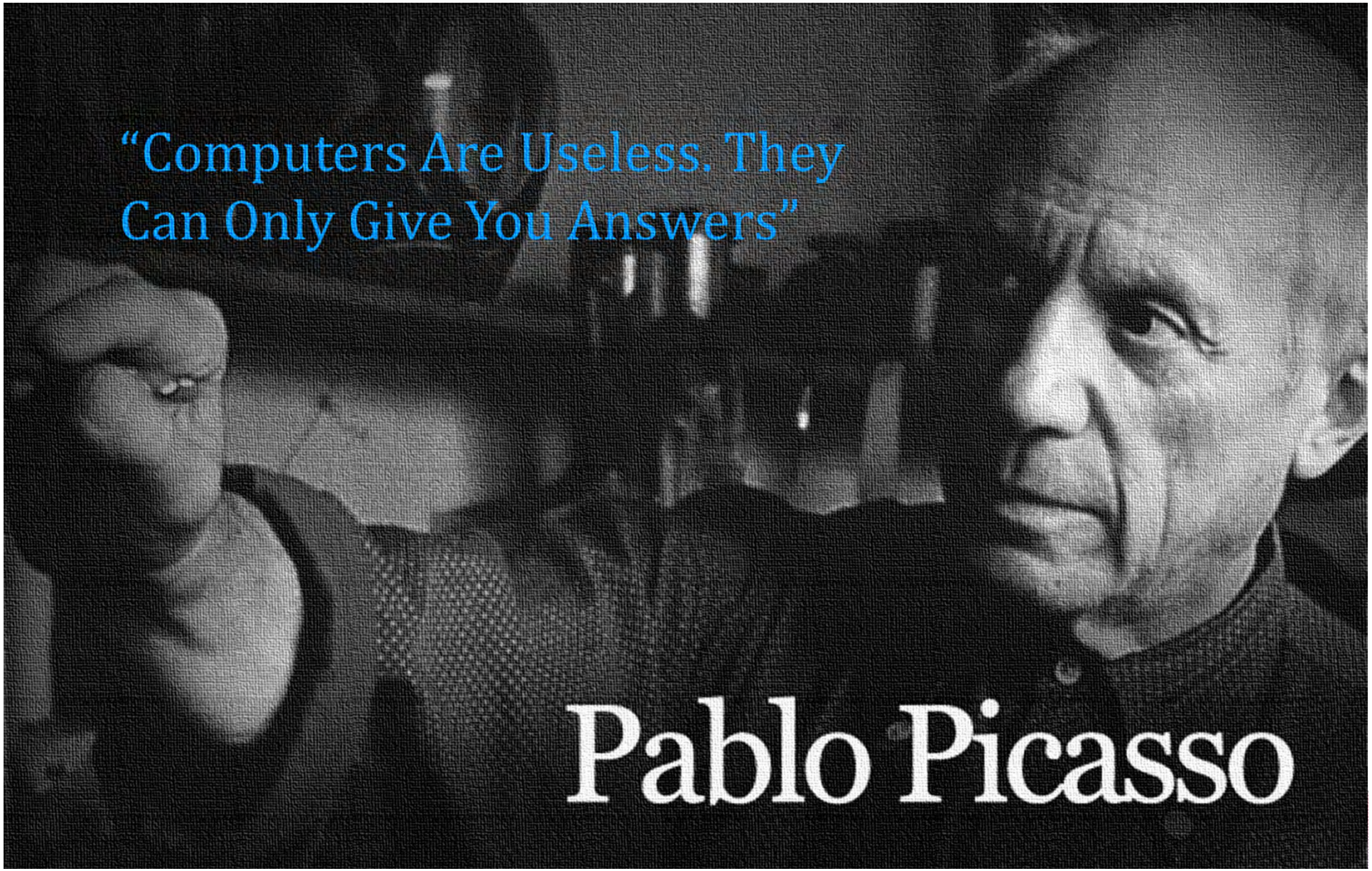
Opening the Black Box with R

JB Crozet & James Simmons

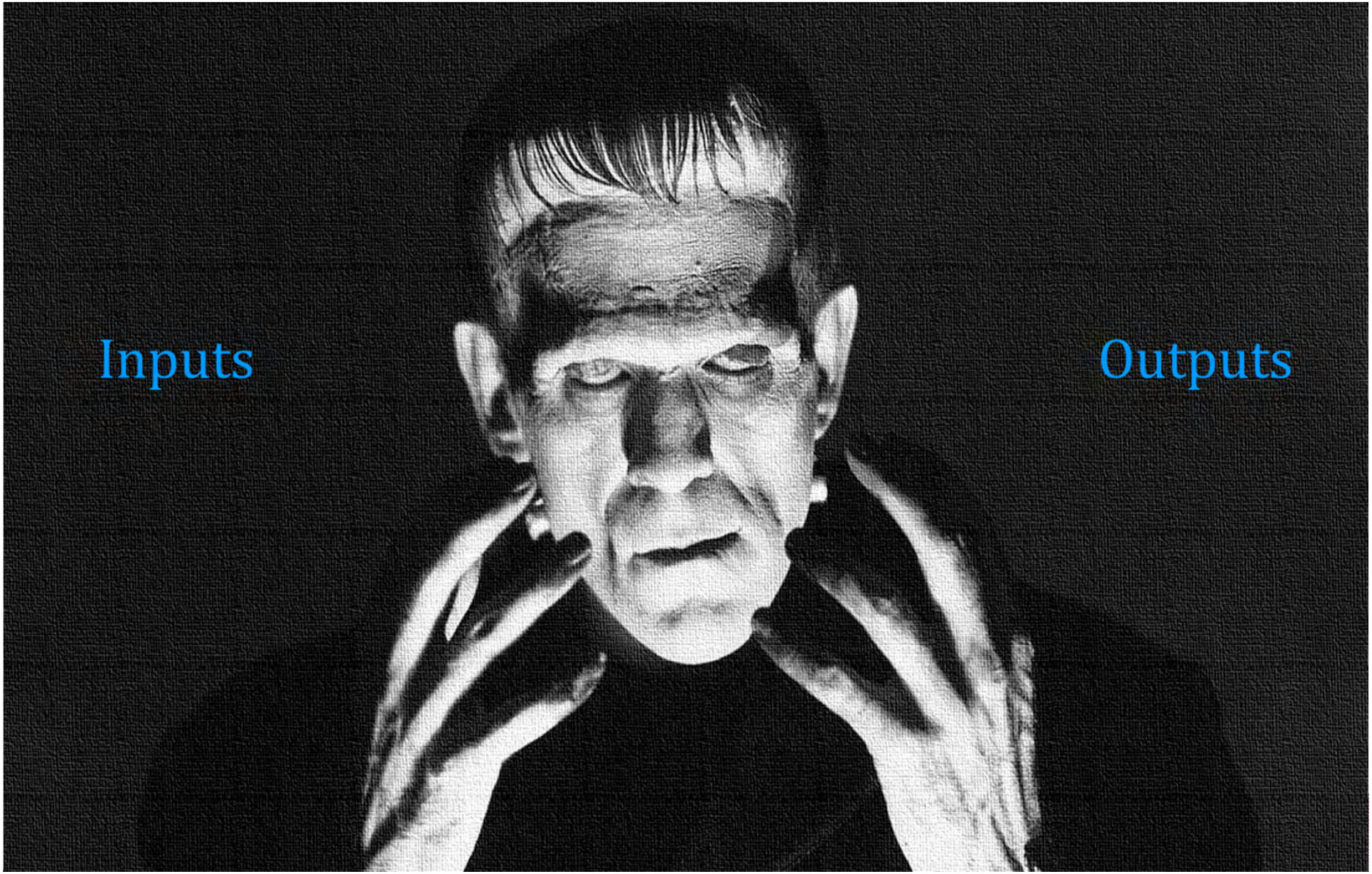
8th June 2017

We have a problem.

MS amlin



We created a man-made monster.



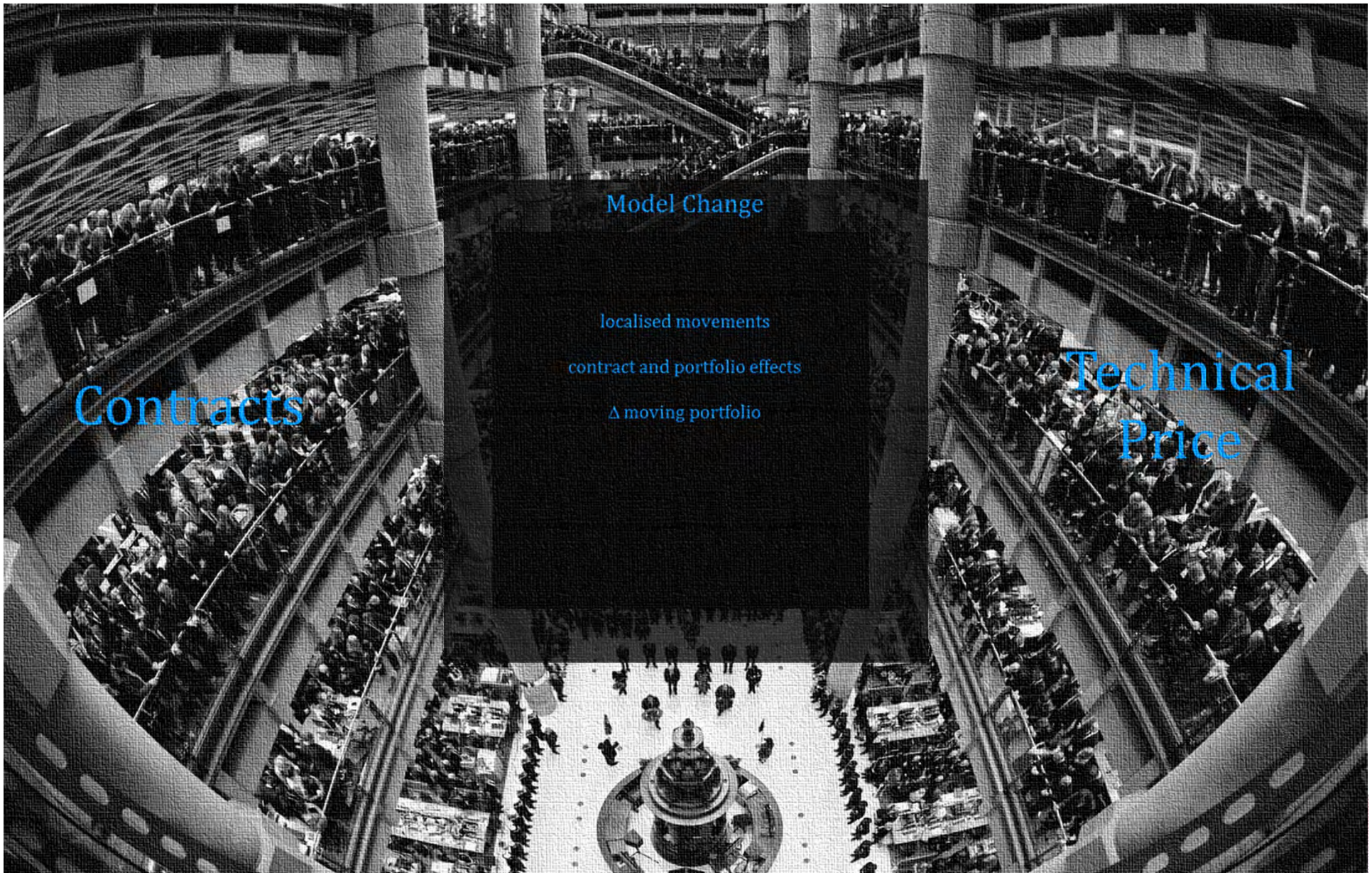
How we did it (1).

How we did it (2).

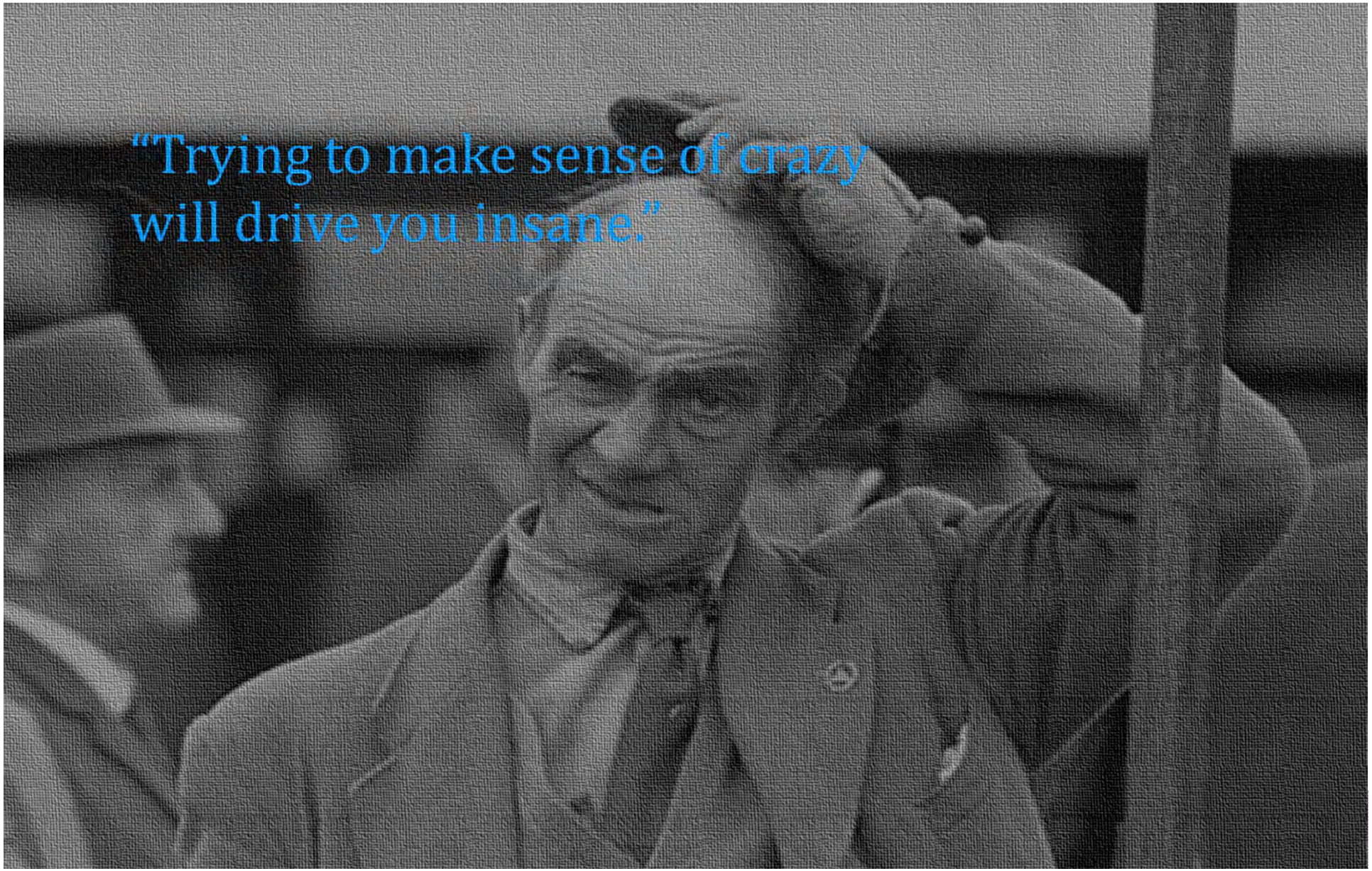
MS Amlin

How we did it (3).

How we did it (4).

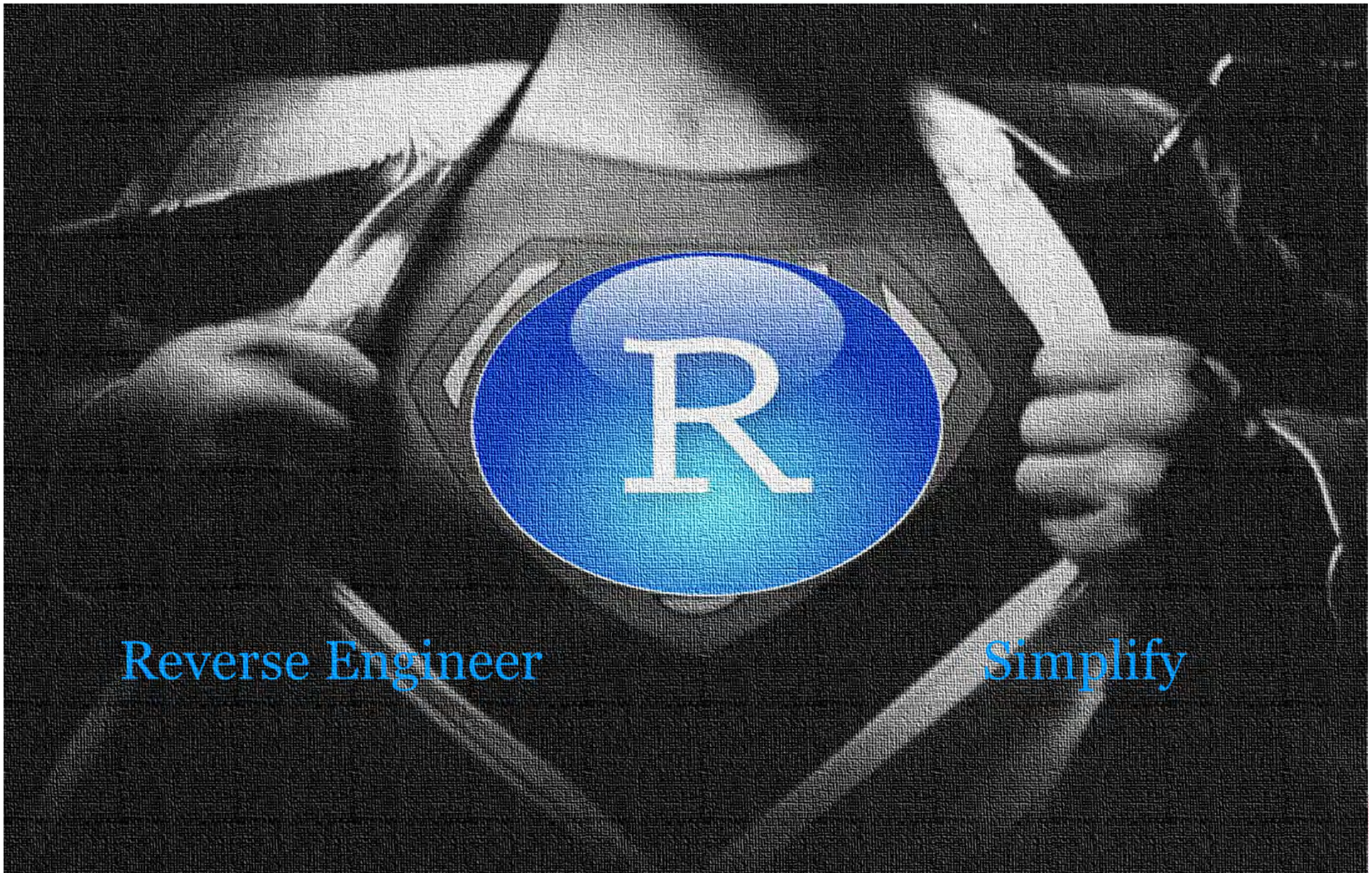


Where we got to.



R to the Rescue.

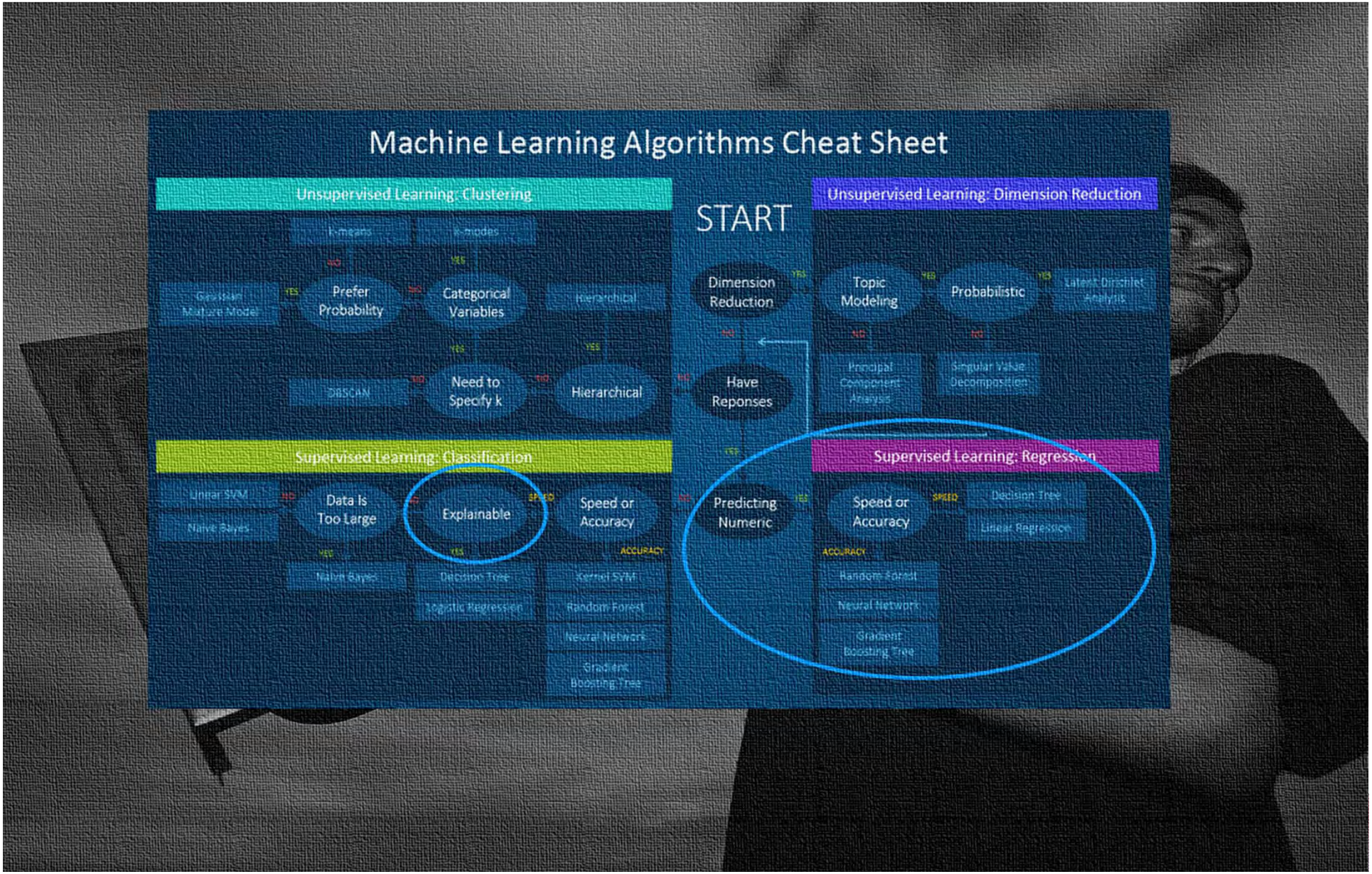
MS Amlin



The Kitchen Sink of Machine Learning? (1)

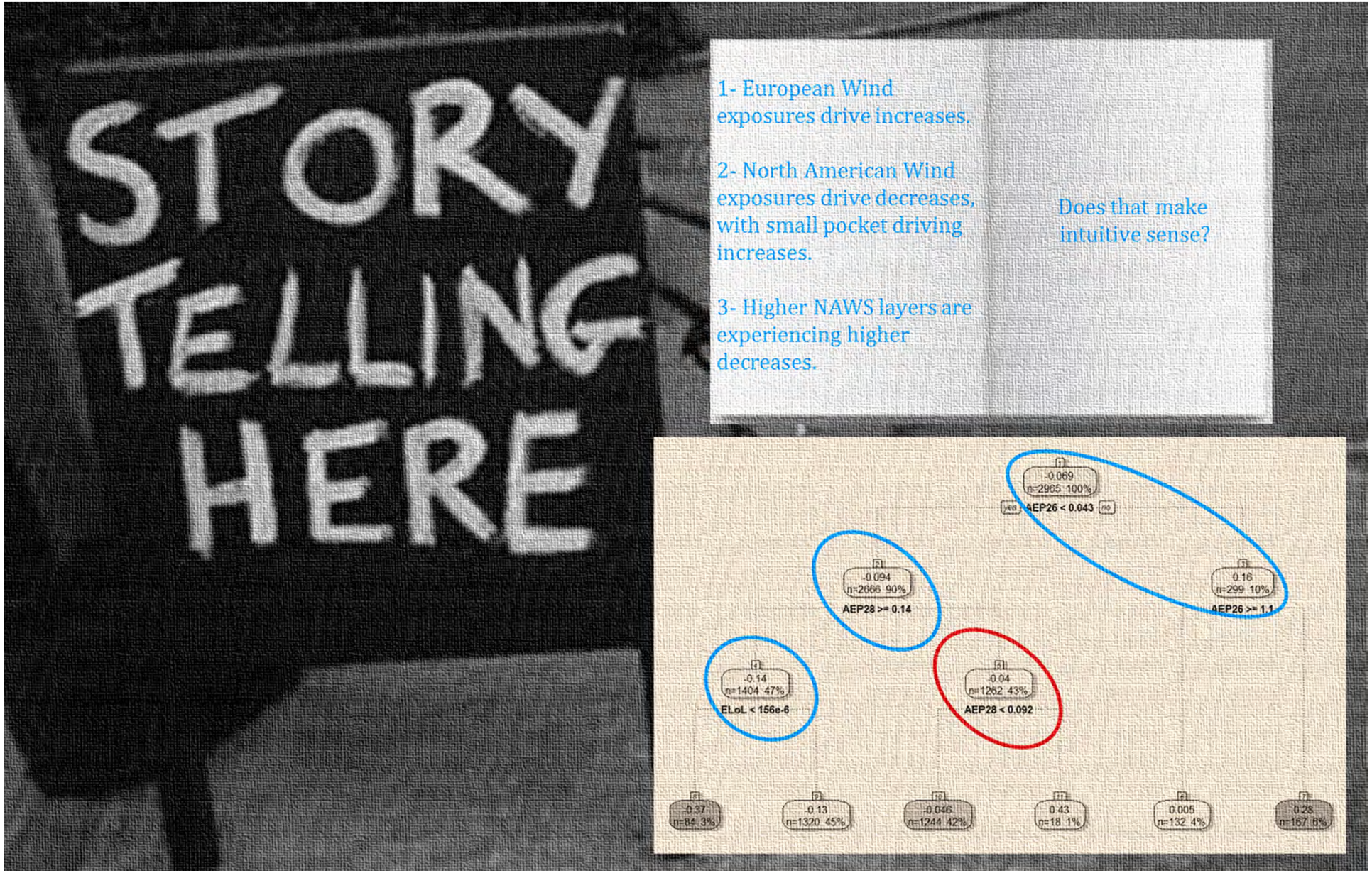
MS Amlin

The Kitchen Sink of Machine Learning? (2)



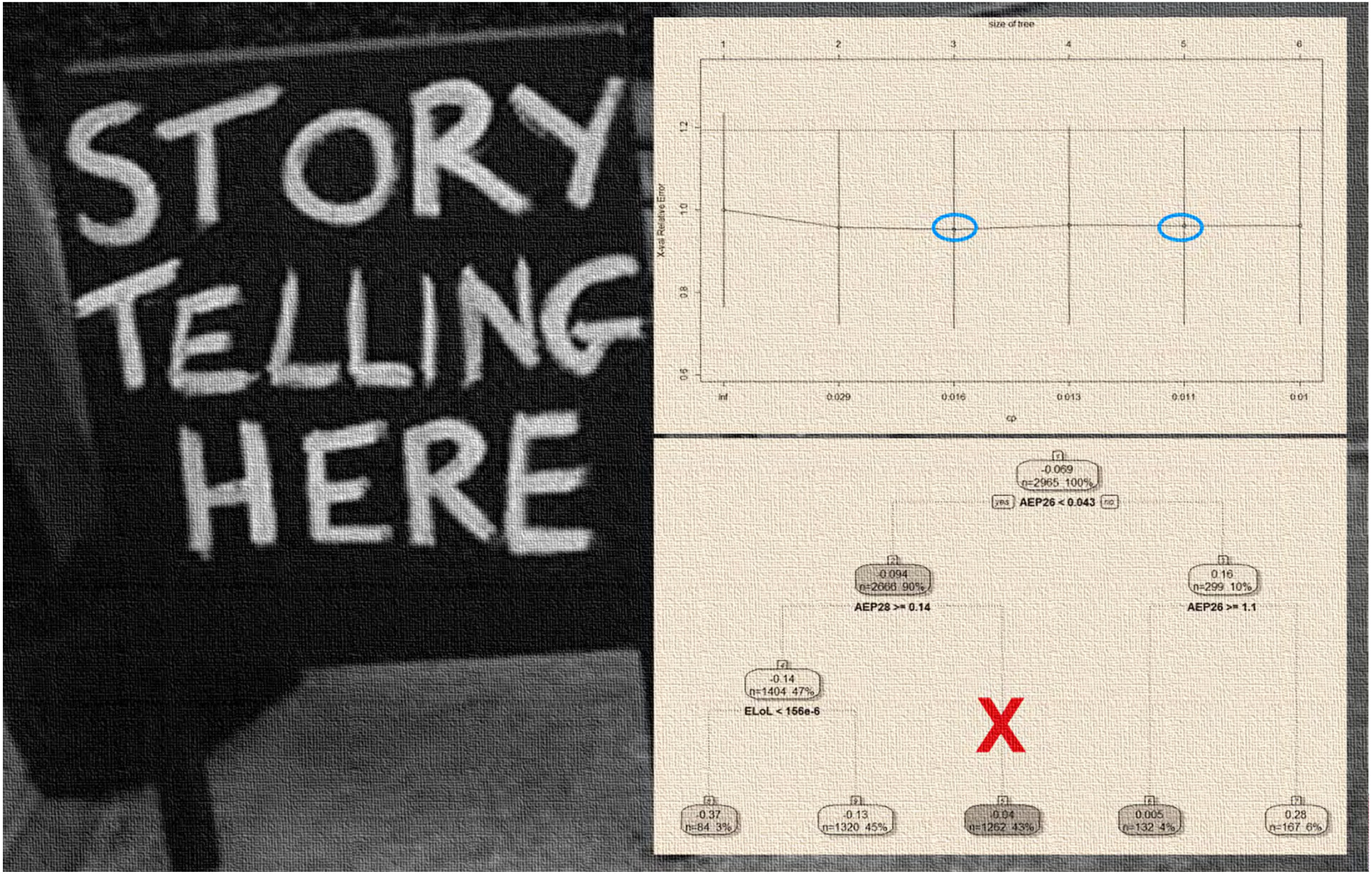
Telling Stories with Pruned Trees (1).

MS Amlin

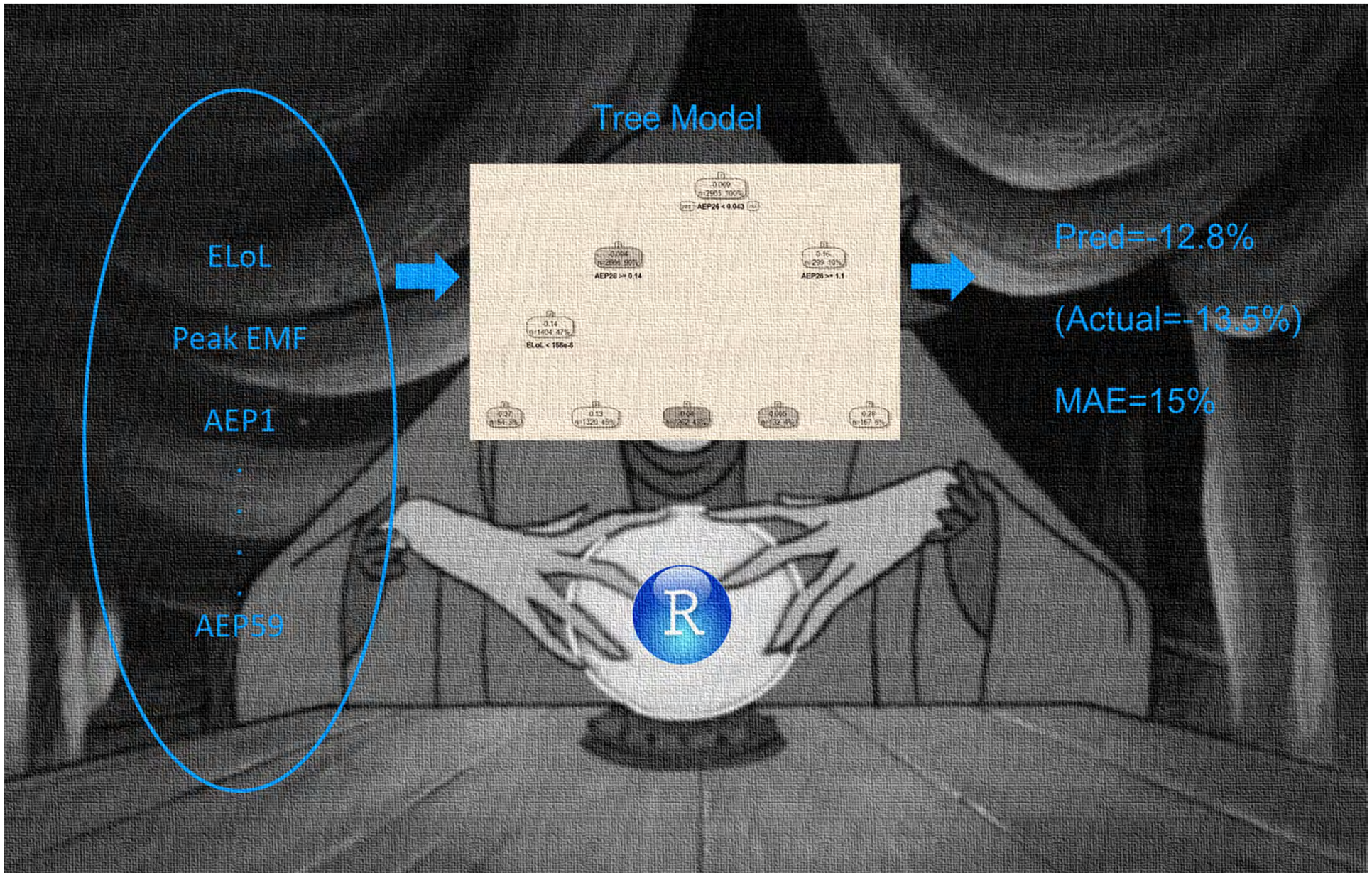


Telling Stories with Pruned Trees (2).

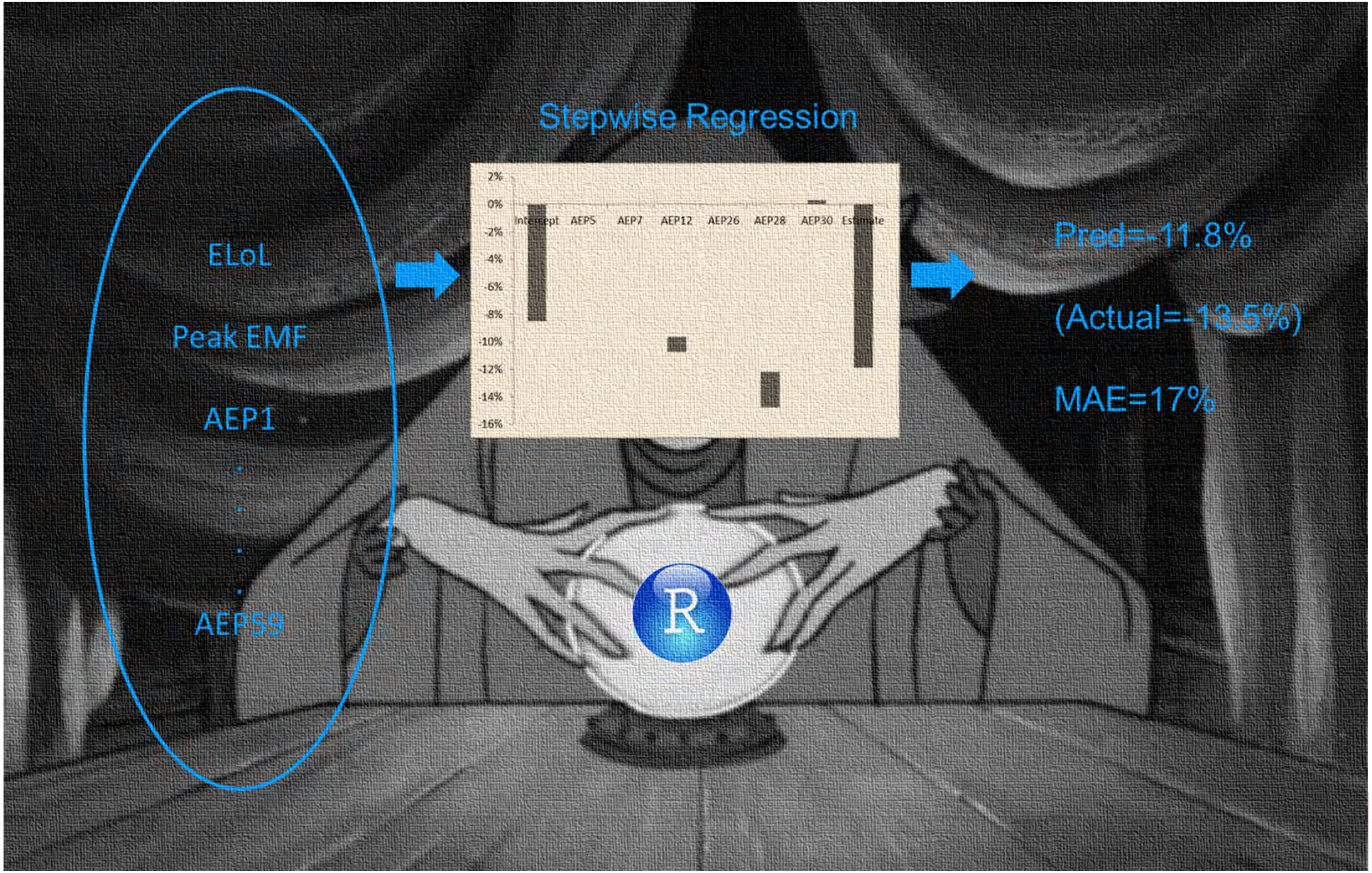
MS amlin



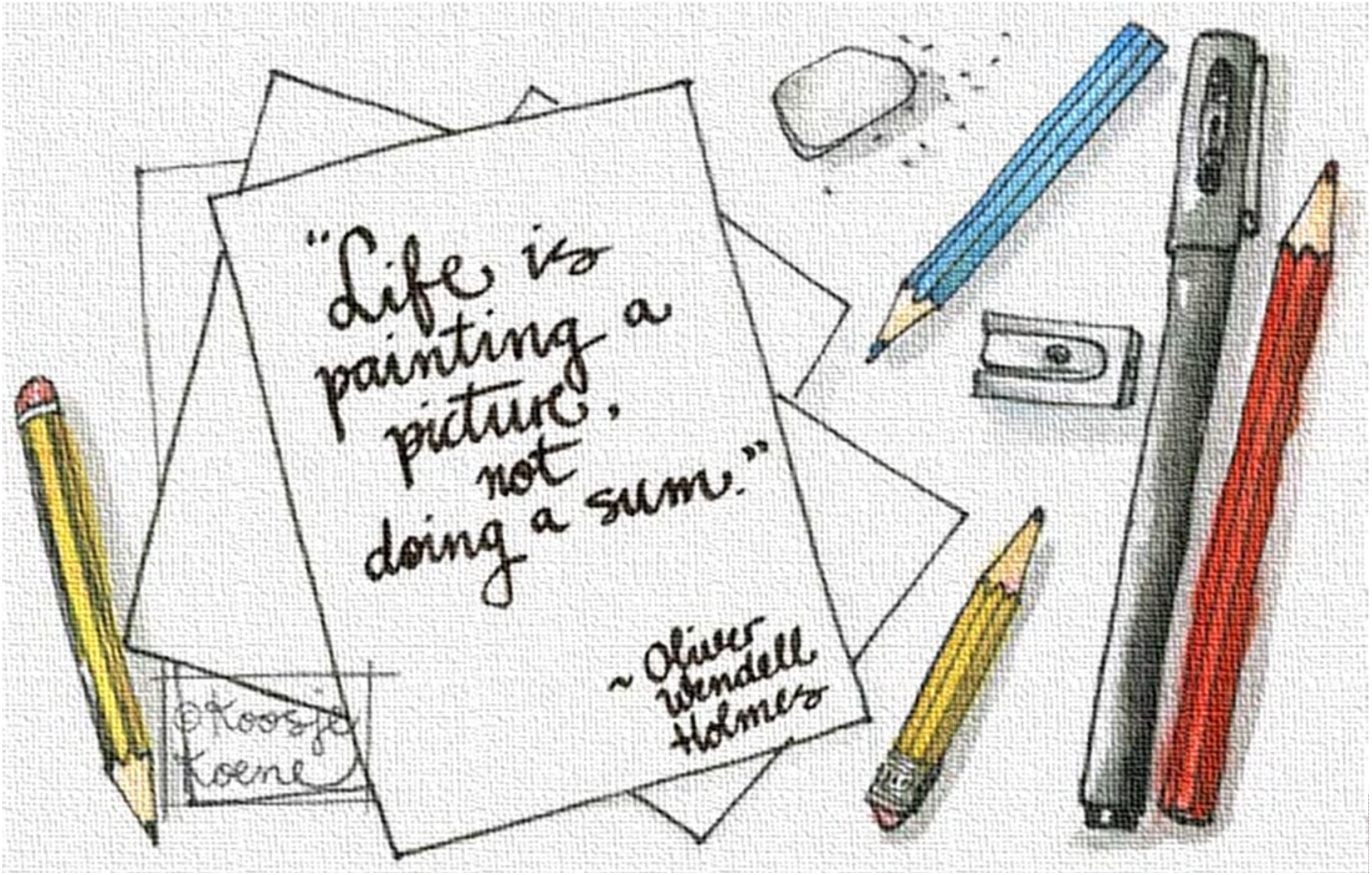
Making Predictions (1).



Making Predictions (2).



Do we have a solution?



Appendix

R Code

```
# Fit regression tree
library(rpart)
fit <- rpart(TPMov ~ ELoL + PeakEMF + AEP4 + AEP5 + AEP6 + AEP7 + AEP8 + AEP10 + AEP12 + AEP13 + AEP14 + AEP15 + AEP18 + AEP25 + AEP26 + AEP28 + AEP30 + AEP56, data=Data,
method="anova")
library(rattle)
library(rpart.plot)
library(RColorBrewer)
fancyRpartPlot(fit)
# Optimising pruning by cross-validation
plotcp(fit)
printcp(fit)
ptree <- prune(fit, cp=fit$cptable[which.min(fit$cptable[,'xerror']),'CP'])
fancyRpartPlot(ptree, uniform=TRUE, main='Pruned Classification Tree')
```

```
library(MASS)
RegressionData <- Data
# Remove outliers
RegressionData <- RegressionData[-c(1199, 1403, 1404),]
# Perform multiple regression
regressionFit <- lm(TPMov ~ ELoL + AEP4 + AEP5 + AEP6 + AEP7 + AEP8 + AEP10 + AEP12 + AEP13 + AEP14 + AEP15 + AEP18 + AEP25 + AEP26 + AEP28 + AEP30 + AEP56, data = RegressionData)
print(regressionFit)
summary(regressionFit)
plot(regressionFit)
# Stepwise regression
step <- stepAIC(regressionFit, direction="both")
step$anova # display results
summary(step)
plot(step)
```

```
MAE <- function(actual, predicted) { mean(abs(actual - predicted)) }
# Regression tree
p.rpart <- predict(fit, Data)
MAE(p.rpart, Data$TPMov)
# Stepwise regression
p.step <- predict(step, RegressionData)
MAE(predict(step, RegressionData), RegressionData$TPMov)
# Uniform allocation
MAE(0, Data$TPMov)
MAE(mean(Data$TPMov), Data$TPMov)
```