*Some examples of already done work can copy from*

[*https://stackabuse.com/overview-of-classification-methods-in-python-with-scikit-learn/*](https://stackabuse.com/overview-of-classification-methods-in-python-with-scikit-learn/)

[*https://towardsdatascience.com/machine-learning-classifiers-comparison-with-python-33149aecdbca*](https://towardsdatascience.com/machine-learning-classifiers-comparison-with-python-33149aecdbca)

*mention potential weakness – not being able to select all parameters due to time, slow to run*

1. *We start with the data set (ingredients) - say where the reader can find the set, and say a little something about what they are, and the variables involved.*

*2. Then we have the data preparation and cleaning (preparing the ingredients - "peel and slice the potatoes" etc.) - describe what the reader should do to the data to get it into a form ready for analysis. This is the point where you should scale all your data so that every variable has a similar mean and variance - this is virtually required by PCA and by neural networks, for example, and I think it makes sense to do this once and for all at the start of the analysis for most tasks.*

1. *Then we have the main analysis (where we cook the meal!): explain what needs to be done to get the results that you got. If it helps your explanation to have some plots or something (think of this as like photos of the food during various stages of the cooking), then include them.*
2. *Finally, wind up with a short paragraph of conclusions at the end.*

*Introduction*

*Although a data set may be easy to find online, a reference to the source would be nice - it's about being helpful for the reader. Then I think that the introduction ought to introduce the data set a bit. Not only should you try to give a source, it is good to summarise the variables (for a reasonably small data set, a full list isn't unreasonable), and say a little about them. Mentioning the size of the data set is good. And so on - it is about getting the reader to feel involved in the project.*

*I don't mind seeing brief introductions to the methods, but you can assume that I know the methods in the course, and you may feel you can use the page limit more wisely (perhaps if you are writing for a client, or don't have a tight page limit, you might want to include something as background).*

**Introduction**

The aim of this project is to undertake various popular classification methods which are used to predict a category of a data point within Python of a dataset, through which a comparison of techniques can be made.

The techniques used in this project are:

* Logistic Regression
* Linear Discriminant Analysis (LDA and QDA)
* Decision Trees
* Random Forests
* Gradient Boosted Decision Trees
* XGBoosted Decision Trees
* Support Vector Machines (SVM)
* Neural Networks

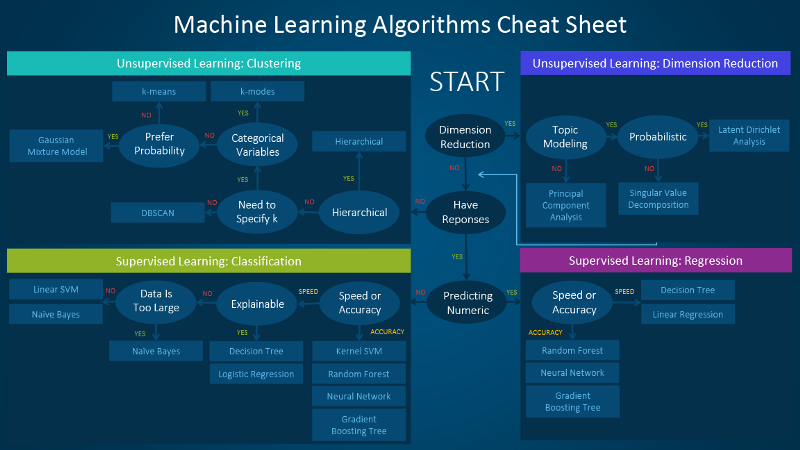
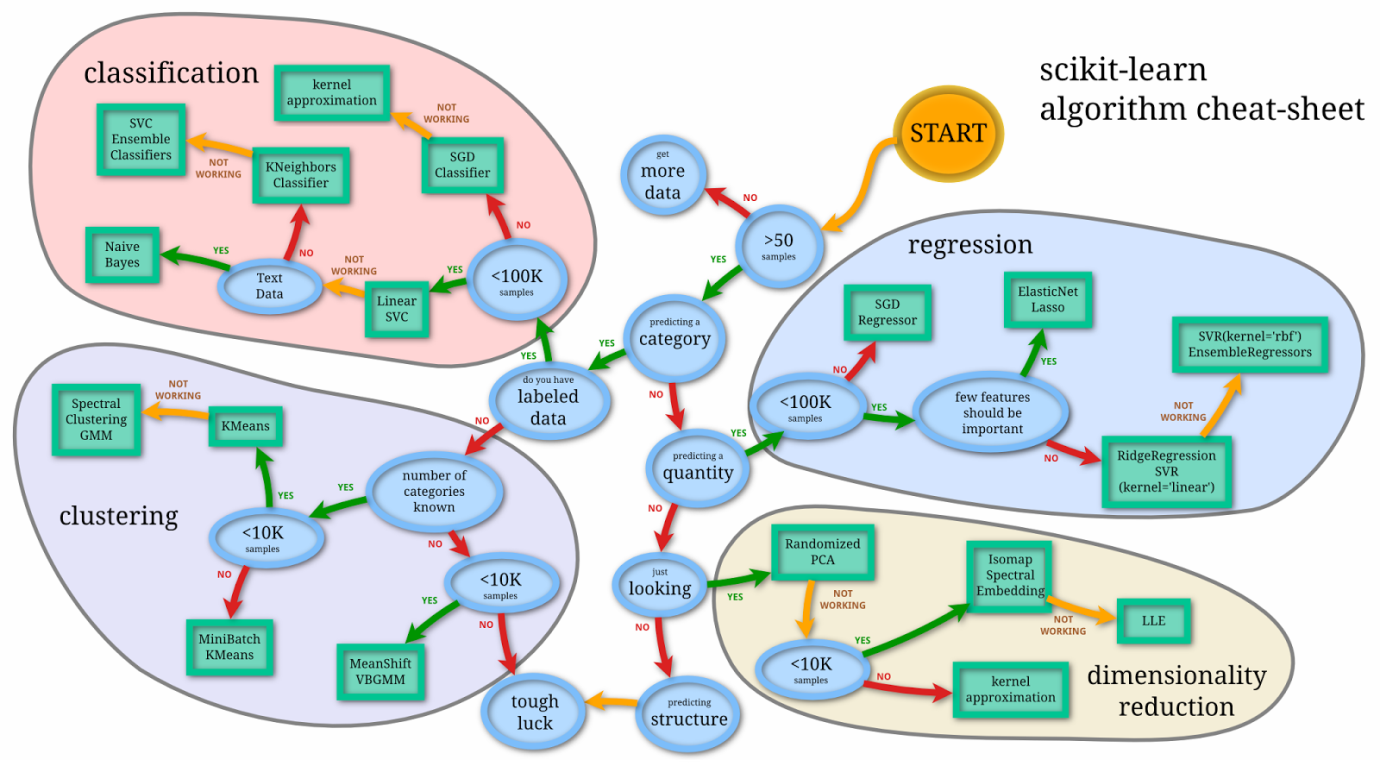
These methods will be compared using different metrics that compare their performance. In particular.

We will use the following metrics to determine the performance of the classifier:

* Accuracy: models’ ability to correctly predict both classes.
* Precision: model’s ability to correctly detect positive classes from all predicted positive classes
* Sensitivity (Recall): models’ ability to correctly detect positive classes from all actual positive classes
* F1 Score: weighted average of precision and recall (used for unbalanced problems).

For each technique a grid search is used – also known as parameter tuning i.e. trying many different parameters for each statistical technique and using the parameters for that technique that yield the best accuracy. Accuracy is used to select the best parameters in the grid search since this is a balanced problem and there is no justification to favour the detection of positive cases or negative (in some medical studies this is the case). Each technique also is carried out without the grid search serving as a comparison between grid search and non-grid-search techniques, although this is more of a side point.

Credit: <https://www.freecodecamp.org/>

Credit: peekaboo-vision.blogspot.com/

We are going to use techniques in the scikit-learn library for python. A popular cheat sheet which guides the statistician which technique best suits their dataset – in order to maintain the best possible performance.

The cheat sheet suggests using 6 of the techniques previously mentioned. This vagueness is a result of the statistician’s choice of using different techniques on a dataset in order to get the best performance. In other words, it is quite often dataset-specific as to what is the best statistical technique to use.

**The Data**

The data set consists of simulated data on high energy gamma particles in an atmospheric Cherenkov telescope, extracted from extracted from the MAGIC Gamma Telescope data set, at <https://archive.ics.uci.edu/ml/datasets/magic+gamma+te>lescope.

As particles pass through the telescope, a shower of electromagnetic radiation is produced, which are approximated by elliptical shapes. The goal is to distinguish the showers which arise from gamma particles from those which come from hadrons.

The data has 11 columns, 10 continuous variables in the first 10 columns, and a class label, g or h, in the final column.

1. Length: the major axis length of the ellipse;

2. Width: the minor axis length of the ellipse;

3. Size: the (log of the) total brightness of the ellipse;

4. Conc: a measure of concentration of the brightness;

5. Conc1: a measure of the maximum brightness to the size;

6. Asym: a measure of how far the brightest pixel is from the centre;

7. M3long: a measure of the concentration along the major axis;

8. M3Trans: a measure of the concentration along the minor axis;

9. Alpha: the angle of the major axis to the axis of the telescope;

10. Dist: the distance from the central point of the telescope to the ellipse;

11. class: either g (for a gamma particle) or h (hadron).

Distance variables (Length, Width, Asym, M3Long, M3Trans and Dist) are measured in millimetres; Size is measured in photons; Alpha is measured in degrees, and the other two numerical variables are dimensionless.

The data is balanced and therefore the Accuracy statistic can be used to judge the performance of the statistical technique.

**Data Preparation**

The data doesn’t contain any missing values and is a balance dataset. Each variable is normalised, making it’s mean and standard variance equal to 0 and 1 respectively. This is necessary to do for a neural network and a support vector machine but for consistency the standardised variables are used for all techniques.

A dataframe is created which only includes the “class” variable – whether it’s a gamma or hadron particle (what we’re trying to classify).

Another dataframe is created including all the standardised predictor variables.

Each of the dataframes’ rows are randomly split by a 75:25 ratio, forming train and test datasets respectively. The analysis will be carried out using the train datasets and evaluated using the test datasets.

**Analysis**