Machine Learning in Julia

A Tragicomedy in Five Acts

Alternative Titles

"Machine Learning in Julia: The Highs, The Lows and Other Things You Should Know"

"Machine Learning in Julia: Everyone Can Master a Grief But He That Has It"
(Much Ado About Nothing, Act 3, Scene 2)

Giorgi's suggestion: "Julia: Even Worse than R"

Dramatis Personae

Pedro, a fool.

Julia, a programming language.

Juno, a Julia environment.

Dramatis Personae

Juno/LT

File Edit View Evaluation Help

Customer Targeting 2.jl* | Tree Testing.jl

```
#1. Installing packages needed.
  #Pkg.add("StatsBase") #Basic statistics and sampling.
  #Pkg.add("DataFrames") #This makes working with tabular data much easier.
  #Pkg.add("DecisionTree") #For decision trees!
  #Pkg.add("Gadfly") #Nice package for plots.
  Pkg.add("Cairo") #To export plots to files.
  using StatsBase ✓
  using DataFrames ✓
  using DecisionTree ✓
  using Gadfly ✓
  using Cairo

#2. Reading data.
  BankFull = readtable("E:/Julia/Pedro/Customer Targeting/bank-additional-full.csv", separator=';')
```

DataFrame (age, job, marital, education, default, housing, loan, contact, month, day_of_week, duration, campaign, pdays, previous, poutcome, e

age	job	marital	education	default	housing	loan
56	"housemaid"	"married"	"basic.4y"	"no"	"no"	"no"
57	"services"	"married"	"high.school"	"unknown"	"no"	"no"
37	"services"	"married"	"high.school"	"no"	"yes"	"no"
40	"admin."	"married"	"basic.6y"	"no"	"no"	"no"
56	"services"	"married"	"high.school"	"no"	"no"	"yes"
45	"services"	"married"	"basic.9y"	"unknown"	"no"	"no"
59	"admin."	"married"	"professional.course"	"no"	"no"	"no"
41	"blue-collar"	"married"	"unknown"	"unknown"	"no"	"no"
24	"technician"	"single"	"professional.course"	"no"	"yes"	"no"
25	"services"	"single"	"high.school"	"no"	"yes"	"no"
⋮	⋮	⋮	⋮	⋮	⋮	⋮

```
  showcols(BankFull) #This is good to check there are no missing values. Julia is pretty unforgiving for that kind of thing.
  describe(BankFull) ✓
  #The variable duration is not to be included in the analysis.
  delete!(BankFull, :duration)
```

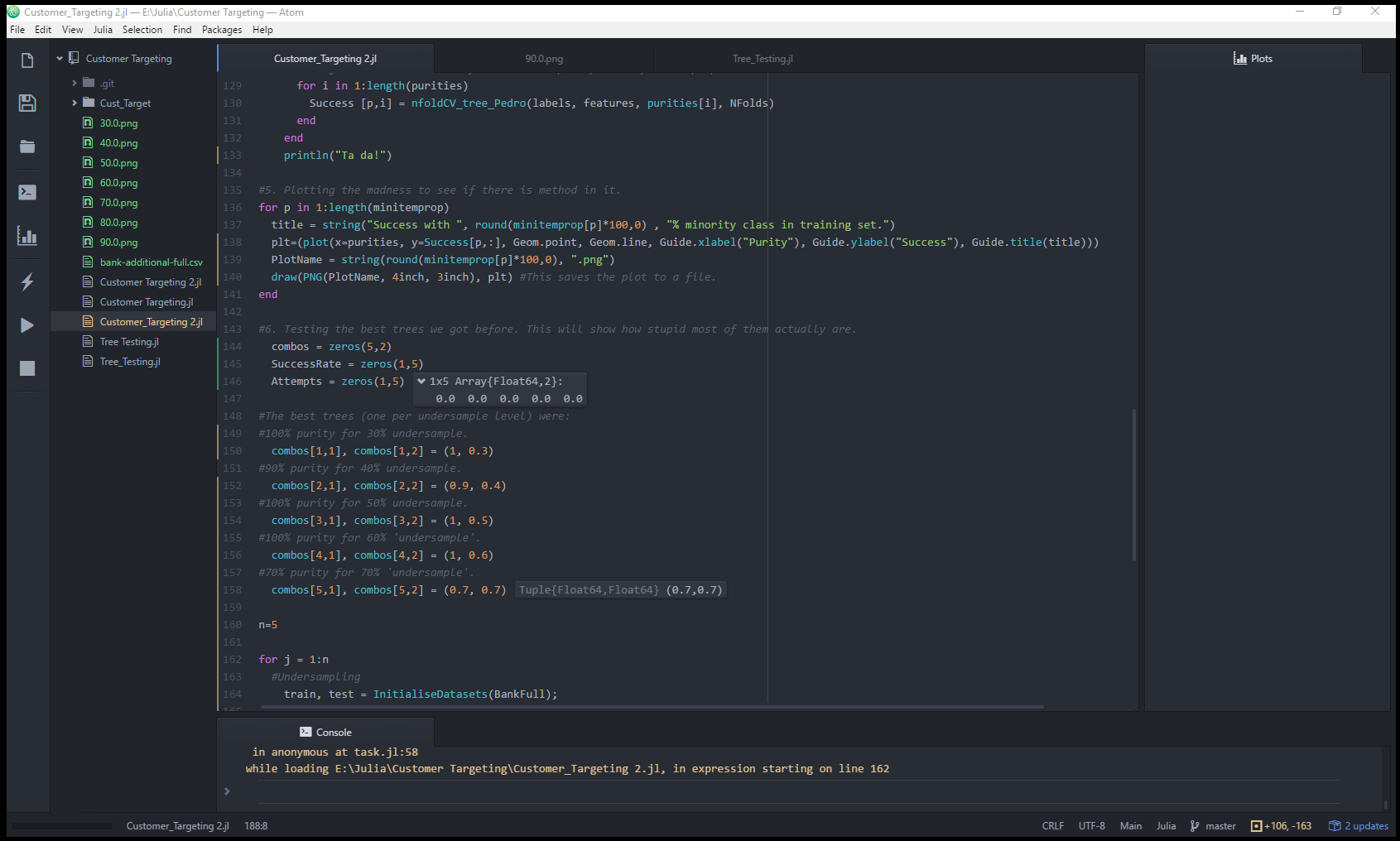
DataFrame (age, job, marital, education, default, housing, loan, contact, month, day_of_week, campaign, pdays, previous, poutcome, emp_var_rate, cons_price_idx, cons_conf_idx, euribor3m, nr_employed, y), 41188

```
#3. Creating training and testing datasets.
  Total = size(BankFull)[1]  41188
  TestProp = 0.3 #Proportion of the dataset reserved for testing.  0.3
  TrainSize = round((1-TestProp)*Total,0)  28832.
  TestSize = Total - TrainSize  12356.
  Total - TrainSize - TestSize #Just checking!  0.
```

Main 15 / 3

```
Type    UTF8String
NAs     0
NA%     0.0%
Unique  2
```

Dramatis Personae



Index

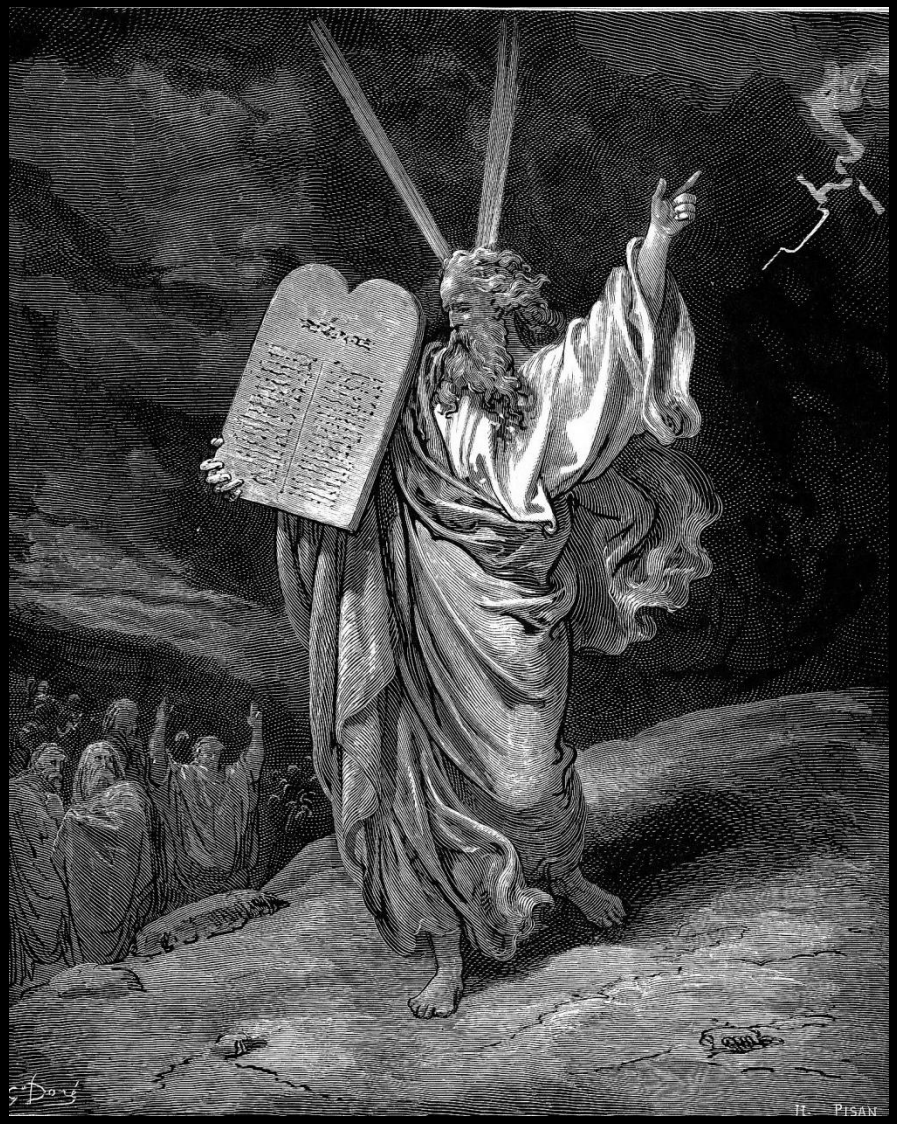
Act 1, The Genesis

"It is the East, and Julia is the sun"
(Pedro and Julia, Act 2, Scene 2)

Act 1, The Genesis

"Oh fool, marvel at all these fine machine learning packages for Julia!

And rejoice in the glory of its data frames!"



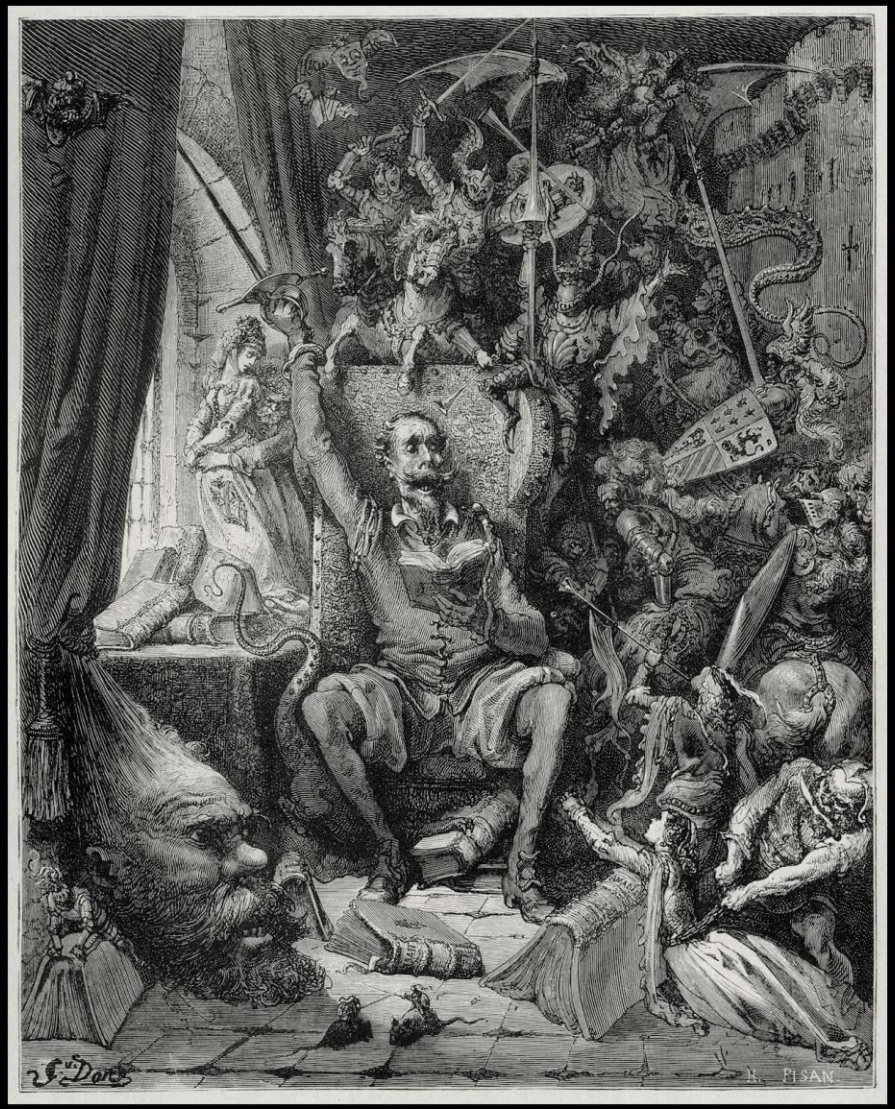
Act 2, The First Fall

```
train = BankFull[1:TrainSize,:]
train = train[1:TrainSize, 1:20] #Dropping the variables Index and RndBuzz.
test = BankFull[TrainSize+1:Total, :]
test = test[1:TestSize, 1:20] #Dropping the variables Index and RndBuzz.
proportionmap(BankFull[:y])
proportionmap(train[:y])
proportionmap(test[:y])
```

Act 1, The Genesis

"I shall slay the Bayes demon and find the fabled random forests of decision trees!"

"And I shall discover optimum proportions for minority class in training datasets when there's class imbalance!"



Act 2, The First Fall

"The course of true love never did run smooth" (A Midsummer Night's Dream, Act 1, Scene 1)

Act 2, The First Fall

"Watch me friend, how I single-handedly defeat the Bayes demon."

Act 2, The First Fall

"What sorcery is this?!"

Act 2, The First Fall

Lesson 1: scarce documentation is common and some packages have been abandoned.

The naïve Bayes package has only two examples with no comments and is no longer maintained.

"Men of few words are the best men" (King Henry the Fifth, Act 3, Scene 2)

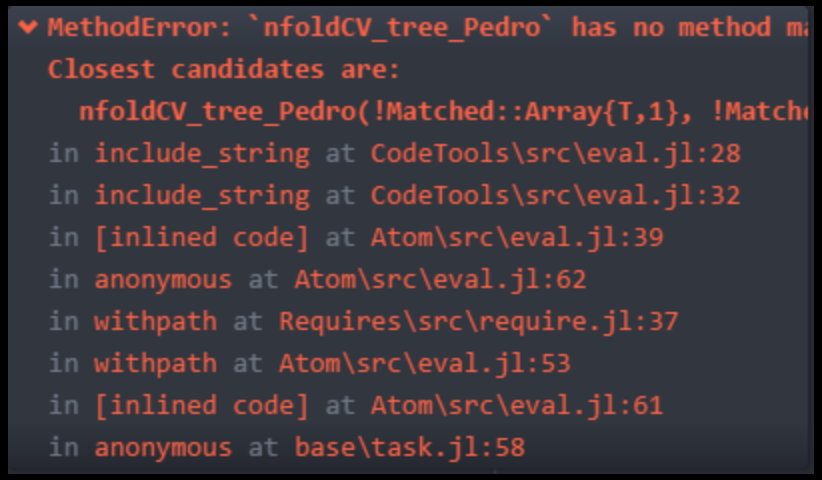
Act 2, The First Fall

```
#------------------Test

train, test = InitialiseDatasets(BankFull); › (28832×20 DataFrames.DataFrame
undertrain=UnderSample (train, 0.5) › 6604×20 DataFrames.DataFrame
#features = convert(Array, undertrain[1:19])
features = undertrain[1:19] › 6604×19 DataFrames.DataFrame
labels = undertrain[20] › 6604-element DataArrays.DataArray{UTF8String,1}:
#labels = convert(Array, undertrain[20])
p = 0.7 0.7
nfoldCV_tree_Pedro(labels, features, p, NFolds)
generic_tree = build_tree(labels, features)
generic_tree = prune_tree(generic_tree, p)
data_test = convert(Array, test[1:19])
results_tree = apply_tree (generic_tree, data_te
ntp = 0
nfp = 0
TestSize = size(data_test)[1]
for i=1:TestSize
  #if test[i,:y]=="yes" && test[i,:first_results
  if test[i,:y]=="yes" && results_tree[i]=="yes"
    ntp=ntp+1
```

```
MethodError: `nfoldCV_tree_Pedro` has no method ma
Closest candidates are:
  nfoldCV_tree_Pedro(!Matched::Array{T,1}, !Matche
in include_string at CodeTools\src\eval.jl:28
in include_string at CodeTools\src\eval.jl:32
in [inlined code] at Atom\src\eval.jl:39
in anonymous at Atom\src\eval.jl:62
in withpath at Requires\src\require.jl:37
in withpath at Atom\src\eval.jl:53
in [inlined code] at Atom\src\eval.jl:61
in anonymous at base\task.jl:58
```

Act 2, The First Fall



MethodError: `nfoldCV_tree_Pedro` has no method matching nfoldCV_tree_Pedro(::DataArrays.DataArray{UTF8String,1}, ::DataFrames.DataFrame, ::Float64, ::Int64)
Closest candidates are:
nfoldCV_tree_Pedro(!Matched::Array{T,1}, !Matched::Array{T,2}, ::Real, ::Integer)
in include_string at C:\Users\New User\.julia\v0.4\CodeTools\src\eval.jl:28
in include_string at C:\Users\New User\.julia\v0.4\CodeTools\src\eval.jl:32
[inlined code] from C:\Users\New User\.julia\v0.4\Atom\src\eval.jl:39
in anonymous at C:\Users\New User\.julia\v0.4\Atom\src\eval.jl:62
in withpath at C:\Users\New User\.julia\v0.4\Requires\src\require.jl:37
in withpath at C:\Users\New User\.julia\v0.4\Atom\src\eval.jl:53
[inlined code] from C:\Users\New User\.julia\v0.4\Atom\src\eval.jl:61
in anonymous at task.jl:58

Act 2, The First Fall

```
#------------------Test

train, test = InitialiseDatasets(BankFull); > (28832×20 DataFrames.DataFrame
undertrain=UnderSample (train, 0.5) > 6604×20 DataFrames.DataFrame
features = convert(Array, undertrain[1:19]) > 6604x19 Array{Any,2}:
#features = undertrain[1:19] > 6604×19 DataFrames.DataFrame
#labels = undertrain[20] > 6604-element DataArrays.DataArray{UTF8String,1}:
labels = convert(Array, undertrain[20]) > Vector UTF8String, 6604
p = 0.7 0.7
nfoldCV_tree_Pedro(labels, features, p, NFolds) 0.503111315971422
```

Act 2, The First Fall

Lesson 2: Julia does not like data frames. Yet.

Data frames have easy syntax and can be used for data manipulation but data must be in vector or matrix format before being used in Julia packages.

"Can one desire too much of a good thing?"
(As You Like it, Act 4, Scene 1)

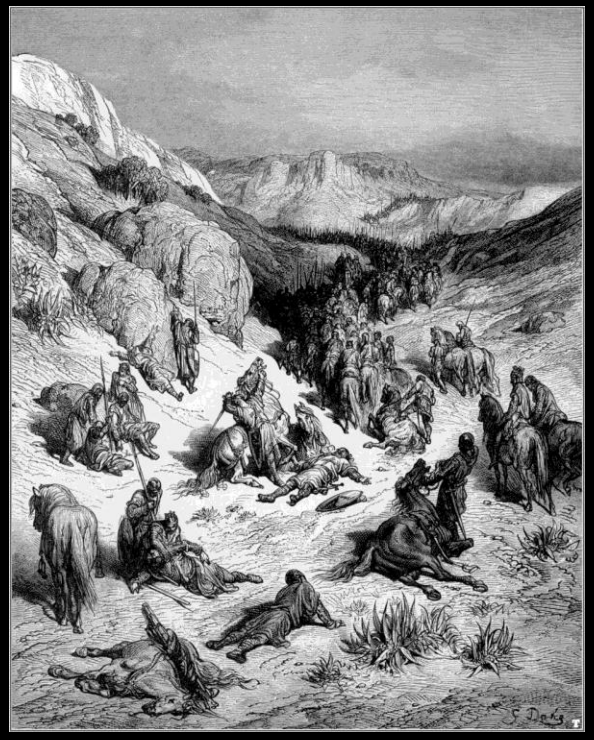
Act 2, The First Fall

Defeated by the Bayes demon, disappointed with data frames and dying of despair.

Act 3, The Second Fall

"When sorrows come, they come not single spies but in battalions"

(Hamlet, Act 4, Scene 5)



Act 3, The Second Fall

"Let us ride forth and find the fable land of the random forests and their mythical decision trees.

What could possibly go wrong?"

Act 3, The Second Fall

"What do you mean the DecisionTrees package returns only one metric and not the confusion matrix?!"

Act 3, The Second Fall

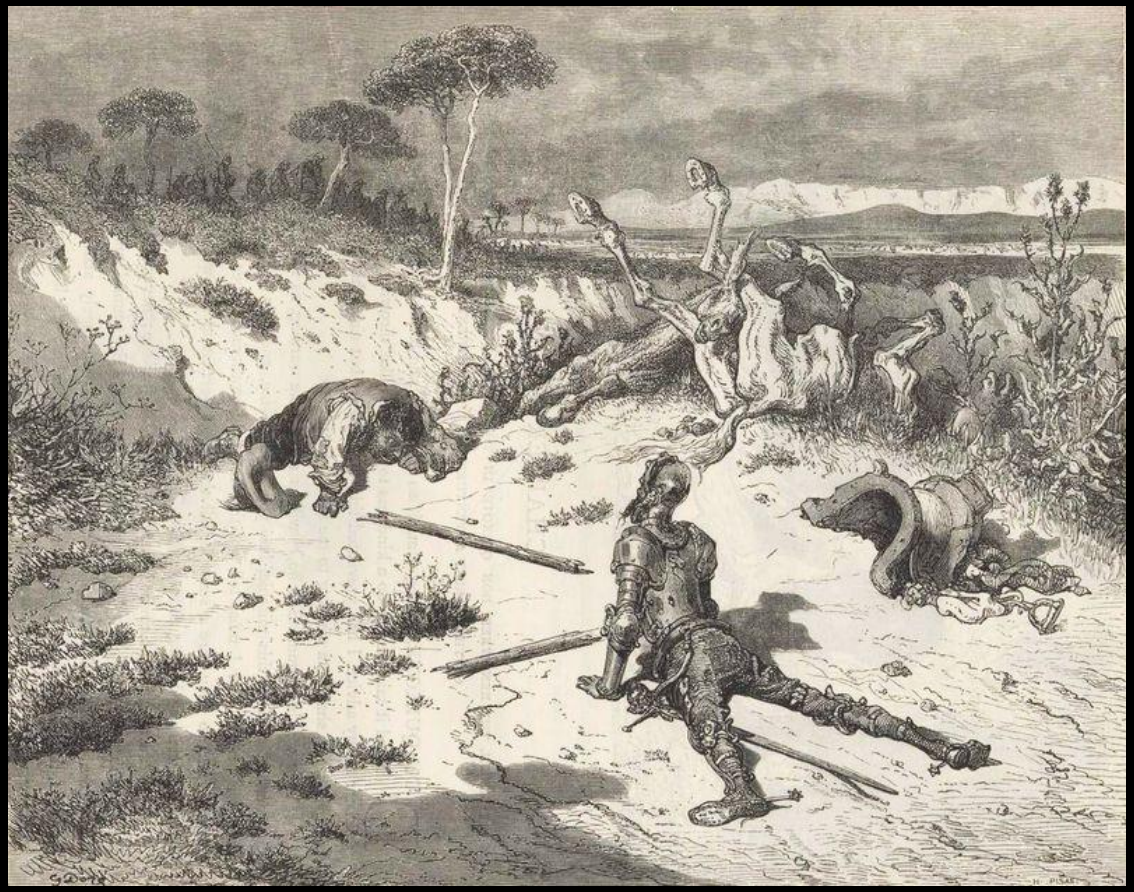
Lesson 3: Many packages are not very comprehensive yet, being designed with very specific purposes in mind, which may not suit your needs.

Be prepared to have to modify some of them.

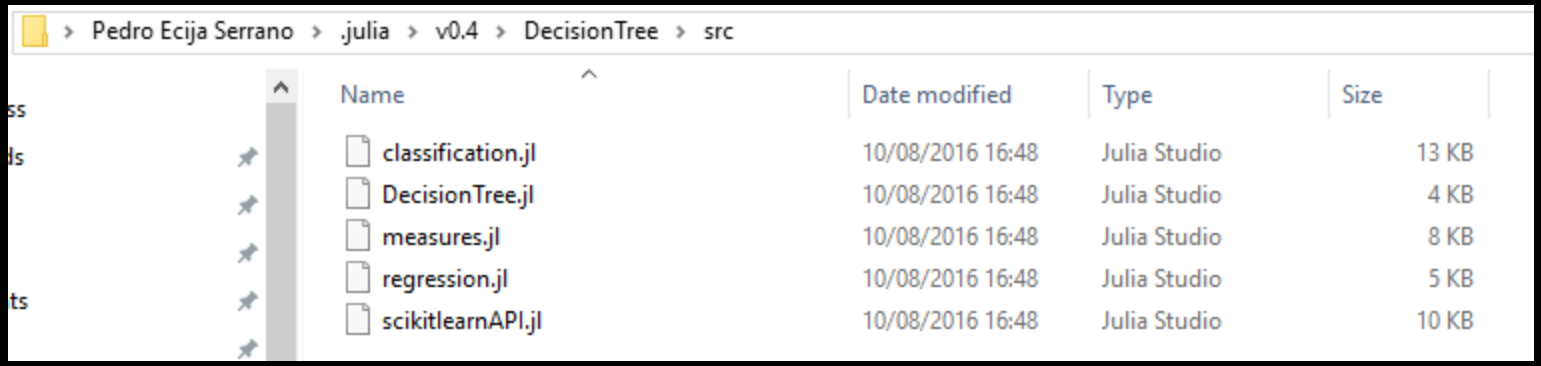
"Brevity is the soul of wit" (Hamlet, Act 2, Scene 2)

Act 3, The Second Fall

"What's this madness? Modifying Julia packages is easy?!"



Act 3, The Second Fall



Act 3, The Second Fall

Customer_Targeting 2.jl • | 90.0.png | Tree_Testing.jl

```julia
type Pedro_ConfusionMatrix
    classes::Vector
    matrix::Matrix{Int}
    accuracy::Float64
    tp::Int32 #True positives
    fp::Int32 #False positives
    tn::Int32 #True negatives
    fn::Int32 #False negatives
    kappa::Float64
end

function Pedro_confusion_matrix(actual::Vector, predicted::Vector)
    @assert length(actual) == length(predicted)
    N = length(actual)
    _actual = zeros(Int,N)
    _predicted = zeros(Int,N)
    classes = sort(unique([actual; predicted]))
    N = length(classes)
    for i in 1:N
        _actual[actual .== classes[i]] = i
        _predicted[predicted .== classes[i]] = i
    end
    CM = zeros(Int,N,N)
    for i in zip(_actual, _predicted)
        CM[i[1],i[2]] += 1
    end
    accuracy = trace(CM) / sum(CM)
    prob_chance = (sum(CM,1) * sum(CM,2))[1] / sum(CM)^2
    kappa = (accuracy - prob_chance) / (1.0 - prob_chance)
    tp = CM[2,2]
    tn = CM[1,1]
    fp = CM[2,1]
    fn = CM[1,2]
    return Pedro_ConfusionMatrix(classes, CM, accuracy, tp, tn, fp, fn, kappa)
end
```

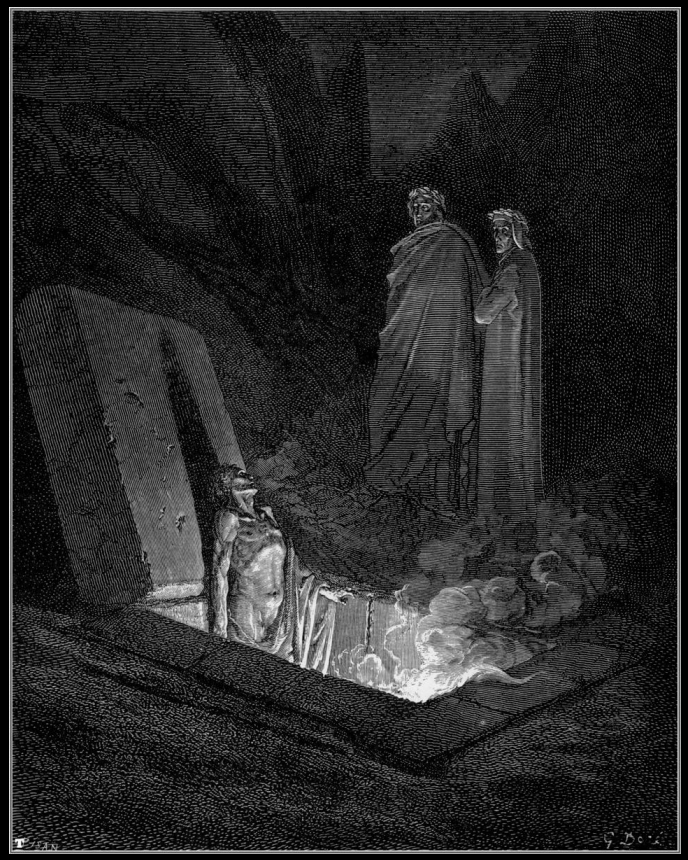
Act 3, The Second Fall

Lesson 4: Editing Julia packages is easy as the code is accessible and written in Julia in most cases. You can add it to your own code for editing and testing in a safe environment.

"Why, then the world's mine oyster" (The Merry Wives of Windsor, Act 2, Scene 2)

Act 4, The Resurrection

"Thought this be madness, yet there is method in't" (Hamlet, Act 2, Scene 2)



Act 4, The Resurrection

"Joyfully programming and defeating random error devils to save the day."

"I just need a ggplot-like package that gives pictures of breathtaking beauty. I shall try Gadfly and plot my many charts with a loop!"

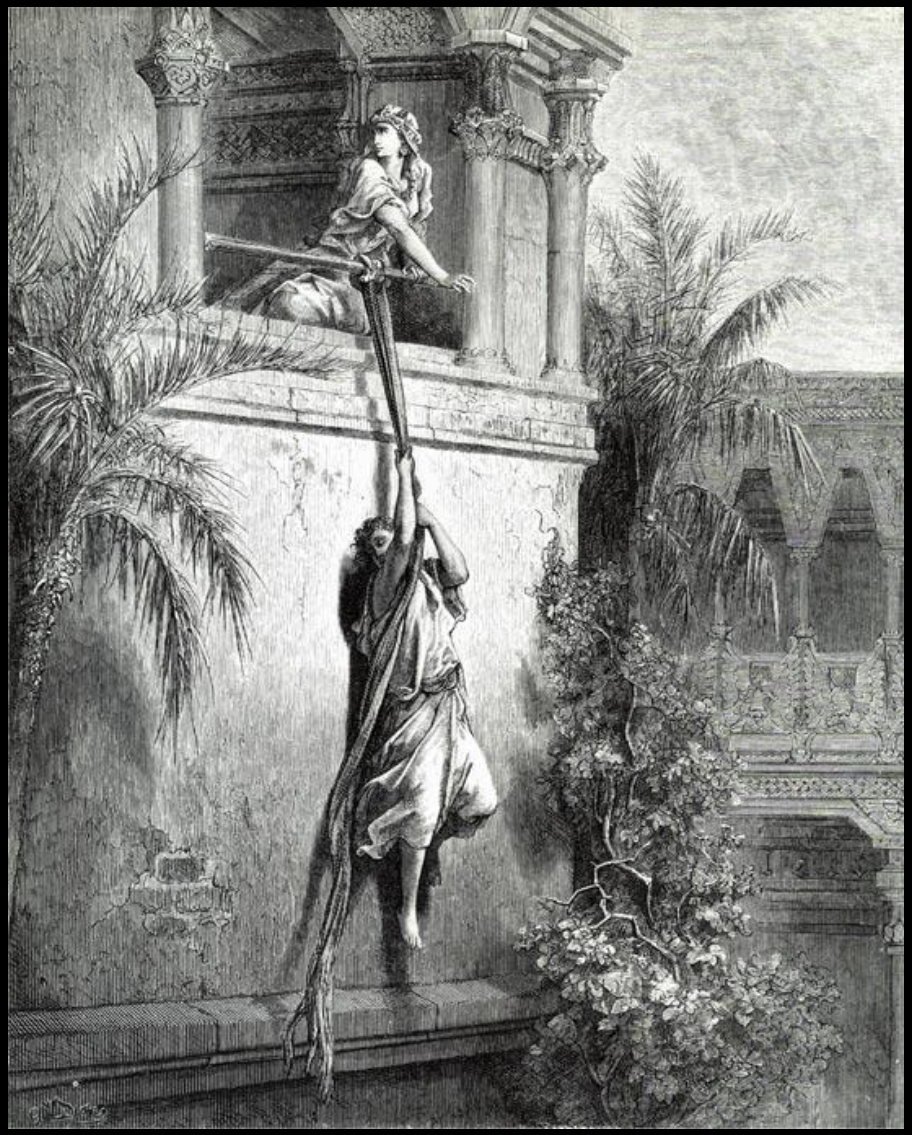
Act 4, The Resurrection

```
#1. Installing packages needed.
  #Pkg.add("StatsBase") #Basic statistics and sampling.
  #Pkg.add("DataFrames") #This makes working with tabular data much easier.
  #Pkg.add("DecisionTree") #For decision trees!
  #Pkg.add("Gadfly") #Nice package for plots.
  #Pkg.add("Cairo") #To export plots to files.
  using StatsBase
  using DataFrames
  using DecisionTree
  using Gadfly
```

```
#5. Plotting the madness to see if there is method in it.
for p in 1:length(minitemprop)
  title = string("Success with ", round(minitemprop[p]*100,0) , "% minority class in training set.")
  plot(x=purities, y=Success[p,:], Geom.point, Geom.line, Guide.xlabel("Purity"), Guide.ylabel("Success"), Guide.title(title))
end
```

Act 4, The Resurrection

"If this was going so well, how did we end up here?"



Act 4, The Resurrection

```
#5. Plotting the madness to see if there is method in it.
for p in 1:length(minitemprop)
  title = string("Success with ", round(minitemprop[p]*100,0) , "% minority class in training set.")
  display(plot(x=purities, y=Success[p,:], Geom.point, Geom.line, Guide.xlabel("Purity"), Guide.ylabel("Success"), Guide.title(title)))
end
```

Act 4, The Resurrection

Lesson 5: Julia (or Juno?) does not actually plot when the statements are in a loop. We must force the plot by using the display.

"But love is blind and lovers cannot see"
(The Merchant of Venice, Act 2, Scene 6)

Act 4, The Resurrection

"Yes, trust me, you can use Cairo to plot directly into a file.

Much better than plotting in the console."

Act 4, The Resurrection

```
#1. Installing packages needed.
  #Pkg.add("StatsBase") #Basic statistics and sampling.
  #Pkg.add("DataFrames") #This makes working with tabular data much easier.
  #Pkg.add("DecisionTree") #For decision trees!
  #Pkg.add("Gadfly") #Nice package for plots.
  #Pkg.add("Cairo") #To export plots to files.
  using StatsBase
  using DataFrames
  using DecisionTree
  using Gadfly
  using Cairo
```

```
Failed to precompile Cairo to C:\Users\New User\.julia\lib\v0.4\Cairo.ji
in error at error.jl:21
```

Act 4, The Resurrection

"I must go on..."

Act 4, The Resurrection

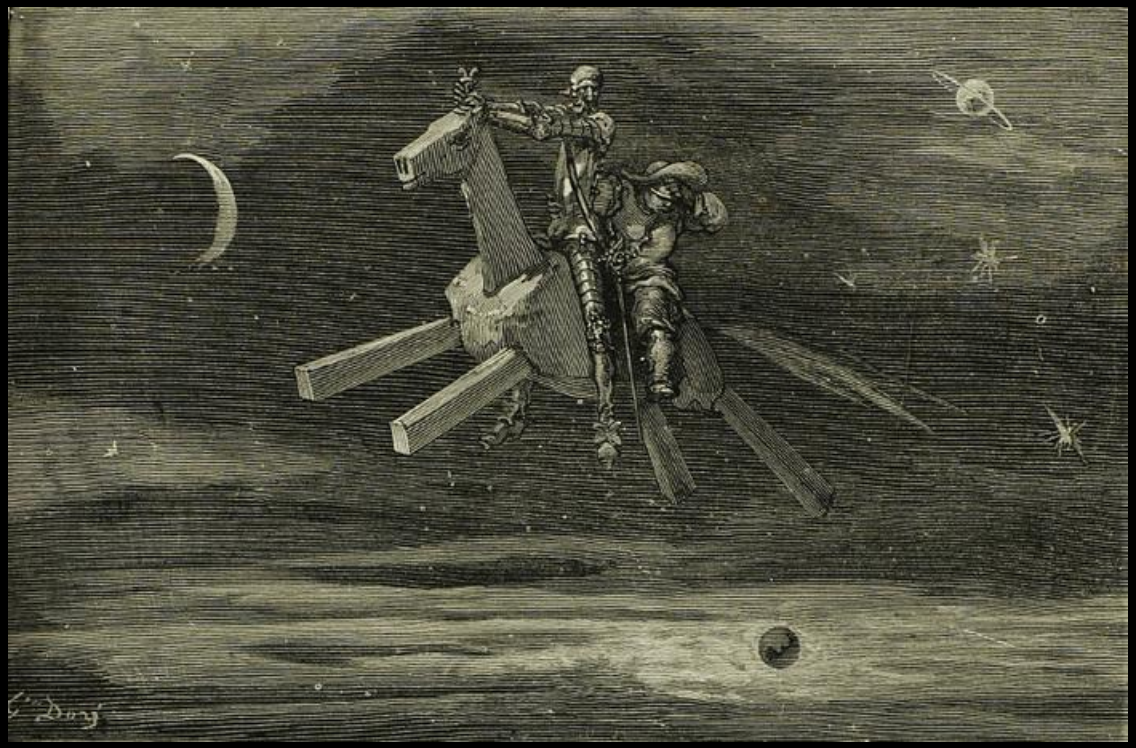
Lesson 6: Cairo allows you to use statement draw to plot directly to a file. However, Cairo fails to compile and there seems to be a bug that the community is failing to fix.

A month later, after updating Juno and Julia, Cairo compiled. The community works fast and some packages might be very sensitive to the version of Julia you are using.

"Temp not a desperate man", (Pedro and Julia, Act 5, Scene 3)

Act 4, The Resurrection

"Fortune smiles! We can do it this time!"



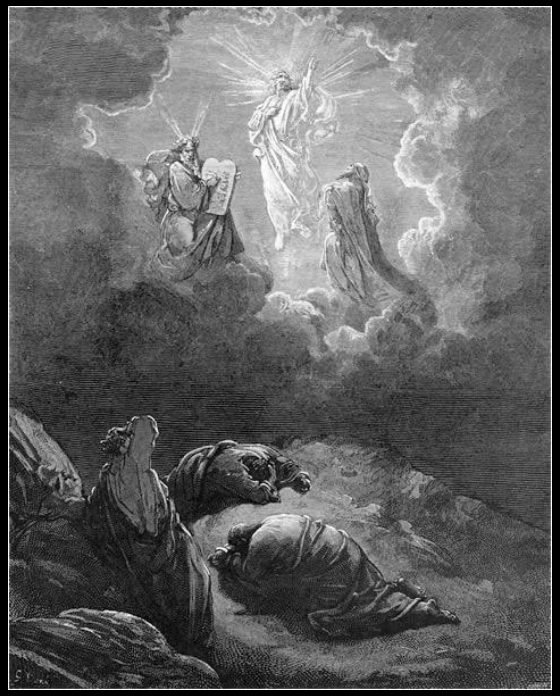
Act 4, The Resurrection

```
#5. Plotting the madness to see if there is method in it.
for p in 1:length(minitemprop)
  title = string("Success with ", round(minitemprop[p]*100,0) , "% minority class in training set.")
  plt=display(plot(x=purities, y=Success[p,:], Geom.point, Geom.line, Guide.xlabel("Purity"), Guide.ylabel("Success"), Guide.title(title)))
  PlotName = string(round(minitemprop[p]*100,0), ".png")
  draw(PNG(PlotName, 4inch, 3inch), plt)
end  ❯ MethodError: `draw` has no method matching draw(:
```

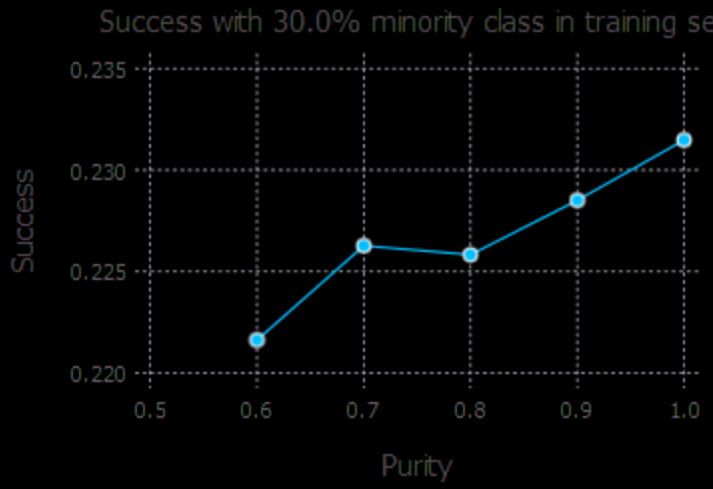
```
#5. Plotting the madness to see if there is method in it.
for p in 1:length(minitemprop)
  title = string("Success with ", round(minitemprop[p]*100,0) , "% minority class in training set.")
  plt=(plot(x=purities, y=Success[p,:], Geom.point, Geom.line, Guide.xlabel("Purity"), Guide.ylabel("Success"), Guide.title(title)))
  PlotName = string(round(minitemprop[p]*100,0), ".png")
  draw(PNG(PlotName, 4inch, 3inch), plt)
end  ✓
```

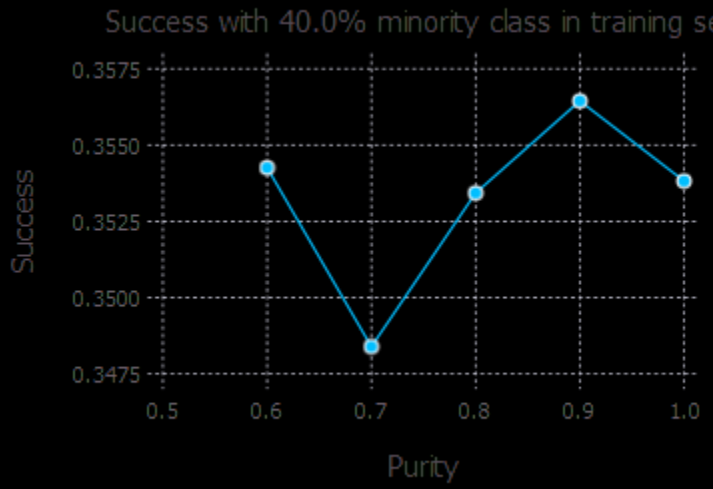
Act 5, Ascendance to Heavens

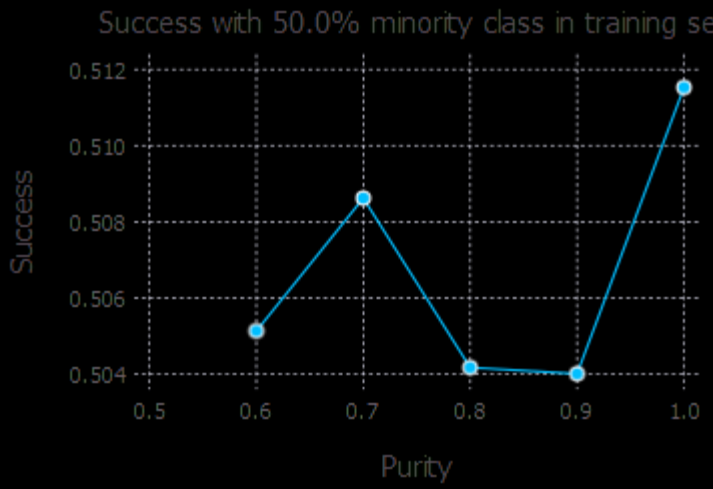
"Small things make base men proud"
(King Henry the Sixth,
Part II, Act 4, Scene 1)

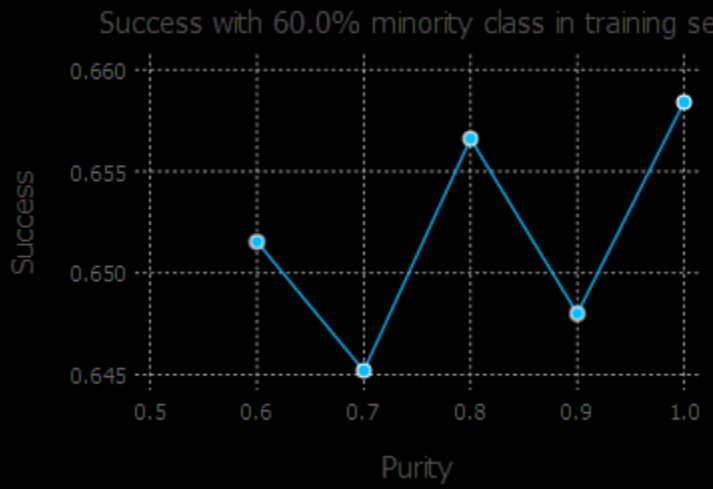


Act 5, Ascendance to Heavens





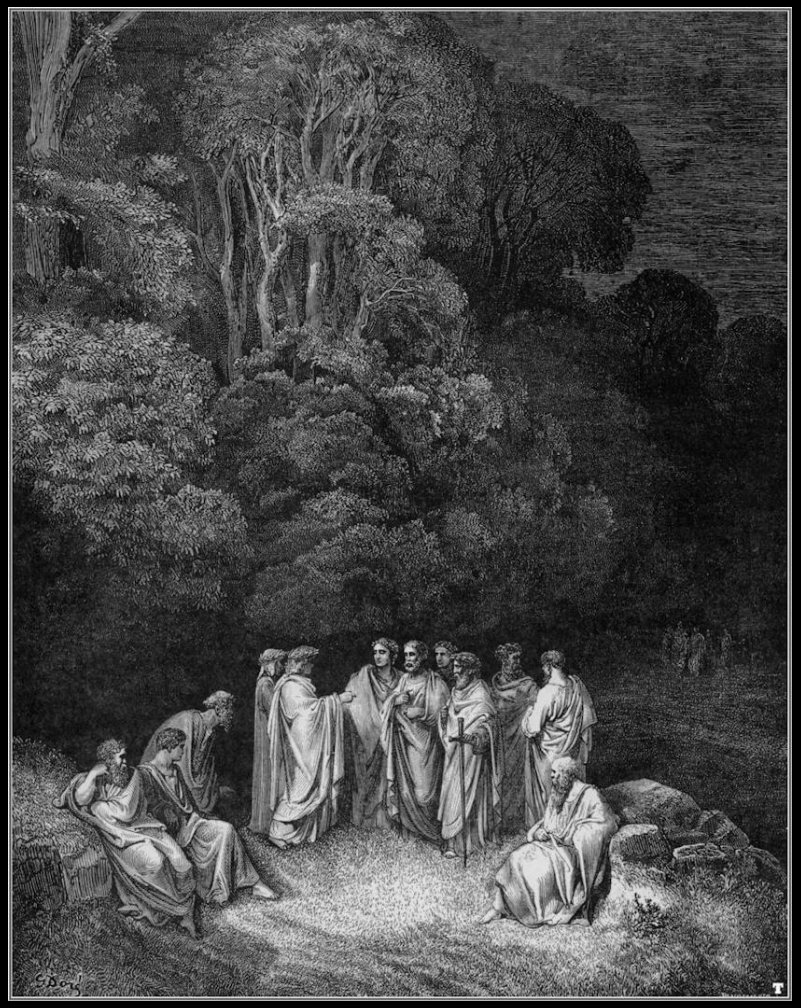




Act 5, Ascendance to Heavens

Epilogue

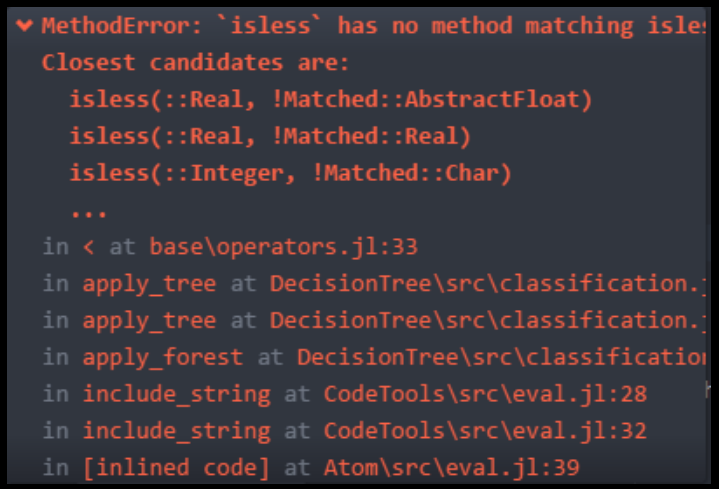
And the random forest?



Epilogue


#6 Going wild here and now trying random forest.
#Undersampling
train, test = InitialiseDatasets(BankFull); (28832×20 DataFrames.DataFrame
data_test = convert(Array, test[1:19]) #Remember, the 20th column is the actual re
TestSize = size(data_test)[1] 12356
prop = 0.5 0.5
undertrain = UnderSample (tra
MethodError: `isless` has no method matching isle
Closest candidates are:
isless(::Real, !Matched::AbstractFloat)
isless(::Real, !Matched::Real)
isless(::Integer, !Matched::Char)
...
in < at base\operators.jl:33
in apply_tree at DecisionTree\src\classification.
in apply_tree at DecisionTree\src\classification.
in apply_forest at DecisionTree\src\classificatio
in include_string at CodeTools\src\eval.jl:28
in include_string at CodeTools\src\eval.jl:32
in [inlined code] at Atom\src\eval.jl:39
#Building a forest
features = convert(Array, und
labels = convert(Array, under
n_feat = 2 #Number of random
n_trees = 10 #Number of trees
prop_sample = 0.5 #Proportion
tree_depth = 6 #Maximum tree
Sherwood = build_forest(label
#Applying the forest
test = convert(Array, test[1:19]) Vector Any, 19
apply_forest(Sherwood, test)

Epilogue



MethodError: `isless` has no method matching isless(::Int64, ::UTF8String)
Closest candidates are:
isless(::Real, !Matched::AbstractFloat)
isless(::Real, !Matched::Real)
isless(::Integer, !Matched::Char)
...
in < at operators.jl:33
in apply_tree at C:\Users\New User\.julia\v0.4\DecisionTree\src\classification.jl:188
in apply_tree at C:\Users\New User\.julia\v0.4\DecisionTree\src\classification.jl:189
in apply_forest at C:\Users\New User\.julia\v0.4\DecisionTree\src\classification.jl:247
in include_string at C:\Users\New User\.julia\v0.4\CodeTools\src\eval.jl:28
in include_string at C:\Users\New User\.julia\v0.4\CodeTools\src\eval.jl:32
in [inlined code] from C:\Users\New User\.julia\v0.4\Atom\src\eval.jl:39
in anonymous at C:\Users\New User\.julia\v0.4\Atom\src\eval.jl:62 in withpath at C:\Users\New User\.julia\v0.4\Requires\src\require.jl:37
in withpath at C:\Users\New User\.julia\v0.4\Atom\src\eval.jl:53 [inlined code] from C:\Users\New User\.julia\v0.4\Atom\src\eval.jl:61
in anonymous at task.jl:58

Epilogue

Bonus Lesson: random forests have an issue with strings in my data set. I have not figured out how to fix it yet. The war goes on...

"For the rain it raineth every day",
(Taming of the Shrew, Act 5, Scene 1)

Epilogue

Summary:

1) There are dataframes in Julia but some work around needed as use is not widespread yet.
2) There is an increasing number of packages for machine learning in Julia.
3) But documentation is scarce.
   However, modifying those packages is easy.
4) Gadly is great but plotting in loops requires "display". Saving to a file requires Cairo and "draw".
5) When there is no hope, update Julia and packages.
6) Do not despair, it is a rapidly evolving community and improvements and solutions appear all the time: you win if you do not lose!
7) Do not be a fool like me and check:
   http://www.juliabloggers.com/
   https://en.wikibooks.org/wiki/Introducing_Julia