merlin: Mixed effects regression for linear, non-linear and user-defined models

Stata Nordic and Baltic Meeting
Oslo, 12th September 2018

Michael J. Crowther

Biostatistics Research Group,
Department of Health Sciences,
University of Leicester, UK,
michael.crowther@le.ac.uk
@Crowther_MJ

Funding: MRC (MR/P015433/1)

the plan

- the motivation
- the past
- the goal
- the example
- the family
- the surprise (at least it was last week)
- the future

the motivation

- More data → more questions
  - need for appropriate statistical modelling techniques, and implementations

the motivation

- More data → more questions
  - need for appropriate statistical modelling techniques, and implementations
- Growth in access to EHR
  - biomarkers $<$ patients $<$ GP practice area $<$ geographical regions...

the motivation

- More data → more questions
    - need for appropriate statistical modelling techniques, and implementations
- Growth in access to EHR
    - biomarkers < patients < GP practice area < geographical regions...
- The standard challenges
    - time-dependent effects, non-linear covariate effects

the motivation

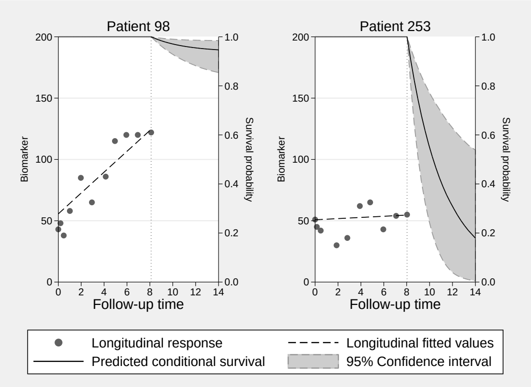
- More data → more questions
  - need for appropriate statistical modelling techniques, and implementations
- Growth in access to EHR
  - biomarkers $<$ patients $<$ GP practice area $<$ geographical regions...
- The standard challenges
  - time-dependent effects, non-linear covariate effects
- The neglected challenges
  - Within-patient variability
  - Informative observations times

the motivation

- More data → more questions
  - need for appropriate statistical modelling techniques, and implementations
- Growth in access to EHR
  - biomarkers $<$ patients $<$ GP practice area $<$ geographical regions...
- The standard challenges
  - time-dependent effects, non-linear covariate effects
- The neglected challenges
  - Within-patient variability
  - Informative observations times

We need modelling frameworks that can accommodate a lot of different things

Joint longitudinal-survival models



Linking via - current value, gradient, AUC, random effects...

Joint longitudinal-survival models - extensions

- Competing risks
- Different types of outcomes
- Multiple continuous outcomes
- Delayed entry
- Recurrent events and a terminal event
- Prediction
- Many others...

Joint longitudinal-survival models - software

- stjm in Stata
- gsem in Stata
- frailtypack in R
- joineR in R
- JM and JMBayes in R
- Many others...

(My) Methods development - software

- stjm - joint longitudinal-survival models
- stmixed - multilevel survival models
- stgenreg - general parametric survival models
- ...

(My) Methods development - software

- stjm - joint longitudinal-survival models
- stmixed - multilevel survival models
- stgenreg - general parametric survival models
- ...

Each new project brings a new code base to maintain...could I make my life easier?

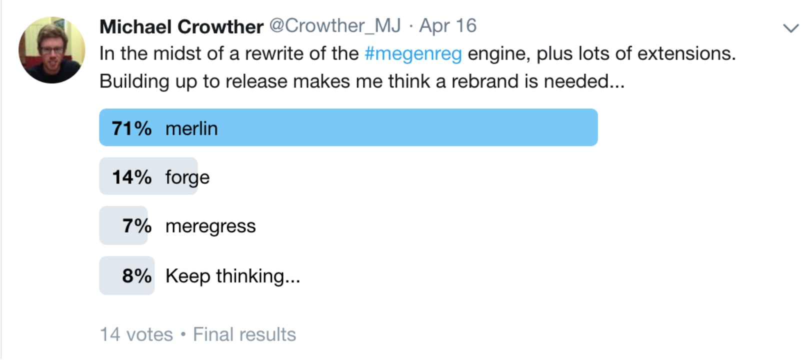
the past

- last year I introduced `megenreg`
- `megenreg` fitted mixed effects generalised regression models
- `megenreg` was awesome...but

the past

- last year I introduced `megenreg`
- `megenreg` fitted mixed effects generalised regression models
- `megenreg` was awesome...but

I really hated the name



Michael Crowther @Crowther_MJ · Apr 16

In the midst of a rewrite of the #megenreg engine, plus lots of extensions. Building up to release makes me think a rebrand is needed...

71% merlin

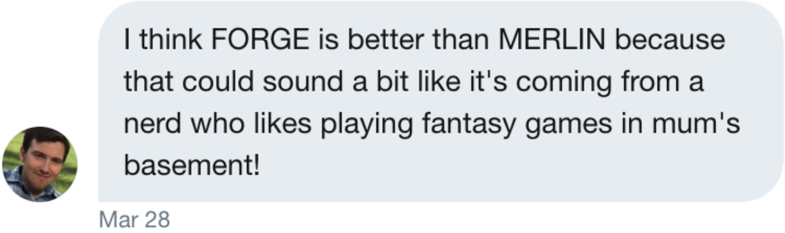
14% forge

7% meregress

8% Keep thinking...

14 votes • Final results

Some people were not so keen...



I think FORGE is better than MERLIN because that could sound a bit like it's coming from a nerd who likes playing fantasy games in mum's basement!

Mar 28

Mixed Effects Regression for LInear, Non-linear and user-defined models

merlin

the goal

- multiple outcomes of varying types
- measurement schedule can vary across outcomes
- any number of levels and random effects
- sharing and linking random effects between outcomes
- sharing functions of the expected value of other outcomes
- a reliable estimation engine
- easily extendable by the user
- ...

a unified framework for data analysis and methods development

the example

- there's no equations in this talk
- there's 14 models
- each of them is applied to the same dataset
- most of them can be considered *new* models
- we can fit all of them with a single line of code

- data from 312 patients with PBC collected at the Mayo Clinic 1974-1984 (Murtaugh et al. (1994))
- 158 randomised to receive D-penicillamine and 154 to placebo
- survival outcome is all-cause death, with 140 events observed
    - we're going to pretend we have competing causes of death - cancer and other causes
- 1945 measurements of serum bilirubin, among other things

the data

```
  +-----------------------------------------------------------------------------+
  | id      time       logb   prothr~n         trt     stime   cancer  other |
  |-----------------------------------------------------------------------------|
  |  1         0   2.674149       12.2   D-penicil   1.09517        1      0 |
  |  1   .525682   3.058707       11.2   D-penicil         .        .      . |
  |-----------------------------------------------------------------------------|
  |  2         0   .0953102       10.6   D-penicil   14.1523        0      1 |
  |  2   .498302  -.2231435         11   D-penicil         .        .      . |
  |  2   .999343          0       11.6   D-penicil         .        .      . |
  |  2   2.10273   .6418539       10.6   D-penicil         .        .       . |
  |  2   4.90089   .9555114       11.3   D-penicil         .        .       . |
  |  2   5.88928   1.280934       11.5   D-penicil         .        .      . |
  |  2   6.88588   1.435084          .   D-penicil         .        .       . |
  |  2    7.8907   1.280934          .   D-penicil         .        .       . |
  |  2   8.83255   1.526056          .   D-penicil         .        .       . |
  +-----------------------------------------------------------------------------+
```

a model

```
merlin (logb                                  /// log serum bilirubin
              time                            /// covariate
              ,                               /// options
              family(gaussian)                /// distribution
       )
```

a model

```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
```

a model

```
merlin (logb                            /// log serum bilirubin
               time                     /// covariate
               time#trt                 /// interaction
               M1[id]@1                 /// random intercept
               ,                        /// options
               family(gaussian)         /// distribution
       )                                ///
```

a model

```
merlin (logb                                    /// log serum bilirubin
               time                             /// covariate
               time#trt                         /// interaction
               M1[id]@1                         /// random intercept
               time#M2[id]@1                    /// random slope
               ,                                /// options
               family(gaussian)                 /// distribution
       )
```

a model

```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               M1[id]@1                     /// random intercept
               time#M2[id]@1                /// random slope
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
       (pro                                 /// prothrombin index
               rcs(time, df(3))             /// covariate
               , family(gamma)              /// distribution
       )                                    ///
```

a model

```
merlin (logb                            /// log serum bilirubin
               time                     /// covariate
               time#trt                 /// interaction
               M1[id]@1                 /// random intercept
               time#M2[id]@1            /// random slope
               ,                        /// options
               family(gaussian)         /// distribution
       )                                ///
       (pro                             /// prothrombin index
               rcs(time, df(3))         /// covariate
               M3[id]@1                 /// random effect
               , family(gamma)          /// distribution
       )                                ///
```

a model

```
merlin (logb                            /// log serum bilirubin
               time                     /// covariate
               time#trt                 /// interaction
               M1[id]@1                 /// random intercept
               time#M2[id]@1            /// random slope
               ,                        /// options
               family(gaussian)         /// distribution
       )                                ///
       (pro                             /// prothrombin index
               rcs(time, df(3))         /// covariate
               M3[id]@1                 /// random effect
               , family(gamma)          /// distribution
       )                                ///
       ,                                /// main options
       covariance(unstructured)         //  vcv
```

a model

```
merlin (logb                                      /// log serum bilirubin
               time                               /// covariate
               time#trt                           /// interaction
               M1[id]@1                           /// random intercept
               time#M2[id]@1                      /// random slope
               ,                                  /// options
               family(gaussian)                   /// distribution
       )                                          ///
       (pro                                       /// prothrombin index
               rcs(time, df(3))                   /// covariate
               M3[id]@1                           /// random effect
               , family(gamma)                    /// distribution
       )                                          ///
       ,                                          /// main options
       covariance(unstructured)                   /// vcv
       redistribution(t) df(5)                    //  re dist.
```

a model

```
merlin (logb                            /// log serum bilirubin
               time                     /// covariate
               time#trt                 /// interaction
               M1[id]@1                 /// random intercept
               time#M2[id]@1            /// random slope
               ,                        /// options
               family(gaussian)         /// distribution
       )                                ///
       (pro                             /// prothrombin index
               rcs(time, df(3))         /// covariate
               M3[id]@1                 /// random effect
               , family(gamma)          /// distribution
       )                                ///
       (stime  trt                      /// response + covariate
              , family(rp, df(3)        /// distribution
                        failure(other)) /// event indicator
       )                                ///
       ,                                /// main options
       covariance(unstructured)         /// vcv
       redistribution(t) df(5)          //  re dist.
```

a model

```
merlin (logb                            /// log serum bilirubin
               time                     /// covariate
               time#trt                 /// interaction
               M1[id]@1                 /// random intercept
               time#M2[id]@1            /// random slope
               ,                        /// options
               family(gaussian)         /// distribution
       )                                ///
       (pro                             /// prothrombin index
               rcs(time, df(3))         /// covariate
               M3[id]@1                 /// random effect
               , family(gamma)          /// distribution
       )                                ///
       (stime  trt                      /// response + covariate
               dEV[logb] EV[pro]        /// associations
               , family(rp, df(3)       /// distribution
                        failure(other)) /// event indicator
       )                                ///
       ,                                /// main options
       covariance(unstructured)         /// vcv
       redistribution(t) df(5)          //  re dist.
```

a model

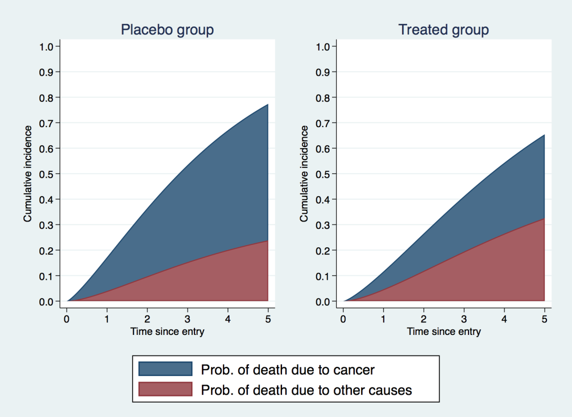
```
merlin (logb                            /// log serum bilirubin
               time                     /// covariate
               time#trt                 /// interaction
               M1[id]@1                 /// random intercept
               time#M2[id]@1            /// random slope
               ,                        /// options
               family(gaussian)         /// distribution
       )                                ///
       (pro                             /// prothrombin index
               rcs(time, df(3))         /// covariate
               M3[id]@1                 /// random effect
               , family(gamma)          /// distribution
       )                                ///
       (stime  trt                      /// response + covariate
               trt#fp(stime, power(0))  /// tde
               dEV[logb] EV[pro]        /// associations
               , family(rp, df(3)       /// distribution
                        failure(other)) /// event indicator
       )                                ///
       ,                                /// main options
       covariance(unstructured)         /// vcv
       redistribution(t) df(5)          //  re dist.
```

a model

```
merlin (logb time time#trt M1[id]@1        /// model 1
               time#M2[id]@1 ,             ///
               family(gaussian)            ///
       )                                   ///
       (pro  rcs(time, df(3)) M3[id]@1     /// model 2
               , family(gamma)             ///
       )                                   ///
       (stime  trt                         ///
               trt#fp(stime, power(0))     /// model 3 - cause 1
               dEV[logb] EV[pro]           /// tde
               , family(rp, df(3)          /// distribution
                        failure(other))    /// event indicator
       )                                   ///
       (stime  trt                         /// model 4 - cause 2
               trt#rcs(stime, df(3) log)   /// tde
               EV[logb] iEV[pro]           /// associations
               , family(weibull,           /// distribution
                        failure(cancer))   /// event indicator
       )                                   ///
       ,                                   ///
       covariance(unstructured)
```

predictions

```
predict cif1, cif marginal outcome(3) at(trt 0)
predict cif1, cif marginal outcome(4) at(trt 0)
```



a user-defined model

```
real matrix gauss_logl(gml)
{
     y        = merlin_util_depvar(gml)         // dep. var.
     linpred  = merlin_util_xzb(gml)            // lin. pred.
     sdre     = exp(merlin_util_ap(gml,1))      // anc. param.
     return(lnnormalden(y,linpred,sdre))       // logl
}

merlin (logb ... , family(user, llfunction(gauss_logl) nap(1)))
       ...
       ...
       ...
```

a user-defined model

```
real matrix gauss_logl(gml)
{
     y         = merlin_util_depvar(gml)          // dep. var.
     linpred   = merlin_util_xzb(gml)             // lin. pred.
     sdre      = exp(merlin_util_xzb_mod(gml,2)) // anc. param.
     return(lnnormalden(y,linpred,sdre))       // logl
}

merlin (logb ...    , family(user, llfunction(gauss_logl)))
       (age M1[id]@1, family(null))
       ...
       ...
```

a user-defined nonlinear model - Yulia's talk

```
webuse orange, clear
menl circumf = (b1+U1[tree])/(1+exp(-(age-b2)/b3))

mata:
real matrix logl(transmorphic gml)
{
    y    = merlin_util_depvar(gml)

    b1   = merlin_util_xzb(gml)
    b2   = merlin_util_xzb_mod(gml,2)
    b3   = merlin_util_xzb_mod(gml,3)

    sdre = exp(merlin_util_ap(gml,1))

    xb   = b1 :/ (1 :+ exp(-b2 :/ b3))

    return(lnnormalden(y,xb,sdre))
}
end

merlin (circumf M1[tree]@1, family(user, llf(logl) nap(1)))
       ( age@1            , family(null))
       (                  , family(null))
```

stuff I didn't show

- random effects at arbitrary levels - `M4[centre>id]@1`
- B-splines - `bs(time, df(3) order(4))`
- `d2EV[]`, `?XB[]`
- `linterval(varname)` - interval censoring
- `ltruncated(varname)` - left-truncation
- 9 (so far) other inbuilt families, e.g. `beta`, `ologit`
- `bhazard(varname)` - relative survival
- `mf(func_name)` - user-defined element function

the family

- `merlin`'s syntax is not simple
- we can develop more user-friendly shell files to allow a simpler syntax for special cases
- `merlin`'s minions...
    - `excalibur` (`stmixed`) for multilevel survival analysis (SJ under revision)
    - `lancelot` - meta-analysis
    - `arthur` - to be revealed next!
    - `galahad` - maybe next year
    - ...

the surprise

Two useful features of `merlin` are:

- `EV[depvar/#]` element type
  - implemented for their use in joint longitudinal-survival models
- `family(null)`
  - implemented for use with user-defined models

their combination gives merlin some new capabilities

the surprise

```
merlin (y x1 x2 EV[2] EV[3], family(bernoulli) link(logit))
       (x1 x2, family(null) link(logit))
       (x1 x2, family(null) link(logit))
```

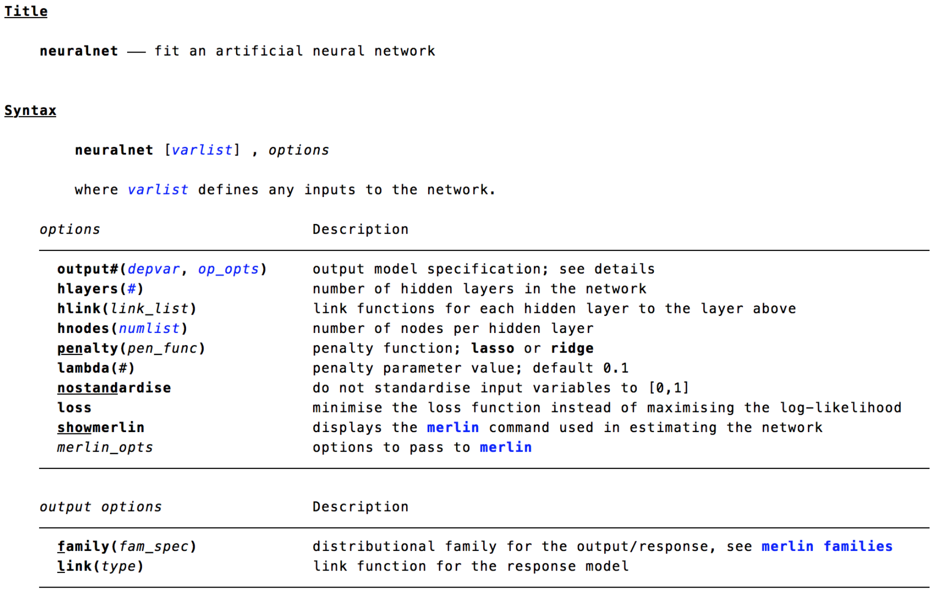
any idea what this is?

the surprise

```
merlin (y x1 x2 EV[2] EV[3], family(bernoulli) link(logit))
       (x1 x2, family(null) link(logit))
       (x1 x2, family(null) link(logit))
```

any idea what this is?

It's an artificial neural network!



Title

neuralnet — fit an artificial neural network

Syntax

neuralnet [*varlist*] , *options*

where *varlist* defines any inputs to the network.

options	Description
output#(*depvar*, *op_opts***)**	output model specification; see details
hlayers(*#***)**	number of hidden layers in the network
hlink(*link_list***)**	link functions for each hidden layer to the layer above
hnodes(*numlist***)**	number of nodes per hidden layer
penalty(*pen_func***)**	penalty function; **lasso** or **ridge**
lambda(#)	penalty parameter value; default 0.1
nostandardise	do not standardise input variables to [0,1]
loss	minimise the loss function instead of maximising the log-likelihood
showmerlin	displays the **merlin** command used in estimating the network
merlin_opts	options to pass to **merlin**

output options	Description
family(*fam_spec***)**	distributional family for the output/response, see **merlin families**
link(*type***)**	link function for the response model

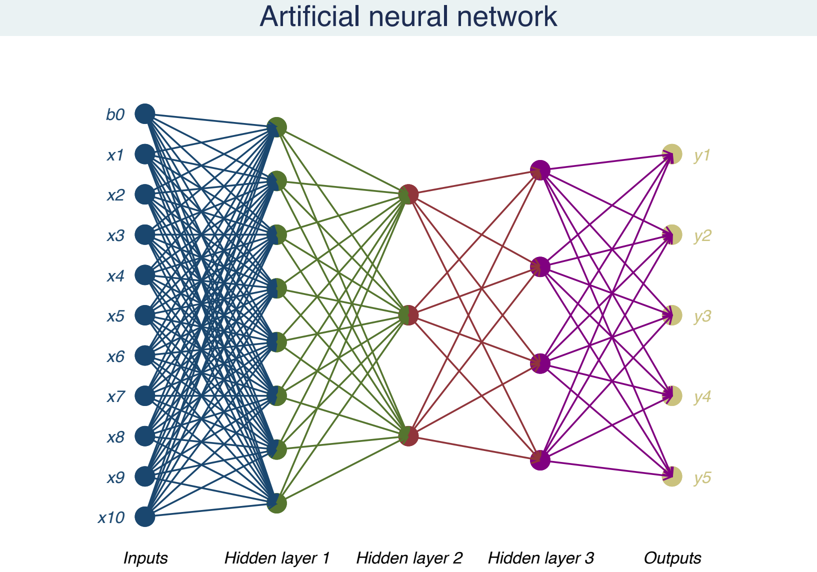
the surprise

```
merlin (y x1 x2 EV[2] EV[3], family(bernoulli) link(logit))
       (x1 x2, family(null) link(logit))
       (x1 x2, family(null) link(logit))

neuralnet x1 x2, output1(y, family(bernoulli) link(logit))
                 hlayers(1) hlink(logit) hnodes(2)
                 penalty(ridge) lambda(1e-07)
```

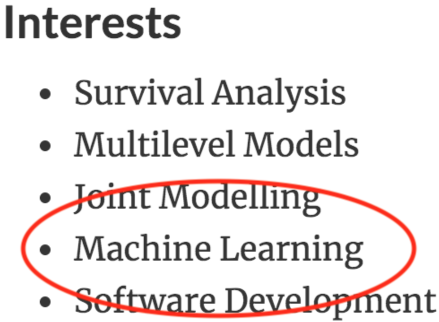
the surprise

```
merlin (y x1_nn x2_nn EV[4] EV[5] EV[6]
                           , family(bernoulli) link(logit))
       (x1_nn x2_nn, family(null) link(atanh))
       (x1_nn x2_nn, family(null) link(atanh))
       (EV[2] EV[3], family(null) link(atanh))
       (EV[2] EV[3], family(null) link(atanh))
       (EV[2] EV[3], family(null) link(atanh))

neuralnet x1 x2, output1(y, family(bernoulli) link(logit))
                 hlink(atanh) hlayers(2) hnodes(2 3)
                 penalty(lasso) lambda(1e-07)
```

```
. nnplot , inputs(10) outputs(5) hlayers(3) hnodes(8 3 4)
```



From my website - I'm now a data scientist!



Interests

- Survival Analysis
- Multilevel Models
- Joint Modelling
- Machine Learning
- Software Development

the future

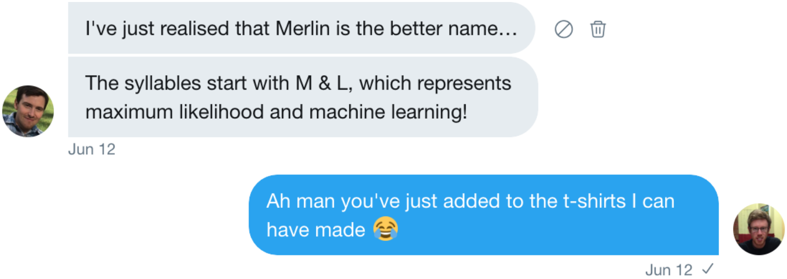
- merlin can do a lot of things, hopefully in a usable way
- merlin is easily extended
- I continue to discover more and more things it can do
- arthur (neuralnet)
  - It's a rubbish implementation of neural networks
  - Needs analytic gradients to be useful
  - penalisation
  - But - all capabilities of merlin can be used in a neural network, and vice versa
  - predict *newvar*, *statistic* ci

www.mjcrowther.co.uk/software/merlin

the papers

- Extended multivariate generalised linear and non-linear mixed effects models. https://arxiv.org/abs/1710.02223
- merlin - a unified framework for data analysis and methods development in Stata. https://arxiv.org/abs/1806.01615
- Multilevel mixed effects parametric survival analysis. https://arxiv.org/abs/1709.06633
- Deep learning neural networks and regression modelling: A general penalised likelihood framework for estimation, prediction and quantifying uncertainty. (In Prep.)

the reversal



I've just realised that Merlin is the better name...

The syllables start with M & L, which represents maximum likelihood and machine learning!

Jun 12

Ah man you've just added to the t-shirts I can have made 😂

Jun 12 ✓