ML for Astronomy

Epoch X Cepheid

22nd March 2023

Random Variables

Random variables with their value resulting from the measurement of a quantity through experiments. In simple words, random variables map outcomes of an experiment to a values held by a variable.
Considering a random variable X, we represent the probability that X holds the value k as $p_X(k)$.
Random variables can take on discrete values or continuous values depending on the experiment and the way we map outcomes to values.

Probability Mass + Density Function

Consider a discrete random variable X which takes values from the set $\mathcal{X}$.
For some $k \in \mathcal{X}$, we have the probability that $X = k$ represented by the probability mass function, $p_X(k)$.
For a continuous random variable X taking values from the domain $\mathcal{X}$, one finds that the probability of X exactly being x for some $x \in \mathcal{X}$ is 0.
Hence, we define the probability density function, $p_X(k)$ as:

$$Pr(x \leq X < x + dx) = p_X(x)dx \tag{1}$$

Cumulative Distribution Function

For a random variable X, one can define the cumulative distribution function $F_X(x)$ as follows:

$$F_X(x) = Pr(X < x) \tag{2}$$

$$F_X(x) = \int_{-\infty}^{x} p_X(t)\, dt \tag{3}$$

The second equation is valid in the case when X is continuous.

Transforming Random Variables

Consider two random variables X and Y related as $Y = f(X)$. One can find the probability density function for Y, given the probability density function for X as follows:

$$p_Y(y) = p_X(f^{-1}(y))|\frac{df^{-1}(y)}{dy}| \tag{4}$$

Random Vectors

Consider a set of random variables X_1, X_2, X_3 ... X_n. One can consider a vector $X = [X_1, X_2, X_3, ...X_n]^T$, where X is a random vector. The probability that $X_1 = k_1$, $X_2 = k_2$, $X_3 = k_3$, ... $X_n = k_n$ is represented by $p_X(k_1, k_2, k_3, ..., k_n)$ or $p_{X_1,x_2,x_3,...,x_n}(k_1, k_2, k_3, ..., k_n)$.

Error Propagation

Consider a function G which takes in random variables X_1, X_2, ... X_n as input. One can compute the error from the output of the function G as follows (with or without covariance):

- Without covariance:
  $\sigma_G^2 = \sum_{i=1}^{N} (\frac{\partial G}{\partial x_i})^2 (\sigma_{x_i})^2$
- With covariance:
  $\sigma_G^2 = \sum_{i=1}^{N} (\frac{\partial G}{\partial x_i})^2 (\sigma_{x_i})^2 + 2\sigma_{X_1 X_2}^2 (\frac{\partial G}{\partial x_1})(\frac{\partial G}{\partial x_2}) + ...$

Machine Learning

- Achieving tasks without explicit code from users and programmers.
- A situation where a computer after experience in a task can do significantly better in the same task.

Machine Learning again can be split into two main types (although there are others like reinforcement learning, recommender systems) - supervised and unsupervised learning.

Machine Learning

- **Supervised Learning:** When the 'input data' is provided/labelled with the right answers. Supervised learning can be a regression problem (continuous) or a classification problem (discrete).
- **Unsupervised Learning:** When the 'input data' is unlabelled.

Machine Learning

One can find various use-applications of the above machine learning algorithms. Some of them include:

- Estimation of stellar atmospheric parameters from their Spectra
  `https://www2.mpia-hd.mpg.de/homes/calj/amla_ss2009/introduction.pdf`
- Source Classification with Images
- Galaxy Clustering
  `http://ned.ipac.caltech.edu/level5/March19/Baron/Baron3.html`

Model Representation

Consider a problem where we are supposed to predict an output value from a combination of features, given some sample data. This is our training set. Following are some terminologies associated from our training set:

- m: Number of training examples
- n: Number of features
- $\bar{x}$: Input variable (a vector with multiple features)
- y: Output variable

Every pair $(\bar{x}^{(i)}, y^{(i)})$ represents the i^{th} single training example.

Hypothesis

We now hypothesise our output value y to be a function of x which is represented as $h_\theta(x)$. θ represents the parameters or weights of the model. In case of a linear model, one can write down our hypothesis function as:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{5}$$

$$h_\theta(x) = \Theta^T X \tag{6}$$

where

- $\Theta = [\theta_0 \theta_1 ... \theta_n]^T$
- $X = [x_0 x_1 ... x_n]^T$ and $x_0 = 1$

Cost Function

Cost function is a way of evaluating the closeness of your model to the actual output values. This cost function can be represented as $J(\Theta)$.
In case of linear models, one can try to minimise the square of differences as follows:

$$J(\Theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\Theta(x^{(i)}) - y^{(i)})^2 \tag{7}$$

Our end-goal is mostly to minimise our cost function.

Gradient Descent

One algorithm to minimise the cost function would be to continuously update the feature vector Θ, based on how off we are from the actual output values.

Algorithm

Repeat Till Convergence $\{$

$$\Theta := \Theta - \alpha\delta$$

$\}$

where

- α is the learning rate
- δ to be $\nabla_\Theta J$

Normal Form

In case of linear multivariate regression, one can solve for Θ algebraically with the help of calculus. As we want to minimise the cost function J, we try to solve for:

$$\nabla_{\Theta} J = 0 \tag{8}$$

Note that one can define a vector $\bar{e}$ as follows:

$$\bar{e} = Y - X\Theta \tag{9}$$

And consequently, we can define J in terms of $\bar{e}$ as follows:

$$J = \frac{1}{2m} \bar{e}^T \bar{e} \tag{10}$$

Normal Form

We now solve for Θ as follows:

$$\nabla_\Theta(\frac{1}{2m}\bar{e}^T\bar{e}) = 0 \tag{11}$$

$$\nabla_\Theta((Y - X\Theta)^T(Y - X\Theta)) = 0 \tag{12}$$

$$\nabla_\Theta((Y^T - \Theta^T X^T)(Y - X\Theta)) = 0 \tag{13}$$

$$\nabla_\Theta(Y^T Y - \Theta^T X^T Y - Y^T X\Theta + \Theta^T X^T X\Theta) = 0 \tag{14}$$

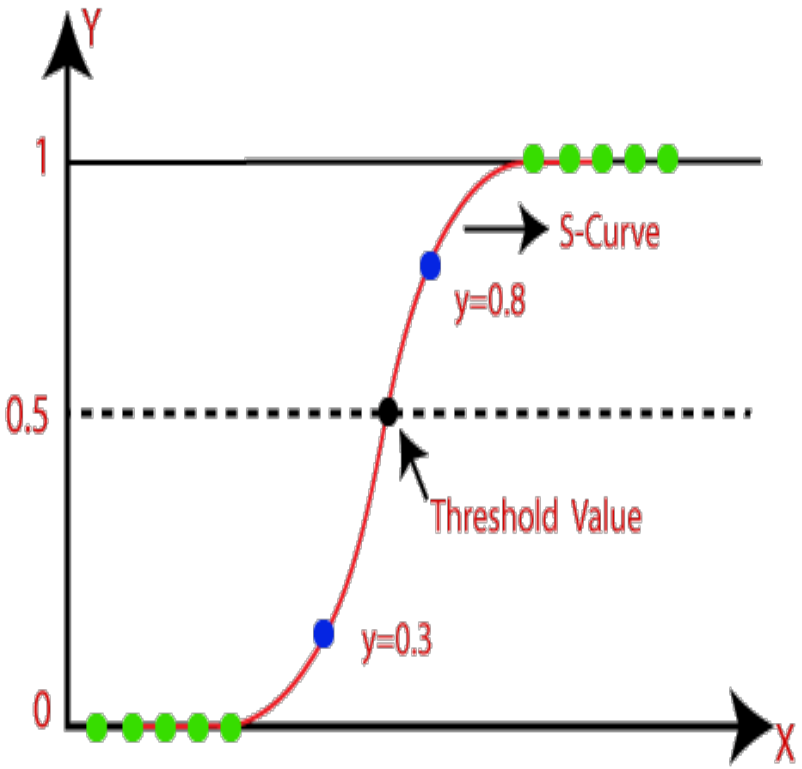
$$-2X^T Y + 2X^T X\Theta = 0 \tag{15}$$

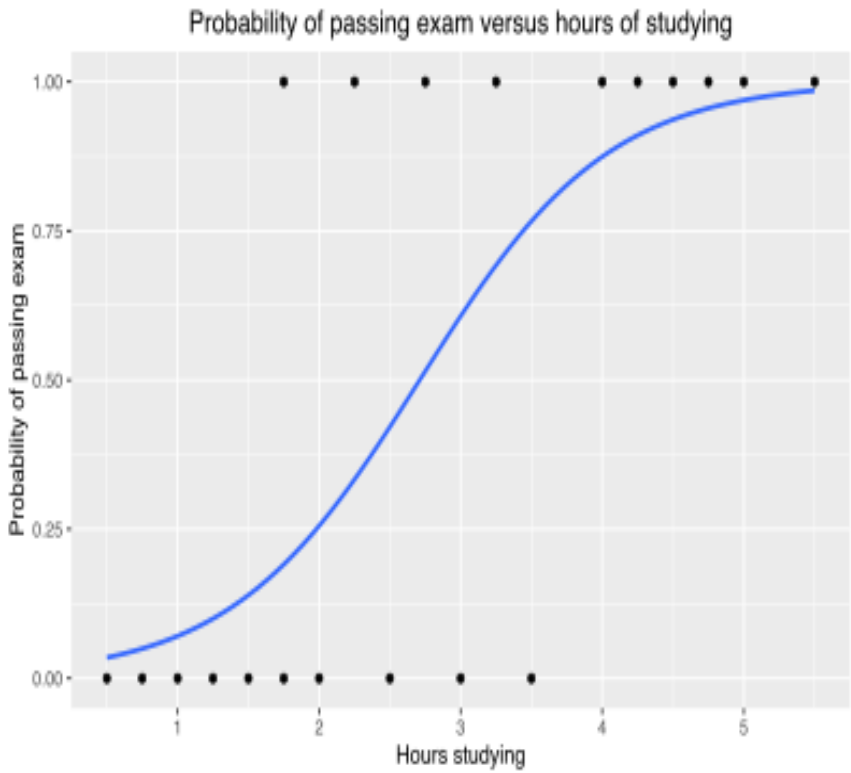
$$\Theta = (X^T X)^{-1}(X^T Y) \tag{16}$$

Logistic Regression

- Technique used in traditional statistics & ML
- Form of supervised learning
- Mainly deals with binary classification problems
- Difference b/w Linear Regression & Logistic Regression?
- Binary logistic regression & Multinomial logistic regression

Binary Logistic Regression





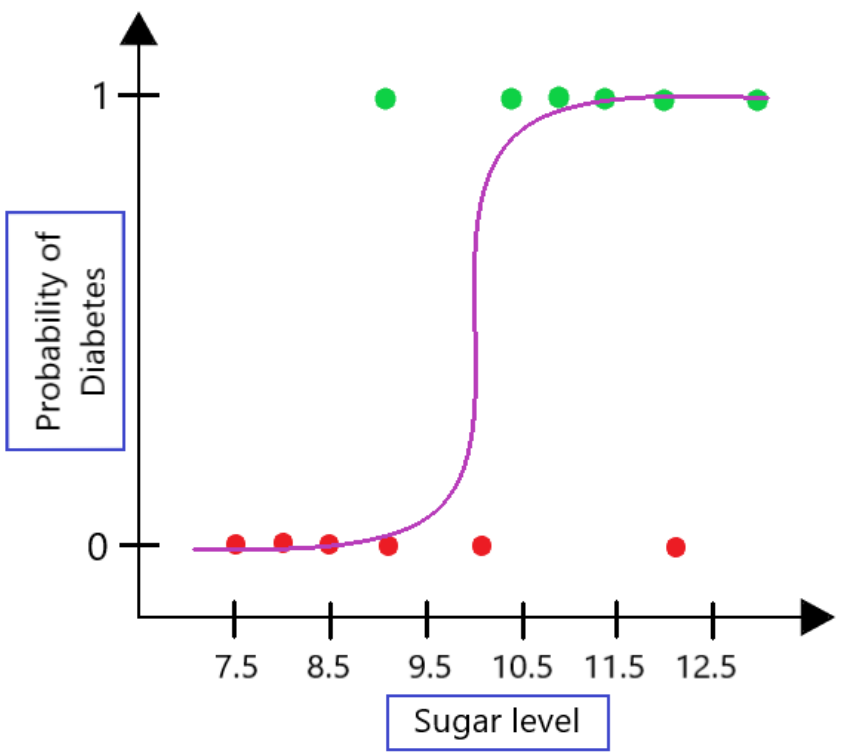
Logistic Regression

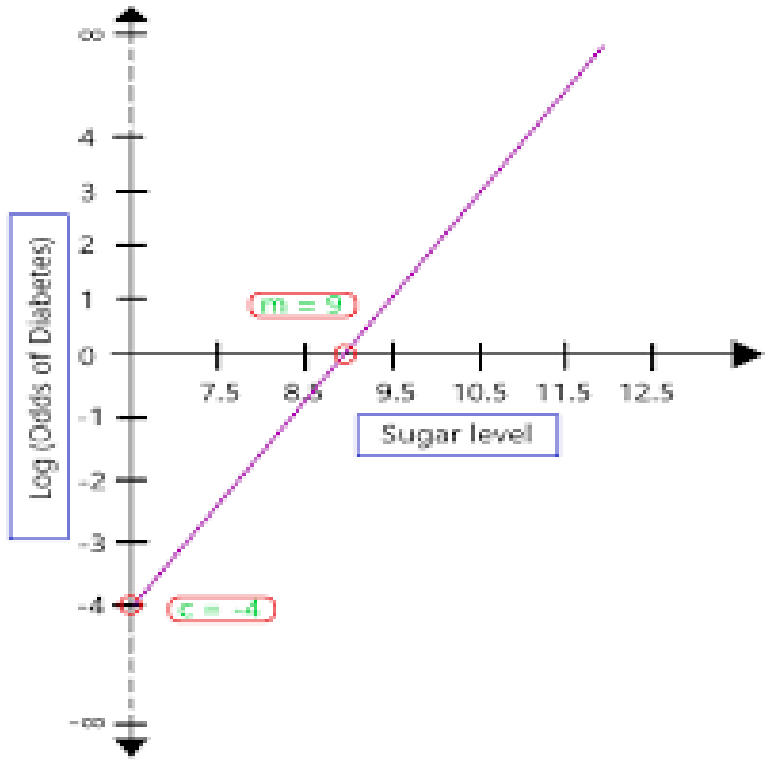
- Odds of a success, $Odds(\theta) = \frac{P(success)}{P(notsuccess)} = \frac{p}{1-p}$
- $p \in (0,1) \implies Odds(\theta) \in (0,\infty)$
- For regression analysis, we use something known as log odds, defined as $log(\frac{p(x)}{1-p(x)})$, also known as the Logit function
- Log odds plays an important role as it changes our regression analysis from probability based to likelihood based model

Maximum Likelihood Estimation(MLE)

- Logit function lies in the range of $(-\infty, \infty)$, and has a linear curve similar to linear regression of the form $y = a_1 + a_2 x$(y is the log odds)

$$p(x) = \frac{e^{(a_1 + a_2 x)}}{1 + e^{(a_1 + a_2 x)}} \tag{17}$$





Maximum Likelihood Estimation(MLE)

- We calculate the likelihood of each of the data point by projecting the point on the line and transforming it on the probability based model, which gives us the likelihood
- Then, we take the summation of all of the log of the individual likelihoods obtained, giving us the log of likelihood
- The values of the coefficients(of the line) which give us the max of log(likelihood) is taken as the solution to the algorithm

Random Forest

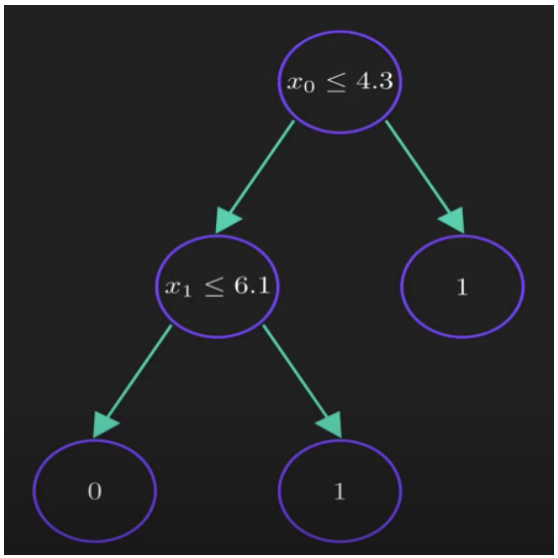
- Another form of supervised learning
- Can be used for both classification as well as regression problems
- Uses multiple decision trees to arrive at the decision(considering the majority decision)
- Greater number of decision trees leads to a higher accuracy

Decision Trees

- Decision Tree is a sequence of comparisons which is used to classify the data
- It is highly sensitive to the test data
- Difficult to generalize
- To overcome this difficulty, we used multiple decision trees, known as a forest

Decision Tree

id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1



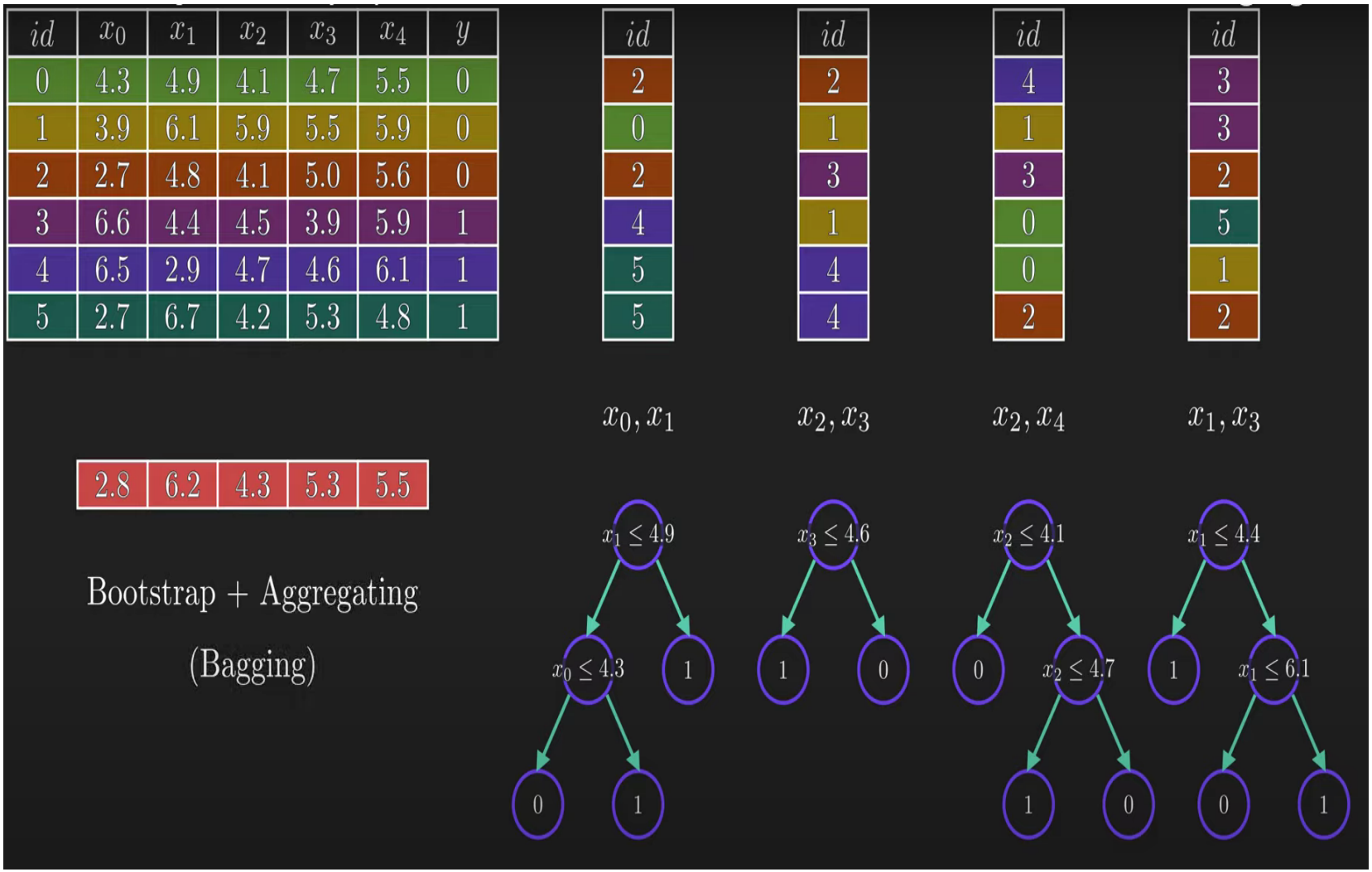
Random Forest

- For a data set, we select multiple random samples with replacement
- For each random sample, we then choose few parameters
- We construct a decision tree for each random sample for the corresponding chosen parameters
- This collection of decision trees is called random forest

Random Forest

- To classify a particular data point, we feed it into all the trees and then take the value that appears most times.
- For regression problems we take the mean of the values of the output of trees.

Random Forest



Social Media - Epoch

Follow our social media handles for regular updates

Instagram

Twitter

YouTube

WhatsApp

Any Questions or Suggestions?

Thank You!