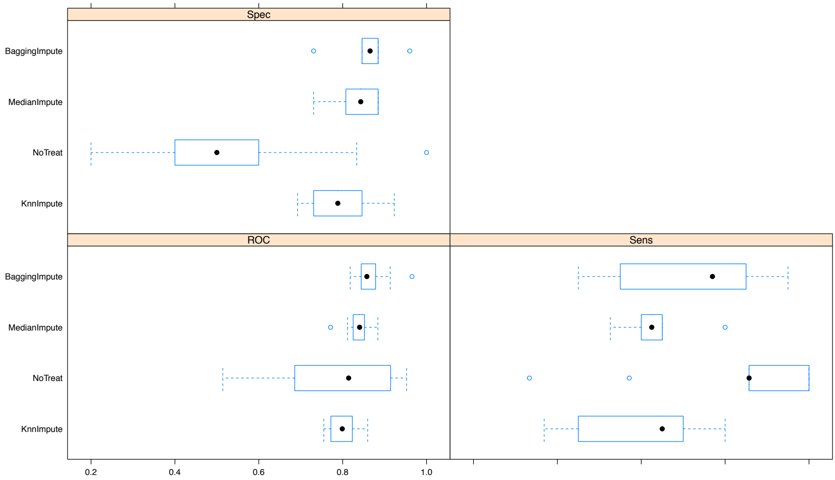
Results on the testing dataset

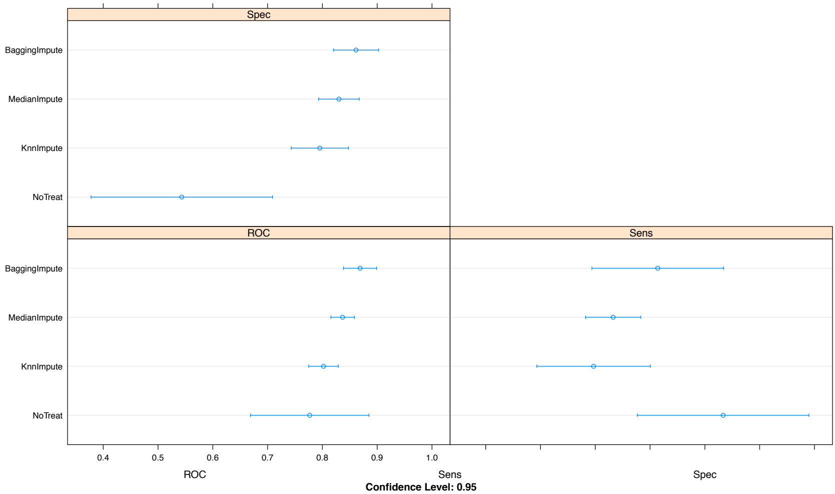


Comments:
The best cutoff is the one that only keeps rows with less than 34,5% NAs (i.e. less then 10 missing values over 29 predictors).

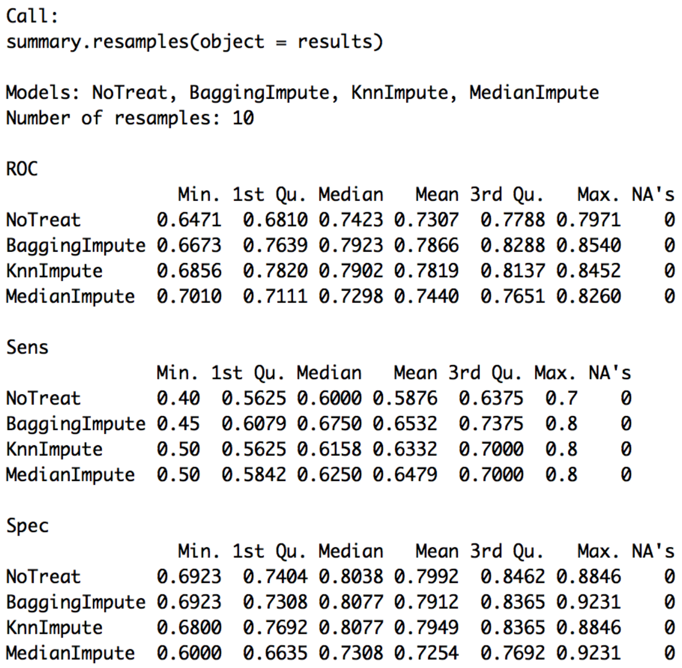
3.2) imputation methods

After reducing the dataset by deleting those raws with more than 34,5% NAs, I proceeded to find the best imputation method: I trained and tested my dataset with no imputation, bad imputation, knn imputation and median imputaton. The model used is SVM. Below are the results:





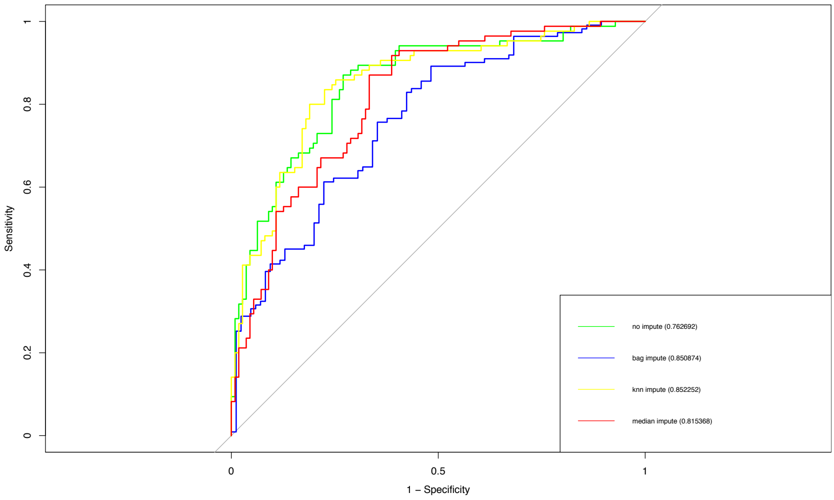
Summary of the training



```
Call:
summary.resamples(object = results)

Models: NoTreat, BaggingImpute, KnnImpute, MedianImpute
Number of resamples: 10

ROC
                Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
NoTreat       0.6471  0.6810 0.7423 0.7307  0.7788 0.7971    0
BaggingImpute 0.6673  0.7639 0.7923 0.7866  0.8288 0.8540    0
KnnImpute     0.6856  0.7820 0.7902 0.7819  0.8137 0.8452    0
MedianImpute  0.7010  0.7111 0.7298 0.7440  0.7651 0.8260    0

Sens
              Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
NoTreat       0.40  0.5625 0.6000 0.5876  0.6375  0.7    0
BaggingImpute 0.45  0.6079 0.6750 0.6532  0.7375  0.8    0
KnnImpute     0.50  0.5625 0.6158 0.6332  0.7000  0.8    0
MedianImpute  0.50  0.5842 0.6250 0.6479  0.7000  0.8    0

Spec
                Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
NoTreat       0.6923  0.7404 0.8038 0.7992  0.8462 0.8846    0
BaggingImpute 0.6923  0.7308 0.8077 0.7912  0.8365 0.9231    0
KnnImpute     0.6800  0.7692 0.8077 0.7949  0.8365 0.8846    0
MedianImpute  0.6000  0.6635 0.7308 0.7254  0.7692 0.9231    0
```

Comments: it would be highly risky not to treat the missing values at all, because this would likely affect the performance of the final model in predicting the class of new records. Infact its results are within a wider range of numbers, and its sensitivity is higly skewed left.
Sensitivity also seems the most difficult point for all the resampled datasets, since the range is wider almost everywhere. We’ll try to solve this problem later, by balancing the classes outcome.

Results on the testing dataset



Comments:
So the best imputation method is knn imputation.
Since in the training it only seemed the third best model, following median and bagging methods, it's clear that the two latters were overfitting the data, and knn imputated dataset developed a better generalization model.

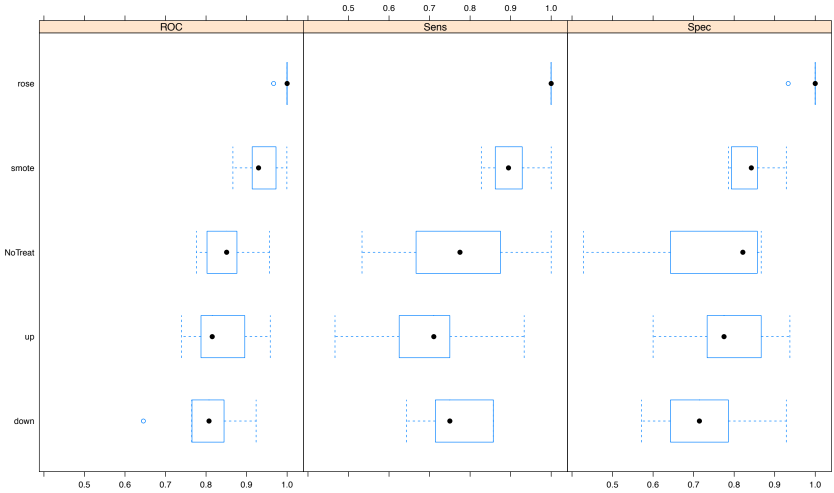
3.3) class balance

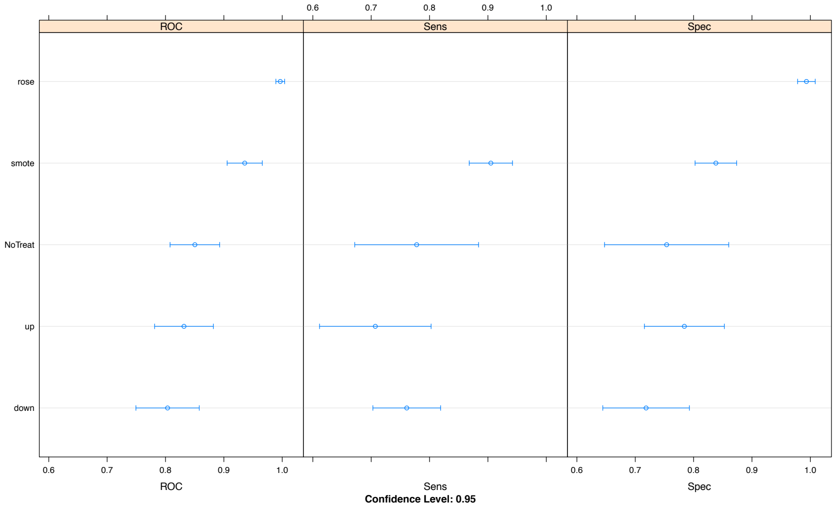
After reducing the dataset by deleting those raws with more than 34,5% NAs and imputing the missing values with the knn, I proceeded with balancing the classes, in order to detect with equally (good) results both classes, ie to have the same Specificity and Sensitivity.
I tryied 4 sampling methods for class balancing: none, down, up, SMOTE and ROSE.

Basically, down sampling reduces the size of the dataset by downsampling records from themajority class to the number of the minority one;
upsampling works in the opposite way;
The SMOTE (Synthetic Minority Over-sampling Technique,) and ROSE (Random Over-Sampling Examples) function oversamples your rare event by using bootstrapping and k-nearest neighbor to synthetically create additional observations of that event;

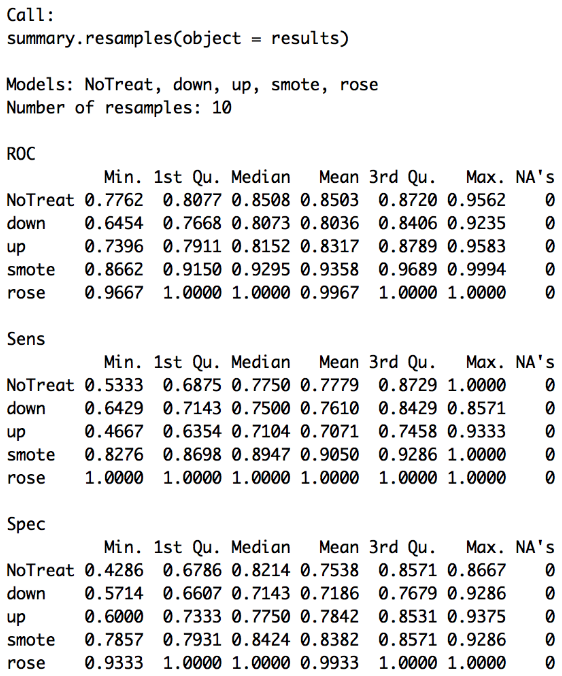
So we're starting with a dataset of size Control 154 Patient 142;
down sampling would count Control 142 Patient 142
up sampling would count Control 154 Patient 154
SMOTE sampling would count Control 284 Patient 284
ROSE sampling would count Control 149 Patient 147

Results on the training dataset





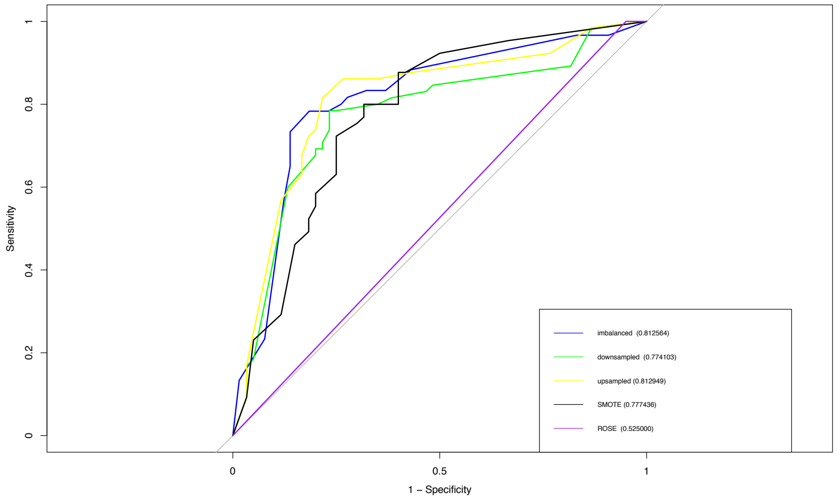
Summary of the training



```
Call:
summary.resamples(object = results)

Models: NoTreat, down, up, smote, rose
Number of resamples: 10

ROC
          Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
NoTreat 0.7762  0.8077 0.8508 0.8503  0.8720 0.9562    0
down    0.6454  0.7668 0.8073 0.8036  0.8406 0.9235    0
up      0.7396  0.7911 0.8152 0.8317  0.8789 0.9583    0
smote   0.8662  0.9150 0.9295 0.9358  0.9689 0.9994    0
rose    0.9667  1.0000 1.0000 0.9967  1.0000 1.0000    0

Sens
          Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
NoTreat 0.5333  0.6875 0.7750 0.7779  0.8729 1.0000    0
down    0.6429  0.7143 0.7500 0.7610  0.8429 0.8571    0
up      0.4667  0.6354 0.7104 0.7071  0.7458 0.9333    0
smote   0.8276  0.8698 0.8947 0.9050  0.9286 1.0000    0
rose    1.0000  1.0000 1.0000 1.0000  1.0000 1.0000    0

Spec
          Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
NoTreat 0.4286  0.6786 0.8214 0.7538  0.8571 0.8667    0
down    0.5714  0.6607 0.7143 0.7186  0.7679 0.9286    0
up      0.6000  0.7333 0.7750 0.7842  0.8531 0.9375    0
smote   0.7857  0.7931 0.8424 0.8382  0.8571 0.9286    0
rose    0.9333  1.0000 1.0000 0.9933  1.0000 1.0000    0
```

Comments: both ROSE and SMOTE gave great results in the training, not only for the mean value of ROC, Sensitivity and Specificity but also because of a very small range of values among the mean; but ROSE is very likely to be overfitting the data. The testing will show if that is the case.

Results on the testing dataset



Comments:
Our suspects were correct: when applied on new cases, ROSE’s detecting precision of new cases’ outcome fails dramatically: this happens because it was too good on predicting the previous dataset only.So the best resampling method for class balance is up sampling, which outstanded not only the above mentioned SMOTE, but also ROSE.

In the next chapter I’m going to do a feature selection. The dataset to start with is going to be preprocessed with the three points above:
1) rows with less than 34,5% NAs
2) bag imputation
3) upsampling for class balance

And the three feature selection methods are:
1) Relief Feature Selection
2) Calculation Of Filter-Based Variable Importance
3) Extract Variable Importance Measure from RandomForest

4) Filter methods for ML in bioinformatics

In bioinformatics and related scientific fields, such as statistical genomics and genetic epidemiology, an important task is the prediction of a categorical response variable (such as the disease status of a patient or the properties of a molecule) based on a large number of predictors.

The aim of this chapter is on one hand to predict the value of the response variable from the values of the predictors, i.e. to create a diagnostic tool, and on the other hand to reliably identify relevant predictors from a large set of candidate variables.

From a statistical point of view, one of the challenges in identifying these relevant predictor variables is the so-called “small n large p” problem: Usual data sets in genomics often contain hundreds or thousands of genes or markers that serve as predictor variables X1,..., X p , but only for a comparatively small number n of subjects or tissue types.

Later I’m going to apply three methods to filter the predictors of my preprocessed dataset in order to train and test different subsamples with the most important variables, selected in turn by each of the method. These methods are:

1)Relief Feature Selection,
2) Calculation Of Filter-Based Variable Importance,
3) Extract Variable Importance Measure from RandomForest.

For each filter method I’m training three different models, ie those that are more likely to be positively affected by the predictor reduction: SVM, NN and DL.

The dummified dataset consists of 50 predictors, 22 are cannabis related and 28 are non cannabis related (ethnical, education, age sex and non-cannabis drugs ones).

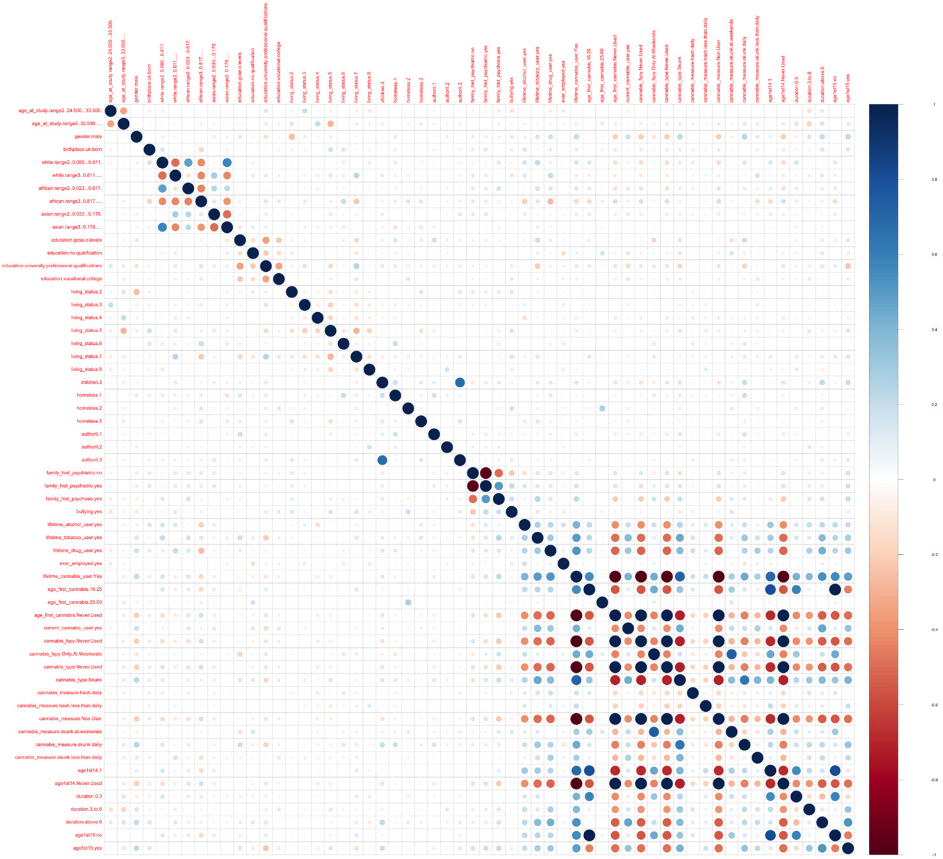


FIG Correlation plot of dummified dataset. The Cannabis variables are among all those with a higher predictive value.

The figure shows a correlation matrix of the training set. Each pairwise correlation is computed from the training data and colored according to its magnitude. This visualization is symmetric: the top and bottom diagonals show identical information. Dark blue colors indicate strong positive correlations, dark red is used for strong negative correlations, and white implies no empirical relationship between the predictors. In this figure, the predictor variables have been arranged according to their position in the dummified dataset, so that collinear groups of predictors are adjacent to one another.

In general, there are good reasons to avoid data with highly correlated predictors. First, redundant predictors frequently add more complexity to the model than information they provide to the model. In situations where obtaining the predictor data is costly (either in time or money), fewer variables is obviously better.
So is no surprise that when it comes to filtering the predictors, not all the 22 cannabis related factors are selected for building the resampled models.

As it can be easily noticed in the correlation plot from the Figure above, the cannabis attributes are the most predictive ones in predicting the disease outcome. In the following featuring part, these are highlighted with the colour red.

Our 9 cannabys factors are:

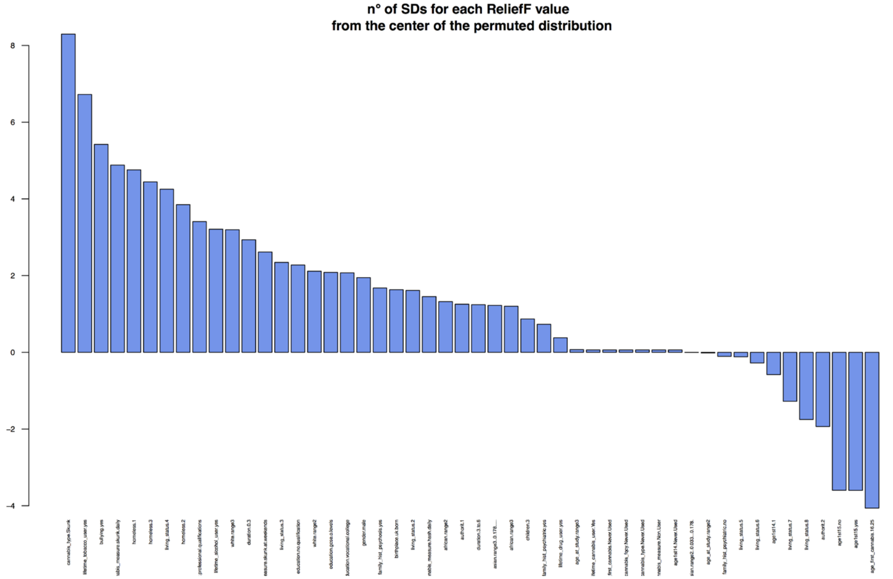
[1] “lifetime_cannabis_user”	(2 levels)
[2] “age_first_cannabis”	(4 levels)
[3] “current_cannabis_user”	(2 levels)
[4] “cannabis_fqcy”	(3 levels)
[5] “cannabis_type”	(3 levels)
[6] “cannabis_measure”	(7 levels)
[7] “age1st14”	(3 levels)
[8] “duration”	(4 levels)
[9] “age1st15”	(3 levels)

After dummifying them (so each factor is spread among its levels -1, to which is assigned not a categorical value but a numerical (1/0) one) we obtain these 22 variables:

[1]“lifetime_cannabis_user.Yes”
[2] “age_first_cannabis.16.25”
[3] “age_first_cannabis.25.60”
[4] “age_first_cannabis.Never.Used”
[5] “current_cannabis_user.yes”
[6] “cannabis_fqcy.Never.Used”
[7] “cannabis_fqcy.Only.At.Weekends”
[8] “cannabis_type.Never.Used”
[9] “cannabis_type.Skunk”
[10] “cannabis_measure.hash.daily”
[11] “cannabis_measure.hash.less.than.daily”
[12] “cannabis_measure.Non.User”
[13] “cannabis_measure.skunk.at.weekends”
[14] “cannabis_measure.skunk.daily”
[15] “cannabis_measure.skunk.less.than.daily”
[16] “age1st14.1”
[17] “age1st14.Never.Used”
[18] “duration.0.3”
[19] “duration.3.to.6”
[20] “duration.above.6”
[21] “age1st15.no”
[22] “age1st15.yes”

4.1) Relief Feature Selection

This function implements the RELIEF feature selection algorithm. The general idea of this method is to choose the features that can be most distinguished between classes. These are known as the relevant features. At each step of an iterative process, an instance x is chosen at random from the dataset and the weight for each feature is updated according to the distance of x to its Nearmiss and NearHit. The dataset must have complete cases therefore imputation must be performed in advance.



THRESHOLDS:
|1.96| and |0.65|

19 predictors selected from relief score |1.96|
32% of cannabis attributes selected,
43% of non cannabis attributes selected.
ratio cannabis-non cannabis: 1 : 1.7

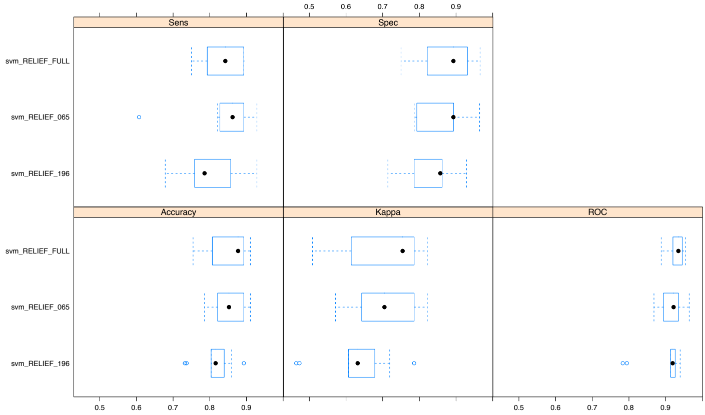
```
[1] “white.range2”                    “white.range3”
 [3] “education.gcse.o.levels”          “education.no.qualification”
 [5] “education.vocational.college”      “living_status.3”
 [7] “living_status.4”                  “homeless.1”
 [9] “homeless.2”                       “homeless.3”
[11] “bullying.yes”                     “lifetime_tobacco_user.yes”
[13] “age_first_cannabis.16.25”          “cannabis_type.Skunk”
[15] “cannabis_measure.skunk.at.weekends” “cannabis_measure.skunk.daily”
[17] “duration.0.3”                     “age1st15.no”
[19] “age1st15.yes”
```

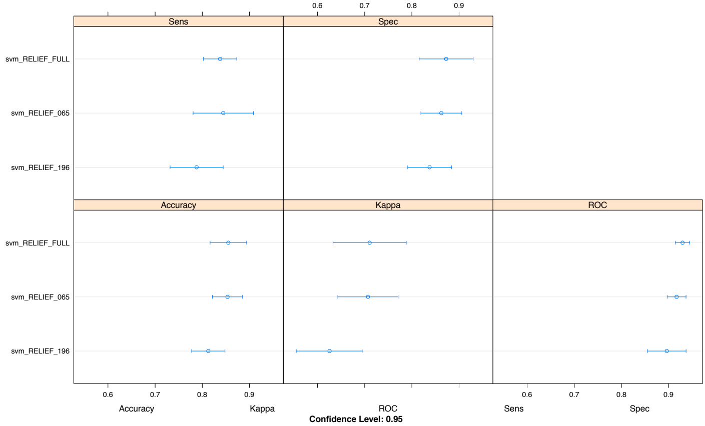
34 predictors selected from relief score |0.65|
41% of cannabis attributes selected,
89% of non cannabis attributes selected.
ratio cannabis-non cannabis: 1 : 2.8

```
[1] “gender.male”                       “birthplace.uk.born”
 [3] “white.range2”                      “white.range3”
 [5] “african.range2”                    “african.range3”
 [7] “asian.range3..0.178.....”            “education.gcse.o.levels”
 [9] “education.no.qualification”          “education.vocational.college”
[11] “living_status.2”                   “living_status.3”
[13] “living_status.4”                   “living_status.7”
[15] “living_status.8”                   “children.3”
[17] “homeless.1”                        “homeless.2”
[19] “homeless.3”                        “authorit.1”
[21] “authorit.2”                      “family_hist_psychiatric.yes”
[23] “family_hist_psychosis.yes”           “bullying.yes”
[25] “lifetime_tobacco_user.yes”           “age_first_cannabis.16.25”
[27] “cannabis_type.Skunk”                “cannabis_measure.hash.daily”
[29] “cannabis_measure.skunk.at.weekends” “cannabis_measure.skunk.daily”
[31] “duration.0.3”                      “duration.3.to.6”
[33] “age1st15.no”                        “age1st15.yes”
```

RESULTS ON THE TRAINING DATASET

SVM results

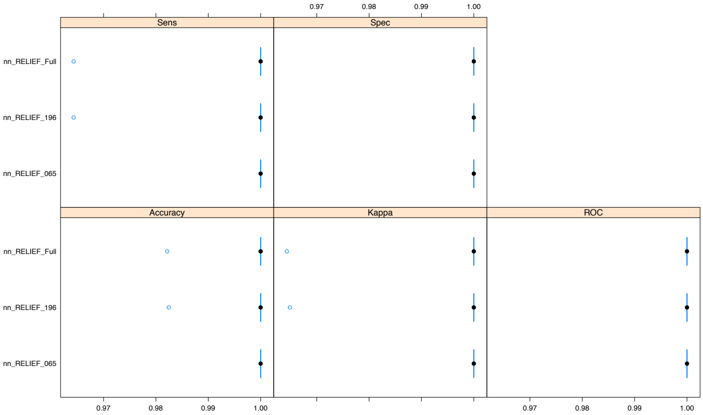


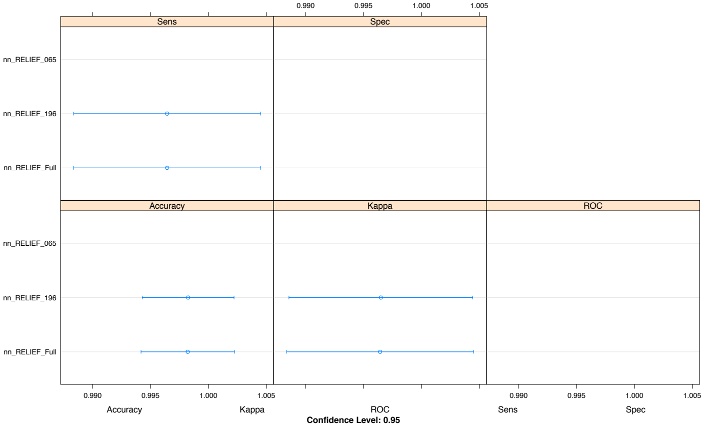


Comments: svm models give us a first insight into data: for training, the more variables the better the precision in detecting the classes outcome.

RESULTS ON THE TRAINING DATASET

NN results

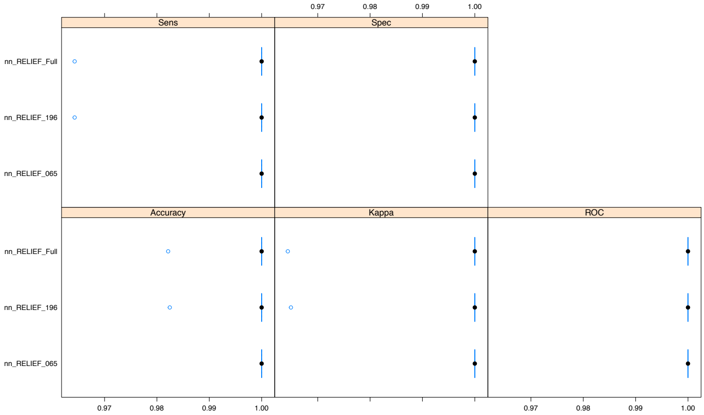


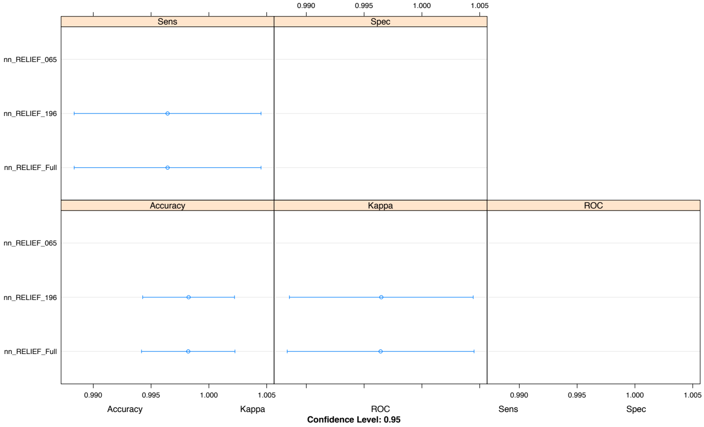


Comments: ANNs are known as powerful ML methods, so it’s relatively surprising that their performance would seem so good: 100% of correct class predicted on the training. But might be overfitting, as common habits of this particular model.

RESULTS ON THE TRAINING DATASET

DL results (10 hiddens)

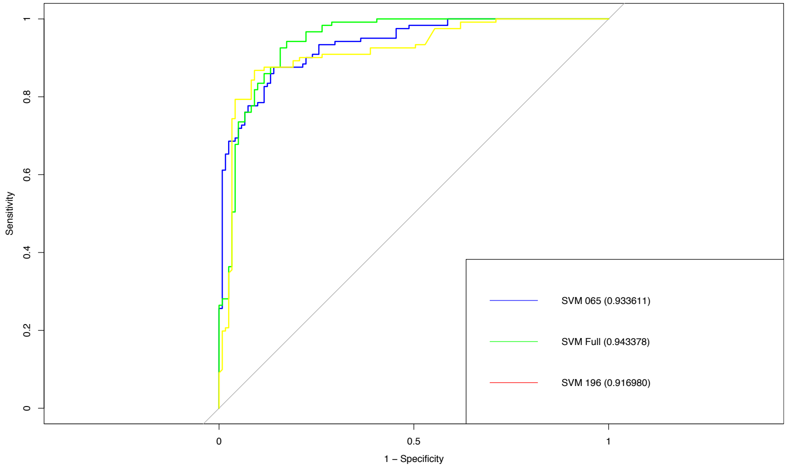




Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

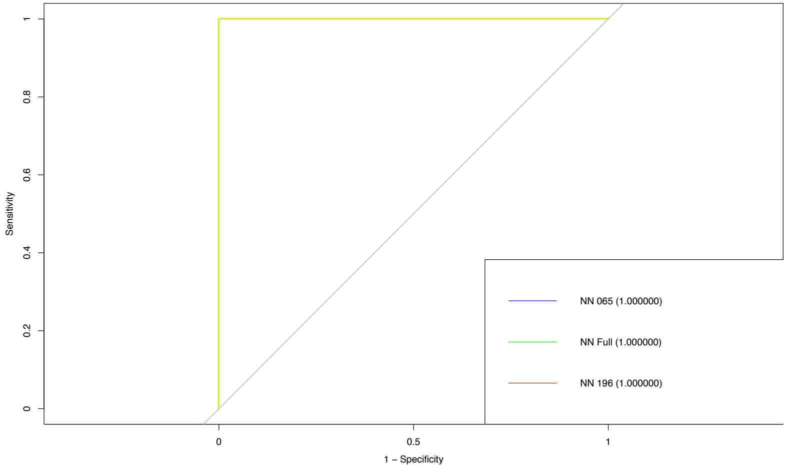
RESULTS ON THE TESTING DATASET

SVM results:



Comments: results on the testing confirm no improvement in the performances of the models when filtering the variables via Relief methods. Also, since the proportion of cannabis and non cannabis variables decreases in favour of the non cannabis related ones when testing the model with more features, it may seems that the former are as good predictors as the other 28 variables from the whole dataset.

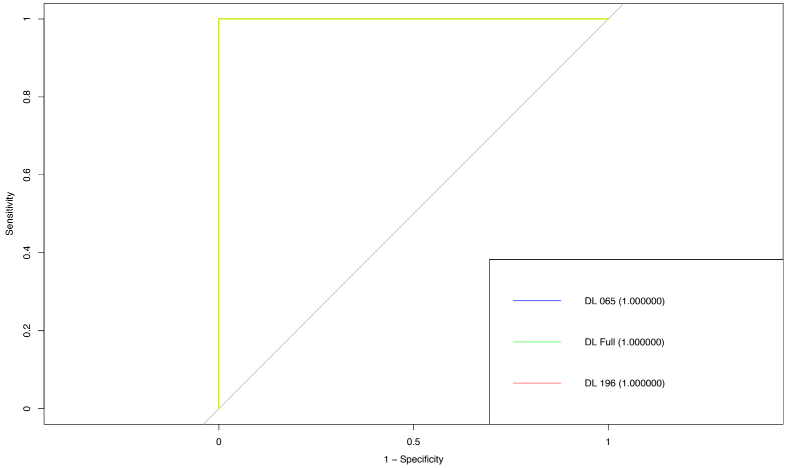
RESULTS ON THE TESTING DATASET
NN results:



Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

RESULTS ON THE TESTING DATASET

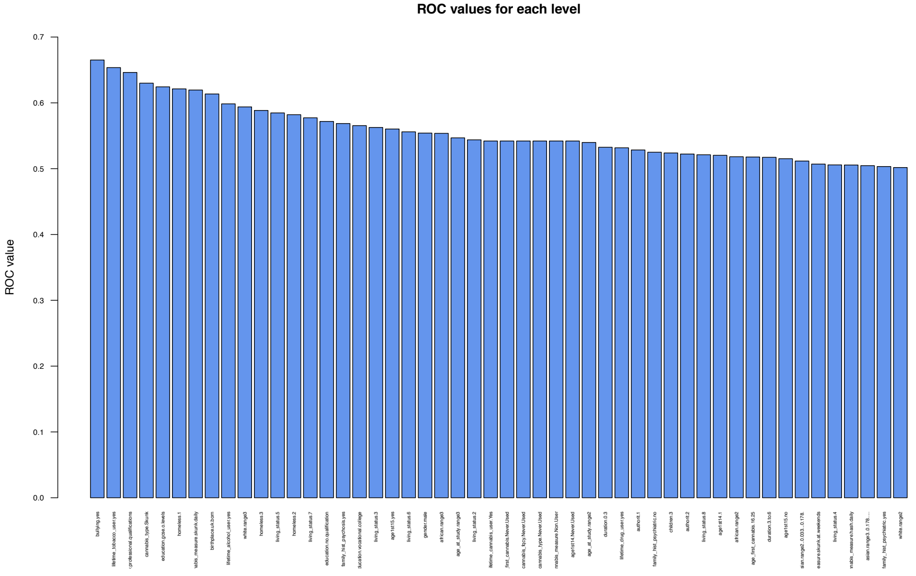
DL results:



Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

4.2) Calculation Of Filter-Based Variable Importance

Specific engines for variable importance on a model by model basis. The importance of each predictor is evaluated individually using a “filter” approach. For classification, ROC curve analysis is conducted on each predictor. For two class problems, a series of cutoffs is applied to the predictor data to predict the class. The sensitivity and specificity are computed for each cutoff and the ROC curve is computed. The trapezoidal rule is used to compute the area under the ROC curve. This area is used as the measure of variable importance. For multi-class outcomes, the problem is decomposed into all pair-wise problems and the area under the curve is calculated for each class pair (i.e class 1 vs. class 2, class 2 vs. class 3 etc.). For a specific class, the maximum area under the curve across the relevant pairwise AUC’s is used as the variable importance measure.



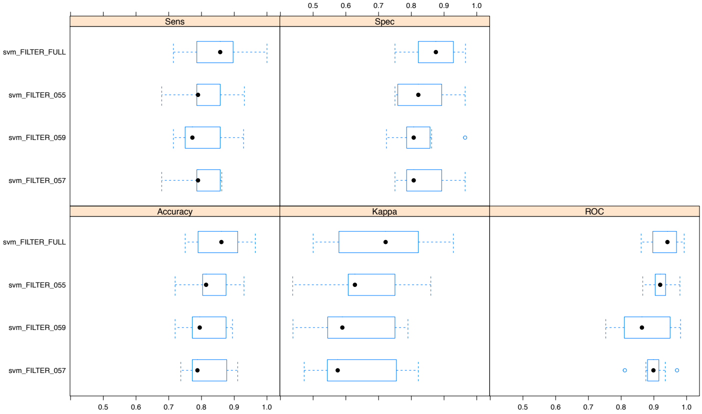
THRESHOLDS:
ROC > 0.59, 0.57, 0.55

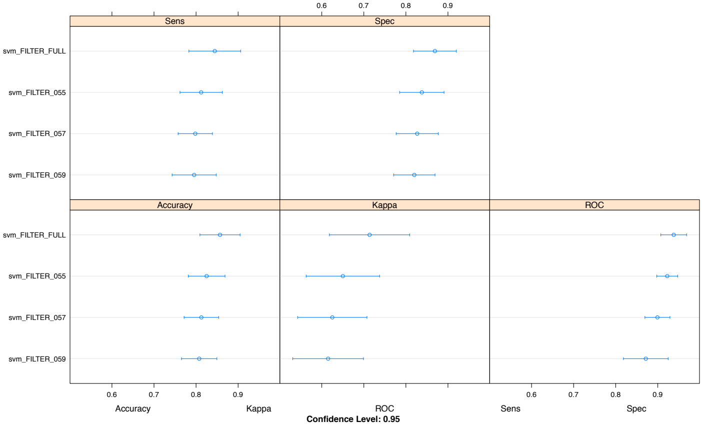
From the above list of selected predictors, our first models would pick up the all set of predictors (ie 51 names); the seconds the first 22 names, the thirds the first 15 and the fourths the first 10 names, each corresponding to the ROC values thresholds 0.55, 0.57 aand 0.59 respectively.

51 predictors from all dataset
    100% of cannabis attributes selected,
    100% of non cannabis attributes selected. ratio: 1 : 1.27
22 predictors selected from ROC > 0.55
    14% of cannabis attributes selected,
    68% of non cannabis attributes selected. ratio: 1 : 6.3
15 predictors selected from from ROC > 0.57
    0% of cannabis attributes selected,
    54% of non cannabis attributes selected. ratio: 0 : 5.4
10 predictors selected from from ROC > 0.59
    0% of cannabis attributes selected,
    36% of non cannabis attributes selected. ratio: 0 : 3.6

```
[1] “gender.male”
 [2] “birthplace.uk.born”
 [3] “white.range3”
 [4] “african.range3”
 [5] “education.gcse.o.levels”
 [6] “education.no.qualification”
 [7] “education.university.professional.qualifications”
 [8] “education.vocational.college”
 [9] “living_status.3”
[10] “living_status.5”
[11] “living_status.6”
[12] “living_status.7”
[13] “homeless.1”
[14] “homeless.2”
[15] “homeless.3”
[16] “family_hist_psychosis.yes”
[17] “bullying.yes”
[18] “lifetime_alcohol_user.yes”
[19] “lifetime_tobacco_user.yes”
[20] “cannabis_type.Skunk”
[21] “cannabis_measure.skunk.daily”
[22] “age1st15.yes”
```

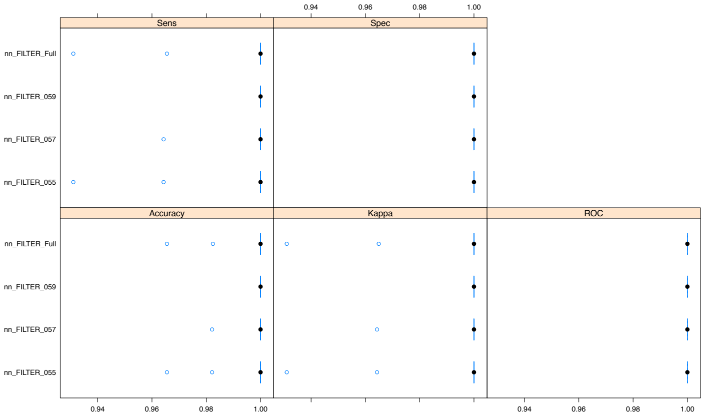
RESULTS ON THE TRAINING DATASET
SVM results:

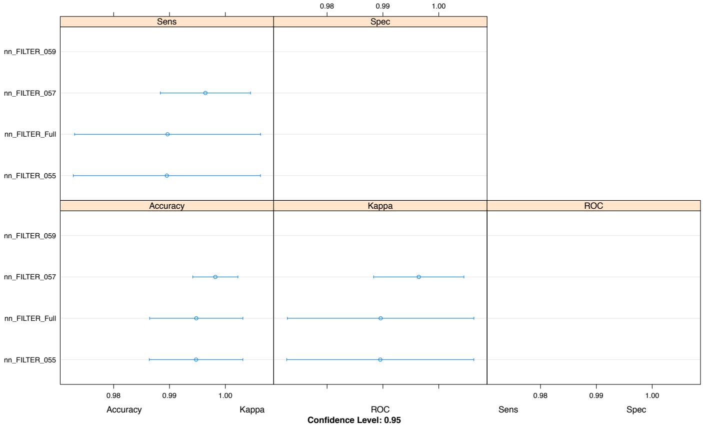




Comments: as one would expect, since our 50 predictors are very good ones, for the training the more we keep the better the learning.

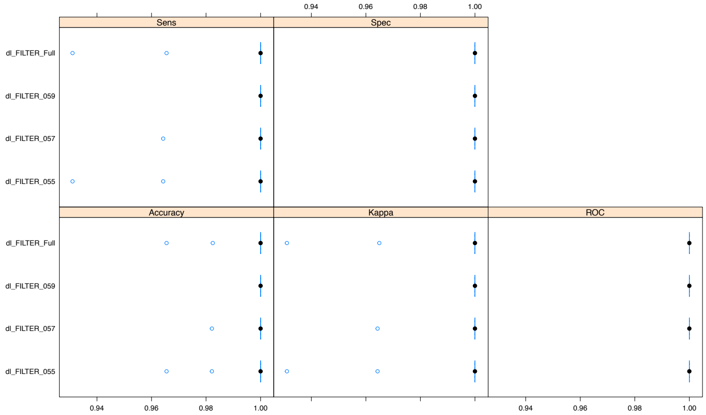
RESULTS ON THE TRAINING DATASET
NN results:

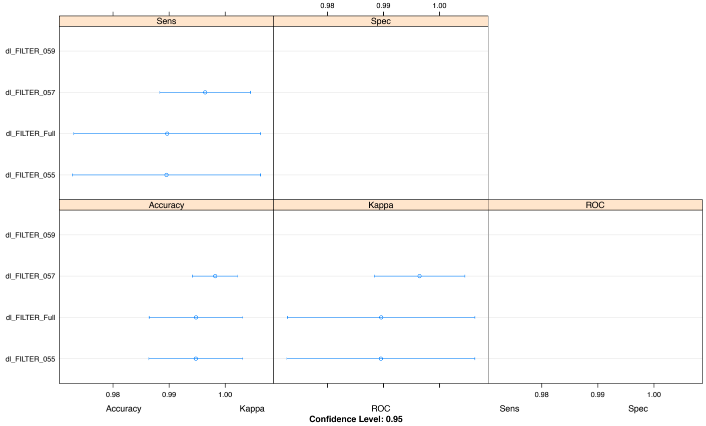




Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

RESULTS ON THE TRAINING DATASET
DL results:

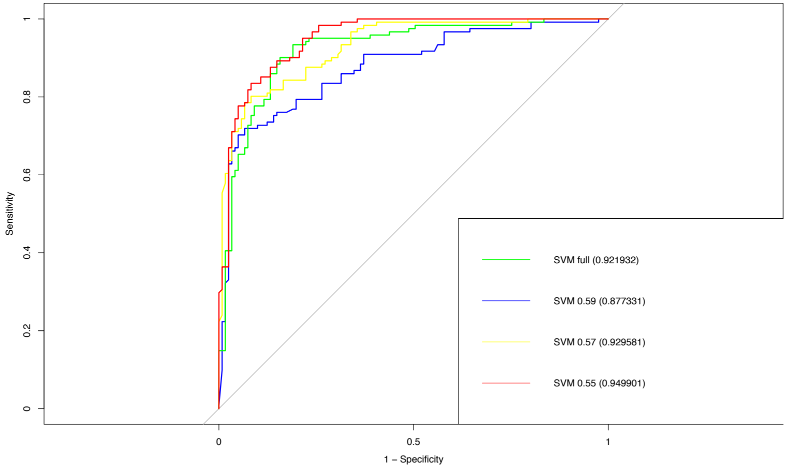




Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

RESULTS ON THE TESTING DATASET

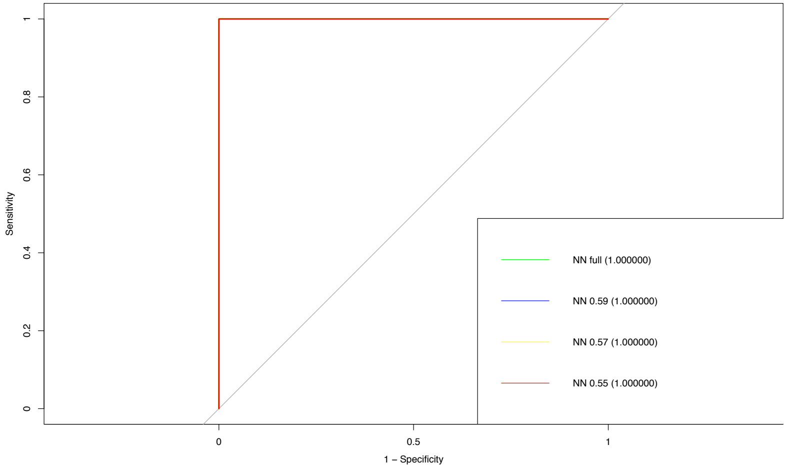
SVM results:



Comments: with the test results of the svm models, we can see that the best scores are associated with the only resampled dataset which actually had kept few cannabis predictors, instead of the other two which had none (plus the full dataset one).

RESULTS ON THE TESTING DATASET

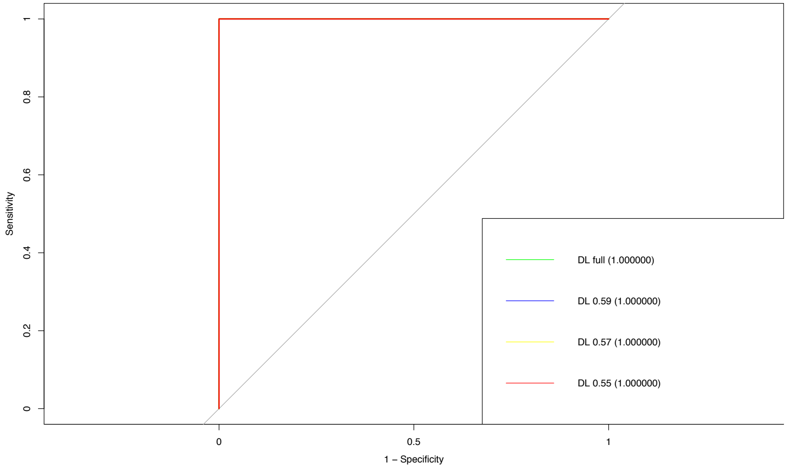
NN results:



Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

RESULTS ON THE TESTING DATASET

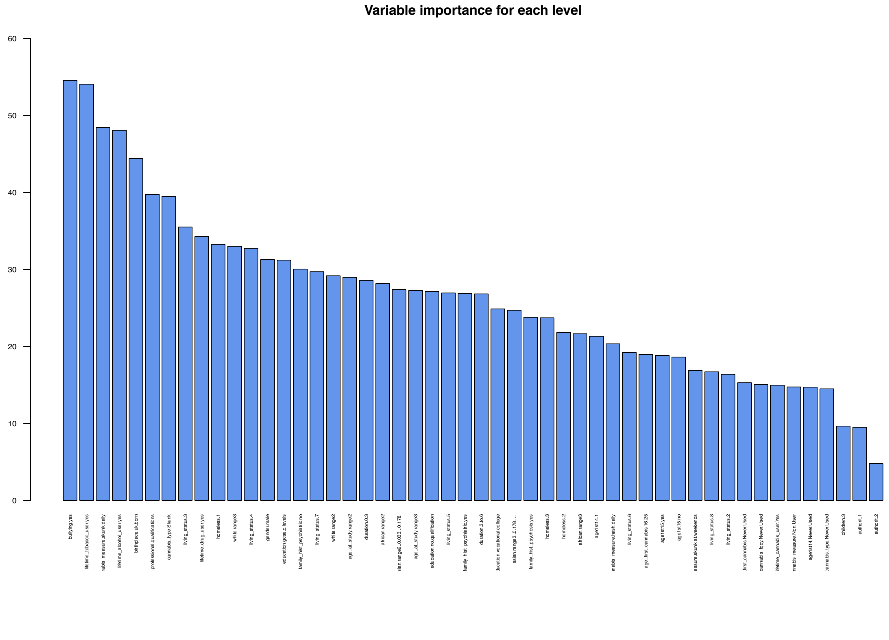
DL results:



Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

4.3) Extract Variable Importance Measure

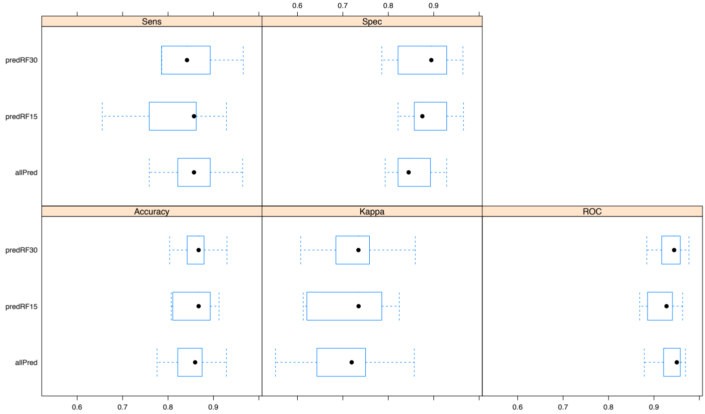
This is the extractor function for variable importance measures as produced by randomforest.
The first measure is computed from permuting OOB data: For each tree, the prediction error on the out-of-bag portion of the data is recorded (error rate for classification, MSE for regression). Then the same is done after permuting each predictor variable. The difference between the two are then averaged over all trees, and normalized by the standard deviation of the differences. If the standard deviation of the differences is equal to 0 for a variable, the division is not done (but the average is almost always equal to 0 in that case).
The second measure is the total decrease in node impurities from splitting on the variable, averaged over all trees. For classification, the node impurity is measured by the Gini index. For regression, it is measured by residual sum of squares.

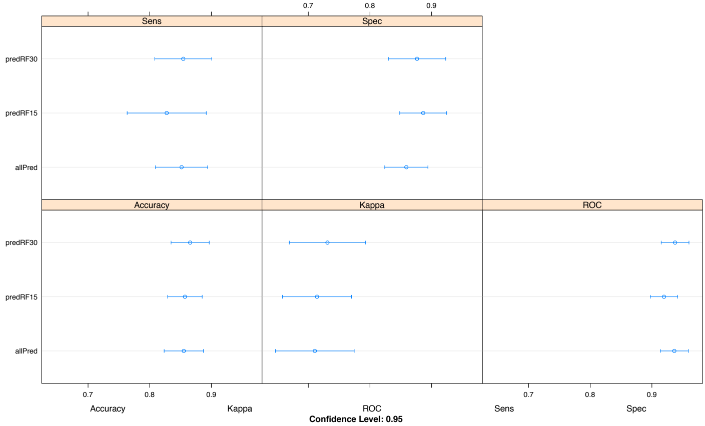


THRESHOLDS:
Top 15 and 30 most predictive variables, corresponding to the scores >30 (9% of cannabis attributes selected, 46.4% of non cannabis attributes selected. ratio: 1 : 6.5) and >23 (18% of cannabis attributes selected, 92.8% of non cannabis attributes selected. ratio: 1 : 6.5), respectively) from the list below:

```
[1] bullying.yes
 [2] lifetime_tobacco_user.yes
 [3] cannabis_measure.skunk.daily
 [4] lifetime_alcohol_user.yes
 [5] birthplace.uk.born
 [6] education.university.professional.qualifications
 [7] cannabis_type.Skunk
 [8] living_status.3
 [9] lifetime_drug_user.yes
[10] homeless.1
[11] white.range3
[12] living_status.4
[13] gender.male
[14] education.gcse.o.levels
[15] family_hist_psychiatric.no

[16] living_status.7
[17] white.range2
[18] age_at_study.range2
[19] duration.0.3
[20] african.range2
[21] asian.range2..0.033...0.178.
[22] age_at_study.range3
[23] education.no.qualification
[24] living_status.5
[25] family_hist_psychiatric.yes
[26] duration.3.to.6
[27] education.vocational.college
[28] asian.range3..0.178.....
[29] family_hist_psychosis.yes
[30] homeless.3
```

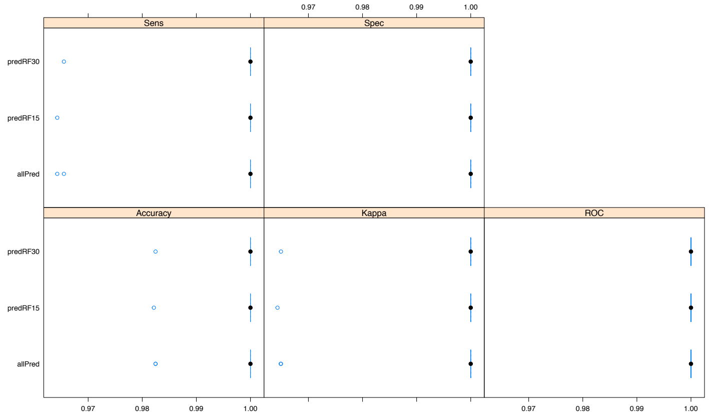
RESULTS ON THE TRAINING DATASET
SVM results:

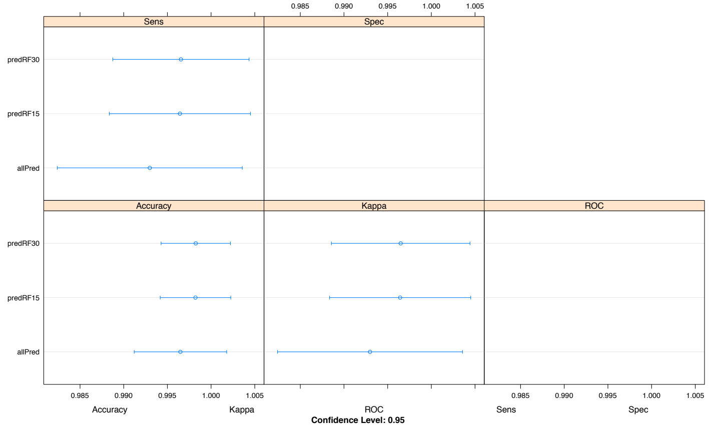




Comments: keeping less variables here seems to improve results, but with the less severe thresholded dataeset.

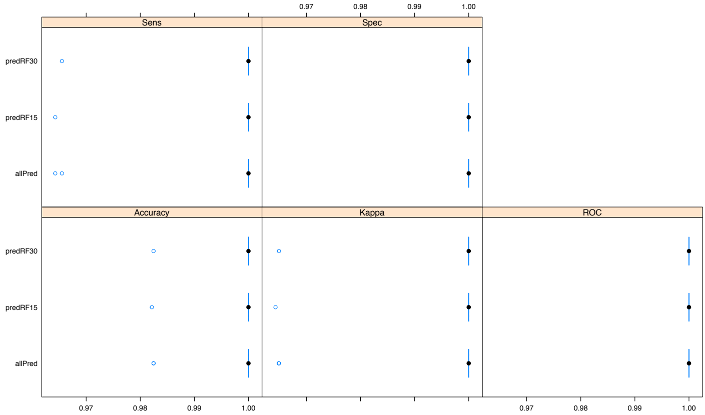
RESULTS ON THE TRAINING DATASET
NN results:

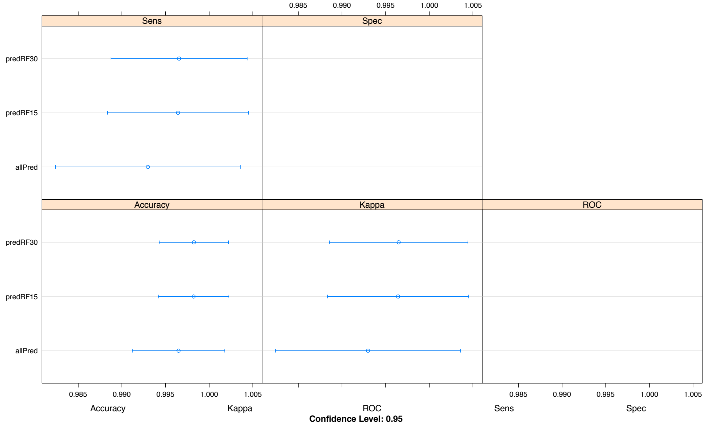




Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

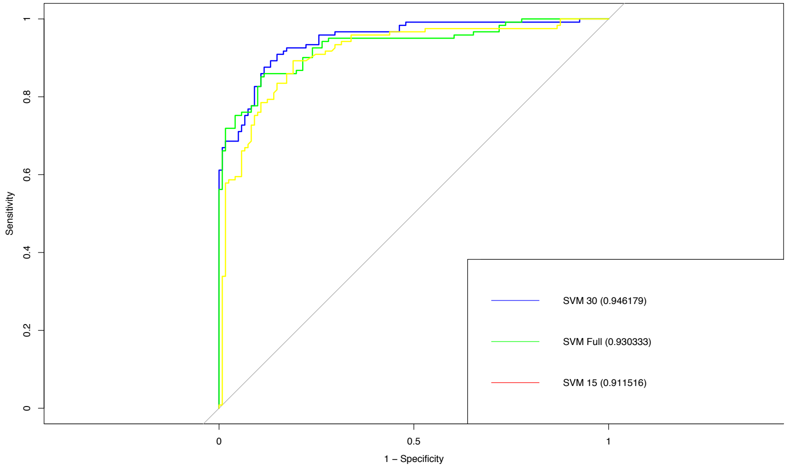
RESULTS ON THE TRAINING DATASET
DL results:





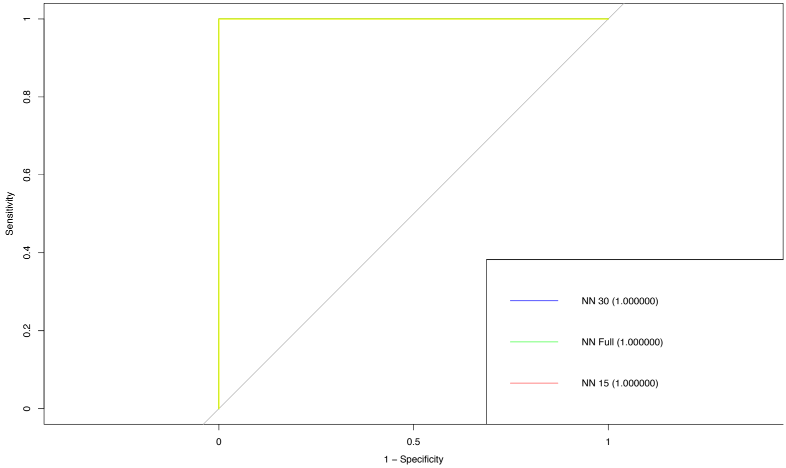
Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

RESULTS ON THE TESTING DATASET
SVM results:



comments: results on the testing confirm the results on the training. Yet the proportion between cannabis and non cannabis related feature was the same in both undersized datasets, making a hard guess to comment on the importance of the former confronted to the latter.

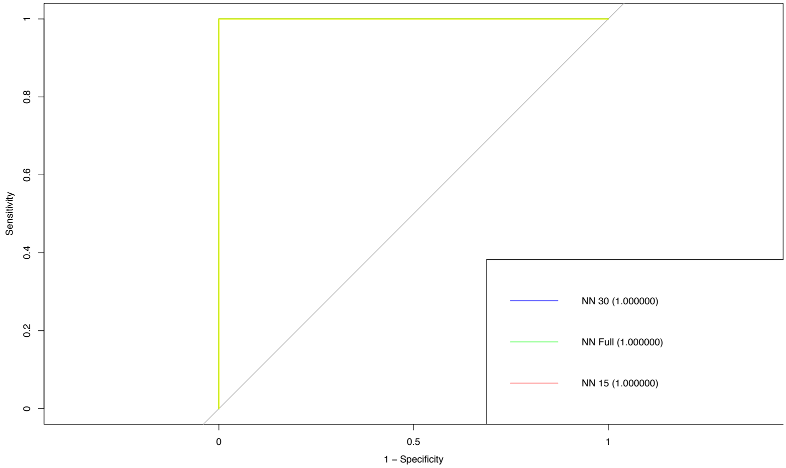
RESULTS ON THE TESTING DATASET
NN results:



Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

RESULTS ON THE TESTING DATASET

DL results



Comments: the model is very valuable but doesn not inform us much about the effectiveness of including cannabis variables.

5) technologies

For the project I used several R libraries like caret, AppliedPredictiveModeling, CORElearn, minerva, pROC, randomForest and stats. Other useful tools to make DL modeling fast and easy would have been h2o (also under R environments) and Tensorflow, though I didn't apply it for this particular project.

5.1) H2O

H2O is fast, scalable, open-source machine learning and deep learning for smarter applications. With H2O, enterprises like PayPal, Nielsen Catalina, Cisco, and others can use all their data without sampling to get accurate predictions faster.

Using in-memory compression, H2O handles billions of data rows in-memory, even with a small cluster. To make it easier for non-engineers to create complete analytic workflows, H2O's platform includes interfaces for R, Python, Scala, Java, JSON, and Cofee-Script/JavaScript, as well as a built-in web interface, Flow. H2O is designed to run in standalone mode, on Hadoop, or within a Spark Cluster, and typically deploys within minutes.

H2O includes many common machine learning algorithms, such as generalized linear modeling (linear regression, logistic regression, etc.), Naïve Bayes, principal components analysis, k-means clustering, and others. H2O also implements best-in-class algorithms at scale, such as distributed random forest, gradient boosting, and deep learning.

H2O is nurturing a grassroots movement of physicists, mathematicians, and computer scientists to herald the new wave of discovery with data science by collaborating closely with academic researchers and industrial data scientists. Stanford university giants Stephen Boyd, Trevor Hastie, Rob Tibshirani advise the H2O team on building scalable machine learning algorithms. With hundreds of meetups over the past three years, H2O has become a word-of-mouth phenomenon, growing amongst the data community by a hundred-fold, and is now used by 30,000+ users and is deployed using R, Python, Hadoop, and Spark in 2000+ corporations.

5.2) Tensorflow

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well. (https://www.tensorflow.org)

6) conclusions and achievements

1)The filtering approach has not provided clear evidence for a causal link between smoking strong cannabis and the risk of mental illness. This might be due to the fact that the non cannabis-reated predictors selected are as good predictors as cannabis related ones are.
2) Also, the cleaning part before the filtering may have slightly modified the results, since preprocessing for missing values means in fact to start from a modified dataset.
3) In almost all the resampled datasets, at least one cannabis level was selected among all the respective factors and, since collinearity was avoided from filtering and therefore keeping all the levels of each cannabis related factor was impossible, this proves the strong link between cannabis consumption and the risk of having a first episode of psychosis.
4) There are two main reasons that we may wish to estimate f: prediction and inference. While SVM gave us insights of the predictors inferences with the outcome, NNs and DLs performed greatly but as black boxes, so no actual impressions could be drawn from their results.

7) references

T. Moore, S. Zammit, A. Lingford-Hughes, et al., “Cannabis use and risk of psychotic or affective mental health outcomes: a systematic review”, The Lancet, vol. 370, no. 9584, pp. 319- 328, 2007.

M. Di Forti, C. Morgan, P. Dazzan, et al., “High-potency cannabis and the risk of psychosis”, The British Journal of Psychiatry, vol. 195, no. 6, pp. 488-491, 2009.

S. Dragt, D. Nieman, F. Schultze-Lutter, et al., “Cannabis use and age at onset of symptoms in subjects at clinical high risk for psychosis”, Acta Psychiatrica Scandinavica, vol. 125, no. 1, pp. 45-53, 2011.

M. Di Forti, A. Marconi, E. Carra, et al., “Proportion of patients in south London with first-episode psychosis attributable to use of high potency cannabis: a case-control study”, The Lancet Psychiatry, vol. 2, no. 3, pp. 233-238, 2015.

R. Radhakrishnan, S. Wilkinson and D. DSouza, “Gone to Pot: A Review of the Association between Cannabis and Psychosis”, Frontiers in Psychiatry, vol. 5, 2014.

Filippo Amato, Alberto López, Eladia María Peña-Méndez2, Petr Vaňhara3, Aleš Hampl3, 4, Josef Havel, “Artificial neural networks in medical diagnosis”, Journal of APPLIED BIOMEDICINE, 2012.

Gareth James Daniela Witten Trevor Hastie Robert Tibshirani, “An Introduction to Statistical Learning with Applications in R”, Springer Science+Business Media New York 2013 (Corrected at 6th printing 2015).

Max Kuhn, Kjell Johnson, “Applied Predictive Modeling” , Springer Science+Business Media New York 2013.

Wajdi Alghamdi, Daniel Stamate, Katherine Vang,Daniel Stahl, Marco Colizzi, Giada Tripoli, Diego Quattrone, Olesya Ajnakina, Robin M. Murray and Marta Di Forti , “A Prediction Modelling and Pattern Detection Approach for the First-Episode Psychosis Associated to Cannabis Use”

Tabea Schoeler, Natalia Petros, Marta Di Forti, Ewa Klamerus, Enrico Foglia, Olesya Ajnakina, Charlotte Gayer-Anderson, Marco Colizzi,
Diego Quattrone, Irena Behlke, Sachin Shetty, Philip McGuire, Anthony S David, Robin Murray, Sagnik Bhattacharyya, “Effects of continuation, frequency, and type of cannabis use on relapse in the first 2 years after onset of psychosis: an observational study “, Lancet Psychiatry 2016. Published Online
August 22, 2016 http://dx.doi.org/10.1016/ S2215-0366(16)30188-2

Saman Sarraf, Ghassem Tofighi, “Classification of Alzheimer’s Disease Using fMRI Data and Deep Learning Convolutional Neural Networks”, 29 Mar 2016.

Hannah Devlin, science correspondent, “Smoking skunk cannabis triples risk of serious psychotic episode, says research”, The Guardian article, Monday 16 February 2015 11.42 GMT, https://www.theguardian.com/society/2015/feb/16/skunk-cannabis-triples-risk-psychotic-episodes-study

J. Premaladha & K. S. Ravichandran, “Novel Approaches for Diagnosing Melanoma Skin Lesions Through Supervised and Deep Learning Algorithms”, Springer Science+Business Media New York 2016.

Daniele Ravì, Charence Wong, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo, and Guang-Zhong Yang, Fellow, IEEE, “Deep Learning for Health Informatics”, ieee journal of biomedical and health informatics, vol. 21, no. 1, january 2017.